

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEINEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

UNIVERSITE FERHAT ABBAS DE SETIF
UFAS, ALGERIE

MEMOIRE

Présentée à la faculté des sciences
Département D'informatique

Pour l'obtention du diplôme de

MAGISTER

Option : Ingénierie Des Systèmes Informatique

Présentée par : GHOUL KHALID

Application des systèmes multi-agents pour des enchères
Optimales au sens de Pareto

Soutenue le : 03/07/2011

Devant la commission d'examen :

Mr. A.MOUSSAOUI

Maitre de conférences U.F.A SETIF

Président

Mr. A.KHABABA

Maitre de conférences U.F.A SETIF

Rapporteur

Mr. M. TOUAHRIA

Maitre de conférences U.F.A SETIF

Examineur

Dedicaces

En déposant ce mémoire, je tiens d'abord à exprimer ma profonde gratitude envers mon encadreur Khababa Abdallah pour sa confiance, ses encouragements continuels et son suivi de près de nos travaux en dirigeant ce mémoire.

Je remercie également tous les enseignants d'Informatique de l'université de Ferhat Abbas-Sétif qui m'ont accueilli et qui m'ont permis de m'épanouir aussi bien dans mes recherches que dans mes enseignements.

Je remercie ma famille et plus particulièrement mes parents, qui m'ont toujours soutenue dans mes études. Je remercie enfin également tous ceux qui ont aidé de près ou de loin à la réalisation de notre projet.

Remerciements

À mes très chers parents

Que Dieu les garde

À toute ma famille et mes amis

À tous ceux qui sont proches de mon cœur

et dont je n'ai pas cité les noms

À mon pays

Je dédie ce modeste travail

Table des matières

Dedicaces	ii
Remeiciements	iii
Introduction	1
1 Agents et Systèmes Multi Agent	5
1.1 Introduction	5
1.2 Qu'est ce qu'un agent ?	6
1.3 Typologie d'agents	7
1.4 Qu'est ce qu'un Système multi-agents ?	8
1.5 L'Environnement dans les systèmes multi-agents	10
1.5.1 Différents types d'environnement	11
1.5.2 Organisation	12
1.6 Communication entre les agents	13
1.6.1 Communication par envoi des messages	14
1.6.2 Communication par partage d'informations	14
1.6.3 Les langages de communication	15
1.7 Les protocoles d'interaction dans les systèmes multi agents	18
1.7.1 Les protocoles de coordination	18
1.7.2 Les protocoles de coopération	18
1.7.3 Les protocoles de négociation	19
1.7.4 Les mécanismes du commerce électronique	19
1.7.5 Les protocoles de FIPA	19
1.8 Utilités des systèmes multi-agents	22
1.9 Conclusion	23

2	L'ingénierie logicielle orientée agents	24
2.1	Introduction	24
2.2	Méthodologies	24
2.2.1	MAS CommonKADS	25
2.2.2	MaSE (Multi-agent System Engineering).	27
2.2.3	La méthodologie GAIA	29
2.2.4	La méthodologie Aalaadin(ou AGR : Agent-Groupe-Rôle)	29
2.3	Outils	31
2.3.1	AUML (Agent based Unified Modeling Language)	31
2.3.2	AgentTool	32
2.3.3	JADE	33
2.3.4	Madkit	35
2.4	Conclusion	35
3	La négociation aux systèmes multi agents	36
3.1	Introduction	36
3.2	Qu'est ce que la négociation ?	36
3.3	Domaines de la négociation	38
3.4	Négociation automatisée : difficultés	39
3.4.1	Besoin d'une ontologie	40
3.4.2	La Formulation de la négociation	40
3.4.3	L'exploitation de la Stratégie	42
3.4.4	L'arrêt de la négociation	43
3.5	Négociation pour l'allocation des tâches	43
3.6	Négociation aux enchères	44
3.6.1	La théorie d'enchères	44

3.6.2	Protocoles d'enchère	44
3.7	Négociation heuristique	45
3.8	La négociation par argumentation	47
3.9	SMACE (Sistema multi - Agente Para. Comércio Electrónico)	48
3.9.1	Modèle De Négociation dans SMACE	48
3.9.2	Protocole de négociation de SMACE	49
3.10	Conclusion	50
4	L'aide à la décision multicritères	51
4.1	Introduction	51
4.2	Notion de base	53
4.3	La formulation multicritère d'un problème de décision	54
4.4	Méthodes multicritères	56
4.4.1	Les méthodes de sur-classement	56
4.4.2	Les méthodes basées sur la théorie de l'utilité	60
4.4.3	Autre méthodes	62
4.5	Conclusion	63
5	Modèle et réalisation	64
5.1	Introduction	64
5.2	Concepts de base pour les enchères multicritères	65
5.3	Vente aux enchères fondée sur la théorie le l'utilité	67
5.3.1	La somme Pondérée	68
5.3.2	Les moyennes	71
5.3.3	Goal Programming	73
5.4	Vente aux enchères fondée sur les méthodes de sur-classement	73
5.4.1	ELECTRE I	73

5.4.2	Promethee 1	75
5.5	Analyse des résultats	78
5.6	Le Modèle proposé	79
5.6.1	Description	79
5.6.2	Notations	80
5.6.3	Processus d'enchère	81
5.6.4	Etude Comparative	83
5.7	Conception du système	89
5.7.1	Choix de la méthodologie	89
5.7.2	Identification des agents	90
5.7.3	Les diagrammes de classes des agents	91
5.7.4	Diagrammes de séquences pour la vente à l'enchère	91
5.7.5	Diagrammes AUML de vente aux enchères anglaise inversée	95
5.7.6	Diagramme d'activités	95
5.8	Réalisation	95
5.8.1	Présentation de logiciel	95
	Conclusion et Perspectives	102
	Abstract	111

Liste des tableaux

1.1	Les primitives pour d'écrire un message ACL	17
3.1	Les différentes procédures d'enchères	46
4.1	Les problématiques de référence	52
4.2	Les valeurs de $p(d)$ selon le critère	59
5.1	Les préférences de l'utilisateur	69
5.2	Résultats du modèle de la somme pondérée	70
5.3	Résultat de contre exemple	70
5.4	Résultat des différentes moyennes	72
5.5	Résultat de GP	74
5.6	Résultats de Electre1	76
5.7	Résultats de Promethee1	78
5.8	La comparaison entre les modèles	86
5.9	Résultat d'un contre exemple pour le modèle de point de référence au premier tour	87
5.10	Résultat d'un contre exemple pour le modèle de point de référence au 2ème tour	88
5.11	Résultat du modèle proposé	89

Table des figures

1-1	L'agent, un processus à 3 phases : perception, délibération puis action . . .	7
1-2	Vue canonique d'un SMA[Woo00a]	11
1-3	Un programme générique de l'environnement [Rus03]	12
1-4	Caractérisation d'un SMA [Zam00]	13
1-5	Communication par envoi de messages	14
1-6	Communication par partage d'informations	15
1-7	Structure globale d'un message KQML	16
1-8	Le protocole FIPA-Contract-Net	21
2-1	Schéma générale de la méthodologie MaSe	28
2-2	Vue générale de phases de GAIA[Woo00b]	30
2-3	Relation entre agents, groupes et rôles[Jac98a] [Jac98b]	31
2-4	L'environnement AgentTool	33
5-1	Diagramme de classe AUML de l'agent Acheteur	92
5-2	Diagramme de classe AUML de l'agent Vendeur	93
5-3	Diagramme de séquence correspond à l'enchère	94
5-4	Diagrammes AUML de vente aux enchères anglaise inversée	96
5-5	Diagramme d'activité Acheteur	97
5-6	Diagramme d'activité Vendeur	98
5-7	L'environnement du système	99
5-8	L'interface de l'acheteur	99
5-9	L'interface pour les préférences	100
5-10	Le processus de négociation	100

5-11 L'interface des vendeurs 101

Introduction

Notre travail se positionne au sein de l'intelligence artificielle et plus particulièrement les systèmes multi agent qui fournissent une nouvelle approche pour analyser, designer et implémenter des applications sophistiquées. Aujourd'hui, la plupart des applications nécessitent de distribuer des tâches entre des " entités " autonome afin d'atteindre leurs objectifs d'une manière optimale. Il y a évidemment plusieurs domaines d'application pour les agents dû au fait que les architectures basées sur les agents fournissent une manière bien particulière afin d'aborder des problèmes rapidement. C'est pour cette raison que les agents sont largement utilisés dans le domaine de commerce électronique.

Les agents sont classifiés selon plusieurs critères, La taxinomie que nous utilisons dans ce travail est similaire à celle de Guttman [Gut98]. Elle s'appuie comme critère de classification sur l'étape du processus d'achat. On trouve également les agents de recommandation, les agents de recherche et les agents de transaction. Les agents de recommandation permettant de conseiller les consommateurs sur l'achat de certains produits. Ces agents servent à retranscrire sur le réseau une fonction d'aide ou de conseil à l'achat, remplie dans le commerce traditionnel, par le personnel de vente, ainsi facilitent la sélection des produits qui se trouvent dans le catalogue.

Comme le temps nécessaire à l'obtention d'une information est proportionnel à la quantité d'information disponible, la durée d'accès à l'information augmente donc également à un taux exponentiel. Dans ce cadre, le recours à des agents de recherche et d'analyse de l'information devient indispensable. En termes plus économiques, ces agents diminuent significativement les coûts de recherche d'information.

Les agents de transaction permettent à des acheteurs et des vendeurs de déterminer les termes d'un échange, sans avoir à se rencontrer physiquement, en interagissant unique-

ment par l'intermédiaire de leurs agents. Il existe principalement deux catégories d'agents de transaction : les agents d'enchères par procuration et les agents de négociation. Dans le cadre d'une vente aux enchères, l'utilisation d'agents permet de mettre en place un système d'enchères par procuration. Un consommateur qui désire participer à une vente aux enchères à la possibilité de se faire représenter par un agent dont il détermine au préalable la stratégie en fixant un certain nombre de paramètres (valeur de réservation, durée maximale de participation, etc.).

Sur la plate-forme Auction Bot¹, les utilisateurs ont non seulement la possibilité de déléguer leur processus de surenchère à un agent mais également de choisir les paramètres de la procédure d'enchères dans laquelle leur agent va interagir (durée de la vente, nombre de partis autorisés à participer, etc.). Les agents de négociation permettent aux acheteurs et vendeurs désirant acquérir ou vendre un bien de négocier les termes de la transaction. L'interaction entre ces agents « vendeurs » et ces agents « acheteurs » se déroulera sur une place de marché électronique (C.-à-d. un système multi-agent). Pour résoudre un conflit, les agents doivent être capables de communiquer entre eux en s'échangeant de manière itérative des offres et des contre-offres. Le marché Kasbah² [Cha96] utilise également ce type d'agents qui négociaient de manière bilatérale le prix d'un produit sous la forme d'offres à prendre ou à laisser. Plus précisément, dans une étape de négociation, un agent « acheteur » propose un prix d'échange à un agent « vendeur » que ce dernier ne peut qu'accepter ou refuser. Dans des systèmes multi-agents plus récents, le protocole de négociation a été amélioré pour permettre aux agents de ne plus se limiter à la négociation du seul prix d'une transaction.

Sur le marché tête-à-tête³ [Gut98], les agents peuvent négocier de manière multilatérale plusieurs dimensions d'une transaction. Un autre système multi-agent qui permet aux agents de négocier plusieurs dimensions d'une transaction est ADEPT⁴ (Advance Decision Environment for Process Tasks) [Far98].

¹auction.eecs.umich.edu, développée par l'université du Michigan

²kasbash.media.mit.edu, développé au MIT, est l'un des tous premiers systèmes multi-agents

³ecommerce.media.mit.edu/tête-à-tête

⁴Ce marché a été développé conjointement par l'université Queen Mary at Westfield College et British Telecom

Notre recherche se place dans le domaine de protocole de négociation multicritères, notre objectif est de formaliser une vente aux enchères optimale aux sens de Pareto s'appuyant sur la notion d'agents et de système multi agents. On a vocalisé sur l'enchère multicritères anglaise inversée. Ce système est constitué d'agents de transaction qui utilisent l'envoi de message comme modèle de communication, on parle d'un modèle d'interaction directe entre les agents C.-à-d. sans l'intervention du commissaire priseur, l'interaction se fait entre les agents vendeurs et l'agent acheteur. Afin d'atteindre notre objectif, le document est organisé comme suit.

Organisation du document

Nous avons choisi de structurer ce document en 5 parties

Chapitre 1 : Agents et Systèmes Multi Agents

Chapitre 2 : L'ingénierie logicielle orientée agent

Chapitre 3 : Négociation dans les systèmes multi agents

Chapitre 4 : Aide à la décision multicritères

Chapitre 5 : Modèle et réalisation

Le premier chapitre qui est considéré comme un état d'art, permet de donner au lecteur une vue sur la notion d'agent et de systèmes multi agents, ainsi tous les aspects relatifs à ces notions. Dans le deuxième chapitre nous introduisons les principales méthodologies existantes pour la modélisation des systèmes multi-agent, ainsi que les outils associés à ces méthodologies qui nous aidons à analyser et à concevoir tels systèmes.

Le troisième chapitre contient les différentes définitions et modèles pour la négociation dans les systèmes multi agents et permet ainsi de comprendre le besoin de ces agents dans le processus de négociation, ainsi les types de ventes aux enchères connues et les caractéristiques de chaque type.

Dans le quatrième chapitre nous donnons une vue suffisante sur les principales notions d'aide à la décision multicritères et nous détaillons les différentes méthodes d'analyse. Le cinquième chapitre présente notre étude qui comprend à la fois l'étude théorique pour donner une vue sur l'aspect théorique et mathématique des enchères multicritère. Et l'étude

conceptionnelle afin de développer notre système en utilisant la méthodologie AUML (Agent Unified Modelling Language). Enfin nous terminons ce document par une conclusion générale.

Chapitre 1

Agents et Systèmes Multi Agent

1.1 Introduction

A la différence de l'Intelligence artificielle classique (IA) qui modélise le comportement intelligent d'un seul agent, l'intelligence artificielle distribuée (IAD) s'intéresse à des comportements intelligents qui sont le produit de l'activité coopérative entre plusieurs agents. Le passage du comportement individuel aux comportements collectifs est considéré non seulement comme une extension, mais aussi comme un enrichissement de l'IA, d'où émergent de nouvelles propriétés et de nouveaux comportements. L'IAD s'intéresse entre autre à la modélisation de comportement intelligent entre plusieurs agents pour des activités coopératives, d'où la réalisation des systèmes dit « multi-agents ». Gerhard Weiss[Wei99] à donner une définition pour L'IAD assez large : « LIAD est, en partie, l'étude, la conception et la réalisation de systèmes Multi-Agents, c'est-à-dire de systèmes dans lesquels des agents intelligents interagissent, poursuivent un ensemble de buts ou réalisent un ensemble d'actions ».

Le concept d'agent est devenu très important dans le domaine de l'Intelligence Artificielle et plus généralement dans celui de l'informatique. Nous allons donc aborder, dans ce chapitre, des définitions et des concepts clés dans le domaine des SMA qui nous apparaissent les plus appropriés compte-tenu du point de vue que nous adoptons dans cette thèse.

1.2 Qu'est ce qu'un agent ?

Les agents ont été l'objet d'études pour plusieurs décennies et dans différentes disciplines. Le concept d'agent été non seulement utilisé dans les systèmes à base de connaissances, la robotique, le langage naturel et d'autres domaines de l'intelligence artificielle, mais aussi dans d'autres disciplines. Aujourd'hui, avec l'avènement de nouvelles technologies, ce concept est encore associé à plusieurs nouvelles applications comme agent ressource, assistant personnel, agent interface, agent ontologique, etc. En conséquence, il n'y a pas une définition acceptée en unanimité pour la notion d'agent, selon Nwana[Nwa96], il y a au moins deux raisons qui permettent d'expliquer cette difficulté.

La première réside dans le fait que les chercheurs, dans le domaine des agents, ne sont pas à l'origine de ce terme comme l'on été. La deuxième raison est que même dans la communauté des chercheurs sur les agents logiciels, le mot « agent » est utilisé pour décrire des systèmes très différents les uns des autres. Les chercheurs sont allés même jusqu'à inventer plusieurs synonymes au mot « agent », par exemple, « knowbots » (robots à base de connaissances), « softots » (robots logiciel), taskbots (robots à base de tâche). En ce qui suit, on présente quelques définitions plus importantes :

- Un agent est un processus cyclique composé de trois phases : perception – délibération – action [Pnu86] (Fig1-1)
- Un agent est un système informatique, situé dans un environnement et qui agit d'une façon autonome pour atteindre les objectifs (buts) pour lesquels il a été conçu [Woo95].
- Un agent est une entité autonome, réelle ou abstraite, qui est capable d'agir sur elle-même et sur son environnement, qui, dans un univers multi-agents, peut communiquer avec d'autres agents et dont le comportement est la conséquence de ses observations, de ses connaissances et des interactions avec les autres agents [Fer95]
- Un agent est une entité qui perçoit son environnement et agisse sur celui-ci [Rus97].
- Les agents intelligents sont des entités logiciels qui réalisent des opérations à la place d'un utilisateur ou d'un autre programme, avec une sorte d'indépendance ou d'autonomie et pour faire cela, ils utilisent une sorte de connaissance ou de représentation

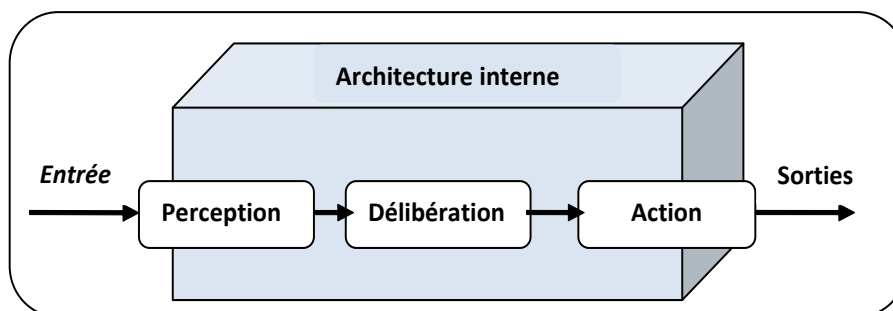


FIG. 1-1: L'agent, un processus à 3 phases : perception, délibération puis action

des buts ou des désires de l'utilisateur. (L'agent IBM).

En partant des définitions citées ci-dessus, on peut identifier les caractéristiques suivantes pour la notion d'agent :

- Situé : l'agent est capable d'agir sur son environnement à partir des entrées sensorielles qu'il reçoit de ce même environnement.
- Autonome : l'agent est capable d'agir sans l'intervention d'un tiers (humain ou agent) et contrôle ses propres actions ainsi que son état interne
- Proactif : l'agent doit exhiber un comportement proactif et opportuniste, tout en étant capable de prendre l'initiative au bon moment
- Capable de répondre à temps : l'agent doit être capable de percevoir son environnement et d'élaborer une réponse dans le temps requis.
- Social : l'agent doit être capable d'interagir avec d'autres agents afin d'accomplir des tâches ou aider ces agents à accomplir les leurs.

1.3 Typologie d'agents

La typologie se réfère à l'étude de type des agents en se basant sur les aspects : dynamique, capacité de raisonnement, caractéristiques et rôles. La dynamique (mobilité) concerne le mouvement des agents suivant un mode d'organisation ; on distingue les agents statiques et les agents mobiles. La capacité de raisonnement fait distinguer :

1. Les agents réactifs, qui sont les plus sommaires. Ils ont un comportement du type

« stimulus – réponse ». L'agent réactif ne possède pas une représentation complète de son environnement et n'est pas capable de tenir compte des actions passées. Les systèmes multi-agents constitués uniquement d'agents réactifs possèdent un grand nombre d'agents. La convergence du comportement de l'ensemble des agents vers un état décisionnel stable n'est pas forcément assurée et si un état stable est atteint, il n'est pas sûr qu'il s'agisse de la solution optimale.

2. Les agents cognitifs, sont plus évolués. Ils sont le résultat direct des recherches menés dans le domaine de l'intelligence artificielle. Les agents cognitifs ont une représentation globale de leur environnement et des autres agents avec lesquels ils communiquent. Ils savent tenir compte de leur passé et s'organisent autour d'un mode social d'organisation. Les systèmes multi-agents constitués uniquement d'agents cognitifs sont constitués d'un nombre d'agents assez faible. Ils réclament des ressources plus importantes que les systèmes d'agents réactifs. La convergence du système vers un état décisionnel stable n'est pas non plus assurée par l'utilisation de ce type d'agents, mais ils permettent de résoudre des problèmes plus complexes et nécessitant une plus grande abstraction
3. L'agent proactif, tendu vers la poursuite de ses buts.

L'agent peut posséder à divers degrés des caractéristiques comme l'autonomie, la coopération, l'apprentissage. La combinaison de deux ou plusieurs aspects caractérise les agents hybrides.

1.4 Qu'est ce qu'un Système multi-agents ?

Le système multi-agents peut être défini comme étant un ensemble d'agents. Ceci signifie que dans un SMA il existe une ou plusieurs organisations qui structurent les règles de cohabitation et de travail collectif entre les agents (définition des différents rôles, partage de ressources, dépendance entre tâches, ...). Les agents sont généralement situés dans un environnement contenant également des entités passives communément appelés objets.

En 1995, une approche intégrée (l'approche « Vowels ») des SMA est proposée par Yves Demazeau. Elle est basée sur une décomposition en quatre parties [Dem97].

- **Agents A**, qui concernent les modèles ou architectures utilisées pour la partie active de l'agent, depuis une simple automate jusqu'à un système complexe à base de connaissances.
 - **Environnement E**, qui sont des milieux dans lesquels évoluent les agents. Ils sont généralement spatiaux.
 - **Interaction I**, qui englobent les infrastructures, les langages et les protocoles d'interactions entre agents, depuis les interactions physiques jusqu'aux interactions par actes de langages.
 - **Organisation O**, qui structurent les agents en groupes, hiérarchies, relation, . . . , etc.
- De plus, cette approche est guidée par deux principes :

1. **L'approche déclarative**, qui peut se résumer à travers cette équation :

$$SMA = Agents + Environnement + Interactions + Organisation$$

2. **L'approche fonctionnelle**, d'un point de vue informatique, les fonctionnalités des SMA incluent les fonctionnalités individuelles des agents enrichies des fonctionnalités qui résultent de la valeur ajoutée par le système multi-agents lui-même, parfois appelée intelligence collective. Une deuxième équation peut alors être formulée :

$$Fonction(SMA) = \sum Fonctions(Agents) + \sum FonctionsCollectives$$

Pour Ferber, un système multi-agents doit être composé des éléments suivants[Fer95].

- **Un Environnement E**, c'est-à-dire un espace.
- **Un ensemble d'objets O**, Ces objets sont situés, c'est-à-dire que pour tout objet, il est possible, à un moment donné, d'associer une position dans **E**. Ils sont passifs, peuvent être perçus, détruits, créés et modifiés par les agents.
- **Un ensemble A d'agents**, qui sont des objets particuliers, lesquels représentent les entités.
- **Un ensemble de relation R**, qui unissent des objets (et donc des agents) être eux.
- **Un ensemble d'opérations Op**, permettant aux agents de percevoir, produire,

consommer, transformer et manipuler des objets. Cela correspond à la capacité des agents de percevoir leur environnement.

- **Des opérateurs** chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification (que l'on appelle les lois de l'univers).

Jennings donne un sens plus général au terme système multi-agents et l'utilise pour tous types de systèmes formés de plusieurs composants autonomes qui respectent les caractéristiques suivantes[Syc98].

- Chaque agent à un restreint de capacité et de ressource, donc plusieurs agents doivent interagir pour résoudre le problème général.
- Le système multi-agent est distribué. Il n'y a pas de contrôle central : chaque agent agit selon son objectif, il n'est pas contrôlé par les autres.
- Il est distribué les ressources : chaque agent s'occupe d'une partie de travail et il possède aussi une partie de ressource du système.
- Le traitement dans un système multi-agent est asynchrone. Chaque agent traite automatiquement son travail selon son plan. Il ne doit pas attendre les autres sauf qu'ils ont besoin d'interagir pour terminer leurs travaux.

Enfin, conformément au point de vue -généralement utilisé-, un système multi-agent est composé de différents agents qui interagissent, mais qui sont limités et différent au sein de leurs capacités sensorielles et cognitives aussi bien que dans leurs connaissances de l'environnement. Une vue canonique d'un SMA est présentée dans la Fig1-2.

1.5 L'Environnement dans les systèmes multi-agents

Selon [Pes97], l'environnement peut être considéré comme la représentation du monde dans lequel les agents se situent. L'environnement est modifiable par les agents, soit de façon globale, soit en faisant la distinction entre objets passifs (soumis aux actions des agents) et entités actives (les agents)[Fer95]. L'environnement d'un SMA fournit un

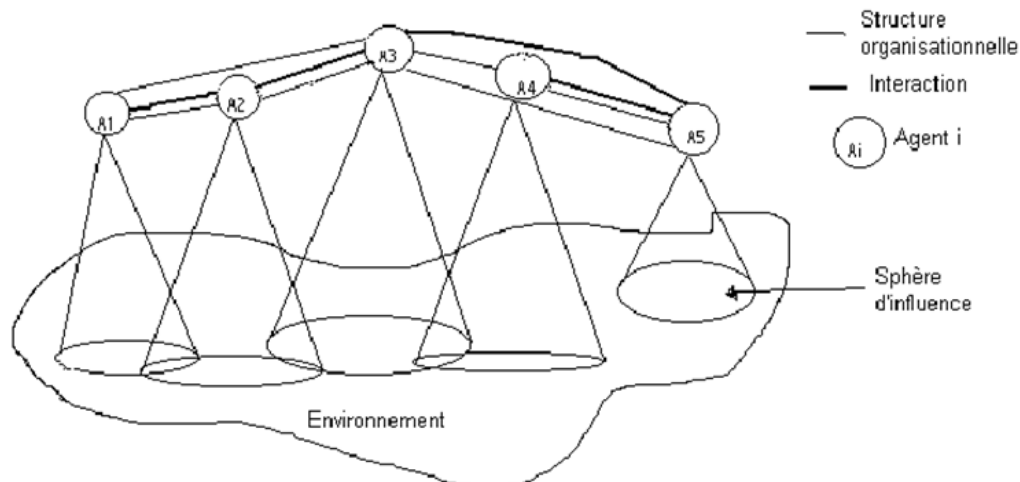


FIG. 1-2: Vue canonique d'un SMA[Woo00a]

support commun aux actions des agents, permettant ainsi l'interaction dans le système.

1.5.1 Différents types d'environnement

Dans [Rus03], les auteurs ont présenté un programme générique d'environnement (Fig1-3). Ce programme est trop générique pour capturer toutes les spécificités d'un environnement mais il illustre bien la relation entre les agents et leur environnement. La réalité maintenue par l'environnement est matérialisée par un état (la variable state) et l'environnement est décrit comme un processus cyclique en trois phases : (1) les perceptions sont données aux agents, ces perceptions sont construites par la fonction Get-Percept et dépendent de l'état de l'environnement et de l'agent lui-même, (2) les actions des agents sont collectées, un agent est ici représenté par un programme (Program[agent]) qui prend en paramètre des perceptions et retourne des actions, (3) l'état de l'environnement est mis à jour par la fonction Update-Fn qui prend en paramètre les actions des agents, les agents et l'état courant de l'environnement.

Russell et Norvig [RUS95] distinguent pour un environnement les propriétés suivantes :

- Accessible/Inaccessible : un environnement est dit accessible si les agents ont accès

```

Algorithme : RUN-ENVIRONMENT (state, UPDATE-FN, agents,
termination)
Input : state, l'état initial de l'environnement
          UPDATE-FN : la fonction de mise-à-jour de l'environnement
          agents, un ensemble d'agents
          termination, un prédicat pour tester la fin du programme

repeat
  foreach agent in agents do
    PERCEPT[agent] ← GET-PERCEPT(agent, state);
  end
  foreach agents in agents do
    ACTION [agent] ← PROGRAM[agent](PERCEPT[agent]);
  end
  state ← UPDATE-FN (actions, agents, state);
Until termination (state);

```

FIG. 1-3: Un programme générique de l'environnement [Rus03]

à l'intégralité de son état. L'accessibilité de l'environnement n'est pas une propriété souhaitable pour un SMA où la portée des actions et de la perception est locale.

- Déterministe/Indéterministe : un environnement est déterministe si un changement de son état est déterminé uniquement par son état courant et les actions des agents. Un environnement indéterministe pourra produire un résultat différent pour une même action.
- Statique/Dynamique : un environnement est dit statique s'il ne change d'état que sous l'effet des actions des agents. Au contraire, un environnement dynamique possède ses propres processus d'évolution qui peuvent modifier son état sans intervention des agents.
- Discret/Continu : l'environnement est discret lorsque le nombre de perceptions et d'actions possibles est limité. Si ce n'est pas le cas, l'environnement est alors continu.

1.5.2 Organisation

On peut définir une organisation comme une structure décrivant comment les membres de l'organisation sont en relation et interagissent afin d'atteindre un but com-

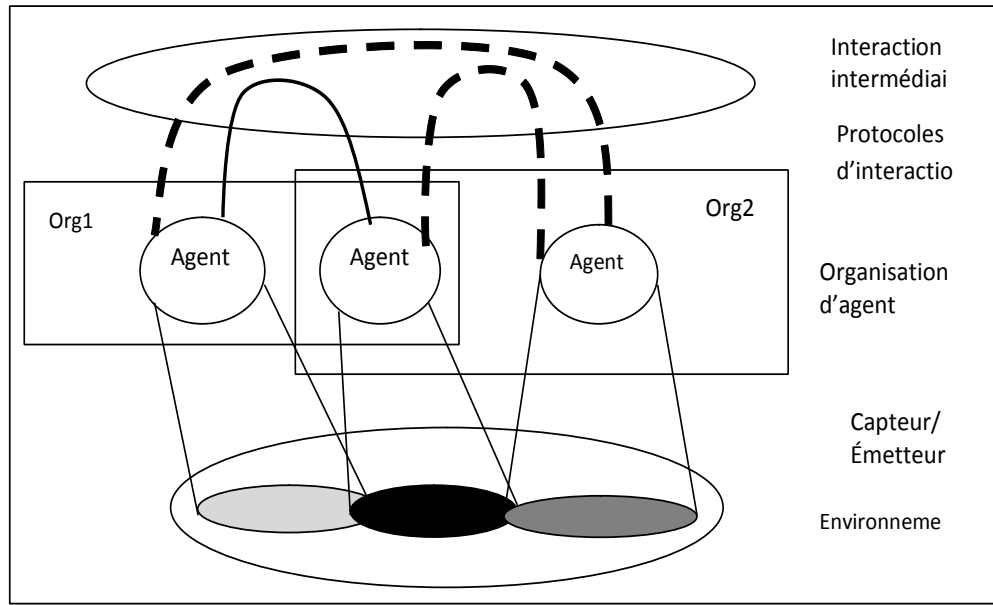


FIG. 1-4: Caractérisation d'un SMA [Zam00]

mun [FOX81]. L'autonomie et le comportement proactif des agents constituant les SMA suggèrent que la conception de ces applications peut être réalisée en imitant le comportement et la structure des organisations humaines, car l'une des missions principales des SMA est de supporter et de contrôler des organisations du monde réel. Le système peut être décomposé en des sous-organisations distinctes. Un agent peut jouer un ou plusieurs rôles tout en coopérant et en respectant ses sous-organisations. Selon [ZAM00], la perspective organisationnelle conduit à une caractérisation générale d'un SMA décrite dans la Figure 1-4.

1.6 Communication entre les agents

Une communication peut être définie comme une forme d'action locale d'un agent vers d'autres agents. Dans les SMA deux stratégies principales ont été utilisées pour supporter la communication entre agents : les agents peuvent échanger des messages directement ou ils peuvent accéder à une base de données partagées (appelée tableau noir ou "blackboard")

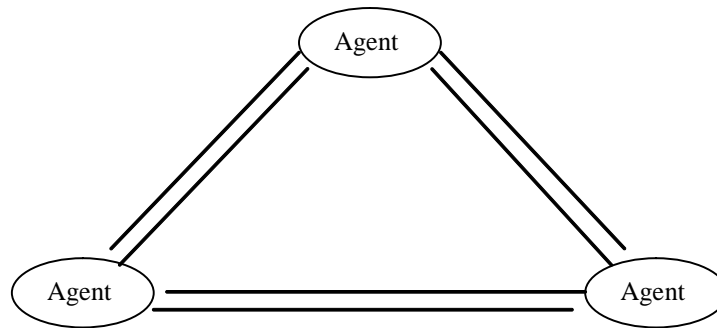


FIG. 1-5: Communication par envoi de messages

dans laquelle les informations sont postées.

1.6.1 Communication par envoi des messages

Dans cette approche la communication se fait à l'aide des messages qui sont transférés entre les agents qui veulent communiquer. Malheureusement ce type de communication souffre de plusieurs inconvénients, le transfert de messages est coûteux en ressources de communication et c'est une approche difficile à mettre en œuvre dans des applications réelles, car il y a en général un fort degré d'incertitude au sujet des états actuel et futurs du monde. Le transport des messages est un point important dans les SMA : les agents "dialoguent" uniquement par échange de message, donc les performances du transport de messages influencent directement les performances du SMA.

1.6.2 Communication par partage d'informations

En intelligence artificielle la technique du tableau noir ("blackboard") est très utilisée pour spécifier une mémoire partagée par divers systèmes. Ce mécanisme est constitué généralement de trois éléments principaux : (1) Les Sources de Connaissances qui est le résultat du partage des connaissances du domaine, (2) Le blackboard qui contient une description de l'état de la résolution sous formes d'entités appelées souvent faits et (3) Le Contrôle qui permet de choisir, parmi les sources de connaissances dont la partie condition

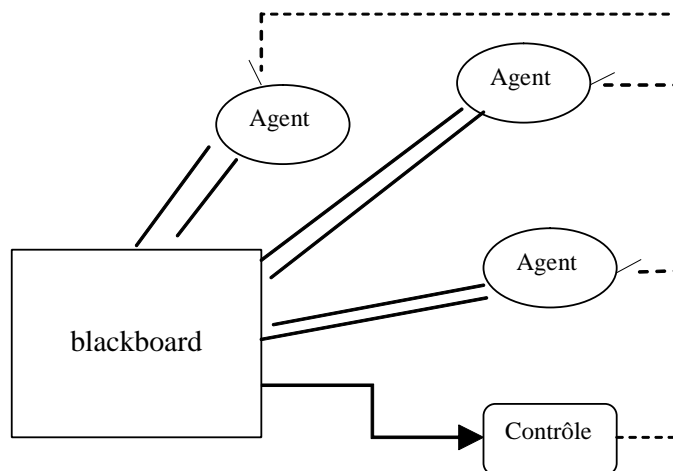


FIG. 1-6: Communication par partage d'informations

est vérifiée, celle dont la partie action sera exécutée, créant ainsi de nouvelles hypothèses dans le blackboard et permettant à la résolution de se poursuivre.

1.6.3 Les langages de communication

Le langage **KQML (Knowledge Query and Manipulation Language)** [Fin93] a été développé au sein du groupe de travail «External Interfaces Group» du projet «Knowledge Sharing Effort» financé par l'ARPA («Advanced Research Projects Agency») [Pat92]. Ce langage peut être considéré comme composé de trois couches : une couche de communication qui décrit le niveau des paramètres de communication ; une couche de message qui contient un performatif et indique le protocole d'interprétation (les performatifs indiquent quel type d'acte de langage l'agent désire employer et définissent les opérations possibles que les agents peuvent demander sur les connaissances et les objectifs des autres agents [Pat92]) ; et une couche du contenu qui contient des informations relatives au performatif soumis. Pour plus de détails vous pouvez consulter [Fin93].

Même si à sa parution KQML a servi à l'implémentation de beaucoup d'applications à base d'agents cognitifs, plusieurs critiques sévères ont été faites à son sujet. La principale critique concerne l'imprécision de sa sémantique et la non-fermeté de sa spécification. De

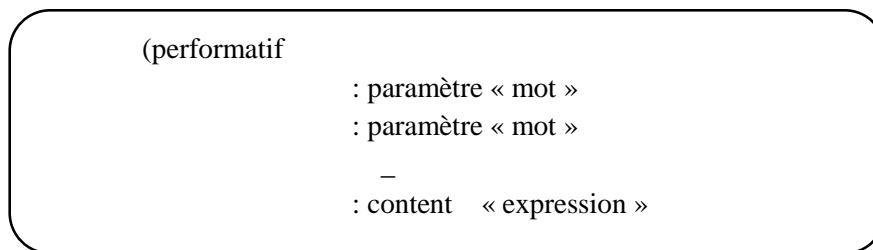


FIG. 1-7: Structure globale d'un message KQML

plus le langage ne possède pas du tout de performatifs concernant les « commissifs » ou « promissifs » (promettre de faire quelque chose pour un autre agent et gérer de telles promesses). Et enfin, l'ensemble des performatifs est vaste et surtout redondant.

FIPA-ACL (FIPA Agent Communication Language), est proposé dans le cadre d'un travail de standardisation mené au sein l'organisation FIPA (Foundation of Intelligent Physical Agents). FIPA-ACL est une extension du langage KQML. FIPA-ACL est fondé sur (21) vingt et un actes communicatifs, exprimés par des performatifs, qui peuvent être groupés selon leurs fonctionnalités de la façon suivante :

- Passage d'information : Inform, Inform-if, Inform-ref, Confirm, Disconfirm,
- Requisition d'information : Query-if, Query-ref, Subscribe,
- Négociation : Accept-proposal, Cfp, Propose, Reject-proposal,
- Distribution de tâches (ou exécution d'une action) : Request, Request-when, Request-whensoever, Agree, Cancel, Refuse.
- Manipulation des erreurs : Failure, Not-understood.

Un message FIPA-ACL peut contenir une partie ou la totalité des éléments décrits dans le tableau 1.1. Les éléments nécessaires pour la transmission d'un message changent selon la situation. Si un agent ne reconnaît pas ou ne peut pas traiter un ou plusieurs éléments, alors il peut répondre avec le message Not-understood.

Elément	Signification
performatif	Le type de l'acte communicatif
sender	l'émetteur du message
receiver	Le destinataire du message
reply-to	Le participant à l'acte de communication
content	Le contenu du message
language	Le langage dans lequel le contenu est représenté
encoding	Décrit le mode d'encodage du contenu du message
ontology	Le nom de l'ontologie utilisé
Protocol	Le nom du protocole d'interaction
conversation-id	L'identifiant de la conversation
reply-with	Identifiant du message, en vue d'une référence ultérieure
in-reply-to	Identifiant du message entrain de se répondre
reply-by	Un délai pour répondre au message

TAB. 1.1: Les primitives pour d'écrire un message ACL

1.7 Les protocoles d'interaction dans les systèmes multi agents

Une des principales propriétés de l'agent dans un SMA est celle d'interagir avec d'autres agents. Ces interactions sont généralement définies comme toute forme d'action exécutée au sein du système d'agents et qui a pour effet de modifier le comportement d'un autre agent. On appellera situation d'interaction un ensemble de comportements résultant du regroupement d'agents possédant des compétences particulières et qui doivent agir pour satisfaire leurs objectifs en tenant compte des contraintes provenant des ressources plus ou moins limitées dont ils disposent [Erc91]. Il existe différents types de protocoles d'interaction, on cite :

- Les protocoles de coordination,
- Les protocoles de coopération,
- Les protocoles de négociation, ainsi que
- les mécanismes du commerce électronique.

1.7.1 Les protocoles de coordination

La coordination se traduit (dans des environnements à ressources limitées) par un comportement individuel visant à servir ses propres intérêts tout en essayant de satisfaire le but global du système. Elle est caractérisée par deux aspects étroitement liés, à savoir les engagements et les conventions [Jen93]. Les engagements fournissent la structure nécessaire pour des interactions prévisibles. Pendant que les situations changent, les agents doivent évaluer si les engagements existants sont encore valides. Les conventions fournissent des moyens pour contrôler les engagements dans des circonstances changeantes.

1.7.2 Les protocoles de coopération

On dit que deux agents sont en situation de coopération si les conditions suivantes surviennent :

1. Ils poursuivent chacun des buts qui peuvent entrer en interférence, soit au niveau des résultats, soit au niveau des procédures.

2. Ils font en sorte de traiter ces interférences pour que les activités de chacun soient réalisées de façon à faciliter la réalisation de celles de l'autre ou la réalisation de la tâche commune (si elle existe).

Les protocoles de coopération consistent à décomposer un problème en tâches puis à les distribuer [Dur90]. Cette approche a l'avantage de réduire la complexité d'un problème. Mais il risque d'avoir des interactions entre les tâches et par conséquent des conflits entre les agents. Il existe plusieurs mécanismes pour distribuer les tâches. On cite les mécanismes d'élection où les tâches sont attribuées à des agents suite à un accord ou un vote, les réseaux contractuels où des tâches sont attribuées aux agents suite à des cycles d'appels d'offres ou de propositions.

1.7.3 Les protocoles de négociation

Les protocoles de négociation sont utilisés dans le cas où les agents ont des buts différents. Les dispositifs principaux de la négociation sont : le langage utilisé, le protocole suivi dans le principe de négociation et la procédure de décision que chaque agent utilise pour déterminer ses positions, ses concessions et ses critères pour l'accord. Plus de détails dans les chapitres suivants.

1.7.4 Les mécanismes du commerce électronique

Les mécanismes du commerce électronique exigent de l'organisation un nombre important de communications. Ils présentent l'avantage d'être simple, équitable, distribué et très utile dans le cas où il s'agit de décider une action ou une solution parmi plusieurs proposées. Une autre approche consiste à utiliser de mécanismes de vente appelés aussi les ventes aux enchères.

1.7.5 Les protocoles de FIPA

FIPA spécifie un environnement d'existence et de fonctionnement des agents ainsi qu'une infrastructure physique de déploiement des agents. FIPA définit une plate-forme

des agents qui représente l'infrastructure dans laquelle les agents peuvent être déployés, un système de gestion des agents décrit tel les agents peuvent gérer la création, la suppression, la suspension, la reprise, l'authentification et la migration des autres agents tout en assurant un service des pages blanches ainsi qu'un service des pages jaunes.

Le protocole FIPA-request

Il permet à un agent d'inviter d'autres agents à exécuter une certaine action. L'agent récepteur peut refuser ou accepter la tâche. S'il l'accepte, il doit exécuter l'action et informer l'agent qui l'a demandée.

Le protocole FIPA-query

Dans ce protocole, l'agent cherche à connaître un objet correspondant à une description ou interroger un autre agent sur la proposition ou l'état sur l'exécution d'une autre action.

Protocole FIPA-request-when

Ce protocole est utilisé par les agents voulant demander à d'autres agents d'exécuter une certaine action à l'avenir une fois la condition préalable donnée devient vraie. Si l'agent destinataire accepte la demande, il attend l'état vrai de la condition préalable pour exécuter l'action puis il informera l'agent expéditeur de demande que l'action a été exécutée.

Le protocole FIPA-contract-net

L'agent prend ici le rôle du manager. Le manager souhaite faire accomplir une certaine tâche par un ou plusieurs autres agents, et optimiser une fonction qui caractérise la tâche. Cette caractéristique est généralement exprimée comme coût, spécifique au domaine(Fig1-8).

Le protocole FIPA-Iterated-contract-net

C'est une extension du protocole précédent. Il diffère de la version de base en permettant plusieurs itérations. Une fois que l'agent Manager reçoit les propositions des contractants, il peut accepter une ou plusieurs offres, rejeter d'autres ou peut réitérer le processus en émettant une nouvelle offre 'révisée'. L'objectif est de permettre au manager d'obtenir de meilleures offres des contractants en modifiant l'appel et en demandant de nouvelles offres

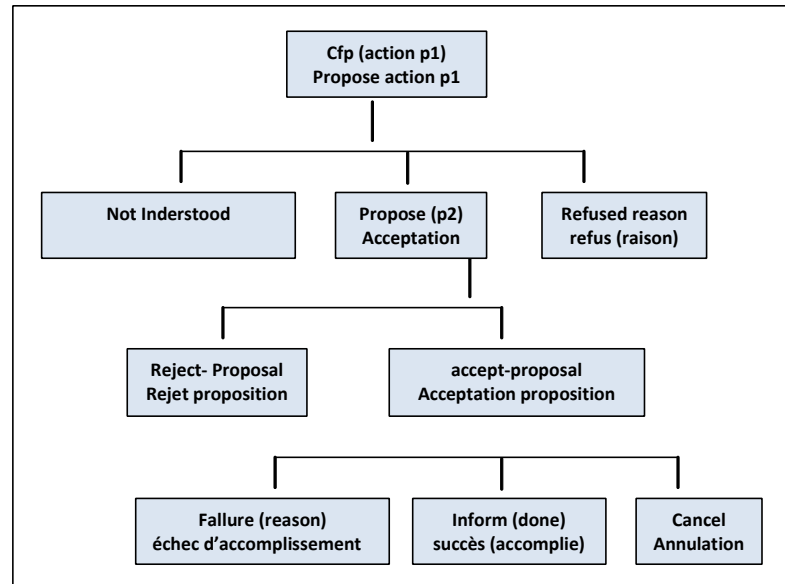


FIG. 1-8: Le protocole FIPA-Contract-Net

(d'une manière équivalente mais révisée).

Le protocole FIPA-Auction-English

Dans l'enchère anglaise, le manager cherche à trouver le prix du marché d'une marchandise en proposant initialement un prix au dessous de celui de la valeur de la marchandise supposée, et en augmentant alors graduellement le prix. Chaque fois que le prix est annoncé, le manager attend des acheteurs pour signaler leur accord sur le prix proposé. Dès qu'un acheteur indique qu'il accepte le prix, le manager émet un nouvel appel d'offres avec un prix légèrement supérieur. L'enchère continue jusqu'à ce qu'aucun acheteur ne soit disposé à payer le prix proposé, auquel cas l'enchère termine. Si le dernier prix qui a été reçu par un acheteur excède un prix de réserve (connu en privé par le manager), la marchandise est vendue à cet acheteur pour le prix convenu. Sinon la vente est annulée.

Le protocole FIPA-Auction-Dutch :

Dans l'enchère hollandaise, les tentatives du manager de trouver le prix de vente d'une marchandise se concrétisent en commençant par un prix beaucoup plus haut que la valeur prévue, puis en réduisant progressivement le prix jusqu'à ce qu'un des acheteurs

accepte le prix. Le manager a habituellement un prix de réserve au dessous duquel il ne vend pas. Si l'enchère ramène le prix de la marchandise au prix de réserve sans qu'il ait d'acheteurs, l'enchère se termine. Dans cette enchère, la marchandise peut être vendue en plusieurs parties, quelques marchés obligent d'acheter la marchandise toute entière.

1.8 Utilités des systèmes multi-agents

Certains domaines sont fonctionnellement distribués et par conséquent les systèmes multi agents constituent la façon adéquate de les modéliser. Les systèmes multi agents sont également requis dans les situations où lorsque les différents systèmes et les données qui s'y rattachent appartiennent à des organisations indépendantes qui veulent garder leurs informations privées et sécurisées pour des raisons concurrentielles. Ainsi, ils peuvent s'avérer bien utiles pour des problèmes possédant de multiples méthodes de résolution, de multiples perspectives et de multiples solveurs. En particulier, les systèmes multi-agents sont très utiles pour modéliser le raisonnement humain à l'intérieur de grandes simulations de combats aériens[Cli98]. Les systèmes multi-agents possèdent également les avantages traditionnels de la résolution distribuée et concurrente de problèmes, en particulier la modularité, le parallélisme et la fiabilité[Dig00].

On peut aussi citer quelques problèmes liés aux systèmes multi-agents, selon Sycara [Ber99] les six défis du SMA sont :

- La décomposition des problèmes et la distribution des tâches aux agents individuels.
- La coordination des communications entre les agents.
- Faire agir aux agents d'une manière cohérente.
- L'augmentation du nombre des agents et des états de la coordination (l'explosion).
- La conciliation des conflits de but entre les agents
- Le côté génie du système multi-agent

De nos jours, la technologie multi-agent a trouvé sa place dans les systèmes manufacturiers, les systèmes financiers, les loisirs, les télécommunications, le contrôle-commande, les systèmes embarqués, et pas mal d'autres applications. Dans ce qui

va suivre nous n'en exposerons que quelques exemples d'applications utilisant cette technologie.

Exemple d'application : Négociation automatique en commerce électronique

Aujourd'hui, il y a de plus en plus vendeurs sur Internet, leurs produits sont nombreux. Afin d'aider les acheteurs à acheter le produit qui leurs satisfait le plus, on veut un système dans lequel il y a des composant dont chacun joue le rôle d'acheteur pour contacter et négocier automatiquement avec un vendeur pour trouver un produit qui satisfait aux intérêts du consommateur. Puis, les composants doivent collaborer pour sélectionner le produit final qui convient le plus aux intérêts du consommateur. Dans ce cas, on considère aussi chaque composant comme un agent et le système comme un système multi-agent.

1.9 Conclusion

On a présenté dans ce chapitre tous les aspects relatifs au concept agent et système multi-agent, et d'après ce chapitre la technologie agent et multi-agent n'est pas un concept voué à rester sur les tablettes des laboratoires de recherche puisque plusieurs exemples d'applications existent déjà. Comme on peut encore le constater le domaine des systèmes multi-agents demeure encore aujourd'hui un domaine rempli de défis à surmonter, autrement dit un domaine très ouvert pour la recherche.

Chapitre 2

L'ingénierie logicielle orientée agents

2.1 Introduction

Malgré le développement de différentes théories, de langages et d'architectures d'agents, très peu de travaux se sont intéressés aux techniques de développement utilisant la technologie des agents. De ce fait, un nouveau thème de recherche, celui de l'ingénierie logicielle orientée agents, s'est développé ces dernières années. L'objectif principal de l'ingénierie logicielle orientée agents, est de développer des méthodologies et des outils qui facilitent le développement et la maintenance des applications multi-agents [Tve01]. Nous trouvons dans le domaine de l'ingénierie logicielle, trois activités de recherche [Cer05] : (i) le développement d'outils conceptuels (outils de notation ou de modélisation formelle), (ii) le développement des outils d'assistance au cours du processus de développement (les CASE tools) et (iii) les méthodologies multi-agents. Le but de ce chapitre est de répondre aux deux questions suivantes : comment peut-on modéliser un système sous forme d'un système multi-agent ? et quel est l'outil efficace nous aidons à le faire ?

2.2 Méthodologies

Il y a plusieurs recherches qui s'intéressent à la méthodologie d'analyse et de conception orientées agent [Del01]. À ce jour, il n'y a pas encore une méthode standard puisque

chaque groupe de recherche propose une méthode différente. Chaque méthode convient à un groupe particulier des applications.

Une méthodologie est caractérisée généralement par :

- La définition des étapes du projet, des guides de conception pour chacune et les successions possibles entre elles.
- Un langage commun de description ou de modélisation pour faciliter l’expression du problème et la réutilisation éventuelle.
- Le choix d’un niveau de description adéquat et les conséquences opérationnelles qui en découlent.
- Un ensemble structuré de directives, qui inclut la définition des étapes, des conseils de conception pour chacune des étapes, et des règles de transition entre ces étapes.
- Une façon unifiée et standardisée de documenter le processus de conception (elle est utilisée pour partager et transmettre l’expérience acquise durant ce processus).
- L’utilisation d’une terminologie homogène, qui possède une signification à chaque étape et qui facilite les transitions entre étapes.
- L’utilisation d’abstractions conceptuelles opérationnelles, c.-à-d. de structures suffisamment abstraites pour permettre un choix suffisant de techniques au moment de l’implémentation, mais assez concrètes pour éviter au concepteur d’utiliser des techniques dépassées ou non pertinentes.

2.2.1 MAS CommonKADS

MAS CommonKADS [Igl97] (Multi-agent System – Knowledge Analysis and Development System) est étendu de la méthodologie de génie connaissance en prenant les techniques orientées objet et celles de la méthodologie de génie protocole. Cette méthode est une combinaison de CommonKADS[Igl98], OOSE (Object Oriented Software Engineering), OMT (Object Modeling Technique) et des protocoles d’agents SDL (Specification and Description Language) et MSC (Message Sequence Charts). La méthodologie MAS-CommonKADS prévoit une phase de conceptualisation qui est dédiée à recueillir les besoins

de l'utilisateur et à obtenir une première description du système. Ensuite, la méthodologie définit les modèles décrits ci-dessous pour l'analyse et la conception d'un système. Pour chaque modèle, la méthodologie définit les constituants (les entités devant être modélisées) et les rapports entre ces constituants. MAS-CommonKADS définit un spécimen textuel pour décrire chaque constituant et une série d'activités pour construire chaque modèle, basées sur l'état de développement de chaque constituant.

- Le Modèle d'Agent : décrit les caractéristiques principales des agents, y compris les capacités de raisonnement, les compétences, services, les buts.
- Le Modèle d'Expertise : décrit les connaissances nécessaires aux agents pour exécuter les tâches.
- Le Modèle de coordination : permet d'écrire les conversations entre les agents, c.-à-d. leurs interactions, protocoles et les capacités d'interaction exigées.
- Le Modèle de tâche : décrit les tâches (les buts) devant être exécutés par les agents, et la décomposition des tâches, en utilisant les spécimens textuels et les diagrammes.
- Le Modèle de Communication : détaille les interactions entre l'agent humain et le logiciel, et les facteurs humains nécessaires au développement de ces interfaces utilisateur.
- Le Modèle d'Organisation : décrit l'organisation dans laquelle le SMA sera introduit et l'organisation de la société d'agents.
- Le Modèle de conception, ce modèle est subdivisé en trois sous-modèles :
 1. La conception de l'application : la composition ou la décomposition des agents d'analyse, selon des critères pragmatiques et la sélection de l'architecture d'agent la plus convenable pour chaque agent.
 2. La conception de l'architecture : concevoir les aspects pertinents du réseau d'agents.
 3. La sélection de la plate-forme de développement pour chaque architecture d'agent.

2.2.2 MaSE (Multi-agent System Engineering).

Pour Mase les agents sont des objets ayant l'intelligence, elle permet de transformer les exigences de l'utilisateur aux modèles qui décrivent le type d'agent. Cette méthodologie utilise les techniques de OMT (Object Modeling Technique) ou de UML (Unified Modeling Language) avec quelques caractéristiques de plus et quelques modifications de la sémantique du paradigme objet pour pouvoir capter les concepts d'agent et les comportements coopératifs des agents. Elle utilise deux langages pour décrire les agents et les SMA : Agent Modeling Language (AgML) et Agent Definition Language (AgDL). Elle contient des étapes (voir la Figure2-1) :

- Captation des buts : permet d'identifier et de partitionner les buts du système. Puis, elle les organise sous forme hiérarchie dans un diagramme.
- Description des cas d'utilisation : on extrait (identifie) les scénarios qui représentent le comportement du système dans les cas spécifiques. Ensuite on écrit chaque scénario d'une manière détaillée.
- Construire l'ontologie : construit la connaissance dans le domaine d'application du système en modélisant les informations échangées dans les scénarios.
- Identifier les rôles : on identifie les rôles du système. Chaque but est capté par au moins un rôle et réciproquement. Puis on détermine les relations entre ces rôles en basant sur les diagrammes des scénarios.
- Représentation des tâches : décrit (par un diagramme d'état) les tâches qu'on a besoin de réaliser pour obtenir le but.
- Création des classes d'agent : identifie les agents en basant sur les rôles. Chaque rôle est joué par au moins un agent et un agent doit jouer au moins un rôle. Les relations entre les agents correspondent aux relations entre les rôles qu'ils jouent.
- Construire des conversations : décrit les conversations entre les agents. Chaque conversation est un protocole de coordination entre deux agents

Cependant, cette méthode a des points faibles [Del05] :

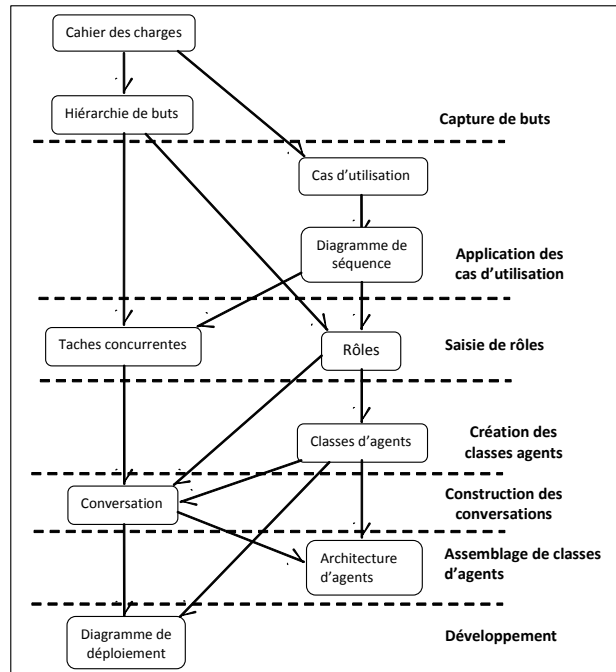


FIG. 2-1: Schéma générale de la méthodologie MaSe

- Elle construit un système fermé dans lequel les agents jouent un nombre de rôles limités et la capacité de changement de rôle est aussi limitée.

- Elle ne permet pas d'intégrer des sous-systèmes à un système multi-agent. Les agents interagissent au même niveau : il n'y a pas de hiérarchie dans le modèle d'interaction entre les agents.

Ainsi, la méthodologie O-MaSE [Del05] (Organization – based Multi-agent System Engineering) est élargie de MaSE, donc elle combine des avantages de MaSE et ceux du génie organisationnel, le but de cette méthode est de construire une société organisationnelle des agents. Elle considère le système multi-agents comme une organisation des agents dans laquelle les agents sont membres. Chaque agent joue un rôle spécifique selon ses capacités pour obtenir ses objectifs.

2.2.3 La méthodologie GAIA

Selon [Woo00b], GAIA est le nom donné aux hypothèses formulées par l'écologiste James Lovelock selon lesquelles les organismes vivants du monde peuvent être compris comme des composantes d'une seule entité, lesquelles composantes régularisent l'environnement mondial.

La méthodologie Gaia est centrée sur deux aspects fondamentaux des agents : spécifier les macro-aspects (aspects sociaux), et les micro-aspects (comportement interne de l'agent). La méthodologie elle-même suit une progression de décomposition du problème du haut-en-bas, qui va des aspects les plus visibles et abstraits de la société d'agents, et qui mène progressivement vers une définition plus détaillée et concrète de chaque élément.

La figure 2-2 présente une vue générale des activités définies par les auteurs de la méthodologie. La première partie, le cahier de charges, est laissée sans définition et au choix du développeur. Ce qui est raisonnable si on considère qu'ils existent plusieurs approches pour sa génération, et que ce document exprime tout simplement, une liste de tout ce qui doit faire le système à développer. Les systèmes auxquels on peut appliquer cette méthodologie contiennent un petit nombre d'agents (moins de 100). Les auteurs de GAIA considèrent un SMA comme une société ou une organisation artificielle ou encore un ensemble institutionnalisé de rôles.

GAIA se compose de deux phases principales, la phase d'analyse qui produit deux modèles : le modèle de rôles et le modèle d'interactions et la phase de conception qui transforme les modèles abstraits de la phase d'analyse en modèles moins abstraits. Le processus de conception se base sur les trois modèles d'agent, modèle de service et modèle de relation.

2.2.4 La méthodologie Aalaadin(ou AGR : Agent-Groupe-Rôle)

Comme son nom l'indique, cette méthodologie repose sur trois notions qui sont : le rôle, groupe et l'agent (Fig 2-3). Ce modèle support conceptuellement la plateforme Madkit qui sera présentée ultérieurement.

Un agent dans cette méthodologie peut être défini comme une entité qui joue des rôles

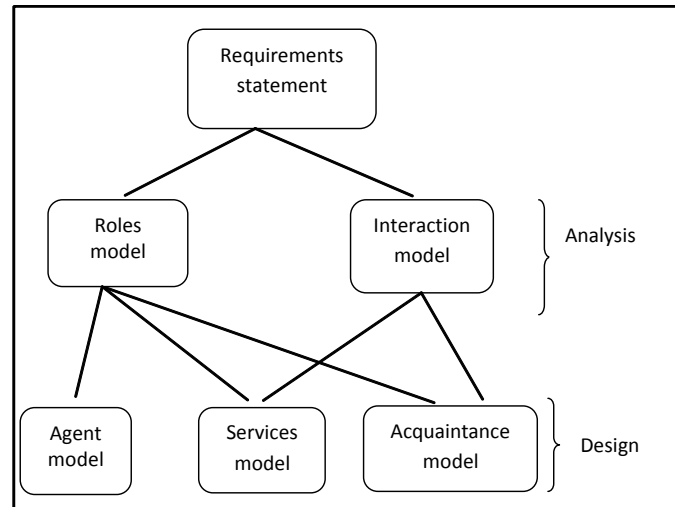


FIG. 2-2: Vue générale de phases de GAIA[Woo00b]

(un ou plusieurs) dans un groupe. Le mécanisme de cette méthodologie ne s'intéresse pas sur la nature (l'architecture interne) des agents puisque le modèle d'agent utilisé est laissé au porté des développeurs. Un groupe est constitué d'un ensemble d'agents, ainsi les groupes peuvent représentés l'idée de but commun.

Les rôles sont une représentation abstraite de la fonctionnalité d'un agent. Cette méthode repose sur trois étapes illustrées ci-dessous :

- L'analyse, permet de définir les principales fonctions dans nos systèmes, ainsi que les dépendances .
- la conception, qui contient l'identification des groupes et des rôles dans des diagrammes de structures organisationnelles.
- La réalisation, correspond à la gestion des entités du domaine qui permet d'implanter le système à partir d'une organisation concrète.

D'après tout ce qui est présenté ci-dessus, nous pouvons conclure que les méthodes orientées agents modélisent le système sur deux aspects : la modélisation des composants du système : les agents, les rôles et les buts, etc. Ces composants sont modélisés par des modèles correspondant (le modèle des agents, le modèle des rôles et le modèle des buts), et la

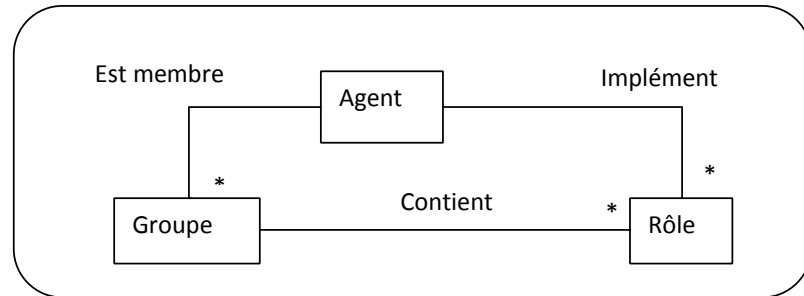


FIG. 2-3: Relation entre agents, groupes et rôles [Jac98a] [Jac98b]

modélisation des interactions (protocole) entre les agents dans le système, ils correspondent au modèle des protocoles (soit être nommé coordination, conversation, interaction) et au modèle d'ontologie (soit être nommé domaine, connaissance, environnement).

2.3 Outils

2.3.1 AUML (Agent based Unified Modeling Language)

La notation standardisée UML dans sa première version peut se révéler insuffisante pour modéliser des agents et des systèmes basés sur des agents pour trois raisons : premièrement, comparé aux objets, les agents sont actifs, car ils peuvent prendre l'initiative dans le contexte d'une interaction. Deuxièmement, les agents n'agissent pas seulement d'une façon isolée, mais en coopération et coordination avec d'autres agents et troisièmement ils communiquent plus fréquemment de façon asynchrone. Pour essayer de combler les possibles déficiences d'UML, il a été proposé une extension d'UML aux concepts agents appelé AUML [kae00]. AUML est justifié vraisemblablement par le fait qu'UML pourrait inclure des spécifications plus riches pour les agents, en proposant des modifications du format de quelques artefacts standard, mais en gardant un certain 'esprit' ou approche visuel facilement compréhensible par n'importe quel développeur habitué à UML.

Alors, les diagrammes de UML deviennent des diagrammes qui peuvent représenter les agents et les interactions entre ces agents, les objets sont remplacés par des agents et des

rôles, pour des interactions complexes des notations sont ajoutées (AND, OR, XOR).

Cependant, selon [Pau03], AUML présente les limitations ci-après :

- Les diagrammes sont désordonnés et peuvent être mal interprétés.
- L’expression de toutes les informations nécessaires sur les protocoles peut les rendre illisibles.
- Les cas de redondance sont difficiles à identifier et à corriger.
- La description des actions temporelles (telles que le timeout, deadline, ...) est difficile à exprimer.
- La terminaison des interactions n’est pas toujours spécifiée.

2.3.2 AgentTool

Est un environnement graphique basé-java, développer pour aider des utilisateurs à analyser, concevoir et mettre en application les systèmes multi-agent, il est associé à la méthodologie MaSE. Il fournit des diagrammes pour représenter les modèles correspondant aux étapes de MaSE, la hiérarchie des buts, la description des cas d’utilisation, le diagramme séquence, le diagramme des rôles, le diagramme des tâches, le modèle d’ontologie, le diagramme des agents, le diagramme des conversations et le diagramme d’exploitation.

La conception de système définit les types d’agents dans le système aussi bien que les communications possibles qui peuvent avoir lieu entre les agents. Ces spécifications au niveau système sont alors raffinées pour chaque type d’agent. Pour raffiner un agent, le concepteur choisit de créer une architecture pour ce dernier puis fournit des spécifications comportementales détaillées pour chaque composant dans l’architecture d’agent. Une fois que le système a été complètement indiqué, le concepteur produit du code. C’est en ce moment que le concepteur définit réellement le cadre fondamental et tous les protocoles spécifiques de communication et de sécurité. Ceux-ci sont automatiquement construits dans le code pendant la synthèse du système. Pour tirer profit des composants de logiciel préexistants, l’agentTool emploie également une bibliothèque des composants. Chaque composant à une définition formelle qui permet à un concepteur d’agentTool de déterminer si le composant

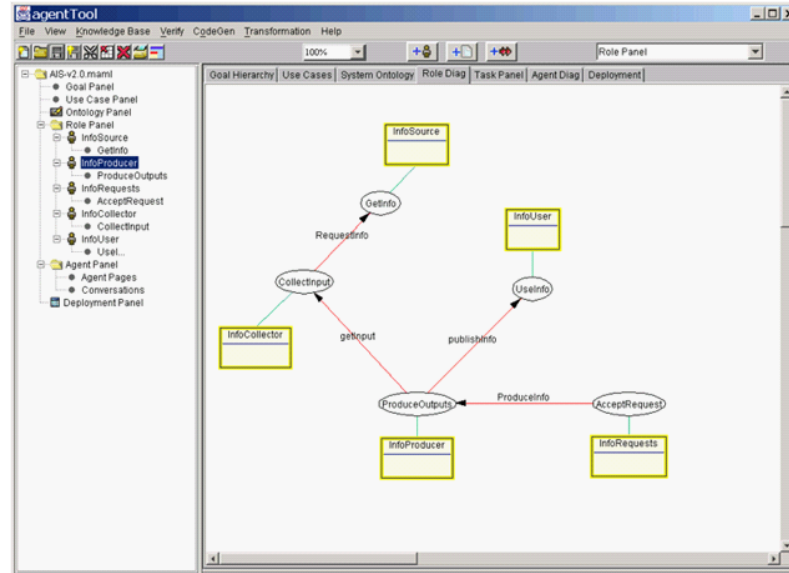


FIG. 2-4: L'environnement AgentTool

répond aux exigences de l'agent indiqué.

Parmi les points qui peuvent être comptés pour AgentTool, celle décrit ci-dessous,

- La vérification des diagrammes d'états.
- De vérifier si les deux diagrammes d'une conversation sont correspondants.
- De générer automatiquement le code selon le modèle conçu des agents. C'est le code préliminaire, le programmeur doit le compléter après.

2.3.3 JADE

JADE (Java Agent DEvelopment framework) est une plate-forme multi-agent créé par le laboratoire TILAB et décrite par Bellifemine dans [Bel00]. JADE permet le développement de systèmes multi-agents et d'applications conformes aux normes FIPA. Elle est implémentée en JAVA et fourni des classes qui implémentent « JESS » pour la définition du comportement des agents. JADE possède trois modules principaux (nécessaire aux normes FIPA).

- DF « Director Facilitator » fournit un service de « pages jaunes » à la plate-forme.

- ACC «Agent Communication Channel » gère la communication entre les agents.
- AMS « Agent Management System » supervise l’enregistrement des agents, leur authentification, leur accès et l’utilisation du système.

Ces trois modules sont activés à chaque démarrage de la plate-forme. JADE offre un ensemble de packages, parmi lesquels le package `jade.proto` qui est un ensemble de classes qui modélisent un certain nombre de protocoles d’interaction standardisés (Fipa-Request, Fipa-Query, Fipa-Contract-Net, Fipa-subscribe et d’autres protocoles définie par FIPA). Cette API aide le programmeur à utiliser les protocoles d’interaction. Les messages sont décrits avec le langage FIPA-ACL. Le package `jade.lang.acl` contient des classes qui implémentent les performatifs de FIPA-ACL.

Pour atteindre ce but, JADE offre les caractéristiques suivantes :

- La plate-forme multi-agents compatible FIPA, qui inclut l’AMS, DF, ACC.
- La plate-forme d’agents distribuée, cette dernière peut être distribuée sur plusieurs machines, à condition qu’il n’y ait pas de pare-feu entre ces machines. Une seule application Java (Machine Virtuelle Java) est exécutée sur chaque machine.
- Un certain nombre de DF.
- Une interface de programmation pour simplifier l’enregistrement de services d’agents avec un ou plusieurs domaines.
- Un mécanisme de transport et une interface pour l’envoi et la réception des messages de et vers les autres agents.
- Le protocole IIOP compatible avec le document FIPA97 pour connecter les différentes plates-formes multi-agents.
- Le transport léger de messages ACL sur la même plate-forme d’agents.
- Une interface graphique utilisateur pour gérer plusieurs agents et plates-formes multi-agents en partant d’un agent unique.

2.3.4 Madkit

Est une plate-forme développée par le Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM) de l'Université Montpellier II. MADKIT est libre pour l'utilisation dans l'éducation. MADKIT est écrit en Java et est fondé sur le modèle organisationnel Aalaadin. Il utilise un moteur d'exécution où chaque agent est construit en partant d'un micro-noyau. Chaque agent a un rôle et peut appartenir à un groupe. Il y a un environnement de développement graphique qui permet facilement la construction des applications. MadKit n'impose aucune architecture particulière aux agents. Il est ainsi possible de développer aussi bien des applications avec des agents réactifs que des agents cognitifs et même de faire interagir aisément tous ces types d'agents. Cela permet ainsi aux développeurs d'implémenter l'architecture de leur choix. Il est ainsi possible de faire communiquer des agents à distance sans avoir à se préoccuper des problèmes de communication qui sont gérés par la plate-forme.

Une application MadKit peut s'exécuter en mode distribué sans qu'il y ait besoin de modifier une seule ligne de code. Le mécanisme de distribution est indépendant de MadKit et il est possible d'en créer de nouveaux si les besoins s'en font sentir ou même d'utiliser des plateformes distribuées existantes, telles que JXTA. Le micronoyau de MadKit est très petit (moins de 100Ko de code), car il ne gère que les organisations (groupes et rôles) et les communications à l'intérieur de la plate-forme.

2.4 Conclusion

Ce chapitre a présenté l'état de l'art sur les méthodologies de modélisation de système multi-agent et sur les outils aidant à analyser et concevoir le système multi-agent. Dans les dernières années, ce domaine a attiré de plus en plus les chercheurs. Il y avait plusieurs méthodes et outils proposés pour modéliser le système multi-agent. Cependant, à l'instance, il n'y a pas encore une méthode et un outil standard pour tous les types d'application.

Chapitre 3

La négociation aux systèmes multi agents

3.1 Introduction

L'interaction dans un système multi agent vise essentiellement à la réalisation des tâches ou pour atteindre un but, l'interaction a lieu, d'habitude dans un environnement commun où les agents ont diverses zones d'influence, notamment diverses parties de l'environnement sur lesquelles ils peuvent agir. Par contre, l'interaction dans un environnement partagé, oblige les agents de coordonner leurs actions et avoir des mécanismes pour la résolution des conflits. La négociation joue un rôle fondamental dans les activités de coopération en permettant aux personnes de résoudre des conflits qui pourraient mettre en péril des comportements coopératifs. En général les chercheurs en intelligence artificielle distribuée utilisent la négociation comme un mécanisme pour coordonner un groupe d'agents. Différentes approches ont été développées en s'appuyant sur la riche diversité des négociations humaines dans divers contextes.

3.2 Qu'est ce que la négociation ?

On peut citer plusieurs définitions pour la négociation, Durfee et ses collègues [Dur89] définissent la négociation comme le processus d'améliorer les accords (en réduisant

les inconsistances et l'incertitude) sur des points de vue communs ou des plans d'action grâce à l'échange structuré d'informations pertinentes.

David et Smith (1980) disent que : "Par négociation, on entend une discussion dans laquelle des individus intéressés échangent des informations et arrivent à un accord en commun."

Pruitts (1981) donne une définition qui s'appuie sur des considérations psychologiques et pour laquelle le conflit est l'élément de base, "La négociation est le processus par lequel plusieurs individus prennent une décision commune. Les participants expriment d'abord des demandes contradictoires, puis ils essaient de trouver un accord par concession ou par la recherche de nouvelles alternatives »[Pru81].

D'après Philippe Mathieu et Alain Taquet [Flo02] : une négociation est un octuplet $(G, R, P, L, S, G, H, C_t)$ où :

1. G est l'ensemble des individus impliqués dans la négociation.
2. R est l'ensemble des ressources disponibles.
3. P est un ensemble de relations de préférences locales à chaque agent
4. L est l'ensemble des actes de langage autorisés pour la négociation.
5. S est un ensemble de règles de séquençement.
6. G est un graphe dont les nœuds sont des règles de séquençement
7. H est l'histoire de la négociation
8. C_t est un contrat qui est l'objet de la négociation.

Si on regarde les définitions citées, on peut identifier deux aspects essentiels de la négociation : la communication et la prise de décisions. Dans le cas des agents intelligents et dans les systèmes multi-agents, la négociation est une composante de base de l'interaction surtout parce que les agents sont autonomes.

Müller [Mul96] distingue les catégories suivantes dans le processus de négociation :

- **Langue** : résume, les primitives de communication (proposer, affiner, confirmer, etc) et leurs sémantiques, l'objet de négociation qui comprend les attributs qu'on veut né-

gocier, dans certains cas il s'agit de négocier uniquement le prix, mais dans d'autres cas il faut aussi négocier plusieurs attributs (multicritères) et les protocoles de négociation.

- **Décision** : À propos de quels aspects sont pris en compte les joueurs de décider sur la négociation à maximiser une fonction d'utilité, les préférences, les stratégies de négociation (être coopératif, concurrence).
- **Processus** : une étude des modèles généraux du processus de négociation et de la conduite générale des participants.

3.3 Domaines de la négociation

Selon Rosenschein et Zlotin [ROS94], la négociation peut être catégorisée en trois domaines :

1. Domaines orientés tâches (« Task Oriented Domains »)

Ces domaines coopératifs sont caractérisés par le fait, que les agents ont tous les moyens pour remplir leur but, dans ce cas un ensemble de tâches. Toutefois, chaque partie pourrait être meilleure si la tâche pourrait être redistribuée entre eux. La négociation est considérée comme un processus de coordination coopérative visant à trouver une redistribution des tâches. Les questions clés sont que les tâches sont indivisibles et que chaque agent ne peut accomplir ses tâches seul.

2. Domaines orientés état (« Etat Oriented Domains »)

Ces domaines sont caractérisés par le fait que les agents peuvent ne pas avoir toutes les ressources nécessaires pour accomplir leurs buts, car ils ont besoin des ressources des autres agents. Dans des mauvais cas, ils pourraient être en concurrence avec les buts des autres agents (d'où le besoin de ressources). En outre, le monde est considéré comme un domaine d'état, alors que le but d'un agent est de faire avancer le monde d'un état initial à un état objectif (but) avec un coût minimal.

3. Domaines orientés valeur (« Worth domains orientés »)

Les Domaines orientés valeur sont une généralisation du domaine orientés état. En domaines orientés valeur, les agents associer une valeur à chaque état du monde en termes de valorisation et de coûts. Selon Rosenschein et Zlotkin, ces états avec des plus hautes valeurs pourrait dire que le but est satisfais complètement, tandis que d'autres, avec des valeurs inférieures, ne satisfont que partiellement l'objectif. Dans ce contexte, chaque agent tente de maximiser leur gain en atteignant les états du monde avec une valeur maximale par rapport à chacun d'entre eux.

La négociation a été utilisée pour : attribution des tâches [Smi88] [Ste88], l'allocation des ressources [Adl89][Sat89], la résolution des conflits et la prévention [Syc89]. La négociation a été utilisée dans des circonstances différentes : les agents non-coopératifs [Syc89], les agents coopératifs [Ste88], par l'intermédiaire d'un médiateur [Syc89] ou d'une organisation hiérarchique [Smi88].

3.4 Négociation automatisée : difficultés

Nous notons que l'automatisation de la négociation peut se faire à deux niveaux différents : 1) Au niveau des participants, c'est-à-dire prenant en entrée des objectifs des négociateurs et des stratégies et de fournir des moyens automatisés pour atteindre les objectifs en utilisant les stratégies prescrites.

2) Au niveau des processus, par exemple, définir la manière des négociations et de fournir l'infrastructure.

En partant des difficultés inhérentes au processus de négociation lui-même, la négociation consiste à automatiser les trois problèmes suivants comme indiqué par Beam et Segev dans [Bea96][Bea97].

1. La nécessité d'une ontologie
2. La formulation de la négociation
3. L'exploitation de la stratégie de négociation, à ces problèmes, nous ajoutons un quatrième problème,

4. La terminaison de la négociation, il aborde la question de quand et comment terminer le processus itératif de la négociation.

3.4.1 Besoin d'une ontologie

Une ontologie est une représentation sémantique formelle et aussi la spécification des objets dans un domaine. En d'autres termes, c'est une façon normalisée de nommer et classer les choses afin de lever l'ambiguïté. Afin de s'assurer que les deux parties communiquant "parlent" de la même chose lors de la négociation, le besoin d'une telle représentation pour les biens et services qui seront l'objet d'échanges est obligatoire.

En outre, l'ontologie doit fournir un niveau suffisant de détails pour décrire les produits ou services dans leur intégralité. Pour certains produits tels que les CD musicaux, seul un petit nombre d'attributs (tels que l'artiste et le nom d'album) pourraient être nécessaires pour décrire le produit offert. D'autres produits tels que les ordinateurs ou les voitures ne se prêtent pas facilement les mêmes à de telles spécifications simples. Il est crucial que l'ontologie capture tous attributs importants et les caractéristiques d'un objet, par exemple couleur, taille, options, etc. Nous voyons deux raisons à cette exigence : (1) pour distinguer un produit d'une autre, (2) on peut compter sur ces attributs pour déterminer la valeur du produit.

Prenons l'exemple du Code universel des produits (Universal Product Code (UPC) system), une ontologie globale pour l'e-commerce est nécessaire. Pour plus de détails, Voir les travaux sur KIF [Gen95] et Ontolingua [Gru93] à l'Université Stanford pour des exemples de recherche menés dans ce domaine.

3.4.2 La Formulation de la négociation

La Formulation de la négociation est particulièrement difficile car la négociation des êtres humains manque, un énoncé clair, de définir d'une façon bien les idées de leurs propres objectifs ou des préférences pour la négociation et ne peut exprimer à l'avance quelle sera la stratégie souhaitée ou la réponse à une situation donnée.

Lors de la négociation, les humains ont souvent la stratégie implicite d'extraire le plus possible des autres parties, les humains basent aussi sur les sentiments. En outre, il est souvent difficile de considérer des conditions dans lesquelles on pourrait envisager une offre raisonnable. Parfois, une offre raisonnable est tout simplement la meilleure offre que l'autre partie peut donner, considérant qu'un accord est mieux que pas d'accord du tout.

À notre avis, le problème de la formulation est le plus grand obstacle à une automatisation de la négociation. Les deux sous-sections suivantes abordent les problèmes liés à l'informatisation des objectifs des négociateurs et des stratégies plus en détail.

Buts

Lorsqu'on représente mathématiquement les préférences, on peut énoncer différentes préférences avec une représentation de mesure ordinaire ou cardinale. La négociation pour un prix est impliquée, une représentation cardinale est souvent plus appropriée qu'une représentation ordinaire parce qu'il y a déjà une notion de valeur impliquée et parce que les préférences affecte directement le paiement.

Généralement, la représentation cardinale est une représentation plus utile et plus complète, mais il est plus difficile d'obtenir la connaissance requise pour cette représentation. Avec une représentation ordinaire, le problème devient rapidement complexe, car elle croit proportionnellement avec le nombre d'attributs et les valeurs des attributs puisque chaque point dans l'espace des commerces doit être évalué les uns contre les autres. Avec la représentation cardinale, nous pouvons simplement ajouter la valeur des différents attributs. Étant donné que les attributs sont indépendants. En plus des difficultés ci-dessus, il y a le risque, lorsque l'utilisateur énonce simplement des préférences, il pourrait aboutir à des contradictions logiques ou des exceptions dans les préférences.

Stratégies

Selon Robinson et Volkov (1998), "une stratégie de négociation se rapporte au plan par lequel les agents prévoient pour agir l'un sur l'autre avec d'autres agents tout en utilisant

un protocole de négociation particulier, pour atteindre les résultats souhaités". Noter que le protocole de négociation influence souvent le choix d'une stratégie de négociation.

De plus, les interactions entre les participants peuvent changer selon le protocole de négociation employé. En général, la négociation est basée sur un échange des offres et les contre-offres et nous devons coder explicitement la stratégie. Pour encoder une stratégie, nous avons besoin de plusieurs fonctions de décision :

- de décider qu'elle devrait être la première offre pour déterminer la formulation de contre-offres
- de décider l'acceptation ou le refus d'une offre.

3.4.3 L'exploitation de la Stratégie

Cette section prend en compte la considération suivante, lorsqu'on remplace une stratégie de négociation par un algorithme, il y a un risque que l'algorithme peut être exploité par les autres parties (agents). Le risque d'exploitation est encore plus haut dans les systèmes ouverts parce que les parties sont d'intention inconnue et la sécurité est non assurée.

Dans le monde réel, le vendeur peut toujours dire 'n'est pas grave' et de refuser de vendre ou de demander au client de faire une offre plus raisonnable. Une situation dangereuse, c'est lorsqu'une partie essaie d'en déduire notre stratégie afin d'acquérir un avantage significatif. Toujours dans le monde réel, ces considérations ne sont pas un problème parce que dans ces cas, le vendeur va probablement mettre fin aux négociations et refuser d'engager des négociations supplémentaires avec ce client.

Les systèmes de négociation automatisés n'ont pas les mêmes capacités humaines pour s'adapter. Dans ces contextes, la conception du protocole de négociation est l'aspect critique de l'ensemble du système, parce que toutes les situations doivent être prises en compte [ROS94]. Enfin, nous constatons que même des algorithmes complexes ne sont pas totalement à l'abri de l'inférence, car l'autre partie pouvait être un agent logiciel avec tout le temps nécessaire et la puissance des ordinateurs pour casser l'algorithme.

3.4.4 L'arrêt de la négociation

Un autre problème fondamental qui se pose en négociation à plusieurs tours entre deux agents logiciels est de savoir comment déterminer la fin des négociations. Intuitivement on pourrait dire que la fin de la négociation se produit lorsque :

- a) Un agent fait une offre dans la zone de la non-convention ou
- b) On conclue qu'il n'y a pas de zone d'accord. Bien que cette affirmation soit correcte, elle pose quelques problèmes informatiques faisables sous les différentes stratégies que les agents peuvent adopter.

3.5 Négociation pour l'allocation des tâches

Un des premiers protocoles pour l'allocation de tâches dans les systèmes multi agents est le réseau contractuel (Contract Net). La coordination de tâches est faite grâce à l'établissement de contrats afin d'atteindre des buts spécifiques. Deux rôles pour les agents dans ce protocole : gestionnaire ("manager") et contractant. L'agent qui doit exécuter une tâche (le gestionnaire) commence par décomposer cette tâche en plusieurs sous-tâches. Le gestionnaire annonce chaque sous-tâche sur un réseau d'agents (les contractants). Les agents qui reçoivent une annonce de tâches à accomplir évaluent l'annonce. Les agents qui ont les ressources appropriées, l'expertise ou l'information requise pour accomplir la tâche, envoient au gestionnaire des soumissions (« bids ») qui indiquent leurs capacités à réaliser la tâche. Le gestionnaire rassemble toutes les propositions qu'il a reçues et alloue la tâche à l'agent qui a fait la meilleure proposition. Ensuite, le gestionnaire et les contractants échangent les informations nécessaires durant l'accomplissement des tâches.

Les rôles des agents ne sont pas spécifiés d'avance ce qui permet de nouvelles décompositions des tâches : un contractant pour une tâche spécifique peut agir comme un gestionnaire en décomposant sa tâche et en annonçant les sous-tâches à d'autres agents. Les protocoles FIPA-Contract-Net et FIPA-Iterated-Contract-Net utilisent le protocole réseau contractuel pour permettre aux agents l'exécution de la négociation. Un problème qui peut survenir avec le réseau contractuel est qu'une tâche puisse être attribuée à un contractant moins

qualifié pour la résoudre, si un contractant plus qualifié est occupé au moment de l'annonce des tâches.

3.6 Négociation aux enchères

3.6.1 La théorie d'enchères

La théorie d'enchères consiste en un vendeur souhaitant maximiser son gain et un ensemble d'acheteurs souhaitant minimiser leur perte en fonction de l'estime qu'il porte à l'objet de la vente. Les enchères étaient assez restreintes auparavant, car elles s'organisaient à des moments bien définis et avec un nombre limité de participants. Avec l'explosion du commerce électronique, les enchères sont devenues un mécanisme d'achat et vente des produits disponibles à une grande échelle. Dans le commerce électronique, en fonction de la nature de la transaction, on peut identifier plusieurs types d'activités de e-commerce :

business-to-business (B2B) : désigne le commerce inter entreprise, i.e., les activités dans lesquelles les clients sont des entreprises. Internet est un terrain particulièrement propice au B to B, car le taux de pénétration du client y est plus fort que chez les particuliers.

business-to-consumer (B2C) : par opposition au B to B, le B to C d'occupe des échanges ou les transactions s'effectuent entre l'entreprise et le consommateur final.

Consumer-to-consumer (C2C) : L'échange inter consommateur, désigne l'ensemble des échanges de biens et de services entre plusieurs consommateurs sans passer par un intermédiaire.

Dans les trois types de e-commerce les enchères sont la méthode la plus étudiée. Comme dans d'autres types de négociation, les agents intelligents utilisés dans les enchères peuvent faciliter pour l'utilisateur la participation et les décisions à prendre.

3.6.2 Protocoles d'enchère

Plusieurs dimensions d'une enchère existent ce qui les différencie les unes des autres :

Détermination du gagnant : qui remporte l'enchère et combien il doit payer

- Premier prix : celui qui mise le plus haut remporte l'objet et paie sa mise
- Second prix : celui qui mise le plus haut remporte l'objet, mais paie la deuxième plus haute mise.

Un facteur important d'une enchère est sur quoi se base l'agent pour obtenir la valeur qu'il attribue à l'objet mis aux enchères

- Valeur privée : la valeur de l'objet dépend seulement des préférences de l'agent
- Valeur commune : la valeur de l'objet dépend seulement des préférences des autres agents
- Valeur corrélée : la valeur de l'objet dépend en partie des préférences de l'agent et pour l'autre partie, des préférences des autres agents

Est ce que les mises d'un agent sont connues des autres agents ?

- Ouverte : tous les agents savent les mises des autres agents, ils savent que les autres savent.
- Secrète : tout agent ne connaît que ses propres mises

Comment les mises sont annoncées

- Directe : les agents ne misent qu'une fois ?
- Ascendant : les agents essaient de miser plus haut que la dernière plus haute mise
- Descendant : le commissaire-priseur annonce des mises en ordre décroissant et le premier agent à signaler son intérêt remporte l'objet au dernier prix annoncé.

Envoie une offre

- secrète et non modifiable. À la clôture de l'enchère.
- le commissaire priseur ouvre les différentes offres.
- prix initial, le commissaire priseur fait diminuer le prix à intervalle de temps régulier.

3.7 Négociation heuristique

La négociation heuristique permet aux agents de dépasser la phase d'acceptation et de refus des propositions en fournissant des réactions utiles. Ces réactions peuvent prendre

	Déroulement	Gagnant	Prix payé
Anglaise	Le vendeur fixe un prix initial. Les acheteurs peuvent surenchérir les uns sur les autres jusqu'à l'adjudication du commissaire priseur	L'acheteur ayant fait l'offre la plus élevée	Le montant correspondant à son offre
Sous plis scellés	Chaque acheteur envoie une offre secrète et non modifiable. À la clôture de l'enchère, le commissaire priseur ouvre les différentes offres	L'acheteur ayant fait l'offre la plus élevée.	Le montant correspondant à son offre
Vickrey	Chaque acheteur envoie une offre secrète et non modifiable. À la clôture de l'enchère, le commissaire priseur ouvre les différentes offres.	L'acheteur ayant fait l'offre la plus élevée.	Le montant de la seconde meilleure offre majoré d'un incrément.
Hollandaises	Le vendeur fixe un prix initial (élevé). Le commissaire priseur fait diminuer le prix à intervalle de temps régulier	Le premier acheteur acceptant l'offre proposée	Le montant correspondant à son offre

TAB. 3.1: Les différentes procédures d'enchères

deux formes.

- La critique : c’est un commentaire sur la partie de la proposition que l’agent accepte ou refuse.
- La contre-proposition : c’est une proposition alternative engendrée en réponse à une proposition.

Dans les méthodes heuristiques de négociation, le problème central est de modéliser la décision de l’agent au cours de la négociation. Généralement, le protocole choisi ne prescrit pas une stratégie optimale pour l’agent. Dans ce cas, le succès relatif de deux agents est déterminé par l’efficacité de leur modèle de raisonnement. Les agents doivent découvrir d’une manière heuristique quelle est la meilleure proposition ou contre-proposition qu’ils doivent faire. Ils doivent découvrir cela en partant du modèle de décision qu’ils emploient.

3.8 La négociation par argumentation

L’argumentation est un processus itératif qui émerge des échanges entre agents pour persuader l’autre et lui faire changer ses intentions. Un agent pourra ainsi expliquer les raisons de sa réaction à une proposition. De même, un agent pourra accompagner une proposition d’arguments destinés à convaincre un autre agent de répondre favorablement.

Un dernier type d’arguments vise à modifier le sous-espace consensuel d’un autre agent en modifiant ses préférences ou les moyens de modifier l’espace de négociation lui-même. Une négociation commence toujours par une proposition qui peut être une offre ou une demande. Cette étape est suivie par un échange d’illocutions qui peuvent être l’émission de contre-propositions ou d’arguments de persuasion. Enfin, une illocution de fin de processus est invoquée.

Dans la négociation, les agents peuvent utiliser différents types d’arguments. Chaque type d’arguments est défini par des pré-conditions qui doivent être satisfaites pour que l’agent puisse utiliser l’argument. Les arguments peuvent se présenter sous les formes suivantes :

- Appel à une promesse passée : le négociateur A rappelle à B une promesse passée

concernant l'objet de négociation, autrement dit l'agent B a promis dans une négociation antérieure à l'agent A d'offrir ou effectuer un objet de négociation.

- Promesse d'une récompense future : le négociateur A promet de faire l'objet de négociation pour un autre agent B à un moment futur.
- Appel au propre intérêt : l'agent A croit que la conclusion d'un accord sur l'objet de négociation est dans l'intérêt de B et essaye de convaincre B de cela.
- Menace : le négociateur menace de refuser de faire/offrir quelque chose à B ou il menace de faire quelque chose qui contredit les désirs de B [Bau00].
- Appel à une pratique fréquente : l'agent A croit que B a refusé la proposition parce que B croit que la proposition contredit un de ses buts. Dans ce cas, l'agent A donne à B l'exemple d'une pratique fréquente qui démontre que l'acceptation de la proposition ne contredit pas le but de B .

3.9 SMACE (Sistema multi - Agente Para. Comércio Eletrónico)

Pour terminer ce chapitre, nous décrivons l'approche de négociation et d'architecture du système SMACE. Qui est un système multi-agent pour le commerce électronique, où les utilisateurs peuvent créer les agents acheteurs et vendeurs qui négocient de façon autonome, SMACE à été employé comme banc d'essai pour différents paradigmes de négociation.

3.9.1 Modèle De Négociation dans SMACE

Le modèle de négociation adopté dans ce système est multilatéral et basé sur beaucoup de questions (c'est-à-dire, multidimensionnelles), comme décrit dans [Far98]. En termes d'enchère, il se relie à l'enchère double scellée continue, où les acheteurs et les vendeurs soumettent des offres simultanément. Le modèle bilatéral de négociation employer est basé sur celui défini dans [Rai82]. Ce modèle de négociation s'appelle également un modèle orienté-

service, puisqu'il implique deux rôles qui sont, en principe : vendeurs des services et des acheteurs des services. Un service est quelque chose qui peut être fourni par un agent et demandé par un autre agent.

La séquence des propositions et des contre-propositions en deux parties est référée à la négociation des Thread. Ces propositions et des contre-propositions sont générées par des fonctions de combinaisons linéaires, appelées tactiques. Utilisez des tactiques d'un certain critère (temps, ressources, etc) pour générer une proposition pour une question donnée. Différents critères peuvent être assignés à chaque tactique utilisée, ce qui représente l'importance de chaque critère pour la prise de décision. L'agent peut désirer changer leur avis sur les critères au cours du temps. Pour ce faire, ils utilisent une stratégie, d'ont lequel un agent change le critère relatif des différentes tactiques au cours du temps.

Pour chaque question $j \in \{1, \dots, n\}$ en cours de négociation, chaque agent dispose d'une gamme de valeurs acceptables $[\min_j^i, \max_j^i]$ et une fonction de marquage

$V_j^i : [\min_j^i, \max_j^i] \rightarrow [0, 1]$ qui donne le score d'un agent i attribue à une valeur de J dans la gamme de ses valeurs acceptable. Plus le score, meilleure, l'utilité de l'agent est aussi.

L'Agents attribuer un poids w_j^i à chaque question de négociation qui représente son importance. En supposant que le poids normalisé ($\sum_j w_j^i = 1$), la fonction de score de l'agent pour une proposition donnée $X = (x_1, \dots, x_n)$ combine les scores des différentes questions évoquées dans le caractère multidimensionnel défini par les chaînes de valeur : $V_i(x) = \sum_j W_j^i \cdot V_j^i(x_j)$. La proposition globale est évalué à un score de zéro si aucune de ces valeurs est en dehors de sa gamme (intervalle).

3.9.2 Protocole de négociation de SMACE

À un moment donné, chaque agent à un objectif qui spécifie son intention d'acheter ou de vendre un service spécifique. Cet objectif doit être atteint dans un certain durée de temps, spécifiée par une date limite. La Négociation s'arrête lorsque ce délai est atteint. Une négociation bilatérale commence après que les deux parties - l'acheteur et le vendeur

– répondre sur le marché et correspondent à leurs objectifs (autrement dit, il ya un accord sur ce que le vendeur a l'intention de vendre avec ce que l'acheteur veut acheter). Les agents, ensuite échangent une séquence de propositions et de contre-propositions. Lors de la réception d'une proposition, un agent va générer une contre-proposition. Les deux - la proposition reçue et de la contre-proposition générés - seront évaluée en utilisant la fonction d'évaluation décrit ci-dessus, et l'agent répond par l'un des trois façons :

- Il retire de la négociation si la date limite a été atteint ou si une entente a été faite avec un autre agent.
- Il accepte la proposition reçue, si elle a une scores plus élevés que celui généré.
- Autrement, envoyez la proposition générée.

SMACE permet aux utilisateurs de créer des agents acheteur et vendeur qui négocient dans le cadre du modèle décrit ci-dessus. Le système a été mis en œuvre avec l'API JDK1.1.4, et utilise le JATLite (Java Agent Template, Lite) paquet pour créer facilement des agents, que l'échange des messages KQML .

3.10 Conclusion

En basant sur ce que nous avons présentés, il est clair qu'il n'y a pas une meilleure approche universelle ou une technique toujours appropriée pour la négociation automatisée. Il y a plutôt un ensemble éclectique de méthodes avec des caractéristiques et des propriétés de performances qui varient significativement en fonction du contexte de négociation.

Chapitre 4

L'aide à la décision multicritères

4.1 Introduction

La décision est le choix d'une action portant sur la mise en œuvre de ressources ou la détermination des objectifs, compte tenu d'un ou plusieurs critères d'évaluation des solutions. Dans notre vie quotidienne, parfois, lorsqu'un on veut prendre une décision, il faut obligatoirement consulter des experts qui sont basés sur des cas similaires ou encore utiliser des techniques de la recherche opérationnelle. Les méthodes d'analyse multicritères sont des outils d'aide à la décision développés depuis les années 1960, de nombreuses méthodes ont été proposées afin de permettre aux décideurs de faire un « bon » choix. Pour certains experts du domaine, ce choix existe dans l'esprit du décideur, et le processus d'aide à la décision doit faire ressortir. Pour d'autres, le processus d'aide à la décision doit créer ce choix. L'intérêt des méthodes multicritères est de considérer un ensemble de critères de différentes nature, sans nécessairement les transformer en critères économiques, ni en une fonction unique. Ainsi d'aider à prendre une décision ou à évaluer entre elles plusieurs solutions, sans avoir forcément de choix à effectuer au final, dans les situations de choix où aucune possibilité n'est parfaite.

B.Roy définit l'aide à la décision comme suit : "L'aide à la décision est l'activité de celui qui, par des voies dites scientifiques, aide à obtenir des éléments de réponse à des questions que se posent des acteurs impliqués dans un processus de décision, éléments concourant à éclairer la décision en vue de favoriser un comportement des acteurs qui soit de nature à accroître la cohérence entre l'évolution du processus d'une part, les objectifs et

Problématique	Objectif	Résultats	Procédure
α	Choix d'un sous-ensemble contenant les actions « les meilleurs » ou, à défaut, « satisfaisantes »	Choix	Sélection
β	Tri par affectation des actions à des catégories prédéfinies	Tri	Affectation
δ	Rangement de classes d'équivalence, composées d'actions, ces classes étant ordonnées de façon complète ou partielle	Rangement	Classement

TAB. 4.1: Les problématiques de référence

ou les systèmes de valeurs au service desquels ces acteurs se trouvent placés d'autre part." [Roy92]. Avant l'apparition de l'analyse multicritère, les problèmes de décision se ramenaient le plus souvent à l'optimisation d'une fonction économique. Cette approche avait le mérite de déboucher sur des problèmes mathématiques bien posés mais qui n'étaient pas toujours représentatifs de la réalité car :

- La comparaison de plusieurs actions possibles se fait rarement suivant un seul critère.
- Les préférences sur un critère sont, dans bien des cas, difficilement modélisables par une fonction.
- Lorsqu'il y a plusieurs objectifs, il est impossible de les atteindre tous à la fois.

Pour [Roy96] les problèmes réels peuvent être formulés à l'aide des méthodes d'analyse multicritère, selon trois formulations de bases : problématique de choix, notée P_α , la problématique de tri ou d'affectation notée P_β et la problématique de rangement noté P_δ (TAB.4.1).

4.2 Notion de base

Critères (ou attributs) : aspects suivant lesquels les alternatives sont examinées. Les critères représentent les paramètres de décision et circonscrivent les objectifs du décideur. On distingue :

- vrai critère : la structure de préférence sous-jacente est un pré ordre total (relation d'ordre qui admet plusieurs éléments équivalents.)
- quasi-critère : la structure de préférence sous-jacente est un quasi ordre.
- critère d'intervalle : la structure de préférence sous-jacente est un ordre d'intervalle
- pseudo-critère : la structure de préférence est un pseudo-ordre

Poids (des critères) : importance attribuée aux critères.

Indifférence : Elle correspond à l'existence de raisons claires et positives qui justifient une équivalence entre les deux actions.

Préférence stricte : Elle correspond à l'existence de raisons claires et positives qui justifient une préférence significative en faveur de l'une (identifiée) des deux actions.

Préférence faible : Elle correspond à l'existence de raisons claires et positives qui infirment une préférence stricte en faveur de l'une des deux actions mais ces raisons sont insuffisantes pour en déduire soit une préférence stricte en faveur de l'autre soit une indifférence entre ces deux actions.

In comparabilité : Elle correspond à l'absence de raisons claires et positives justifiant l'une des trois situations précédentes.

Seuil Vêto : valeur à partir de laquelle la différence des performances entre deux actions A et de B est considérée comme trop criarde pour accepter un sur classement de B par A .

Indice de concordance : La somme des critères où une action est supérieure ou égale à une autre multiplié par la pondération de ce critère le tout divisé par la somme des pondérations.

Indice de discordance : Le maximum pondéré des écarts entres les critères défavorable à l'action étudiée par rapport à une autre action.

Relation de sur classement : Une action surclasse une autre si et seulement si l'indice

de concordance est supérieur à un seuil et l'indice de discordance est inférieur à un autre seuil. Autrement dit, Sur classement : L'action A surclasse l'action B , si A est au moins aussi bonne que B sur une majorité de critères sans être trop nettement plus mauvaise relativement aux autres critères.

4.3 La formulation multicritère d'un problème de décision

Dans la terminologie introduite par Vansnick [Van90], la formulation d'un problème de décision multicritère peut être définie comme le modèle " $A, A/F, E$ " où :

A : est l'ensemble des actions potentielles (envisageables, admissibles,...). D'après [Roy85], "une action 'a' est la représentation d'une éventuelle contribution à la décision globale, susceptible, eu égard à l'état d'avancement du processus de décision, d'être envisagée de façon autonome et de servir de point d'application à l'aide à la décision."

D'après toujours [Roy85], "Une action potentielle est une action réelle ou fictive provisoirement jugée réaliste par un acteur au moins ou présumée comme telle par l'homme d'étude en vue de l'aide à la décision". Cet ensemble peut être défini explicitement (ensemble fini), les contraintes étant implicites, ou implicitement (en général ensemble infini).

Soit $A = \{a_1, \dots, a_p\}$ l'ensemble des actions potentielles, A peut être :

- **Stable** : si A peut être défini a priori et être non modifiable pendant le processus de décision.
- **évolutive** : si, au contraire, A est susceptible de changer pendant le processus de décision.

A/F est l'ensemble fini des attributs ou critères à prendre en considération, généralement conflictuels, à partir desquels les actions seront évaluées. Ces critères découlent des conséquences des actions, c'est-à-dire de "tout effet ou attribut de l'action susceptible d'interférer avec les objectifs ou avec le système de valeurs d'un acteur du processus de décision, en tant qu'élément primaire à partir duquel il élabore, justifie ou transforme ses préférences" [Roy85].

Une action a plusieurs conséquences, ainsi la conséquence d'une action selon un critère donné est évaluée par une fonction g définies sur l'ensemble A des actions potentielles de telle sorte qu'il soit possible de décrire le résultat de la comparaison de deux actions a et b relativement à partir des nombres $g(a)$ et $g(b)$.

Les critères peuvent classés en deux types : les vrai-critères et les pseudo-critères. Pour le vrai critère, en considérant deux actions a et b à comparer, deux situations sont possibles :

$$g(a) = g(b) \iff bI_g a \text{ (Indifférence)}$$

et

$$g(b) > g(a) \iff bP_g a \text{ (Préférence stricte)}$$

Pour le pseudo-critère on associe à la fonction g deux fonctions seuils $q_g(g(a))$ exprimant un seuil d'indifférence et $p_g(g(a))$ exprimant un seuil de préférence.

$$g(b) \geq g(a) \implies bS_b a$$

S_b : « Aussi bon que » ou, S est une relation de sur-classement, on dira dans se cas que b surclasse a , on notera $bS_b a$. On introduit des seuils (constants ou fonction de g) tels que :

$$g(b) - g(a) \leq q_g(g(a)) \iff bP_g a$$

$$P_g(g(a)) < g(b) - g(a) \iff bP_g a$$

Où q_g est un seuil dit d'indifférence et p_g un seuil dit de préférence. La situation non couverte par ces deux éventualités, à savoir :

$$q_g(g(a)) < g(b) - g(a) \leq q_g(g(a))$$

Correspond à une situation d'hésitation (indétermination) entre l'indifférence et la préférence stricte appelée préférence faible.

E : est l'ensemble des évaluations de performances des actions selon chacun des attributs ou critères, c'est-à-dire l'ensemble des vecteurs de performances, un vecteur par action.

Les évaluations des actions selon chacun des attributs ou critères peuvent s'effectuer en ayant recours à divers moyens (des formules analytiques, des instruments de mesure, des jugements...), être plus ou moins subjectives et être entachées d'imperfections plus ou moins importantes [Roy89].

En général, cette formulation ne permet pas de répondre totalement au problème de

décision et nécessite l'usage d'une méthode multicritère pour dégager les préférences globales du décideur.

4.4 Méthodes multicritères

Les méthodes multicritères sont nombreuses parce qu'il est dans la nature des choses qu'aucune méthode ne respecte la totalité des exigences qu'un utilisateur pourrait trouver "normales" dans l'idée du multicritère. Les spécialistes de l'aide à la décision multicritère s'accordent sur trois grandes familles[Vin89] :

4.4.1 Les méthodes de sur-classement

Les méthodes ELECTRE (ELimination Et Choix Traduisant la REalité)

Ces méthodes ont été développées par Bernard Roy⁵ au début des années 1970. Il a ainsi initié toute une série de méthodes, dites de sur-classement, basées sur des comparaisons d'actions deux à deux. Celles-ci demandent peu d'information pour pouvoir être implémentées, de plus cette information est facilement accessible au décideur (en effet, il est plus facile de comparer deux actions, que de donner une évaluation précise de leurs performances).

La méthode Electre I Cette méthode permet d'identifier le sous-ensemble d'actions offrant le meilleur compromis possible. Dans cette méthode, on définit de vrai-critères, on retrouve également une notion de concours dans cette méthode, retenir les meilleurs. On considère un ensemble A de m actions, qui représentent l'objet de la décision, dont le but est d'identifier un sous-ensemble d'actions offrant un meilleur compromis parmi l'ensemble de départ. Pour chaque critère, on définit une fonction d'évaluation g_j où $j = 1$ à n où n est le nombre de critères, pour chaque critère, on évalue un poids k_j qui augmente avec

⁵Bernard Roy, né le 15 mars 1934 est un chercheur français, professeur émérite de mathématiques appliquées aux sciences de gestion à l'Université Paris-Dauphine. Il est considéré comme l'un des pionniers de la recherche opérationnelle en France

l'importance du critère. L'indice de concordance⁶ pour deux actions a et b est noté $C(a, b)$, compris entre 1 et 0, mesure la pertinence de l'assertion « a surclasse b », comme suit :

$$C(a, b) = \frac{\sum_k \mathbb{1}_{\forall j: g_j(a) \geq g_j(b)}}{k} \text{ avec } k = \sum_{j=1}^n k_j$$

L'indice de discordance $D(a, b) = \left\{ \begin{array}{l} 0, \text{ si } g_j(a) \geq g_j(b) \\ \frac{1}{\delta} \max_j [g_j(b) - g_j(a)] \end{array} \right\}$, avec δ est la différence maximale entre le même critère pour deux actions donnée.

La relation de sur-classement pour Electre I est construite par la comparaison des indices de concordance et de discordance à des seuils limites de concordance \hat{c} et de discordance \hat{d} .

Ainsi, a surclasse b , si :

$$aSb \iff C(a, b) \geq \hat{c} \text{ et } D(a, b) \leq \hat{d}.$$

ELECTRE IS Est une adaptation d'ELECTRE I qui intègre la notion de flou décisionnel par la définition de pseudo-critères et de seuils.

ELECTRE II Elle relève des problématiques de classement. Elle vise à classer les actions depuis les meilleurs jusqu'aux moins bonnes. En se basant sur un principe de pré ordre total, Electre II postule que toutes les actions sont comparables (l'in comparabilité est exclue, en d'autre termes, le décideur peut toujours opérer un choix entre une action A et une action B).

ELECTRE III Elle relève aussi des problématiques de classement. Le but est de classer les actions des meilleures aux moins bonnes. L'originalité de cette méthode est d'admettre une part de flou dans les choix du décideur. Cet aléa est pris en compte par la définition de seuils de préférence et de surclassement, et par l'introduction d'un seuil veto.

ELECTRE IV Relève des problématiques liées aux procédures de classement. Comme ELECTRE III, elle associe à chaque critère des seuils de préférences, mais l'originalité réside dans le fait de supprimer la pondération attachée à chaque critère.

⁶Cet indice mesure en quelque sorte les arguments en faveur de l'affirmation précédente.

ELECTRE TRI Relève des problématiques d'affectation. Ici, des actions de références sont classées par catégorie. Chaque catégorie est bornée inférieurement et supérieurement par des actions. Le but du jeu est de classer les actions qui seront proposées au décideur dans une des catégories prédéfinies.

Les méthodes PROMETHEE

PROMETHEE est l'acronyme de Preference Ranking Organisation Method for Enrichment Evaluations. PROMETHEE se distingue d'ELECTRE par le fait qu'elle construit une relation de sur classement value traduisant une intensité de préférence : les méthodes PROMETHEE sont à mi-chemin entre l'approche de sur classement dont elles conservent la relation S de sur classement et les méthodes de MAUT dont elles utilisent les méthodes de construction des fonctions d'utilité partielles g_i . La comparaison des actions selon cette méthode se fait deux à deux via la mesure d'intensité de préférence $P(a, b)$ de l'une par rapport à l'autre sur chaque critère au moyen d'une fonction $p(d)$ où d représente la différence des évaluations $g(a) - g(b)$ sur le critère.

Pour un critère donné, une valeur $P(a, b) = 0$ traduit la non préférence de a sur b , une valeur $P(a, b) = 1$ traduit la préférence forte de a sur b , les valeurs intermédiaires croissantes entre 0 et 1 traduisant une préférence d'abord faible puis de plus en plus marquée pour a . La fonction $p(d)$ peut prendre différentes formes et dépendre de certains paramètres au choix (Tab 4.2).

PROMETHEE calcule ensuite un indicateur de préférence $\pi(a, b)$ comme la moyenne des intensités $P_c(a, b)$ éventuellement pondérées par l'importance w_c accordée au critère c :

$$\pi(a, b) = \sum w_c \cdot P_c(a, b) / \sum w_c$$

Les indicateurs de préférence pour chaque couple sont calculés dans chaque sens $\langle \pi(a, b)$ et $\pi(b, a) \rangle$ et peuvent être représentés dans un graphe value de surclassement. Ce graphe n'est pas exploité en tant que tel mais au travers des flux sortants $\Phi^+(a)$ et entrants $\Phi^-(a)$ de chaque action a :

$$\Phi^+(a) = \sum P(a, x) \text{ et } \Phi^-(a) = \sum P(x, a).$$

Vrai critère	$p(d) = 0$ si $d \leq 0$ et $p(d) = 1$ sinon
Quasi critère	$p(d) = 0$ si $d \leq q$ et $p(d) = 1$ sinon ; q est le seuil d'indifférence
Critère à préférence linéaire	$p(d) = 0$ si $d \leq 0$, $p(d) = \min(d/p, 1)$ sinon ; p est le seuil de préférence stricte
Critère à niveaux	$p(d) = 0$ si $d \leq q$, $p(d) = 1/2$ si $q < d \leq p$ et $p(d) = 1$ sinon ; q et p sont les seuils classiques
Critère mixte	$p(d) = 0$ si $d \leq q$, $p(d) = (d - q)/(p - q)$ si $q < d \leq p$ et $p(d) = 1$ sinon ; combine les deux précédents
Critère gaussien	$p(d) = 1 - \exp(-d^2/2\sigma^2)$; σ^2 est un paramètre analogue à la variance d'une distribution normale

TAB. 4.2: Les valeurs de $p(d)$ selon le critère

Les flux représentent donc l'intensité des avantages et des désavantages qu'une action a possède vis-à-vis de l'ensemble des autres actions candidates.

Il existe 4 méthodes PROMETHEE qui se distinguent à ce stade par la manière d'exploiter des flux. PROMETHEE I et II, se distinguent par le type de relation d'ordre souhaité pour le classement des actions. PROMETHEE I construit un pré-ordre partiel et permet l'incomparabilité :

$$aPb \text{ Ssi } \Phi^+(a) > \Phi^+(b) \text{ et } \Phi^-(a) < \Phi^-(b)$$

$$bPa \text{ Ssi } \Phi^+(b) = \Phi^+(a) \text{ et } \Phi^-(b) < \Phi^-(a)$$

$$aIb \text{ Ssi } \Phi^+(a) > \Phi^+(b) \text{ et } \Phi^-(a) = \Phi^-(b)$$

$$aRb \text{ sinon}$$

L'inconvénient de PROMETHEE I est que l'indifférence est en pratique très rare vu les nombreux calculs pour obtenir les flux. Cet inconvénient est réduit par PROMETHEE III qui introduit des seuils d'indifférence sur les flux.

PROMETHEE II construit un pré-ordre total excluant l'incomparabilité et réduisant

fortement l'indifférence : il s'agit souvent d'un ordre total obtenu sur la base du flux $\Phi(a) = \Phi^+(b) - \Phi^-(a)$.

$$aPb \text{ Ssi } \Phi(a) - \Phi(b) > 0 \text{ et } aIb \text{ Ssi } \Phi(a) = \Phi(b).$$

Dans sa variante II, la plus utilisée, PROMETHEE apparaît clairement comme une méthode d'utilité, les comparaisons 2 à 2 ne servant qu'à masquer le calcul du score final $\Phi(a)$ de chaque action.

PROMETHEE III est une variante de PROMETHEE II qui étend la possibilité d'indifférence en introduisant des seuils d'indifférence sur les flux. Pratiquement, on calcule l'écart-type des flux soit (s) et on choisit une constante (α) (assez petite) pour construire autour de chaque flux $\Phi(a)$ un intervalle symétrique [$X(a) = \Phi(a) - s.\alpha, Y(a) = \Phi(a) + s.\alpha$]. On définit ensuite la relation ($I - P$) par :

$$aPb \text{ Ssi } X(a) > Y(b)$$

$$aIb \text{ Ssi } X(a) \leq Y(b) \text{ et } X(b) \leq Y(a)$$

PROMETHEE IV a un intérêt plutôt théorique et concerne des ensembles infinis d'actions, elle nécessite le calcul d'intégrales au lieu de sommes.

4.4.2 Les méthodes basées sur la théorie de l'utilité

La Méthode du Goal Programming

Les techniques de « Goal Programming » ont été développées par CHARNES et COOPER [Cha61] pour adapter la programmation mathématique à la résolution de problèmes concrets. « Goal Programming » est une extension de la programmation linéaire [Cha61] [Iji65] qui est capable de traiter des problèmes de décision qui traitent un seul but avec de multiples buts secondaires ainsi que des problèmes avec des objectifs multiples avec des sous objectifs multiples. Le principe de base du Goal Programming consiste à introduire des variables d'écarts par rapport aux objectifs et de minimiser une fonction linéaire ou non de ces écarts.

Soient c_1, c_2, \dots, c_n les critères et soient $C^* = (c_1^*, c_2^*, \dots, c_n^*)$ le point idéal dans l'espace des critères correspondant au vecteur des niveaux d'aspiration. En général, $c_i \leq c_i^*, \forall a \in A$

car le niveau d'aspiration est trop élevé. Il est classique mais non nécessaire de prendre pour point idéal des valeurs c_i^* correspondant à l'optimisation du critère c_i sans tenir compte autres critères :

$$c_i^* = \max_{(a \in A)} c_i(a) \text{ pour } i = 1, \dots, n$$

Plusieurs distances peuvent alors être utilisées. Une distance souvent utilisée à des coefficients de pondération près est la distance de MINKOWSKI :

$$d(c(a), c^*) = \left\{ \sum_{i=1}^n |c_i^* - c_i(a)|^p \right\}^{\frac{1}{p}}$$

Remarque : Pour $p = 2$, on obtient la distance Euclidienne et la fonction à minimiser d est quadratique en $c_i(a)$. Par contre, pour $p = 1$ et $p = \infty$, la distance correspond à la norme distance en valeur absolue et à la norme, $\max |c_i^* - c_i(a)|$ pour $i = 1, \dots, n$. On peut alors résoudre le programme de minimisation de la distance en utilisant les algorithmes de la programmation linéaire ou ceux de la programmation mathématiques.

MAUT (Multi Attribute Utility Theory)

MAUT est une méthode développée vers la fin des années 60 par Ralph Keeney et Howard Raiffa. L'idée est assez simple, le décideur doit associer une utilité à chacune des actions considérées. Pour ce faire, il va considérer séparément chacun des critères, et observer quelle utilité dégage chaque critère pour considérée. En effet, l'utilité $V_A(x_1, x_2, \dots, x_n)$ associée à l'action A , évaluée sur les critères $1, 2, \dots, n$ peut se décomposer sous la forme :

$V_A(x_1, x_2, \dots, x_n) = \sum_{i=1}^n v_i(x_i)$ ($v_i(x_i)$: étant l'utilité générée au vu du critère si l'action considérée a la performance x_i) ; à condition que les critères considérés soient indépendants.

Il faut noter une grande attention aux réponses fournies car tous les critères doivent être évalués sur une même échelle. Des chercheurs travaillant dans le domaine de MAUT ont été principalement concentrés sur deux problèmes distincts :

- Quelles sont les conditions dans lesquelles les préférences d'un décideur peuvent être représentées par une fonction de d'utilité spécifique ?
- Comment évaluer cette fonction (et ses paramètres relatifs) ?

Somme pondérée Cette méthode n'est pas à proprement parler une théorie basée sur l'utilité; mais elle s'en reproche. Le fonctionnement de cette méthode est très simple. Elle définie comme suit :

$$g(a) = \sum_{j=1}^n k_j \cdot g_j(a)$$

Les poids k_j sont des coefficients strictement positifs, Il n'est pas restrictif d'imposer $\sum_{j=1}^n k_j = 1$.

Cette méthode permet de se passer des difficultés inhérentes à MAUT, et propose au décideur de noter directement les différentes actions relativement à tous les critères. L'avantage de cette méthode est qu'elle permet d'obtenir un résultat numérique, et un classement complet des actions, sans la lourdeur de MAUT. Si l'on dispose de données chiffrées sur les performances des actions, on peut les implémenter directement, sans avoir à convertir ces données sur une échelle abstraite. Toutefois, la somme pondérée demande beaucoup d'attention car elle est compensatoire.

4.4.3 Autre méthodes

AHP

La méthode AHP (Analytical hierarchy Process) à été développé par Thomas Saaty⁷ dans les années 80. Elle vise à affiner de processus de décision en examinant la cohérence et la logique des préférences du décideur. Le point de départ est définir une arborescence hiérarchique de critères et de sous critères. Chaque critère doit être identifié, avec sa définition et son intensité, il faut que la somme des poids de tous les critères fils d'un même critère père soit égale à 1 (relation d'interdépendance).

⁷Thomas L. Saaty (né en 1926 à Mossoul, Irak) est un mathématicien américain qui est un éminent professeur d'université à l'Université de Pittsburgh, où il enseigne dans le Joseph M. Katz Graduate School of Business

MACBETH

La méthode MACBETH (Measuring Attractiveness by a Categorical Based Evaluation Technique) est une méthode assez récente, développée notamment par Carlos Bana e Costa. Cette approche itérative a pour but de quantifier l'attractivité de chaque action par rapport à une autre. Elle ne requiert qu'un jugement qualitatif de préférence, ce qui lui permet de s'affranchir des critiques concernant les notations et leurs références. le système est basé sur un programme, vérifiant la consistance des données fournies, et créant une échelle de poids.

4.5 Conclusion

Le développement extraordinaire depuis le début des années 1970 du domaine d'aide à la décision multicritères explique son importance. Cette importance est expliquée par l'existence de plusieurs groupes de travail ,nous pouvons citer, 1) Le groupe de travail européen "Aide multicritère à la Décision" (il se réunissait à Québec, à l'Université Laval),2) The International Society on MCDM qui publie le bulletin MCDM-Worldscan et qui organise Périodiquement un congrès depuis le début des années 1980, 3) ESIGMA : European Special Interest Group on Multicriteria Analysis qui se réunit une fois par an à l'occasion des congrès européens de RO.

Chapitre 5

Modèle et réalisation

5.1 Introduction

La vente aux enchères la plus connue est celle où un vendeur offre un bien ou un service pour lequel des acheteurs font des offres compétitives, on parle dans ce cas d'enchère d'acheteurs. Si c'est l'acheteur qui est seul, mettant en concurrence des fournisseurs potentiels, on est dans le cas d'une enchère de vendeurs (enchère inverse), et l'objectif de l'agent unique est d'obtenir le prix le plus faible possible pour une prestation donnée. Cette catégorie d'enchères est utilisée en particulier par les États et leurs subdivisions (départements, communes, . . .) pour acquérir des biens et services en provenance du secteur privé (marchés publics).

Ce type d'enchère souffre d'une mauvaise publicité en raison de l'usage abusif d'acheteurs qui ne se focalisent que sur le prix. Pour régler cet inconvénient les directions d'achat ont mis en place l'enchère multicritère (multi-paramètre) qui permet la prise en compte, non seulement du prix, mais de nombreux autres paramètres, ce type d'enchère sera étudié par la suite dans notre modèle de négociation.

Dans cette mémoire, nous portons sur les enchères multicritères anglaises inversée pair à pair (négociation directe entre les vendeurs et l'acheteur (soumissionnaires ou Bidders). L'enchère anglaise inversée pair à pair peut être définie comme un processus itératif sur plusieurs tours (étapes) avec une date de clôture (fin d'enchère) où les agents vendeurs sont en compétition pour vendre un produit à un agent acheteur. L'enchère se commence par l'agent acheteur, au début de l'enchère ce dernier collecte les préférences de l'acheteur, et les

paramètres de l'enchère. Ensuite, il ouvre l'enchère par l'envoi des informations collectées précédemment. Dans chaque tour l'agent acheteur collecte toutes les propositions (une par agent vendeur), sélectionne la meilleure proposition comme proposition de référence pour l'étape suivante, met en attente le vendeur correspondant et formule la contrainte pour le tour suivant. Chaque agent vendeur doit envoyer sa proposition avant la fin du tour, sinon il est éliminé. L'enchère se termine :

- lorsque tous les vendeurs sauf un ont abandonné ou,
- lorsque la date de clôture est atteinte.

Pour avoir une vente aux enchères multicritères, nous avons besoin d'un module pour l'expression des préférences de l'acheteur, d'une méthode pour permettre à l'agent acheteur de sélectionner la meilleure offre et finalement d'un mécanisme qui permet à toute nouvelle proposition de battre la meilleure proposition reçue au tour précédent.

Dans ce système on s'intéresse principalement l'optimalité de Pareto⁸. Un résultat est optimum de Pareto si et seulement si aucun accord existe d'autres, ce qui est préféré par au moins un des deux agents, l'acheteur ou le fournisseur sélectionné gagner, et pour lequel l'agent est d'autres moins indifférents [Var92]. De façon générale, une solution (résultat) x à un problème est efficace Pareto (Pareto optimale), s'il n'y a pas une autre solution x' telle qu'au moins un agent ait une meilleure situation avec x' qu'avec x et aucun agent n'ait une plus mauvaise situation avec x' qu'avec x .

Autrement dit, le résultat (solution) est Pareto optimal s'il n'y a pas un autre résultat qui fasse gagner plus à un agent aux dépens d'un autre agent qui, lui, gagnera moins dans cette solution.

5.2 Concepts de base pour les enchères multicritères

- On a un ensemble de P vendeur noté S ($S = \{s_1, s_2, \dots, s_p\}$) qui sont en concurrence pour vendre un article à un acheteur unique, noté B .

⁸Pareto : économiste, première référence historique à la décision Multicritère (1896)

- V est l'ensemble de solutions (offres).
- Soit $\xi = \{\xi_1, \dots, \xi_q\}$ un ensemble de q critères définis par l'acheteur : $\xi_j : V \longrightarrow E_j$, où E_j est un ensemble ordonné (ces critères sont publics pour les vendeurs).
- Soit \succcurlyeq la relation de dominance définie comme suite :

$$\xi(a) \succcurlyeq \xi(b) \iff \xi_i(a) \geq \xi_i(b), \forall i \in \{1, \dots, q\} \text{ et } \exists k \in \{1, \dots, q\}, \xi_k(a) > \xi_k(b).$$
Ainsi nous définissons \succ comme suite : $\xi(a) \succ \xi(b) \iff \xi_i(a) > \xi_i(b), \forall i \in \{1, \dots, q\}$.
- Soit k la durée maximale de l'enchère et $k_r, \forall k \in \mathbb{N}$ le temps du tour r .
- $(b_i^j(k_r))$ désigne le $j^{\text{ème}}$ attribut de $r^{\text{ème}}$ l'offre proposée par le vendeur s_i , si l'index de l'attribut est omis, tous les attributs sont considérés.
- ε est un pas plus au moins grand choisir par l'acheteur.

Les relations de préférences

– Habituellement, lorsque l'on compare deux solutions multicritères a_i et a_j on suppose que seulement un des trois situations suivantes peut se produire :

1. $(a_i P a_j)$ ou $(a_j P a_i)$: a_i est préféré à a_j ou a_j est préféré à a_i .
2. $(a_i I a_j)$ ou $(a_j I a_i)$: $(a_i \text{ est indifférent à } a_j) \iff (a_i \succcurlyeq a_j \wedge a_j \succcurlyeq a_i) \iff (a_i = a_j)$.
3. $(a_i J a_j)$ ou $(a_j J a_i)$: $(a_i \text{ est incomparable à } a_j) \iff \langle \neg(a_i \succcurlyeq a_j) \wedge \neg(a_j \succcurlyeq a_i) \rangle$. Constitue ce qu'on appelle une structure de préférence.

- Toute amélioration de l'offre actuelle sur au moins un critère conduit à une nouvelle offre.
- Une vente aux enchères multicritères se termine au moment k si $\forall k \geq K, b(k) = \emptyset$.

Caractéristique de l'ensemble des gagnants

– A chaque instant k , l'ensemble des offres gagnantes associés à la vente aux enchères multicritères, noté $WS_{MA}(k)$ est défini par :

$$WS_{MA}(k) = \left\{ b \in H_{MA} \setminus \{ \tilde{b} \in H_{MA}(k) : \tilde{b} P b \} \right\} \text{ i.e. l'ensemble des gagnants } WS_{MA}(k)$$

d'une vente aux enchères à l'instant k , est l'ensemble des offres non préférées dans son historique.

- le calcul d'ensemble des gagnants à l'instant k est uniquement basée sur l'ensemble

des gagnants à $k - 1$ et sur la dernière offre proposée.

- b_i est non dominée Ssi \nexists pas b_k telle que $b_k \succ b_i$.
- b_i est efficace Ssi b_i est non dominée.
- Dans une vente aux enchères multicritères, l'ensemble des gagnants
- $WS_{MA} = \{b_1, b_2, \dots, b_m\}$ est dit optimale aux sens de Pareto s'il n'existe pas un autre ensemble potentiel $\widetilde{WS}_{MA} = \{\tilde{b}_1, \tilde{b}_2, \dots, \tilde{b}_n\}$ tels que $\exists i \in \{1, \dots, m\}, j \in \{1, \dots, n\} \setminus b_i P \tilde{b}_j$ et $b_k(P \cup I) \tilde{b}_l, \forall k \in \{1, \dots, m\}, l \in \{1, \dots, n\}$.
- La solution lui-même est constituée de q critères (sous solutions), c'est à dire $b_i = \{b_i^j, \forall j \in \{1, \dots, q\}, \forall i \in \{1, \dots, m\}\}$.
Donc $WS_{MA} = \{b_i^j, \forall j \in \{1, \dots, q\}, \forall i \in \{1, \dots, m\}\}$, alors WS_{MA} est optimale aux sens de Pareto Ssi $\nexists \widetilde{WS}_{MA} = \{\tilde{b}_k^j, \forall j \in \{1, \dots, q\}, \forall k \in \{1, \dots, m\}\}$, tels que $\exists i \in \{1, \dots, m\}, j \in \{1, \dots, n\}, \forall k \in \{1, \dots, q\} \setminus b_i^k P \tilde{b}_j^k$ et $b_s^j(P \cup I) \tilde{b}_l^j \setminus l, s \in \{1, \dots, m\}$.

Suivant la classification adoptée précédemment pour les méthodes d'analyse multicritères, nous avons trois modèles de vente aux enchères, le premier est basé sur Les méthodes de sur-classement (prenons l'exemple de ELECTRE I et PROMETHEE I), le deuxième est basé sur la théorie de l'utilité (prenons le cas de la somme pondérée, les moyennes et Goal programming), et le dernier modèle sera proposé pour avoir une vente aux enchères instinctivement optimale aux sens de Pareto.

5.3 Vente aux enchères fondée sur la théorie le l'utilité

D'une manière générale une fonction d'agrégation additive est définie comme suit :

$$h(a) = \sum_{i=1}^n u_i(a_i), \text{ avec } u_i \text{ monotones strictement croissantes (} u_i : \text{ fonction d'utilité).}$$

Le maximum d'une fonction d'agrégation additive correspond à des alternatives Pareto-optimales.

Rappel : une fonction d'agrégation additive est de la forme : $h(a) = \sum_{i=1}^n f_i(a_i)$ où les f_i sont des fonctions monotones strictement croissantes.

Soit a^* une alternative maximisant la fonction $h(a)$, et supposons que cette alternative ne soit pas Pareto optimale. Il existe donc une alternative a' qui la domine, et donc :

$$\forall i; a'_i \geq a_i^*, \text{ et } \exists i : a'_i > a_i^*.$$

Utilisons maintenant le fait que les fonctions f_i sont monotones strictement croissantes par définition :

$\forall i : f_i(a'_i) \geq f_i(a_i^*)$ et $\exists i : f_i(a'_i) > f_i(a_i^*)$ et donc on a : $\sum_{i=1}^n f_i(a'_i) > \sum_{i=1}^n f_i(a_i^*)$. D'où une contradiction puisque a' ne maximiserait pas $h(a)$.

Prenons un cas particulier de la combinaison linéaire des critères : $u_i(a_i) = \lambda_i \cdot a_i$

$$h(a) = \sum_i \lambda_i \cdot a_i, \lambda_i > 0.$$

On abouti au cas particulier de la somme pondérée définie comme suit :

5.3.1 La somme Pondérée

Soit a une solution composée de q critères, $u_i(a_i) = \sum_j^q w_j \cdot a_j$, avec $w_j > 0 (j = 1, \dots, q)$, et $\sum_j^q w_j = 1$ où w_j représente l'ensemble des vecteurs de poids qui définissent l'importance relative des critères.

Dans notre enchère multicritères, les propositions sont comparées par l'agent acheteur selon une fonction d'agrégation noté F_w qui définit la classe de relations de préférence P_{SP} :

$$P_{SP} = \left\{ \begin{array}{l} \succsim_w \subset V \times V, \forall b_i, b_j \in V, \exists w \in \Omega, b_i \succsim_w b_j, \forall i \in \{1, \dots, q\} \\ \iff \\ F_w(b_i) \geq F_w(b_j); F_w(b_i) = \sum_{m=1}^q w_m \cdot b_i^m, m \in \{1, \dots, q\} \\ \text{et } \Omega = \{w \in \mathbb{R}_+^*, \sum_j^q w_j = 1\} \end{array} \right\}$$

De la même manière toute proposition du vendeur est définie par une classe de préférence P'_{SP} comme suit :

$$P'_{SP} = \{ \succsim'_w \subset V \times V, \exists \varepsilon \in \mathbb{R}^+; \forall b_i, b_j \in V, b_i \succsim'_w b_j \iff F_w(b_i) > F_w(b_j) + \varepsilon \}.$$

Expérimentation : Nous présentons un exemple qui illustre une enchère où un acheteur négocie avec 3 vendeurs (α, β, δ) l'achat d'un véhicule décrit par 7 critères. L'incrément ε est fixé 0.01. Le poids mentionné ci-dessus reflète l'importance que l'utilisateur donne à chaque paramètre.

L'enchère se déroule en 4 étapes et se terminent sur un accord avec le vendeur β pour la proposition 4 de poids : 11.830625. Les offres associent aux vendeurs, ainsi ces évaluations

	Marque			Prix		Air bague		ABS	
Poids	0.25			0.25		0.1		0.1	
Valeurs	α	β	δ	2000	6000	oui	non	oui	non
Scores	7	10	10	15	5	15	5	15	5

	Nbre portes				Capac.Bagage		Vitesse Max	
Poids	0.1				0.15		0.05	
Valeurs	2	3	4	5	200	1000	150	250
Scores	9	10	14	15	5	20	10	12

TAB. 5.1: Les préférences de l'utilisateur

sont représentées au Tableau 5.2.

Propriétés

- Les poids w_i représentent des taux de substitution entre critères ($w_1 = bw_2$) plutôt qu'une importance.
- La logique d'agrégation sous-jacente est totalement compensatoire ainsi, une proposition avec un très bas score sur un critère peut être sélectionnée si elle a un score élevé sur un critère important. exemple : si les vendeurs β et δ mis l'offres $(5590, O, 5, O, 385, 170)$ et $(3000, N, 4, N, 380, 169)$ respectivement alors l'offre du δ sélectionnée comme gagnante.
- De très légères variations sur les valeurs des poids peuvent conduire à des solutions radicalement différentes.
- Supposons que ce vente n'est pas optimale au sens de Pareto i.e. il existe une offre au premier tour $P(3060, N, 5, N, 380, 170)$ qui est meilleur que l'offre gagnante dans ce tour $(3073, N, 5, N, 380, 170)$ et le système ne prend pas en considération l'offre P (Tab 5.3) :

Alors selon ce mécanisme il est impossible qu'il existe une offre P meilleur que celle sélectionnée et que le système ne prend pas en considération. Alors ce mécanisme de vente

Marque	Offre	Prix	Air bague	Nbre de portes	ABS	Capacité ba- gage	Vitesse Max	Poids
α	1	3098	N	3	N	320	185	8.93
α	2	3170	N	3	o	320	185	9.39
α	3	3250	O	5	o	320	185	10.84
α	4	3200	O	5	o	320	185	10.87
β	1	3073	N	5	N	380	170	10.35
β	2	3505	N	5	o	380	170	10.58
β	3	3545	O	5	o	380	170	11.56
β	4	3113	O	5	o	380	170	11.83
δ	1	3085	N	5	N	380	169	10.34
δ	2	3185	N	5	o	380	169	10.78
δ	3	3718	O	5	o	380	169	11.45
δ	4	3533	O	5	o	380	169	11.56

TAB. 5.2: Résultats du modèle de la somme pondérée

	Prix	Air bague	Nbre de portes	Capacité du bagage	Vitesse Max	Poids	ABS
β_1	3073	N	5	380	170	10.35	N
P	3060	N	5	380	170	10.36	N

TAB. 5.3: Résultat de contre exemple

est optimale au sens de Pareto, mais il ne permet pas d'imposer à un vendeur que son offre doit maximiser/minimiser tous les critères s'il s'agit d'une maximisation/minimisation (compensation).

5.3.2 Les moyennes

On à : $h(a) = \sum_{i=1}^n u_i(a_i)$, prenons un cas particulier de la combinaison linéaire des critères : $u_i(a_i) = \frac{1}{n} \cdot a_i$, Alors :

$$h(a) = \frac{1}{n} \sum_i a_i$$

On a plusieurs cas possible :

- Si $h(a) = a$ alors a est la moyenne arithmétique.
- Si $h(a) = h^2$ alors a est la moyenne quadratique.
- Si $h(a) = \frac{1}{a}$ alors a est la moyenne harmonique.
- Si $h(a) = \ln a$ alors a est la moyenne géométrique. (Avec $a_i > 0$).

Dans notre enchère multicritères, les propositions sont comparées par l'agent acheteur selon une fonction d'agrégation noté h qui définit la classe de relations de préférence $P_{h(a)}$.

$$P_{h(a)} = \left\{ \begin{array}{c} \succsim_w \subset V \times V, \forall b_i, b_j \in V, b_i \succsim_w b_j, \forall j \in \{1, \dots, q\} \\ \iff \\ h(b_i) \geq h(b_j) \end{array} \right\}$$

De la même manière toute proposition du vendeur est définie par une classe de préférence $P'_{h(a)}$ comme suit :

$$P'_{h(a)} = \{ \succsim'_w \subset V \times V, \exists \varepsilon \in \mathbb{R}^+; \forall b_i, b_j \in V, b_i \succsim'_w b_j \iff h(b_i) > h(b_j) + \varepsilon \}$$

Expérimentation : nous évaluons les résultats de l'exemple précédant suivant chacun de ces quatre mécanismes (tableau 5.4) :

On remarque d'une façon générale que :

- M. harmonique < M. géométrique < M. arithmétique < M. quadratique
- Le problème de la compensation est toujours posé.
- Même résultats pour tous les mécanismes.

Marque	modèle	Arithmétique	Harmonique	Géométrique	Quadratique
α	1	8.88	8.17	8.54	9.18
α	2	9.57	8.49	9.03	10.08
α	3	11.68	10.63	11.17	12.14
α	4	12.13	11.38	11.77	12.45
β	1	10.15	9.16	9.69	10.55
β	2	10.71	9.48	10.13	11.20
β	3	12.13	11.56	11.846	12.40
β	4	12.28	11.71	12	12.55
δ	1	10.14	9.16	9.68	10.54
δ	2	10.82	9.56	10.23	11.32
δ	3	12.06	11.49	11.77	12.34
δ	4	12.13	11.56	11.84	12.40

TAB. 5.4: Résultat des différentes moyennes

5.3.3 Goal Programming

La meilleure proposition est sélectionnée par l'agent acheteur selon une fonction qui définit la classe de relations de préférence P_{GP} :

$$P_{GP} = \left\{ \begin{array}{l} \succ_{GP} \subset V \times V, \forall b_i, b_j \in V, b_i \succ_{GP} b_j \\ \iff \\ d(b_i, C_*) < d(b_j, C_*) \end{array} \right\} \text{ tels que } d(b_i, C_*) = \left\{ \sum_{j=1}^q |c_*^j - b_i^j|^p \right\}^{\frac{1}{p}}$$

et $C_* = \{c_*^j = \max(score^j), \forall j \in \{1, \dots, q\}\}$:le point idéal dans l'espace des critères correspondant au vecteur des niveaux d'aspiration.

De la même manière toute proposition du vendeur est définie par une classe de préférence P'_{GP} définit comme suit :

$$P'_{GP} = \left\{ \begin{array}{l} \succ_{GP} \subset V \times V, \exists \varepsilon \in \mathbb{R}^+, \forall b_i, b_j \in V, b_i \succ_{GP} b_j \\ \iff \\ d(b_i, C_*) < d(b_j, C_*) - \varepsilon \end{array} \right\},$$

Les enchères se déroulent en 4 étapes et se terminent sur un accord avec le vendeur β pour la proposition 4 de distance :16.08 . Les offres associent aux vendeurs, ainsi ces évaluations sont représentées au tableau 5.5.

Propriétés

- la méthode du goal programming sélectionne la solution réalisable qui permet d'accomplir le mieux possible les objectifs du décideur.
- En se basant sur la solution de satisfaction, la technique du goal programming est très réaliste.
- problème de la compensation.

5.4 Vente aux enchères fondée sur les méthodes de sur-classement

5.4.1 ELECTRE I

La meilleure proposition est sélectionnée par l'agent acheteur selon une fonction qui définit la classe de relations de préférence P_{ELE} :

$p = 1$								
	Modèle	Prix	Air bague	portes	ABS	Capac.bagage	Vitesse.Max	Distance d
α_1	3	2.74	10	5	5	12.75	1.3	39.79
α_2	3	2.92	10	5	0	12.75	1.3	34.97
α_3	3	3.12	0	0	0	12.75	1.3	20.17
α_4	3	3	0	0	0	12.75	1.3	20.05
β_1	0	2.68	10	0	5	11.625	1.6	30.90
β_2	0	3.76	10	0	0	11.625	1.6	26.98
β_3	0	3.86	0	0	0	11.625	1.6	17.08
β_4	0	2.78	10	0	0	11.625	1.6	16.08
δ_1	0	2.71	10	0	5	11.625	1.62	30.95
δ_2	0	2.96	0	0	0	11.625	1.62	16.20
δ_3	0	4.29	0	0	0	11.625	1.62	17.54
δ_4	0	3.83	0	0	0	11.625	1.62	17.07

TAB. 5.5: Résultat de GP

$$P_{ELE} = \left\{ \begin{array}{l} \succ_{ELE} \subset V \times V, \forall b_i, b_j \in V, \exists \hat{c}, \hat{d} \in [0, 1], \\ b_i \succ_{ELE} b_j \\ \iff \\ C(b_i, b_j) \geq \hat{c} \wedge D(b_i, b_j) \leq \hat{d} \end{array} \right\}$$

Où $C(b_i, b_j) = \frac{\sum_{\forall j: b_i^j \geq b_j^j} k_j}{k}$ avec $k = \sum_{j=1}^q k_j$ et,

$$D(b_i, b_j) = \left\{ \begin{array}{l} 0, \forall j \text{ si } b_i^j \geq b_j^j \\ \frac{1}{\delta} \max[b_j^j - b_i^j], \text{ sinon} \end{array} \right\} : \text{ où } \delta \text{ est la différence maximale entre le}$$

même critère pour deux offres données.

Une proposition $b_f(k_r)$ étant sélectionnée comme meilleure au tour r , alors chaque nouvelle proposition $b_i(k_{r+1})$ doit vérifier les deux conditions suivantes :

$$C(b_i(k_{r+1}), b_f(k_r)) > C(b_i(k_r), b_f(k_r)) + \varepsilon \text{ et}$$

$$D(b_i(k_{r+1}), b_f(k_r)) < D(b_i(k_r), b_f(k_r)) - \varepsilon.$$

Nous contenions avec le même exemple précédent, en multipliant les poids par 10, alors on obtient les résultats suivants (Tab.5.6) :

Le résultat de cette vente aux enchères est que, $\beta_4 \succ \delta_4, \delta_4 J \beta_4$ et $\alpha_4 J \beta_4$ à des seuils limites de concordance $\hat{c} = 1$ et $\hat{d} = 0$ de discordance .

5.4.2 Promethee 1

La proposition gagnante suivant cette méthode est définie par une classe de relations de préférence P_{Pro} :

Concordance tour1				Discordance Tour1			
	α_1	β_1	δ_1		α_1	β_1	δ_1
α_1	-	0.25	0.25	α_1	-	0.5	0.5
β_1	0.95	-	1	β_1	0.03-	-	0
δ_1	0.75	0.7	-	δ_1	0.032	0.03	-
Concordance tour2				Discordance Tour2			
	α_2	β_2	δ_2		α_2	β_2	δ_2
α_2	-	0.5	0.5	α_2	-	0.5	0.5
β_2	0.7	-	0.75	β_2	0.083	-	0.083
δ_2	0.7	0.95	-	δ_2	0.032	0.002	-
Concordance tour3				Discordance Tour3			
	α_3	β_3	δ_3		α_3	β_3	δ_3
α_3	-	0.6	0.6	α_3	-	0.3	0.3
β_3	0.7	-	1	β_3	0.073	-	0
δ_3	0.7	0.7	-	δ_3	0.032	0.034	-
Concordance tour4				Discordance Tour4			
	α_4	β_4	δ_4		α_4	β_4	δ_4
α_4	-	0.6	0.6	α_4	-	0.3	0.3
β_4	0.7	-	1	β_4	0.03	-	0
δ_4	0.7	0.7	-	δ_4	0.003	0.083	-

TAB. 5.6: Résultats de Electrel1

$$P_{\text{Pro}} = \left\{ \begin{array}{c} \succ_{\text{Pro}} \subset V \times V, \forall b_i, b_j \in V, \\ \\ b_i \succ_{\text{Pro}} b_j \\ \\ \iff \\ \\ \left[\begin{array}{c} (\Phi^+(b_i) > \Phi^+(b_j)) \wedge (\Phi^-(b_i) < \Phi^-(b_j)) \\ \\ \vee \\ \\ (\Phi^+(b_i) = \Phi^+(b_j)) \wedge (\Phi^-(b_i) < \Phi^-(b_j)) \\ \\ \vee \\ \\ (\Phi^+(b_i) > \Phi^+(b_j)) \wedge (\Phi^-(b_i) = \Phi^-(b_j)) \end{array} \right] \end{array} \right\},$$

où $\Phi^+(b_i) = \frac{1}{p-1} \sum \pi(b_i, b_t)$, $\Phi^-(b_i) = \frac{1}{p-1} \sum \pi(b_t, b_i)$ et

$\pi(b_i, b_t) = \sum w_c \cdot P_c(b_i, b_t)$ tel que

$P_c(b_i, b_t) = b_i^c - b_t^c, \forall c \in \{1, \dots, p\}$.

Une proposition $b_f(k_r)$ étant sélectionnée comme meilleure au tour r , alors chaque nouvelle proposition $b_f(k_{r+1})$ doit vérifier les conditions suivantes :

$$\left\{ \begin{array}{c} \left(\begin{array}{c} \Phi^+(b_i(k_{r+1})) > \Phi^+(b_f(k_r)) + \varepsilon \wedge \\ \Phi^-(b_i(k_{r+1})) < \Phi^-(b_f(k_r)) - \varepsilon \end{array} \right) \\ \\ \vee \\ \\ \left(\begin{array}{c} \Phi^+(b_i(k_{r+1})) = \Phi^+(b_f(k_r)) \wedge \\ \Phi^-(b_i(k_{r+1})) < \Phi^-(b_f(k_r)) - \varepsilon \end{array} \right) \\ \\ \vee \\ \\ \left(\begin{array}{c} \Phi^+(b_i(k_{r+1})) > \Phi^+(b_f(k_r)) \wedge \\ \Phi^-(b_i(k_{r+1})) = \Phi^-(b_f(k_r)) - \varepsilon \end{array} \right) \end{array} \right\}$$

Les résultats de cette méthode sont représentés dans le tableau 5.7. Le gagnant de cette vente selon cette méthode est le vendeur β avec la quatrième proposition.

Propriétés

- Les méthodes de sur classement sont partiellement compensatoires : un handicap important d'une action par rapport à une action a , sur un critère, va empêcher (via l'implémentation du concept de discordance) que , même si dans les autres domaines, prend l'avantage sur b .

	1ier tour	2ème Tour	3ème Tour	4ème Tour
	π	π	π	π
(α, β)	-14.193	-11.943	-7.193	0.417
(α, δ)	-14.108	-13.933	-6.102	-6.946
(β, α)	0.085	-1.99	1.091	-7.365
(β, δ)	14.193	11.943	7.193	-0.418
(δ, α)	14.108	13.933	6.102	6.946
(δ, β)	-7.585	1.99	-1.091	7.365

	1ier tour	2ème Tour	3ème Tour	4ème Tour
$\Phi^+(\alpha)$	-14,15125	-12,93875	-6,648125	-8.26375
$\Phi^-(\alpha)$	14,15125	12,93875	6,648125	8.26375
$\Phi^+(\beta)$	7,139375	4,976875	4,1425	6.108125
$\Phi^-(\beta)$	-10,889375	-4,976875	-4,1425	-6.108125
$\Phi^+(\delta)$	3,261875	7,961875	2,505625	2.155625
$\Phi^-(\delta)$	-7,011875	-7,961875	-2,505625	-2.155625

TAB. 5.7: Résultats de Promethee1

- Les méthodes de sur classement participent de l'esprit de recherche d'une solution "satisfaisante", on y parlera souvent de solution de compromis et non optimales.

5.5 Analyse des résultats

- L'enchère basée sur la somme pondérée à le même résultat que celui basée sur Promethee1 (Promethee1 utilise la somme pondérée dans sa calcule).
- Le mécanisme basé sur les moyennes (arithmétique, harmonique, géométrique et quadratique) aboutit à un même résultat.
- Une vente aux enchères basée sur la théorie de l'utilité donne généralement des solu-

tions optimales aux de Pareto, mais a comme inconvénient le problème de la compensation.

- Dans Electre 1 le problème de l’incomparabilité apparais (défaut de cette méthode, le recours à Electre 2 est indispensable), ainsi le résultat est influencé par les seuils limites de concordance \hat{c} et de discordance \hat{d} .
- Toujours dans Electre1, pour avoir une vente optimale aux sens de Pareto les seuils de concordance et de discordance doivent prennent les valeurs 1 et 0 respectivement.

Remarque : Toutes ces méthodes ne permettent pas d’atteindre une solution qui maximise tous les scores sur tous les critères, pour permettre atteindre cet objectif nous proposons de suivre le modèle d’écrit dans la section suivante.

5.6 Le Modèle proposé

5.6.1 Description

Une enchère anglaise inversée est un processus itératif sur plusieurs étapes, avec une date de clôture où les agents vendeurs sont en compétition pour vendre un produit à un agent acheteur qui représente un acheteur humain.

L’enchère est débutée par l’agent acheteur qui collecte les préférences de l’acheteur. Puis, il ouvre l’enchère en envoyant à l’ensemble des vendeurs ces préférences, ainsi que les paramètres de cette vente (la durée d’une étape, d’un tour et la date de clôture de l’enchère,...). Chaque étape de l’enchère est divisée en plusieurs tours, durant chaque étape en mis une enchère anglaise inversée unicritères de plusieurs tours, c’est-à-dire l’agent acheteur collecte toutes les propositions (Une par agent vendeur) - concernant ce critère seulement-, sélectionne la meilleure proposition comme proposition de référence pour le tour suivant, met en attente le vendeur correspondant et formule la contrainte pour le tour suivant. Chaque agent vendeur doit envoyer sa proposition avant la fin du tour, sinon il est éliminé. Le résultat obtenu à chaque étape est la meilleure valeur atteinte pour le critère négocié durant l’étape (optimale au sens de Pareto). Avant d’entamer l’étape suivante pour négocier

sur un autre critère, les vendeurs doivent mettre l'accord sur la valeur finale du critère négocié à l'étape précédente, si un vendeur n'accepte pas cette valeur alors peut tirer de l'enchère, c'est pour cela il faut dès le départ classer les critères de « moins important » (i.e. qui n'exclut pas au départ les vendeurs) et commencer l'enchère par lui, « au plus important ». Cette classification peut se faire à l'aide des méthodes multicritères (problèmes de type : β, γ).

L'enchère se termine sur la meilleure proposition courante, soit lorsque tous les vendeurs sauf un ont abandonné ou soit lorsque la date de clôture est atteinte. L'enchère est synchrone, chaque agent vendeur doit envoyer sa proposition avant la fin du tour, sinon il est éliminé.

Les valeurs de chaque critère pour l'agent acheteur et vendeurs sont converties en scores et à chaque étape le problème est de maximiser ces scores correspondants aux valeurs pour le critère négocié.

5.6.2 Notations

- S^i contient les identités des vendeurs participés à l'étape i .
- Chaque critère est négocié dans une étape donc on a : q étapes.
- b_i^{jr} représente la proposition du vendeur i pour le critère j durant le tour r .
- ε est un pas plus au moins grand choisir par l'acheteur.
- Chaque étape i est caractérisée par : l'ensemble S des vendeurs, la durée de l'étape T_i , et la meilleure proposition sélectionnée b_{final}^i .
- b_{final}^{ir} est la valeur sélectionnée au tour r de l'étape i .
- ξ^i représente l'ensemble de critères négociés à l'étape i .
- $\xi^i \subset \xi$ donc $\xi \setminus \xi^i = \{\xi^j, \forall j \in \{1, \dots, p\} \text{ et } i \neq j\}$.
- C^i est l'ensemble des accords reçus pour la participation à l'étape i
- C_j contient les identités des accords reçus à l'étape i

5.6.3 Processus d'enchère

Stratégie de l'agent acheteur

1. Classification des critères à négociés

Au départ, on a : $C = \{c_1, c_2, \dots, c_p\}$, et $b_{Réf}^{i_0} = 0, \forall i \in \{1, \dots, q\}$.

2. L'agent acheteur répète les opérations suivantes pour chaque étape i :

- (a) Mettre $\xi^i = \{\xi_i\}$.

- (b) $Start_Timer(T_{étape})$

- (c) Pour chaque Accord C_j reçu, Supprimer C_k de C tel que $k \neq j$ et Mettre j dans S^i .

- (d) L'agent acheteur répète les opérations suivantes pour chaque tour r

- i. $Start_Timer(T_{tour})$

- ii. Réception des propositions $b_j^{i_r}, \forall j \in S^i$.

- iii. Teste $b_j^{i_r} \leq b_{Réf}^{i_{r-1}}$ Alors $\forall j \in S^i$, S_j est éliminé mettre $S_i = S_i - S_j$ et $S^i = S^i - \{j\}$.

- iv. Sélectionné la gagnante $b_j^{i_r} \succ b_k^{i_r} \iff b_j^{i_r} > b_k^{i_r}$.

- v. Calculer $b_{final}^{i_{r+1}} = \max(b_j^{i_r}), \forall j \in S^i$.

- vi. Définir la contrainte pour la prochain tour $b_{Réf}^{i_r} = b_{final}^{i_r} + \varepsilon$.

- vii. Envoyer $b_{Réf}^{i_{r+1}}$.

- viii. $Stop_timer(T_{tour})$.

Stratégie du vendeur

1. Réception des préférences de l'acheteur et les paramètres initiaux de l'enchère

2. L'agent vendeur répète les opérations suivantes pour chaque étape i :

- (a) **Si** la valeur $b_{final}^{i_{r-1}}$ est acceptée **Alors** Envoyer l'accord C_l .

- (b) L'agent vendeur répète les opérations suivantes pour chaque tour r

- i. Formuler et envoyer la proposition $b_j^{i_r}$.

- ii. Réception $b_{Réf}^{i_r}$.
- iii. Si la valeur $b_{Réf}^{i_r}$ n'est pas acceptée alors abandonné.

Le résultat de cette vente est une offre $b_{Enchère}$:

$$b_{Enchère} = (\max(b_j^{i_r}), (\max(b_j^{i+1_r}), \dots, \max(b_j^{q_r})), \forall i \in \{1, \dots, q\} \text{ et } r \in \{1, \dots, n\},$$

alors il est impossible que cette dernière est dominée par aucune autre offre

($b_{Enchère}$) .i.e. elle ne peut être améliorée au regard d'un critère sans la détériorer

pour un autre.

Propriétés d'enchère

1. **Efficacité** : Une enchère est efficace quand aucun vendeur, excepté peut-être le vendeur gagnant, ne peut fournir une proposition meilleure que la proposition gagnante selon la relation de préférence de l'acheteur.

De manière formelle, l'enchère est efficace si et seulement si $\nexists b \in V/b \succ b_{Enchère} \iff$ si $\nexists b \in V/b > b_{Enchère}$, or $b_{Enchère} = (\max(b_j^{i_r}), (\max(b_j^{i+1_r}), \dots, \max(b_j^{q_r})), \forall i \in \{1, \dots, q\}$ et $r \in \{1, \dots, n\} \implies$ donc notre enchère est logiquement efficace. Il faut noter que l'efficacité ne peut être garantie que sous l'hypothèse que l'enchère se termine naturellement et non à cause des limites de temps.

2. **Non-dominance des propositions gagnantes** : Dans les enchères, la proposition gagnante doit être toujours non dominée. De manière formelle, b_i est non dominée si il n'existe pas b_k telle que $b_k \succ b_i$.

A chaque étape i la proposition gagnante est de la forme :

$$b_{final}^1 = \max(b_j^{1_r}, \dots, b_{final}^i, null, \dots, null), \text{ donc il est impossible l'existence de } b_k \text{ tel que } b_k \succ b_i.$$

Le résultat de notre vente aux enchères est l'offre $b_{Enchère}$, alors il est évident que $b_{Enchère}$ ne peut être dominé (la relation de dominance ' \succ ' est '>').

3. **Concurrence équitable des propositions non-dominées** : une vente aux enchères satisfait la concurrence équitable des propositions non-dominées si toute

proposition non-dominée est susceptible d'être sélectionnée. La négociation à chaque étape i concerne un et un seule critère et la sélection est suivant la valeur maximale proposé pour ce dernier, alors il est évident qu'une proposition b qui a une valeur plus grande sur ce critère gagne si elle est proposée par un vendeur $\in S^i$.

4. **L'optimalité au sens de Pareto :** Une enchère est optimale aux sens de Pareto si la proposition gagnante est meilleure à toutes autres propositions sur tous les critères. La proposition gagnante est : $b_{Enchère} = (\max(b_j^{i_r}), (\max(b_j^{i+1_r}), \dots, \max(b_j^{q_r})), \forall i \in \{1, \dots, q\}$ et $r \in \{1, \dots, n\} \implies$ qui Maximise tous les valeurs correspond aux critères.

Note : Dans tous les exemples précédemment la transformation des valeurs en scores

se fait comme d'écrit ci-dessous :

Si le critère est discret **alors**

Prend leur valeur associe

Sinon

On utilise la formule suivante :

$$\frac{(v_i - \min_i) \cdot (T[\max_i] - T[\min_i])}{\max_i - \min_i} + T[\min_i]$$

v_i : Est la valeur a convertie

\min_i : La valeur minimale pour le critère i

\max_i : La valeur maximale pour le critère i

$T[v_i]$: Le score associe à la valeur v_i .

5.6.4 Etude Comparative

Dans Cette étude nous comparons notre modèle (présentée précédemment) avec les deux modèles, Butterfly[Yve06] et le modèle basé sur le point de référence [Mar04]. Avant tout nous commençons par donner une vue sur ces deux dernier modèles.

Le modèle Butterfly [Yve06]

Ce modèle est inspirée à la fois des deux méthodes Prométhée et Electre. Il est basé sur des comparaisons entre les offres multidimensionnelles. Plus formellement, la différence entre les deux offres b et \tilde{b} sur le critère j est définie comme suit :

$$d_j(b, \tilde{b}) = \tilde{b}^j - b^j$$

Sous l'hypothèse que tous les critères doivent être minimisés, une valeur de $d_j(b, \tilde{b}) > 0$ indique que b n'est pas pire que \tilde{b} sur le critère j . Les valeurs $d_j(b, \tilde{b})$ nous aidera à spécifier si l'offre b est au moins aussi bon que \tilde{b} , ils ne peuvent pas être utilisés directement puisque leur interprétation n'est pas évidente, alors sont transformés en degrés de préférence est indispensable.

Deux paramètres sont caractériser cette transformation, Un seuil d'indifférence noté q_j , et Un seuil de préférence noté p_j . Une fois que les degrés de préférence par critère ont été évalués, il faut les agréger pour obtenir une mesure globale afin d'évaluer si b est au moins aussi bon que \tilde{b} . Conformément à la méthode de Prométhée, ce modèle en utilisant une somme pondérée classique :

$$P_w(b, \tilde{b}) = \sum_{j=1}^q w_j \cdot P_j(d_j(b, \tilde{b})), \text{ où } w = \{w_1, w_2, \dots, w_q\}$$

représentent les poids compatibles avec les préférences.

Toutefois, contrairement à la méthode Electre et Prométhée, les paramètres sont fixe par le commissaire-priseur avant la vente aux enchères. Ces paramètres appartiennent à un sous-ensemble de poids recevable, a noté $AW(k)$, qui est susceptible d'évoluer pendant la vente aux enchères.

$AW(k) = \{w_1, w_2, \dots, w_q\}$, telle que :

1. $w_l > 0, l = 1, \dots, q$
2. $\sum_{l=1}^q w_l = 1$
3. $a_{i1}w_1 + \dots + a_{iq}w_q < c_i, i = 1, \dots, m_k$
4. $\sum_{l=1}^q w_l \cdot (P_l(b_i, b_j) - P_l(b_j, b_i)) \geq 0, \forall b_i, b_j \in S(k)$

Les deux premiers contraintes considérées ci-dessus sont naturels, bien connu des restrictions pour les poids utilisés dans ce contexte. La troisième contrainte repré-

sente les restrictions imposées directement par l'acheteur. Notons que à l'instant k , il pourrait y avoir différents m_k des contraintes imposées sur le poids. Enfin la dernière contrainte prend en compte non comparaisons triviales imposées par l'acheteur.

Point de référence [Mar04]

Selon ce modèle, l'acheteur doit spécifier un point d'aspiration qui exprime les valeurs souhaitées sur chaque attribut décrivant le produit à acheter et un point d'exigence qui représente les valeurs minimales acceptables sur chaque critère [Wie86]. Une proposition est évaluée (par l'agent acheteur) en fonction de la différence maximale entre les valeurs désirées et les valeurs de la proposition sur tous les critères.

Dans ce modèle, la méthode multicritère utilise la définition d'une déviation au point d'aspiration qui mesure l'écart maximal entre les valeurs des critères. Soit le point d'aspiration a et une proposition b , la déviation de b à a est définie par la relation suivante :

$$\begin{aligned} \text{déviation}(a, b) &= \max_{j=1,q} (\lambda_j (a_j - b_j)), \text{ avec } \lambda_j = \frac{1}{(\text{ideal}_j - \text{antiIdeal}_j)} \text{ et} \\ \text{ideal} &= \text{ideal}_1, \dots, \text{ideal}_q \text{ où } \text{ideal}_j = \max_{i=1,p} (b_i^j) \text{ et} \\ \text{antiIdeal} &= \text{antiIdeal}_1, \dots, \text{antiIdeal}_q \text{ où } \text{antiIdeal}_j = \min_{i=1,p} (b_i^j). \end{aligned}$$

L'évaluation d'une proposition mesure la déviation qui la sépare du point d'aspiration. La déviation retenue est la norme pondérée de Tchebychev.

$$\text{On dit que : } b_m \succ b_i \iff \text{déviation}(a, b_m) < \text{déviation}(a, b_i).$$

L'ensemble des propositions reçues lors d'une enchère est totalement ordonné par cette relation. Ainsi, à chaque étape d'une enchère, la meilleure proposition b^* est celle qui minimise la déviation au point d'aspiration : $b^* = \min\{\text{déviation}(a, b)\}$

Ce mécanisme suit la règle «beat-the- quote» qui impose que toute nouvelle proposition soit meilleure que la meilleure proposition reçue jusqu'alors. Cette règle est mise en œuvre dans ce mécanisme par l'introduction d'un incrément ε qui représente l'amélioration demandée à chaque étape. A l'étape $t + 1$, l'agent acheteur envoie le point d'exigence défini par l'équation ci-dessous comme contre proposition à tous les vendeurs appelés à améliorer leurs propositions. la définition de contre propositions sous forme de points d'exigence assure

	1	2	3	4	5
Butterfly	Vétifier	Vétifier	Vétifier	Vétifier	NonVétifier
Points de références	Vétifier	Vétifier	Vétifier	NonVétifier	Vétifier
Notre modèle	Vétifier	Vétifier	Vétifier	Vétifier	Vétifier

TAB. 5.8: La comparaison entre les modèles

la progression des propositions vers le point d'aspiration . De plus, elle permet à l'agent acheteur de garder privé son point d'aspiration et son modèle d'agrégation.

$$\forall j \in \{1, \dots, q\} e^{(t+1)} = a_j - (dt - \varepsilon)/\lambda_j.$$

La comparaison

Nous prenons les propriétés suivantes pour faire la comparaison entre les trois modèles, 1 : l'efficacité, 2 : Non-dominance des propositions gagnantes, 3 : Concurrence équitable des propositions non-dominées ,4 : L'optimalité au sens de Pareto ,5 : la non compensation. Le tableau 5.8 illustre bien cette comparaison.

Le modèle de points de référence n'est pas optimale aux sens de Pareto, cela peut être justifié par un contre exemple, pour ça nous prenons $\varepsilon = 0.3$ et les points de référence comme suit(Tab 5.9) :

Il s'arrive dans ce mécanisme de vente aux enchères que l'ensemble des gagnants ne contient pas une seule offre mais plusieurs alors le conflit de choix est survenu (se qu'on appelle Frontière de Pareto). Supposons que ce dernier est réglé (sous l'intervention humaine) dans ce mécanisme de vente, parfois la meilleur proposition n'est pas sélectionnée comme gagnante, alors son mécanisme de sélection (l'acheteur) n'est pas optimale au sens de Pareto ce qui conduit que l'enchère n'est pas optimale au sens de Pareto. Il faut signaler que ce mécanisme assure l'amélioration des critères d'une étape à une autre.

Points d'aspiration		
<i>A</i>	<i>B</i>	<i>C</i>
80	70	75

Points d'exigence_1		
<i>A</i>	<i>B</i>	<i>C</i>
20	25	15

Premier Tour			
Idéal	36	40	30
Anti_ideal	15	25	20
λ	0.04761905	0.06666667	0.1

	<i>Propositions₁</i>			<i>Résultats</i>			<i>Max</i>	<i>d_t</i>
P_1	30	25	30	5	3	2.14285714	5	5
P_2	20	25	15	6	3	2.85714286	6	
P_3	20	25	20	6	3	2.61904762	6	
p_4	20	30	15	6	2.67	2.85714286	6	
P_5	25	25	15	5.5	3	2.85714286	5.5	
P_6	25	30	15	5.5	2.67	2.85714286	5.5	
P_7	20	40	36	6	2	1.85714286	6	
P_8	30	40	30	5	2	2.14285714	5	
P_9	30	30	30	5	2.67	2.14285714	5	

TAB. 5.9: Résultat d'un contre exemple pour le modèle de point de référence au premier tour

Points d'exigence_2		
<i>A</i>	<i>B</i>	<i>C</i>
33	25	15

	<i>Propositions₂</i>			<i>Résultats</i>			<i>Max</i>	<i>d_t</i>
P_1	33	40	30	1.88	-0.75	1.5	1.88	1.25
P_2	35	45	30	1.8	1.25	1.5	1.8	
P_3	35	25	25.3	1.8	2.25	1.65666667	2.25	
p_4	33	30	30	1.88	2	1.5	2	
P_5	25	30	15	2.2	2	2	2.2	
P_6	40	40	25.3	1.6	1.5	1.65666667	1.65666667	
P_7	40	30	30	1.6	2	1.5	2	
P_8	50	45	45	1.2	1.25	1	1.25	
P_9	60	50	45	0.8	1	2.5	2.5	

TAB. 5.10: Résultat d'un contre exemple pour le modèle de point de référence au 2ème tour

Critères	Marque	Prix	Air bague	Nbre portes	ABS	bagage	Vitesse Max
L'ordre	7	6	1	2	3	4	5
Valeur sélectionnée	β	3073	O	5	O	380	170

TAB. 5.11: Résultat du modèle proposé

Le modèle de Butterfly souffre du problème de la compensation parce qu'il utilise la somme pondérée pour la calculer les indicateurs de préférence (π).

Expérimentation : Nous essayons d'adapter l'exemple décrit dans la section 5.3.1 avec notre modèle. Le vainqueur de cette vente est le vendeur β avec l'offre qui porte les valeurs définies dans le tableau 5.11 (sous l'hypothèse que ce vendeur accepte ces valeurs) :

Le vendeur α est exclu de l'enchère parce qu'il n'a pas de voiture qui a une capacité de bagage 380cc, ainsi le vendeur δ est exclu puisque sa vitesse maximale est de 169Km/H.

5.7 Conception du système

Notre travail vise à appliquer la notion des systèmes multi agents aux enchères optimales au sens de Pareto. La modélisation des enchères s'articule autour de deux composantes principales, à savoir : les agents qui le composent et le monde extérieur ou l'environnement externe. Nous abordons dans cette section la conception du modèle correspondant à notre sujet, en montrant la structure fonctionnelle et opérationnelle du système.

5.7.1 Choix de la méthodologie

Parmi les méthodes qui couvrent le mieux le cycle de développement d'un système multi-agent, que nous avons citées dans le chapitre 3, nous avons choisi la méthodologie AUML pour modéliser notre système. AUML est basée essentiellement sur la notation standardisée UML (Unified Modeling Language) qui est le résultat de la collaboration d'un

groupe de spécialistes du domaine du génie logiciel, pour arrêter la dispersion existante dans les notations utilisées dans le domaine, et créer le standard de facto pour modéliser les systèmes OO. AUML décrit le comportement d'agents grâce à des diagrammes d'activité, des graphes d'états et des diagrammes de séquence. En effet, comme avec UML, il semble qu'un formalisme unique ne soit pas suffisant

5.7.2 Identification des agents

Les agents constituent les acteurs principaux du système. Pour identifier nos agents nous devons nous plier aux exigences réelles d'une vente aux enchères optimale au sens de Pareto. Dans la plupart des enchères, nous retrouvons deux types d'agent, un agent acheteur et un autre vendeur. Au final, notre système sera composé des agents suivants :

L'Agent Acheteur (AA)

C'est l'acteur le plus important dans nos système, car c'est lui le responsable de la vente aux enchères, donc il contrôle le cycle de participation à ses enchères. Un Agent Acheteur pourra effectuer les tâches suivantes :

- Collecte les préférences de l'acheteur (la durée maximale de l'enchère, la durée d'une étape et du tour et bien sure l'incrément ε).
- Envoi à l'ensemble des vendeurs ces préférences, ainsi que les paramètres de cette vente (la durée d'une étape, d'un tour et la date de clôture de l'enchère,..).
- Dans chaque étape de l'enchère, mis une enchère anglaise inversée uni critères de plusieurs tours.
- Dans chaque tour il répète les opérations suivantes :
 - Collecte toutes les propositions (Une par agent vendeur),
 - Sélectionne la meilleure proposition comme proposition de référence pour le tour suivant, met en attente le vendeur correspondant et
 - Formule la contrainte pour le tour suivant.
- Reçoit les accords pour le prochain tour

L'Agent Vendeur

Dans notre modèle l'agent vendeur représente un utilisateur humain qui lui définit ses intérêts, Un Agent vendeur pourra effectuer les tâches suivantes :

- Reçoit les préférences de l'agent l'acheteur
- Définir son proposition.
- Envoi son accord pour le tour suivant
- Attend l'appel offre de l'agent acheteur et réfère de nouveau ce processus.

5.7.3 Les diagrammes de classes des agents

Ces classes des agents montrent les interactions et les messages envoyés et reçus de chaque agent du système avec les autres agents. Un diagramme de classes agent doit supporter tous les concepts liés à l'agent(Fig 5-1 et 5-2).

5.7.4 Diagrammes de séquences pour la vente à l'enchère

Dans notre système, on a considéré deux acteurs (Acheteur, vendeur), et quatre entités. L'entité Agent-Acheteur qui correspond a un humain(Acheteur), L'entité Agent-Vendeur qui correspond a un humain (vendeur).on a représentent la base de données par une entité BDD, et la dernière entité correspond a Interface qui joue le intermédiaire entre l'humain et l'agent Acheteur(Fig 5-3).

- Au début de l'enchère l'acheteur (humain) doit saisir ses préférence dans une interface (1), l'application doit répondre (par une interface) (2).
- Les préférences sont entrées (3) ; ces dernières sont enregistrées dans la base de données (3.1).
- L'enregistrement doit être confirmé par un acquittement, fournit par la base via l'interface (3.2), ensuite de l'interface à l'acheteur(4).
- L'agent acheteur obtient ces informations directement de la base de données (5) (6).
- Ensuite l'agent acheteur invite les vendeurs pour la participation (7)
- Le vendeur, répond par son agent vendeur (7.1) (8),

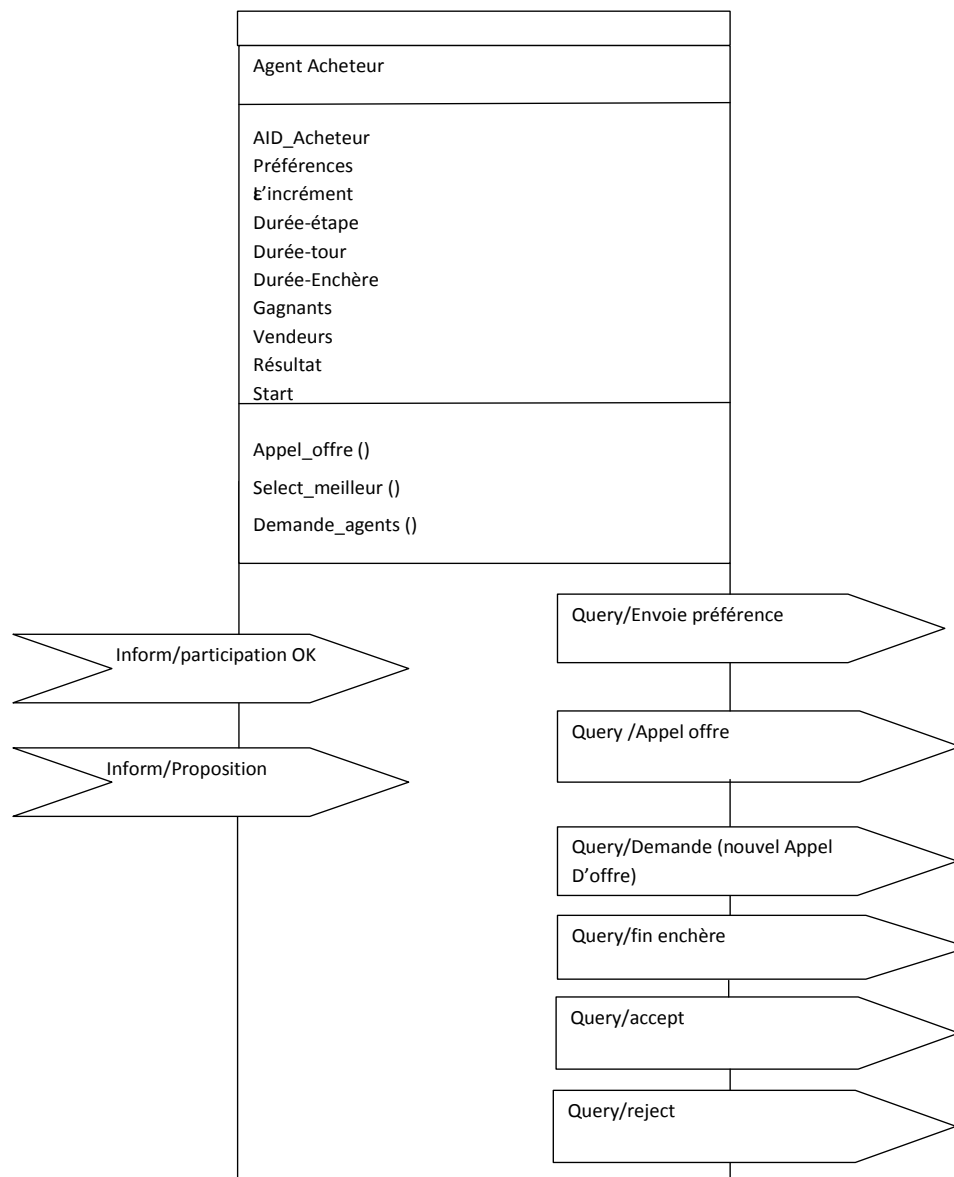


FIG. 5-1: Diagramme de classe AUML de l'agent Acheteur

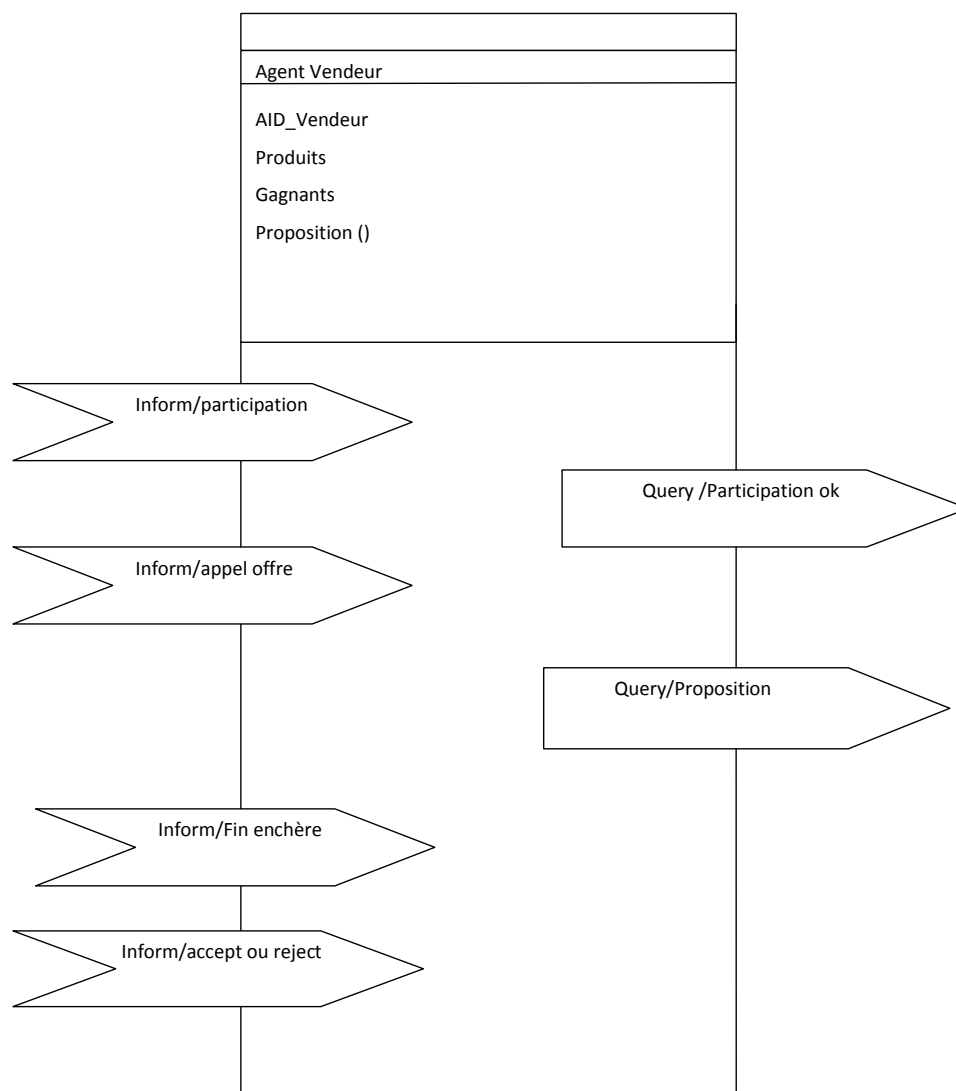


FIG. 5-2: Diagramme de classe AUML de l'agent Vendeur

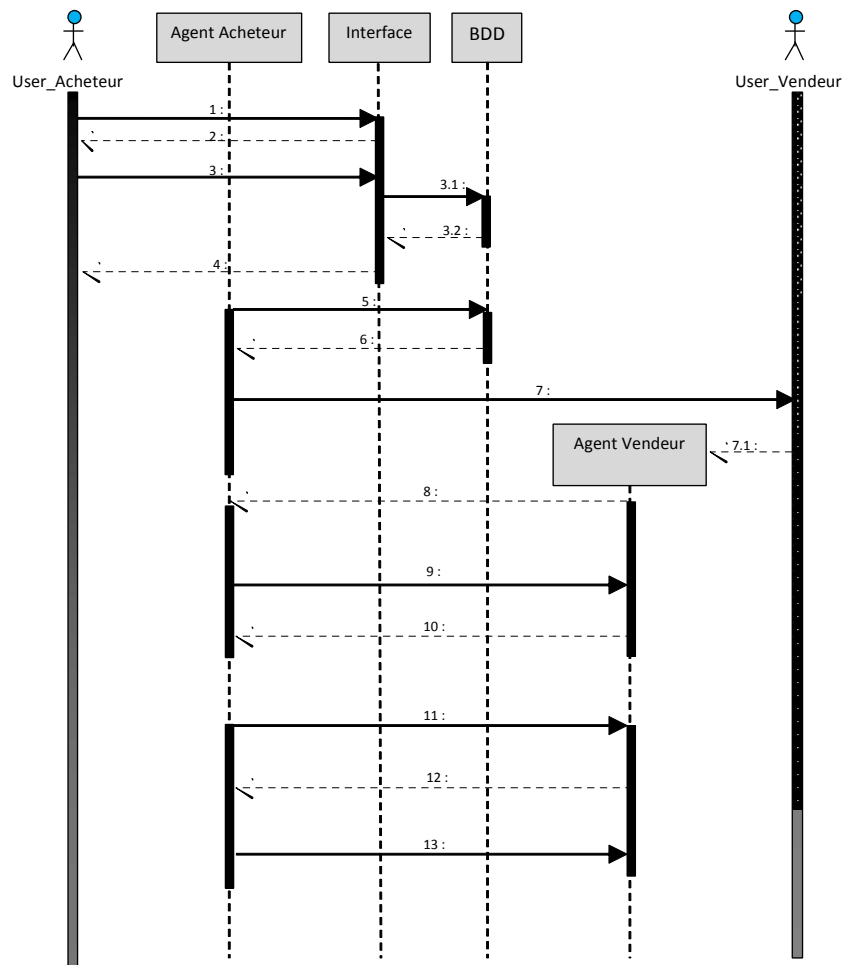


FIG. 5-3: Diagramme de séquence correspond à l'enchère

- Par la suite la négociation aura lieu(9) (10), ce processus se répète (11) (12),et en fin l'agent acheteur envoie aux agents vendeurs le nom du vainqueur(13).

5.7.5 Diagrammes AUML de vente aux enchères anglaise inversée

LA figure 5-4 présente le diagramme AUML qui présente une vente à l'enchère anglaise inversée qui illustre le principe de cette vente.

5.7.6 Diagramme d'activités

Le diagramme d'activité est une sorte d'organigramme correspondant à une version simplifiée du diagramme d'états .Il permet de modéliser des activités qui se déroulent en parallèle les unes des autres, quand ce parallélisme peut poser problème. Le but de ce diagramme est de mettre en évidence les contraintes de séquentialité et de parallélisme qui pèsent sur la tâche globale(Fig 5-5 et 5-6).

5.8 Réalisation

Afin de pouvoir réaliser notre système d'enchère multicritères basé sur la technologie agent, l'ensemble d'outils que nous avons utilisés comme support technique s'est porté sur un environnement Jbuilder2006 et une plateforme orientée agents Jade(Fig5-7)

5.8.1 Présentation de logiciel

L'acheteur

Cette fenêtre (Fig5-8) est utilisée par l'acheteur pour contrôler le processus de ce dernier, ce processus est composé de trois phases :1) initialisation (envoi préférences et les paramètres d'enchères, Fig5-9) ,2) la négociation (Fig5-10)et3) les résultats.

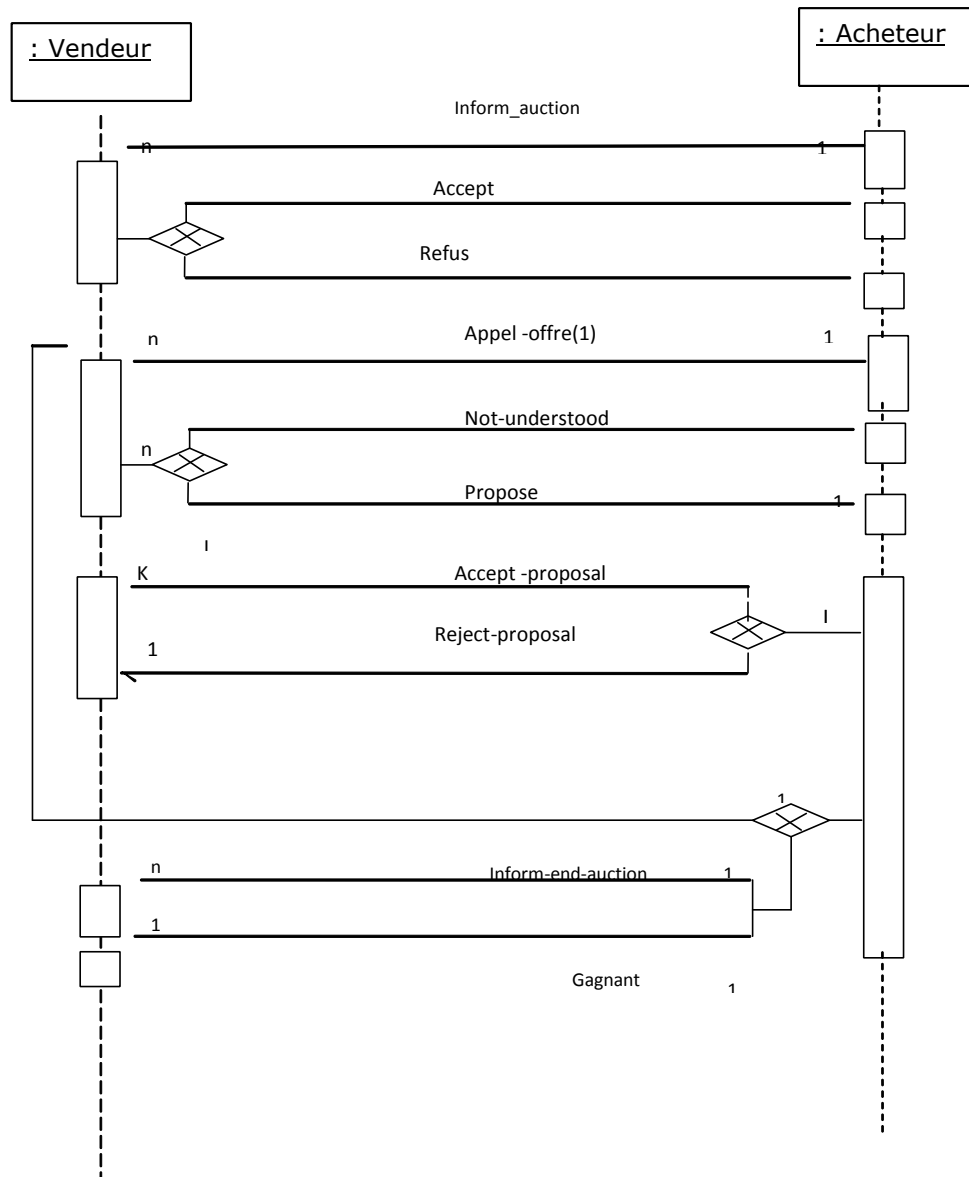


FIG. 5-4: Diagrammes AUML de vente aux enchères anglaise inversée

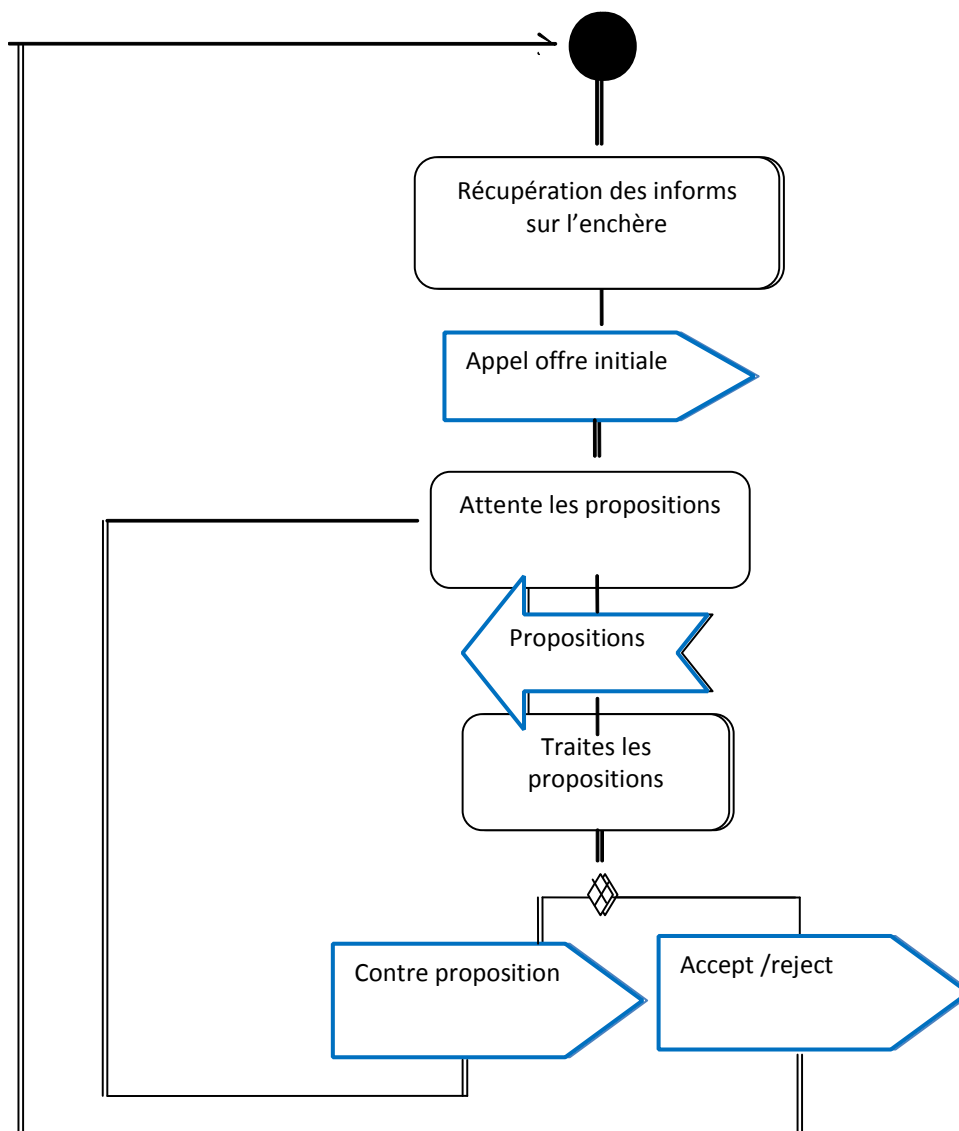


FIG. 5-5: Diagramme d'activité Acheteur

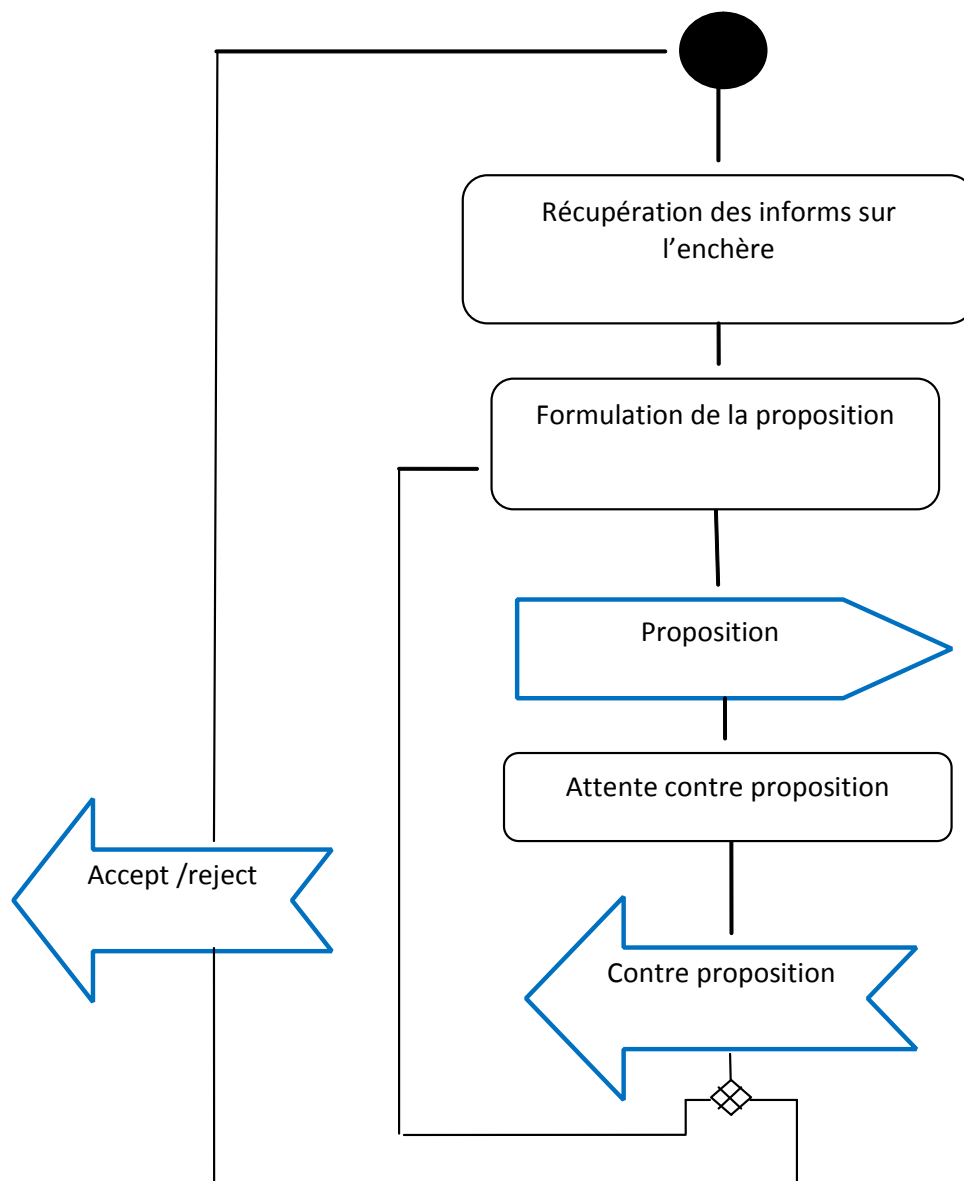


FIG. 5-6: Diagramme d'activité Vendeur

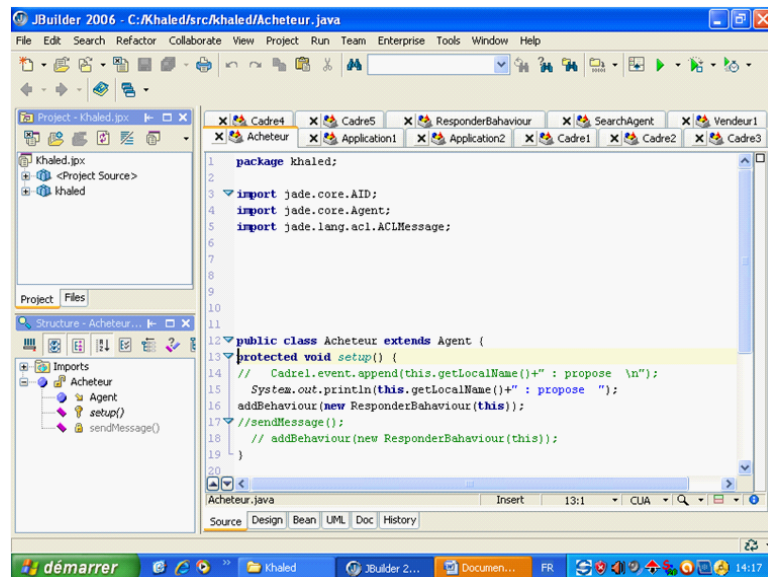


FIG. 5-7: L'environnement du système

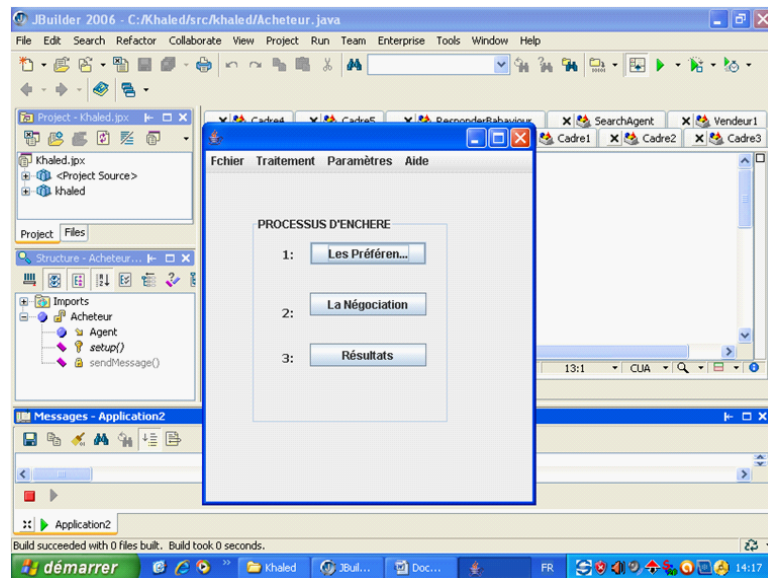


FIG. 5-8: L'interface de l'acheteur

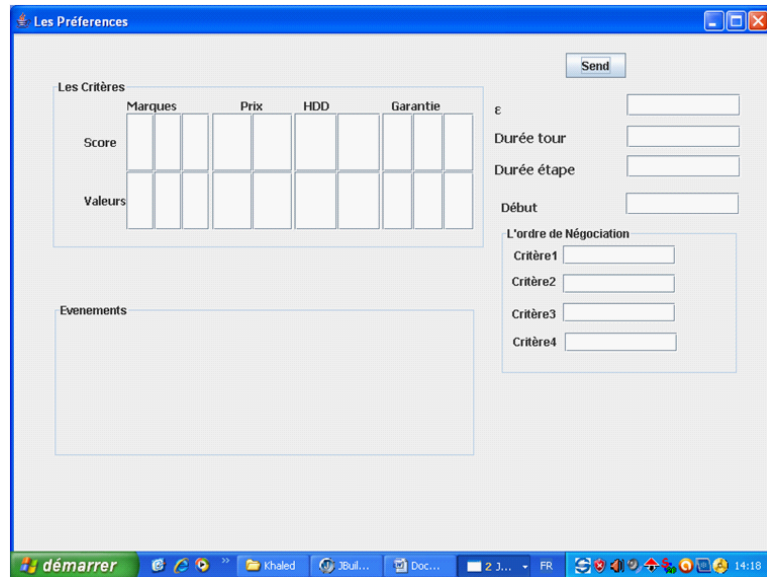


FIG. 5-9: L'interface pour les préférences

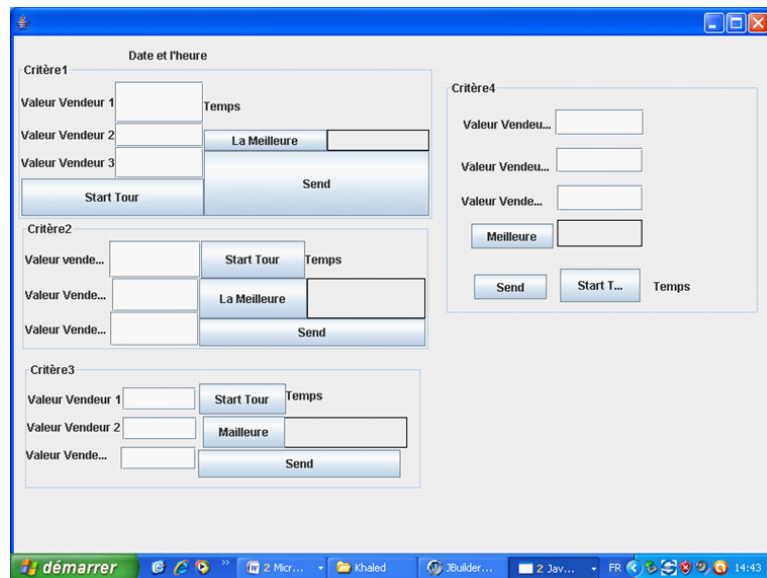


FIG. 5-10: Le processus de négociation

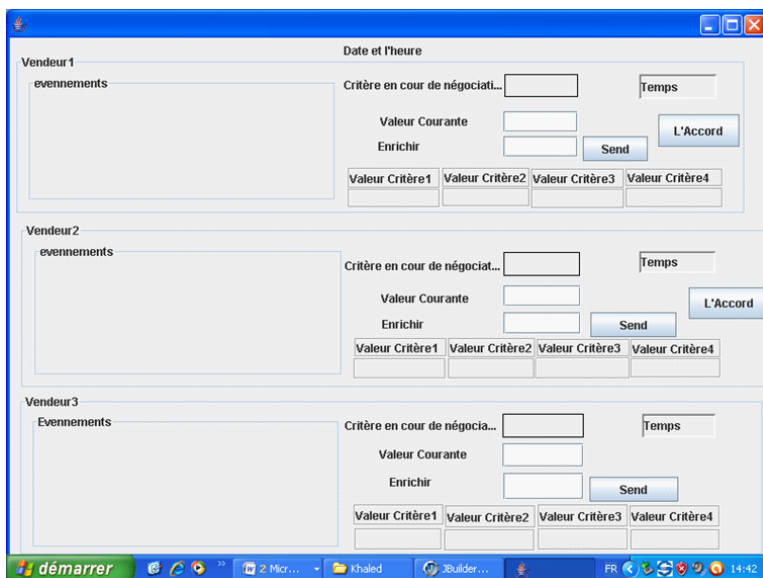


FIG. 5-11: L'interface des vendeurs

Les vendeurs

Dans cette fenêtre (Fig5-11) les vendeurs reçoivent les paramètres et les préférences de l'acheteur, ensuite entrent dans le processus de la négociation.

Conclusion et Perspectives

Dans ce document, on a décrit nos travaux de thèse qui consistent en un modèle de conception pour les ventes aux enchères multicritères anglaises inversées. Ce modèle repose sur l'utilisation du paradigme multi agents en vue d'automatiser le processus de négociation entre les agents. La raison pour laquelle nous utilisons les agents logiciels, est que les champs d'expertise sont fonctionnellement distribués dans différents domaines. Par exemple, afin de vendre un produit, on doit tenir compte de la connaissance du domaine auquel il appartient, de la connaissance et des stratégies de la société commerciale qui le vend et de bien d'autres facteurs.

On a supposé dans ce travail qu'un agent est une entité logicielle autonome qui perçoit partiellement son environnement au travers de capteurs, raisonne à son sujet et produit des changements dessus au moyen d'actions.

En utilisant les critères de Wooldridge [Woo99], l'environnement auquel nous avons affaire est caractérisé par les propriétés suivantes :

- Il est virtuel parce que les agents le perçoivent avec des capteurs logiciels et qu'ils agissent en utilisant des programmes et autres modules.
- Il est impossible que les agents en aient une vue complète.
- L'environnement n'est pas déterministe : les agents agissent mais il n'y a, a priori, aucune garantie de succès.
- Il ne peut pas être considéré comme épisodique : Le passé doit être pris en compte car il a une grande influence sur le présent, particulièrement dans des interactions commerciales.
- Il est fortement dynamique : les marchés, les bases de données ne cessent d'évoluer en reflétant l'activité mondiale humaine.
- Il doit être considéré comme continu car l'ensemble d'actions potentielles est infini.

Pour L'automatisation d'enchères multicritères on a focalisé sur les trois composants suivants, un module pour l'expression des préférences de l'acheteur, une méthode multicritères pour permettre à l'agent acheteur de sélectionner la meilleure offre et finalement un mécanisme qui permet à toute nouvelle proposition de battre la meilleure proposition reçue au tour précédent.

Suivant la classification de Vincke 1989 pour les méthodes d'aide à la décision multicritères on a présenté le déroulement d'enchère en trois modèles, le premier est basé sur la théorie de l'utilité, on a pris l'exemple de la somme pondérée et les moyennes (harmonique, quadratique, géométrique et arithmétique). Le deuxième modèle est basé sur les méthodes de sur classement, on a focalisé les ELECTREI et PROMTHEREI, avec une expérimentation de chaque méthode des deux modèles par un exemple d'achat d'un véhicule caractérisé par sept critères.

Suivant l'analyse des résultats et les critiques de chaque modèle, on a vu que toutes ces méthodes ne permettent pas d'atteindre l'optimalité de Pareto pour notre enchère multicritères. Le recours à l'enchère uni critère est indispensable surtout parce qu'il est intrinsèquement optimale aux sens de Pareto, alors on a proposé d'adapter ce modèle de vente uni critère en un modèle multicritères. Ensuite, on a donné à notre modèle une implémentation via un outil de modélisation orienté agent AUML.

Dans notre travail, on n'a pas tenu compte de la mobilité des agents. Une extension de ce travail concerne la prise en compte de la mobilité des agents lors de la distribution des SMA. Une autre extension est la conception d'un marché électronique basée sur les enchères en étudiant la sécurité, la transaction et paiement. Une perspective concerne l'implémentation. Il serait intéressant de mettre en œuvre ce travail sur autres plateformes de développement de SMA et d'utiliser d'autres outils, notamment pour l'aspect distribué.

Bibliographie

- [Adl89] M. R. Adler, A. B. Davis, R. Weihmayer, and R. W. Worrest. Conflict resolution strategies for non hierarchical distributed agents. In L. Gasser and M. Huhns, editors, *Distributed Artificial Intelligence Volume II*, pages 139–162. Pitman Publishing : London and Morgan Kaufmann : San Mateo, CA, 1989.
- [Bau00] Bauer, B., *Extending UML for the Specification of Interaction Protocols*, submitted for the 6th Call for Proposal of FIPA, 1999. Odell, James ed., *Agent Technology*, OMG, green paper produced by the OMG Agent Working Group, 2000.
- [Bea96] C. Bearn and A. Segev, "Electronic Catalogs and Negotiations", Working Paper 96-WP-1016C; enter for Management and Information Technology, 1996.
- [Bea97] C. Beam and A. Segev, "Automated Negotiations : A Survey of the State of the Art", Working Paper 97-W-1077, Center for Management and Information Technology, 1997.
- [Bel00] Bellifemine F, Giovanni C., Tiziana T. Rimassa G., "Jade Programmer's Guide" Jade version 2.6 .2000.
- [Ber99]] D. P. Bertsekas ET D. A. Castanon : Rollout algorithms for stochastic scheduling problems. *Journal of Heuristics*, 5 :89, 1999.
- [Cer05] L. Cernuzzi, M. Cossentino, and F. Zambonelli. Process models for agent-based development. *Journal of Engineering Applications of Artificial Intelligence*, Elsevier. Vol.18(2) :205–222, March 2005.
- [Cha61] A. Charnes et W. W. Cooper, 1961, *Management models and industrial applications of linear programming*, Vol. 1-2. New York : Wiley.

- [Cha96] A. Chavez et P. Maes, “Kasbah : An Agent Marketplace for Buying and Selling Goods”, Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, London, UK, 1996.
- [Cli98] Clint Heinze, Bradley Smith ET Martin Cross : Thinking quickly : Agents for modeling air warfare. In Proceedings of the 9th Australian Joint Conference on AI (Ai’98), Australie, 1998.
- [Del01] S. A. DeLoach. Analysis and Design using MaSE and agentTool. 12th Midwest Artificial Intelligence and Cognitive Science Conference(MAICS 2001), Miami University, Oxford, Ohio, March 31April 1, 2001.
- [Del05] S. A. DeLoach. Engineering Organization based Multi agent Systems. The 4th International Workshop on Software Engineering for Large scale multi agent Systems (SELMAS’05), May 1516, 2005, St.Louis, MO. Springer, LNCS vol. 3914,, pages 109 – 125.
- [Dem97] Y.DEMAZEAU. Steps toward multi-agent oriented programming .In Proceedings of the first International Workshop on Multi-Agent Systems – IWMASS97, 1997.
- [Dig00] F. Dignum, B. Dunin-Keplicz et R. Vebrugge : Agent theory for team formation by dialogue. In C. Castelfranchi et Y. Lespérance, éditeurs : Proceedings of the 7th International Workshop on Agent Theories, Architectures and Languages (ATAL’00), Boston, USA, 2000.
- [Dur89] 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, 1989.
- [Dur90] E. H. Durfee and A. Montgomery, Thomas. A hierarchical protocol for coordinating multi-agent behaviors. In Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI- 90), Boston, MA, August 1990.
- [Erc91] J. Erceau et J. Ferber. L’intelligence artificielle distribue. La recherche .Juin 1991.

- [Far98] P. Faratin, C. Sierra, and N. R. Jennings : Negotiation Decision Functions for Autonomous Agents. *Int. Journal of Robotics and Autonomous Systems* 1998.
- [Fer95] J. Ferber. *Les systèmes multi-agents : Vers une intelligence collective*. Inter Editions, 1995. [FLO02] A. Florea. Using Utility Values in Argument-based Negotiation. In *Proc. of IC-AI'02, the 2002 International Conference on Artificial Intelligence*, Las Vegas, Nevada, USA, 2002.
- [Fin93] Finin T. et al, *Specification of the KQML Agent-Communication Language*. Draft.1993.
- [Flo02] A. Florea. Using Utility Values in Argument-based Negotiation. In *Proc. of IC-AI'02, the 2002 International Conference on Artificial Intelligence*, Las Vegas, Nevada, USA, 2002.
- [Fox81] Fox M.S., *An organizational view of distributed systems*. *IEEE Trans. Syst.Man. - Univ. Cybern.* Vol. SMC-11 ; 1981, pp. 70-80.
- [Gen95] M. Genesereth, "KIF Interchange Format Specification", Knowledge Systems Laboratories, Stanford University, 1995.
- [Gru93] T.R. Gruber, "A Translation Approach to Portable Ontologies", *Knowledge Acquisition*, vol. 5, no 2, p. 199-220, 1993.
- [Gut98] R. Guttman, A. Moukas et P. Maes, "Agent-mediated Electronic Commerce : A Survey", *Knowledge Engineering Review*, vol. 13, n° 2, 1998, p. 147-159.
- [Igl97] C. Iglesias, M. Garrijo, J. Gonzalez and J. R. Velasco. *Analysis and Design of multi agent systems using MASCommonKADS*. *Intelligent Agents IV : Agent Theories, Architectures and Languages*, 1997, M. P.
- [Igl98] C.A. Iglesias, M. Garjo, J.C. Gonzàlez, and J.R. Velasco. *Analysis and Design of Multi agent Systems using MAS-CommonKADS*. In *AAA'97 Workshop on Agent Theories, Architectures and Languages*, Providence, RI, July 1997. ATAL. An extended version of this paper has been published in *INTELLIGENT AGENTS IV : Agent Theories, Architectures, and Languages*, Springer-Verlag, 1998.

- [Iji65] Ijiri, Y. (1965). *Management Goals and Accounting for Control*. Amsterdam, The Netherlands : North-Holland.
- [Jac98a] Jacques Ferber et Olivier Gutknecht. Aalaadin : a meta-model for the analysis and design of organizations in multi-agent systems. Dans ICMAS'98, juillet 1998.
- [Jac98b] Jacques Ferber et Olivier Gutknecht. A meta-model for the analysis and design of organizations in multi-agent systems. Dans *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS98)*, pages 128-135, Paris, France, 1998.
- [Jen93] Jennings J.R. Commitments and Conventions : The foundation of coordination in multi-agent systems. *The knowledge Engineering review*, 2(3) : 1993.
- [kae00] L.P. Kaelbling et Y. Chang : Playing is believing : The role of beliefs in multi-agent learning. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS'01)*, Canada, 2001.
- [Mar04] Marie Jo Bellosta, Imene Brigui, Sylvie Kornman, Daniel Vanderpooten. « A Multicriteria Model for Electronic Auctions ». LAMSADE, Université de Paris Dauphine .2004.
- [Mul96] H. J. Muller. Negotiation principles. Dans *Foundations of Distributed Artificial Intelligence*, G.M. P. O'Harre et N. R. Jennings (Editeurs), John Wiley & Sons, 1996.
- [Nwa96] Nwana H. S, « Software Agents : An Overview », *Knowledge Engineering Review*, vol. 11, n°3, p. 205–244, 1996.
- [Pat92] Patil R. ; Fikes R. ; Patel-Schneider P. ; Mckay D. ; Finin T. ; Gruber T. ; Neches R., The DARPA knowledge sharing effort : Progress report In : *Principles of Knowledge Representation and reasoning : Proceedings of the Third International Conference*, November 1992.
- [Pau03] S. Paurobally, R. Cunningham and N. R. Jennings, « Developing Agent Interaction Protocols using Graphical Methodologies », In *Workshop on Programming*

MAS, AAMAS, 2003.

- [Pes97] Pesty S., Brassac C., et Ferrent P., Ancrer les agents cognitifs dans l'environnement. In : Actes des 5e Journées Francophones sur l'Intelligence Artificielle Distribuée et les Systèmes Multi-Agents. – La Colle sur Loop, Avril 1997.
- [Pru81] D.G. Pruitts. Negotiation Behaviour. Academic Press, New York, 1981.
- [Pnu86] Pnueli Amir. Specification and development of reactive systems (invited paper). In IFIP Congress, pages 845–858, 1986.
- [Rai82] H. Raiffa : The Art and Science of Negotiation. Harvard University Press, Cambridge, USA, 1982.
- [Ros94] J. Rosenschein and G. Zlotkin. Rules of Encounter : Designing Conventions for Automated Negotiations among Computers MIT Press, 1994.
- [Roy85] Roy B., Skalka JM. Electre Is, aspects méthodologiques et guide d'utilisation. Document 30, Lamsade. Paris : Université de Paris Dauphine.1985.
- [Roy89] Roy, B, “Main Sources of Inaccurate Determination, Uncertainty and Imprecision in Decision Models”, Mathematical and Computer Modelling, 1989.
- [Roy92] Roy, B. “Science de la décision ou Science de l'aide à la décision ?”, Revue Internationale de Systémique, Vol. 6, 5, 497-529 or “Decision science or Decision-aid science ?”, E.J.O.R., Vol. 66, 2, April 1993, 184-203.1992.
- [Roy96] Roy, B. and Vanderpooten, “The European School of MCDA : Emergence, Basic Features and Current Works”, J.M.C.D.A.1996.
- [Rus97] Russell, S.J. Rationality and intelligence. Artificial Intelligence, Vol. 94, 1997. p.57-77.
- [Rus03] Stuart J. Russell et Peter Norvig. Artificial Intelligence. A Modern Approach. Prentice Hall, 2003.
- [Sat89] A. Sathi and M. S. Fox. Constraint-directed negotiation of resource allocations. In L. Gasser and M. Huhns, editors, Distributed Artificial Intelligence Volume II,

- pages 163–194. Pitman Publishing : London and Morgan Kaufmann : San Mateo, CA, 1989.
- [Smi88] R. G. Smith. The contract net protocol : High-level communication and control in a distributed problem solver. In A. H. Bond and L. Gasser, editors, Readings in Distributed Artificial Intelligence. Morgan Kaufmann Publishers : San Mateo, CA, 1988.
- [Ste88] R. Steeb, S. Cammarata, F. A. Hayes-Roth, P. W. Thorndyke, and R. B. Wesson. Distributed intelligence for air fleet control. In A. H. Bond and L. Gasser, editors, Readings in Distributed Artificial Intelligence. Morgan Kaufmann Publishers : San Mateo, CA, 1988.
- [Syc98] K. P. Sycara. Multi agent Systems. AI Magazine, vol. 10, 1998.
- [Syc89] K. P. Sycara. Multi-agent compromise via negotiation. In L. Gasser and M. Huhns, editors, Distributed Artificial Intelligence Volume II. Pitman Publishing : London and Morgan Kaufmann : San Mateo, CA, 1989.
- [Tve01] A. Tveit. A survey of agent-oriented software engineering. May 2001.
- [Van90] Vansnick, J.C, “Measurement theory and decision aid”, in Bana e Costa (ed.), Readings in Multiple Criteria Decision Aid, Springer-Verlog, Berlin,1990.
- [Var92] Varian, H.R, Microeconomic analysis, W.W. Norton & Co, New York City,1992.
- [Vin89] Vincke P. L’aide multicritère à la décision. Bruxelles : Éditions de l’Université de Bruxelles, 179 p, 1989.
- [Wei99] G.Weiss. Multiagent systems.A modern approach to distributed artificial intelligence.MIP Press,Cambridge,United Kigdom, 1999.
- [Wie86] A.P. Wierzbicki. On the completeness and constructiveness of parametric characterizations to vector optimization problems. OR Spektrum, 8(2) :73–87, 1986.
- [Woo95] Wooldridge, M. et N. R. Jennings. Agent theories, architectures, and languages. Intelligent Agents, Springer Verlag, 1995.
- [Woo99] Varian, H.R, Microeconomic analysis, W.W. Norton & Co, New York City,1992.

- [Woo00a] Wooldridge M, Jennings N. R. and. (2000), «Agent-Oriented Software Engineering» in Handbook of Technology (ed. J. Bradshaw) AAAI/MIT Press.2000.
- [Woo00b] Wooldridge M., Jennings N.R., and Kinny D. (2000), «The Gaia Methodology For Agent-Oriented Analysis and Design».
- [Yve06] Yves De Smet. A MULTICRITERIA PERSPECTIVE ON REVERSE AUCTIONS. Université. Libre de Bruxelles.2006.
- [Zam00] Zambonelli F., Jennings N. R., and Wooldridge M. (2000), «Organisational Abstractions for the Analysis and Design of Multi-Agent Systems» Proc. 1st Int. Workshop on Agent-Oriented Software Engineering, Limerick, Ireland, 127-141.

ABSTRACT

Application des Systèmes Multi Agents pour les enchères optimales aux sens de Pareto

par

Khalid Ghoul

Université Ferhat Abbas, Sétif , Avril, 2011

In recent years the one hand, great progress has been made in electronic commerce, especially in auctions. And other multi-agent systems (MAS) are widely used, particularly for applications requiring complex interaction between multiple entities. This research is devoted to the formalization of an optimal auction Pareto using a multi-agent system. This system consists of agents of transaction based on the communication by sending a message ; it is called a model of direct interaction between agents. In this memory we present a real model of negotiation which corresponds to the Multi criteria reverse English auction and the application of the concepts of agent and multi-agent systems to this last, and incorporating Pareto optimality to this auction.

We use a model of direct communication between actors in the system, to express the concept of multi criteria many approaches exist, but we focus on the most used and most consistent with our objective (Pareto optimality). The modeling of multi-agent systems is a very important factor for the development of such a system, we choose to use the most effective method for our system is AUML, For the realization of this multi agent system , we use the JADE platform.

Keywords : Multi-Agent Systems (MAS), Electronic Commerce, AUML, Multi criteria Negotiations, Agents of interactions, methods of Multi criteria analysis. Thesis and dissertation creation is hard work and a poor graduate student needs all the help he or she can get.

Résumé : Ces dernières années d'une part, de grands progrès ont été enregistrés dans le commerce électronique, en particulier dans les ventes aux enchères. Et d'autre part les systèmes multi-agents (SMA) sont très largement utilisés, particulièrement pour les applications complexes nécessitant l'interaction entre plusieurs entités. Ce travail de recherche est consacré à la formalisation d'une vente aux enchères optimale aux sens de Pareto à l'aide d'un système multi agents. Ce système est constitué d'agents de transaction s'appuyant sur la communication par envoi de message, on parle d'un modèle d'interaction directe entre les agents. On présente ici un modèle de négociation réel qui correspond à la vente aux enchères anglaises inversées de type multicritères et l'application des notions d'agent et systèmes multi-agents à cette dernière, ainsi l'intégration d'optimalité de Pareto à cette vente. Nous utilisons un modèle de communication directe entre les acteurs du système, ainsi pour pouvoir exprimer la notion de multicritères différentes approches existent, mais nous concentrons sur les plus utilisées et les plus correspondent à notre objectif (optimalité de Pareto). La modélisation des systèmes multi agent est un facteur très important pour le développement d'un tel système, nous choisissons d'utiliser la méthode la plus efficace pour notre système qui est AUML, pour la réalisation de ce dernier une plateforme multi agent doit être utilisé, alors nous utilisons la plateforme JADE.

Mots-clés : Systèmes Multi-Agents (SMA), Commerce électronique, AUML, Négociations Multicritères, Agents d'interactions, Les méthodes d'analyse multicritères.

ملخص:

في السنوات الأخيرة من جهة، التجارة الالكترونية عرفت تقدما كبيرا و خاصة البيع بالمزاد و من جهة أخرى الأنظمة المتعددة الوكلاء تستخدم على نطاق واسع، وخاصة بالنسبة للتطبيقات المعقدة التي تتطلب التفاعل بين كيانات متعددة. هذا البحث العلمي يتضمن تكوين مزاد يكون مثالي وفق Pareto، يتكون هذا المزاد من مجموعة من الوكلاء المبرمجين و الذين يعتمدون على إرسال الرسائل كطريقة التواصل و التحدث. نقدم نموذجا للتفاوض الذي يتوافق مع المزاد الإنجليزي العكسي المتعدد المعالم و تطبيق نظام الوكلاء المبرمجين على هذا المزاد بحيث يكون مثالي وفق Pareto. نحن استخدام نموذج الاتصال المباشر بين الجهات الفاعلة في النظام، من أجل التعبير عن فكرة تعدد المعالم هناك عدة طرق، ولكن نحن نركز على الأكثر استخداما والأكثر تراسقا مع هدفنا (مثالية Pareto). تحقيق نموذج للأنظمة متعددة الوكلاء هو عامل مهم جدا لتطوير مثل هذا النظام، نختار استخدام الأسلوب الأكثر فعالية لنظامنا AUML و من أجل تحقيق هذا النظام اخترنا أرضية Jade.

الكلمات المفتاحية: النظام المتعدد الوكلاء، التجارة الالكترونية، AUML، المفاوضات المتعددة المعالم، طرق التحليلي المتعددة المعالم، الوكلاء .