



**Faculté des sciences exactes et des sciences de la nature et de la vie  
Département d'informatique**

---

# **Une approche formelle pour la négociation automatique d'un e-commerce à base d'agent**

## **Thèse**

**Pour obtenir le diplôme de  
Docteur en Sciences  
En  
Informatique**

Présentée par  
**Saïd Bachir**

Soutenue le :

**Devant le jury composé de**

**Président :** Professeur BENMOHAMED Mohamed de l'université de Constantine

**Rapporteur :** Professeur KAZAR Okba de l'université de Biskra

**Examineur :** Professeur CHAOUI Allaoua de l'université de Constantine

**Examineur :** Dr. KHELIL Nacer, maître de conférences A de l'université de Biskra

**Examineur :** Dr. TERRISSA Labib Sadek, maître de conférences A de l'université de Biskra

**Invitée:** Dr. BENHARKAT Aicha-Nabila maître de conférences A de l'INSA de Lyon

## **Résumé**

Le système multi-agents est parmi les technologies qui permettent la simulation du comportement humain. Lorsque de nombreux agents interagissent, des conflits peuvent survenir. Pour les résoudre, de différentes méthodes peuvent être utilisées, comme la coordination, le système de vote et la négociation. La négociation peut prendre diverses formes, (les systèmes d'enchères, la négociation heuristique, la négociation par argumentation ... etc.).

La négociation par argumentation, qui est parmi les approches de négociation les plus récentes, est adoptée dans cette thèse, en proposant une formalisation basée sur le web sémantique des mécanismes de négociation du commerce électronique. Cette formalisation comprend la représentation des éléments de l'argument, des forces conduisant la prise de décision et du protocole de négociation. Les technologies du web sémantique sont également utilisées dans la représentation de toutes les connaissances qui concernent les éléments du marché électronique (produits, acheteurs et vendeurs).

## **Abstract**

The multi-agent system is among technologies which allow the simulation of the human behavior. When many agents interact, conflicts can occur. To resolve them, various methods can be used, like coordination, voting system and negotiation. The negotiation can take different forms, (auction systems, heuristic negotiation, argumentation based negotiation ... etc).

The argumentation based negotiation, which is among the most recent negotiation approaches, is adopted in this thesis, by proposing a semantic Web based formalization of the e-commerce negotiation mechanisms. This formalization comprises the representation of the argument elements, the forces leading decision making and the protocol of negotiation. Semantic Web technologies are also used in the representation of all knowledge concerning the e-market elements (goods, buyers and sellers).

## ملخص

يعتبر النظام متعدد العملاء من بين التقنيات التي تسمح بمحاكاة السلوك البشري. عندما يتفاعل العديد من العملاء قد يحدث بينهم خلافات. و لحل هذه الخلافات يمكن استخدام عدة طرق، كالتنسيق، نظام التصويت و التفاوض. التفاوض قد يأخذ عدة أشكال ( أنظمة المزاد، التفاوض بالاستكشاف، التفاوض بالحناجة... الخ). إن التفاوض بالحاجة و الذي يعتبر من أحدث أنواع التفاوض المقترحة، قد تم اعتماده في هذه الأطروحة باقتراح شكلنة مبنية على الشبكة المعنوية لآليات التفاوض في مجال التجارة الإلكترونية. هذه الشكلنة تشمل تمثيل عناصر الحجة، مجموعة القوى المؤثرة في اتخاذ القرار و كذا بروتوكول التفاوض. لقد تم توظيف تقنيات الشبكة المعنوية كذلك في تمثيل جميع المعارف المتعلقة بالسوق الإلكتروني (المنتجات، المشترون و الباعة).

*À mon pays ...*

*À mes parents, mon frère, mes trois sœurs ...*

*À ma chère femme, mes enfants: Hussam, Mouatassim et Amdjed ...*

## Remerciements

Avant de présenter ce travail, je tiens à remercier Dieu tout puissant, de m'avoir permis d'arriver à ce niveau d'étude, et aussi de m'avoir donné beaucoup de patience et de courage.

A travers cette modeste thèse je tiens à présenter mes sincères remerciements et ma profonde reconnaissance à mon directeur de thèse Monsieur Kazar Okba, Professeur à l'université de Mohamed Kheidar de Biskra, que je respecte beaucoup.

J'adresse également mes remerciements aux personnes qui m'ont fait l'honneur de participer, au jury de cette thèse : Monsieur Benmohamed Mohamed, Professeur à l'université de Constantine, Monsieur Chaoui Allaoua, Professeur à l'université de Constantine, Monsieur Khelil Nacer, Maître de conférences « A » à l'université de Biskra, Monsieur Terrissa Labib Sadek, Maître de conférences « A » à l'université de Biskra et Madame Benharkat Aicha-Nabila Maître de conférences de l'INSA de Lyon.

J'adresse aussi mes sincères reconnaissances à tous mes collègues de l'université Kasdi Merbah de Ouargla pour leurs aides, soutiens et

conseils : Mr. le recteur Pr. Boutarfaia Ahmed, les membres de l'équipe  
GRAIN, les enseignants, sans oublier tout le staff administratif.

Grand merci à tous ceux et toutes celles qui ont contribués de près ou  
de loin dans l'accomplissement de ce travail.

# Table des matières

<b>Liste des figures</b>	vii
<b>Liste des tableaux</b>	ix
<b>1. Introduction</b>	1
1.1. Contexte du travail .....	1
1.1.1. Des agents intelligents aux systèmes multi-agent .....	1
1.1.2. Domaines d'application des SMA .....	3
1.1.3. SMA et Commerce électronique .....	3
1.1.4. Négociation à base d'agents en e-commerce .....	4
1.1.4.1. Définitions .....	4
1.1.4.2. Classifications des systèmes de négociation .....	6
1.2. Motivation et objectifs .....	8
1.3. Contributions de l'étude .....	9
1.4. Structuration de la thèse .....	10
<b>2. Chapitre2. Travaux connexes</b>	12
2.1. Approches théoriques de l'argumentation et de la prise de décision .....	12
2.1.1. Modèle d'argument de Toulmin .....	12
2.1.2. Systèmes abstraits d'argumentation .....	15
2.1.3. Technique d'analyse de champ de force .....	18
2.2. Frameworks de négociation à base d'argumentation .....	20
2.2.1. Espace inter-agents .....	21
2.2.1.1. Langage de négociation .....	22
2.2.1.2. Protocole de négociation .....	23
2.2.1.3. Espace de Partage et de stockage .....	24
2.2.2. Espace intra-agent .....	24
2.2.2.1. Mécanisme d'évaluation .....	24
2.2.2.2. Mécanisme de génération .....	25
2.2.2.3. Mécanisme de sélection .....	26
<b>3. Chapitre 3. Contribution 1 : Framework de négociation proposé</b>	27
3.1. Introduction .....	27

3.2. Représentation de l'espace de négociation inter-agents .....	27
3.2.1. Langage de négociation .....	29
3.2.2. Protocole de négociation .....	37
3.3. Représentation de l'espace de négociation intra-agent .....	38
3.3.1. Les connaissances locales .....	39
3.3.1.1. Les croyances .....	40
3.3.1.2. Les désirs .....	40
3.3.1.3. Les pré-requis .....	40
3.3.2. Les mécanismes de négociation .....	41
3.3.2.1. Mécanismes d'évaluation .....	41
3.3.2.2. Mécanismes de génération et de validation .....	45
3.3.2.3. Mécanisme de sélection .....	46
<b>4. Chapitre 4. Contribution 2 : Architecture du système</b> .....	47
4.1. Introduction .....	47
4.2. Le web sémantique en tant qu'outil de représentation et de raisonnement...	47
4.3. Architecture de l'espace de négociation inter-agents .....	50
4.4. Architecture interne des agents (Vendeurs et Acheteurs) .....	55
4.5. La représentation des mécanismes de négociation par les règles SWRL ....	57
4.5.1. Les règles d'évaluation de la locution entrante .....	58
4.5.2. Les règles de génération des locutions candidats .....	61
4.5.3. La règle de sélection .....	64
4.6. Le processus d'achat .....	65
<b>5. Chapitre 5. Contribution 3 : Expérimentation</b> .....	67
5.1. Introduction .....	67
5.2. Les outils et les plateformes utilisés .....	67
5.2.1. Présentation de l'éditeur d'ontologie Protégé .....	67
5.2.2. La plate-forme JADE .....	68
5.2.2.1. Description .....	68
5.2.2.2. Architecture logiciel de la plate-forme JADE .....	69
5.2.2.3. Langage de communication de la plate-forme JADE (FIPA- ACL) .....	70
5.2.3. Présentation du prototype .....	70



5.2.4. Architecture globale .....	70
5.2.5. Présentation des interfaces utilisateur .....	71
5.2.5.1. L'interface d'accueil .....	72
5.2.5.2. L'interface du vendeur (VendeurUI) .....	72
5.2.5.3. L'interface de l'acheteur (AcheteurUI) .....	74
5.2.6. Les ontologies .....	75
5.2.7. Les agents (Vendeur et Acheteur) .....	77
5.3. Exemple d'expérimentation .....	82
5.3.1. Description du scénario .....	82
5.3.2. Résultats .....	86
5.3.3. Discussion .....	87
5.4. Conclusion .....	88

**Conclusion et perspectives**

**Bibliographie**

## Liste des figures

2.1	Diagramme d'argument de Toulmin .....	14
2.2	Exemple de système abstrait d'argumentation .....	17
2.3	Exemple de diagramme d'analyse de champ de force .....	20
2.4	Espace inter-agents dans un framework de NA .....	22
2.5	Espace intra-agent dans un framework de NA .....	25
3.1	Architecture de l'espace inter-agent du framework de négociation .....	28
3.2	Représentation du protocole de négociation .....	38
3.3	Les éléments de l'espace intra-agent .....	39
3.4	Espace de négociation entre deux propositions .....	43
3.5	Les forces définies dans le champ de la négociation .....	44
4.1	Architecture de l'espace inter-agent d'un e-marché de Laptops .....	51
4.2	Description UML de l'ontologie <i>Laptop</i> .....	52
4.3	Description UML de l'ontologie <i>Protocole</i> .....	53
4.4	Description UML de l'ontologie <i>Locution</i> .....	54
4.5	Description UML de l'ontologie <i>Marché</i> .....	54
4.6	Architecture interne d'un agent (Vendeur / Acheteur) .....	55
4.7	Description UML de l'ontologie <i>Acheteur</i> .....	56
4.8	Description UML de l'ontologie <i>Vendeur</i> .....	57
5.1	L'éditeur d'ontologie Protégé version 3.4.4 .....	68
5.2	Architecture logiciel de la plate-forme JADE .....	70
5.3	L'architecteur globale du prototype .....	71
5.4	La fenêtre d'authentification .....	73
5.5	L'interface du vendeur .....	73
5.6	L'interface de l'acheteur .....	75
5.7	FSMBehaviour de l'agent Acheteur sur l'inspecteur de JADE .....	80
5.8	Spécification des ordres de préférence de l'acheteur .....	83
5.9	Spécification des poids des attributs pour l'acheteur .....	83
5.10	Spécification des désirs de l'acheteur .....	84
5.11	Spécification des pré requis de l'acheteur .....	84

5.12 Capture de l'échange de messages par Sniffer de JADE .....	86
5.13 Evaluation des propositions des agents Vendeur .....	87

## Liste des tableaux

2.1	Exemple d'argument formé selon le modèle de Toulmin .....	14
3.1	Description des locutions non persuasives .....	29
3.2	Description des locutions persuasives .....	30
3.3	Les éléments d'une locution persuasive .....	31
3.4	Éléments de persuasion utilisés par l'agent acheteur .....	32
3.5	Éléments de persuasion utilisés par l'agent vendeur .....	33
3.6	Représentation formelle des locutions persuasives d'un agent acheteur ...	35
3.7	Représentation formelle des locutions persuasives d'un agent vendeur ...	37
3.8	Les forces influencées par les locutions persuasives .....	44
5.1	Croyances, désirs et pré requis sur les attributs négociés .....	85
5.2	Ordre de popularité supposé sur les caractéristiques des Laptops .....	85

# 1

## Introduction

Le commerce électronique est l'un des domaines qui bénéficie de l'évolution de l'informatique ainsi de plusieurs autres domaines. Le commerce repose sur trois principaux éléments : d'une part le vendeur qui veut maximiser son profit. D'autre part, l'acheteur qui désire un produit spécifique au moindre coût et le produit au milieu. Cette contradiction dans les désirs des vendeurs et des acheteurs peut les entraîner dans un état de conflit. L'informatique, à partir des systèmes multi-agents, a offert une solution des conflits par l'utilisation de la négociation à base d'agent.

La négociation est un processus grâce auquel plusieurs parties aboutissent à une décision commune. La négociation prend plusieurs formes tels les systèmes de vote, les enchères, la négociation par argumentation...etc.

Dans ce qui suit, nous présentons le contexte du travail, l'objectif et les motivations de l'étude, les contributions de l'étude et enfin un guide de lecture de la thèse.

### **I.1. Contexte du travail**

#### **I.1.1. Des agents intelligents aux systèmes multi-agent**

Le concept d'agent a été étudié pendant plusieurs années dans le domaine d'intelligence artificiel, mais la notion d'agent est inspirée des sciences humaines, tel que la philosophie et la psychologie. Il existe plusieurs définitions du terme agent, la définition de Jacques Ferber [1995] est parmi les définitions les plus citées.

**Définition 1** [Ferber, 1995] : Un agent est une entité réelle ou virtuelle qui :

- est capable de percevoir et d'agir (avec des limites) dans un environnement,
- peut communiquer d'autres agents,
- a des objectifs à atteindre ou des besoins à satisfaire,
- possède des ressources propres et des compétences et offre des services,
- et peut éventuellement se reproduire.

Mais cette définition est très vaste. Jennings, Sycara et Wooldridge [Jennings et al., 1998] ont donné une définition plus précise de l'agent.

**Définition 2** [Jennings et al., 1998]: Un agent est un système informatique, *situé* dans un certain environnement et qui est capable d'agir de façon *flexible* et *autonome* afin d'atteindre des objectifs spécifiés lors de sa conception. Dans cette définition trois concepts clés sont employés pour décrire l'agent, *situé* veut dire qu'il peut interagir avec l'environnement par la réception des entrées sensorielles et l'exécution des actions qui changent l'environnement, *autonome* signifie qu'il peut agir sans avoir besoin d'intervention des humains (ou des autres agents) et *flexible* veut dire:

- *répondre bien*: percevoir et répondre à temps,
- *proactif*: exhiber un comportement opportuniste dirigé par ses objectifs avec la capacité de prendre l'initiative au moment approprié,
- *social*: peut interagir avec les agents et les humains pour résoudre leur propre problème et pour aider d'autres avec leurs activités.

Nous pouvons déduire de cette définition que l'agent a une capacité d'interagir dans un environnement, de participer à la résolution des problèmes et d'aider d'autres agents, ce qui nous permet de concevoir des agents qui ont des objectifs communs à atteindre, ces agents forment ensemble un système, qui s'appelle système multi-agent.

Plusieurs définitions des Système Multi-Agent (SMA) ont été proposées. Nous retenons seulement la définition de Jacques Ferber [1995].

**Définition 3** [Ferber, 1995] : Un SMA est un système composé de six éléments:

1. Un environnement  $E$ , c'est un espace qui dispose généralement d'une métrique.
2. Un ensemble d'objets  $O$  qui sont :
  - *situés* : ils peuvent avoir une position dans  $E$  à un moment donné.
  - *passifs* : ils peuvent être perçus, créés, détruits et modifiés par les agents.
3. Un ensemble d'agents  $A$  ( $A \subseteq O$ ), qui sont des objets actifs.
4. Un ensemble de relations  $R$  qui lient des objets (passifs ou actifs) entre eux.
5. Un ensemble d'opérations  $Op$  permettant aux agents de percevoir, produire, consommer, transformer et manipuler des objets passifs.

6. Des opérateurs, qui représentent l'application des opérations et la réaction du monde.

L'avantage de l'application des SMA [Jennings et al., 1998] est qu'elle permet une résolution distribuée et concourante des problèmes, ainsi qu'elle présente des patterns sophistiqués d'interaction, tel que la coordination, la coopération et la négociation. Mais dans la conception des SMA plusieurs difficultés peuvent se poser [Franklin, 1997 ; Iglesias, 1997] sur:

- La manière de formulation, description, décomposition, et allocation des problèmes et synthèse des résultats?
- La manière de communication et d'interaction.
- L'existence de cohérence dans l'interaction entre agents.
- ...

### **I.1.2. Domaines d'application des SMA**

Les problèmes qui inhérent la conception des SMA n'ont pas empêché l'utilisation des SMA dans de nombreux domaines scientifique, tel que :

- SD (systèmes distribués) [Fagin, 1995; Halpern, 1995; Glaser, 1996].
- BDD et BCD (Bases de Données et Bases de Connaissances Distribuées) [Babin, 1997].
- TALN (Traitement automatique de Langages Naturels) [Allen & Perrault, 1980 ; Cohen & Levesque, 1990; Galliers, 1988; Kaplan, 1998].
- IHM (Interface Homme Machine) [Negroponte, 1995; Kobsa, 1989].
- Simulation [Fischer et al., 1999; Klügl, 2009; Treuil et al., 2008].
- Protocoles de communication et réseaux de télécommunications [Bourron, 2001; Nwana & Ndumu, 1999].
- Génie logiciel [Jennings, 2000]
- Robotique [Lakemeyer, 1999; Lespérance, 1991; Lespérance, 1994; RoboCup].
- Applications Web et Internet [Chaib, 1996; Jennings & Wooldridge, 1998].
- Web Sémantique [Berners-Lee & col., 2001] et Services Web [Huhns, 2003].

### **I.1.3. SMA et commerce électronique**

L'application des SMA dans le Web est accompagnée par la naissance d'un nouveau concept, c'est *l'agent électronique* ou *l'agent web* [Eichmann, 1994].

**Définition 4:** Eichmann [1994] a défini l'agent web comme un programme qui explore, de façon autonome, la structure du Web pour effectuer certaines actions sur les objets qu'il rencontre.

L'agent électronique ne doit pas être conçu seulement pour effectuer des tâches complexes, car il est capable d'effectuer des actions simples sur un très grand nombre d'objets dispersés. En générale, un agent électronique est capable d'être employé dans n'importe quel processus sur le web, tel que le processus d'achat en ligne. Guttman et al. [1998] ont adapté le modèle appelé CBB (Consumer Buying Behaviour) au commerce électronique. Selon eux un processus d'achat en ligne est basé sur six étapes:

1. *Identification de la demande:* Spécification du besoin.
2. *Courtage du produit:* Recherche du produit.  
Exemples : [EShopBooster; Amazon; E-Bay].
3. *Courtage du marchand:* Recherche du vendeur.  
Exemples : [ E-Bay; Price-Minister].
4. *Négociation:* Recherche d'un accord sur les termes de la transaction (prix, ...).  
Exemples : Kasbah [Chavez & Maes, 1996], AuctionBot [AuctionBot; Wurman et al, 1998], Tete-a-Tete [Guttman & Maes, 1998], Magnet [Magnet] et E-CAP (E-commerce Agent Platform) [E-CAP; Bartolini et al, 2005].
5. *Achat et livraison:* Options de paiement et de livraison.
6. *Service après-vente et évaluation :* Rappel et suivie.

Dans les systèmes du commerce électronique existants, les agents ne doivent pas forcément jouer tous les rôles décrits précédemment, au même temps nous avons trouvé que plusieurs plateformes de commerce électronique, basés sur les agents, s'intéressent à la fonctionnalité de négociation ce qui implique l'importance de la négociation dans le processus d'achat en ligne.

### **I.1.4. Négociation à base d'agents en e-commerce**

L'adoption rapide de l'Internet comme milieu commercial a apporté un changement significatif sur la gestion traditionnelle des affaires. Afin de rester en concurrence, beaucoup d'entreprises ont établi leur présence sur les marchés virtuels.

#### **I.1.4.1. Définitions**

He et Leung [2002] ont défini le commerce électronique comme :



**Définition 5 :** Toute forme de transactions commerciales ou administratives ou d'échange d'informations qui sont effectuées en utilisant des moyens de la technologie d'information et de communication.

Les activités de commerce électronique sont augmentées continuellement. Dans les activités commerciales du monde réel, plusieurs commerçants pourraient présenter le même produit ou service avec différentes valeurs des attributs de produit. En général un acheteur ne s'inquiète pas seulement d'un seul terme du produit ou du service, mais Il va considérer plusieurs termes et changer les différents paramètres de ces termes pour obtenir le meilleur choix. Dans l'autre coté, il est nécessaire que les vendeurs soient d'accord avec les modifications proposées par l'acheteur. Donc, pour que les marchés virtuels soient capables de jouer le même rôle des marchés du monde réel ils doivent prendre en compte la phase de négociation.

Aujourd'hui, plusieurs travaux de recherche s'intéressent à l'automatisation de la négociation dans l'e-commerce. Sur un marché électronique à base d'agents, les vendeurs et les acheteurs peuvent déterminer les différents termes des transactions sans avoir besoin de se rencontrer physiquement. Sur ce marché, les agents doivent être capables de communiquer itérativement entre eux et échanger des offres et des contre-offres afin d'arriver à un accord [Faratin et al., 1998]. Lomuscio et al. [2000] ont défini la négociation à base d'agents comme :

**Définition 6 :** un processus par lequel un groupe d'agents communiquent entre eux en essayant de parvenir à un accord mutuellement acceptable sur une certaine matière.

La négociation a été automatisée afin d'augmenter la vitesse de réponse, d'augmenter le volume des réponses et d'augmenter la qualité des réponses, mais la question posée : est ce que les agents peuvent vraiment surpasser des humains [Kephart, 2002]. Kephart à fait des expérimentations sur la négociation des agents et la négociation des être humains, il a trouvé que les agents sont mieux que des novices par ~ 20% et mieux que des expert par ~5-7%, mais ces résultats restent valables pour certains types de négociation. Pour l'instant le problème reste localisé sur la modélisation d'une négociation à base d'agents pour l'e-commerce. La modélisation d'un système de négociation à base d'agent nécessite la détermination de tous les composants de cette négociation. Il existe plusieurs décompositions de la négociation automatisée, Jennings et al. [2001] ont décomposé la négociation en trois éléments :

protocoles de négociation, objets de négociation et modèles de prise de décision des agents (stratégies). Un protocole de négociation est l'ensemble de règles qui gouvernent la négociation [Rosenshien & Zlotkin, 1994]. Les objets de négociation représentent les termes autour desquelles les participants négocient (prix, qualité, ...). Une stratégie est la spécification de la séquence des actions que l'agent prévoit d'effectuer durant la négociation [Lomuscio et al., 2000]. Le protocole de négociation avec les stratégies de négociation forment dans l'ensemble qu'on appelle le mécanisme de négociation.

### I.1.4.2. Classifications des systèmes de négociation

La négociation dans l'e-commerce peut avoir différents types de classification [Lomuscio et al., 2000; Wooldridge, 2002; Wurman et al., 1998], elle peut se faire sur plusieurs dimensions. Les dimensions considérées dans les classifications existantes dépendent directement, à notre avis, des composants de la négociation. Sur la dimension objets, la négociation peut se dérouler sur *un seul attribut* ou sur *plusieurs attributs* [Wooldridge, 2002]. Sur la dimension protocole, la classification dépend de deux facteurs. Le premier facteur est le mode d'interaction entre les agents impliqués, donc il peut avoir des négociations en *un acheteur-à-un vendeur*, en un *acheteur-à-plusieurs vendeurs* ou en *plusieurs acheteurs-à-plusieurs vendeurs* [Lomuscio et al., 2000]. Le deuxième facteur est le type de protocole choisi, nous trouvons que parmi les points les plus importants lors de la modélisation des systèmes de négociation est le choix de protocole, à cette raison, on trouve que les travaux de recherche sur la négociation se concentrent beaucoup plus sur l'optimisation des protocoles existants ou sur la proposition des nouveaux protocoles. Parmi les protocoles de négociation les plus connus dans l'e-commerce: la famille des protocoles d'enchère, par exemple AuctionBot [Wurman et al., 1998] et Fishmarket [Noriega, 1998], et le protocole Contract-Net [Smith, 1980] et ces optimisations. La dernière classification se base sur la stratégie. Les travaux de recherche dans cette dimension se concentrent sur la prise de décision des agents négociateurs. Il existe trois grandes familles des approches [Jennings et al., 2001]:

- **Approches basées sur la théorie des jeux:** La théorie des jeux est une branche de l'économie qui étudie les interactions entre les agents avec des intérêts locaux. Comme la théorie de la décision, avec laquelle il partage tous les concepts, la théorie des jeux a ses racines dans le réseau de Von Neumann et Morgenstern [Neumann & Morgenstern,

1944]. Comme son nom l'indique, les concepts de base de la théorie des jeux ont survécu de l'étude des jeux comme le jeu d'échecs. La théorie des jeux est pertinente pour l'étude de la négociation automatisée parce que les participants à ces négociations peuvent avoir des intérêts locaux (profits). L'émergence de l'Internet et de Web a fourni un impératif commercial énorme pour un développement futur des techniques de négociation basées sur la théorie des jeux [Noriega & Sierra, 1999], mais il y a quelques limitations considérables dans cette théorie [Dash et al., 2003]. Les approches basées sur cette théorie supposent que les agents ont des ressources de calcul illimitées et que l'espace de résultat est complètement connue, tandis que, dans la plupart des environnements, ces hypothèses ne parviennent pas en raison de la limitation des capacités du traitement et de la communication des systèmes d'information.

- ***Approches basées sur les heuristiques*** : les principaux moyens de surmonter les limitations des modèles de théorie des jeux est d'utiliser des méthodes heuristiques qui visent à produire des bonnes solutions. Les méthodes elles-mêmes peuvent être des approximations informatiques des techniques basées sur la théorie des jeux ou elles peuvent être des réalisations informatiques des modèles informels de négociation (i.e. [Pruitt, 1981; Raiffa, 1982]). La préoccupation centrale de cet axe de travail est de modéliser la prise de décision des agents de manière heuristique pendant la négociation. Certains chercheurs ont inspiré des heuristiques observées dans les négociations humaines pour implémenter le mécanisme de prise de décision des agents de négociation [Sierra et al., 1997; Chavez & Maes, 1996; Faratin et al., 1998]. Au contraire des approches basées sur la théorie des jeux, la mise en œuvre des stratégies basées sur les heuristiques ne nécessite pas un grand nombre de calcul. Même lors de déroulement de la négociation il est facile pour les utilisateurs de comprendre le comportement des agents sur le marché. Mais, comme la théorie des jeux, l'utilisation d'heuristiques pose deux inconvénients [Jennings et al., 2001]. Premièrement, les modèles mènent souvent aux résultats qui ne sont pas optimaux parce qu'ils adoptent une notion approximative de la rationalité parce qu'ils n'examinent pas l'espace complet des résultats possibles. Et deuxièmement, il est très difficile de prévoir avec précision le comportement du système et des agents constituants. En conséquence, les modèles ont besoin d'évaluation étendue par des simulations et des analyses empiriques.

- *Approches basées sur l'argumentation* : en plus des limitations mentionnées sur les approches précédentes, les propositions elles-mêmes dénotent généralement de seuls points dans l'espace des accords de négociation; la seule réaction qui peut être faite sur une proposition est une contreproposition, qu'elle-même est un autre point dans l'espace, ou une acceptation ou un retrait ; Il est difficile de changer l'ensemble de termes en négociation au cours d'une négociation. Donc, le but de la négociation à base d'argumentation est d'enlever ces limitations. L'idée fondamentale derrière cette approche est de permettre l'échange d'informations supplémentaires, au delà des propositions. C'est à dire chaque agent argumente au sujet de ces croyances et d'autres attitudes mentales durant le processus de négociation. Dans le contexte de la négociation, un argument est vu comme une pièce d'information qui peut permettre un agent de a) justifier sa position de négociation ; ou b) influencer la position de la négociation d'un autre agent [Jennings & Wooldridge, 1998]. En plus, les arguments peuvent être des justifications sur une proposition, en déclarant les raisons pour lesquelles une proposition a été faite ou les raisons pour lesquelles elle devrait être acceptée. Parmi les notions utilisées dans la négociation par argumentation est la notion de persuasion, qui a apparue dans les travaux de Sycara [Sycara, 1990; 1992], qui a présenté un modèle d'argumentation persuasive qui intègre les méthodes d'intelligence artificielle et de la théorie de décision.

### **I.2. Motivation et objectifs**

Notre étude est focalisée sur la formalisation de la phase de la négociation du processus d'achat dans le commerce électronique. La formalisation de cette phase permet aux agents d'être capables d'effectuer des interactions de négociation plus compliqués. Pendant que le commerce électronique est devenu plus adopté par les marchands ainsi que par les acheteurs, la négociation à base d'agent doit être capable de répondre à tous les besoins des utilisateurs.

Dans la négociation des humains, les vendeurs et les acheteurs utilisent plusieurs tactiques de négociation afin d'augmenter leurs gains. Ces tactiques sont souvent très compliquées, car ils se basent sur l'utilisation des expressions compliquées en visant au changement de l'attitude de la contrepartie ou de passer d'une situation à une nouvelle situation plus préférable.

Nous voyons que pour arriver à formaliser une négociation à base d'agent d'un commerce électronique, plusieurs facteurs doivent être pris en compte : l'état du marché, l'état mental et la situation moral des différentes parties de la négociation.

Dans notre travail, nous avons proposé une formalisation d'une négociation à base d'agent du commerce électronique et nous avons adopté la négociation par argumentation comme forme de négociation à base d'agent. A notre avis la négociation par argumentation est la forme la plus appropriée pour le domaine de commerce électronique, car elle permet à chaque agent de persuader l'agent opposant d'accepter sa proposition.

D'une autre coté, nous avons vu que la formalisation de la négociation dans l'e-commerce nécessite l'emploi de la technologie du web sémantique, à cause du très grand nombre de connaissances qui doivent être représentées et échangées entre les agents via le web.

Donc, l'objectif de notre travail est de proposer une représentation formelle basé sur le web sémantique de tous les éléments de l'espace de négociation par argumentation pour un environnement du commerce électronique à base d'agent.

### **I.3. Contributions de l'étude**

L'objectif de notre thèse est de proposer une approche formelle basée sur le web sémantique pour une négociation à base d'agent d'un commerce électronique. Nous avons fait une étude bibliographique sur l'utilisation de l'approche agent dans le domaine du commerce électronique, ainsi que sur les travaux qui s'intéresse à la négociation par argumentation. Nous avons constaté les différents besoins d'une négociation à base d'agent d'un environnement de commerce électronique ainsi les limites des travaux de recherche dans le domaine de la négociation par argumentation.

Nous avons noté, d'une part, que la majorité des travaux sur la négociation par argumentation proposés sont soit des frameworks généraux, soit des frameworks spécifiques à des domaines différents de l'e-commerce tel que : les délibérations, le diagnostic médical, négociations légales, ...etc. D'autre part, c'est rare de trouver des frameworks de négociation par argumentation qui utilisent le web sémantique comme outil de représentation.

Alors, nous avons essayés de traiter les besoins, d'une négociation à base d'agent du commerce électronique, qui sont considérés comme des défis à notre avis pour les travaux existants.

Donc, les contributions de cette thèse se résument sur les points suivants :

- L'utilisation de l'argumentation pour la modélisation d'une négociation à base d'agent pour un environnement du commerce électronique.
- La proposition d'une représentation formelle d'une locution basée sur le langage d'ontologie web (OWL) [McGuinness & Van Harmelen, 2004], en inspirant ses éléments du modèle original d'argument proposé par le philosophe S. Toulmin [1958].
- La proposition d'une formalisation, basée sur les techniques du web sémantique des mécanismes d'argumentation, en employant la technique d'analyse de champ de force proposée par le psychologue K. Lewin [1947]. Dans cette approche nous avons adopté le langage OWL pour la représentation des éléments de l'espace de négociation (inter-agents et intra-agent). Les mécanismes de négociation représentés sous forme des règles écrites en langage de règle du web sémantique (SWRL) [Horrocks et al., 2004].
- Définition du protocole de négociation dans une ontologie, en basant sur une représentation par les machines des états finis d'UML.
- Le développement d'une méthode pour l'échange du contenu des locutions, décrites par l'ontologie Locution, entre les agents via les messages FIPA-ACL. Cette méthode est basée sur l'approche proposée dans [Schiemann & Schreiber, 2006], pour l'utilisation du langage OWL-DL comme un langage de contenu des messages FIPA-ACL.

### **I.4. Structuration de la thèse**

Cette thèse est structurée en cinq chapitres. Le premier chapitre est une introduction sur l'étude, il présente le contexte du travail. Ce travail traite un problème qui touche principalement trois axes de recherche : les systèmes multi-agents, le commerce électronique et la négociation automatisée. Donc, l'objectif de ce chapitre est d'attirer l'attention du lecteur à la relation entre ces différents axes de recherche.

Le deuxième chapitre présente les travaux connexes à cette étude. Il se divise en deux parties : la première partie décrit plusieurs approches théoriques de l'argumentation et de la prise de décision, la deuxième partie présente une vue des frameworks de négociation à base d'argumentation.

Le troisième chapitre décrit la représentation formelle du système de négociation d'un e-commerce proposé. Les différents éléments du système de négociation, décrits dans le deuxième chapitre, sont formalisés en utilisant des ensembles, des relations et des règles logiques.

Dans le quatrième chapitre, une architecture pour le système de négociation est proposée et détaillée. Cette architecture comprend la représentation interne des agents (espace intra-agent) et la représentation de l'environnement d'interaction (espace inter-agents). Les langages du web sémantique (OWL et SWRL) sont employés pour la représentation des éléments de l'espace inter-agents ainsi que les mécanismes d'argumentation dans l'espace inter-agent.

Le dernier chapitre intitulé « expérimentation », décrit la stratégie suivie dans la validation de l'approche proposée, les différents outils utilisés dans l'implémentation du prototype, les scénarios du test effectués sur le prototype et enfin la discussion des résultats obtenus.

Ce document se termine par la section « conclusions et perspectives », qui présente les différentes conclusions déduites des résultats obtenus de ce travail, les limites, les défis et enfin, les travaux futurs.

# 2

## Travaux connexes

### II.1. Théorie de l'argumentation et de la prise de décision

L'argumentation est le processus qui permet de former des raisons, de justifier des croyances, et de tirer des conclusions dans le but d'influencer les croyances et/ou les actions des autres. van Eemeren [van Eemeren et al., 1996] ont défini l'argumentation comme

**Définition 2.1 :** Une activité verbale et sociale de raison visée augmentant (ou diminuant) l'acceptabilité d'un point de vue controversé pour l'auditeur ou le lecteur, en proposant une constellation de propositions prévues pour justifier (ou réfuter) le point de vue avant un juge raisonnable.

#### II.1.1. Modèle d'argument de Toulmin

Stephen Toulmin est un philosophe anglais, il a développé un modèle pour l'analyse des arguments utilisés tous les jours dans la vie réelle [Toulmin, 1958]. Il a identifié des éléments d'un argument persuasif et a donné les catégories utiles par lesquelles un argument peut être analysé. Il a déclaré que de bons arguments réalistes doivent être caractérisés par six parties décrites par les points suivants :

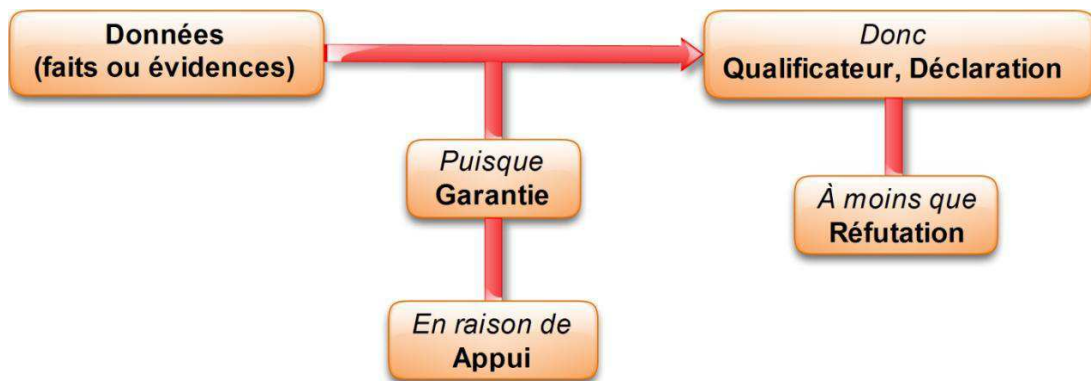
- **Données (en anglais Data) :** les faits ou les évidences employés pour prouver l'argument (par exemple statistiques donnés pour des bénéfices d'utilisation de mon produit). Les données seront comptées comme des prémisses sous l'analyse standard [Freeman, 1991].
- **Conclusion (en anglais Claim) :** la thèse étant argumentée, c'est la position étant discutée pour la conclusion de l'argument. (par exemple les gens devraient acheter mon produit pour obtenir une certaine indemnité). La conclusion est généralement



une conséquence d'un ensemble de prémisses : (*Prémisse1, Prémisse2, Prémisse3, ..., PrémisseN*) par conséquent (*Conclusion*) [Govier, 2010]. La conclusion est souvent caractérisée par le mot « *donc* ».

- **Qualificateur (en anglais Qualifier):** déclaration qui limite la force de l'argument ou qui propose les conditions dans lesquelles l'argument est vrai. Par exemple, si quelqu'un déclare que les taches sur des dents sont provoquées en buvant du café, il a peut-être besoin d'admettre qu'il peut y avoir d'autres causes aussi bien. La déclaration qualifiée sera: boire du café est la cause la plus significative (bien que peut-être pas la seule cause) des dents souillées. La qualification de l'argument peut être exprimée par l'utilisation des mots tel que : *certainement, exactement, probablement, pas forcément, vraisemblablement, ...etc.*
- **Garantie (en anglais Warrant):** la déclaration hypothétique et logique général qui forme un pont entre la déclaration et les données (exemple: ce qui justifierait l'achat de mon produit). Donc, la garantie est implicitement présente dans l'étape des données à la conclusion et, réciproquement, la nature des données dépend de la nature de la garantie [van Eemeren et al., 1996]. Un des aspects importants faits par Toulmin est que la garantie est un genre de règle d'inférence, et en particulier pas une déclaration des faits [Hage, 1997]. La garantie est caractérisée souvent par le mot « *puisque* ».
- **Appui (en anglais Backing):** la déclaration qui sert à soutenir la garantie, c.-à-d. un argument qui ne prouve pas nécessairement le point en discussion, mais qui prouve que la garantie soit vraie. Il est souvent caractérisée par le mot « *en raison de* ».
- **Réfutation (en anglais Rebuttal):** un argument contraire ou une déclaration qui indique les circonstances auxquelles l'argument ne juge pas vrai (c.-à-d. contre-exemple). Selon Freeley et Steinberg [Freeley & Steinberg, 2008], réfuter signifie 'surmonter l'évidence de l'opposition et raisonner en prouvant qu'elle soit fausse ou incorrecte. La réfutation est souvent caractérisée par le mot « *à moins que* ».

Le diagramme d'argument de Toulmin est illustré dans Figure. II.1 :



**Figure. II.1 :** Diagramme d’argument de Toulmin

Le modèle de Toulmin fournit des outils utiles pour analyser les composants des arguments [Meany & Shuster, 2002]. Il stipule que les arguments soient généralement exprimés avec des qualificateurs et des réfutations et qu'ils ne soient pas donnés comme absolus.

L'exemple suivant est utilisé par Toulmin [1958] pour expliquer le modèle qui a présenté :

Données	Garantie	Appui	Qualificateur	Conclusion	Réfutation
Harry est né à Bermudes	un homme né à Bermudes sera généralement un sujet Britannique	les statuts suivants et d'autres dispositions légales : -..... -.....	sans doute	Harry est un sujet Britannique	ses deux parents étaient des étrangers / il est devenu un Américain naturalisé /...

**Tableau II.1 :** Exemple d’argument formé selon le modèle de Toulmin

Cet exemple peut être écrit comme suite :

« Harry est né à Bermudes. *Puisque* un homme né à Bermudes sera généralement un sujet Britannique, *étant donné* les statuts suivants et d'autres dispositions légales : ...,

*donc sans doute, Harry est un sujet Britannique, à moins que ses deux parents étaient des étrangers/ il est devenu un Américain naturalisé/... ».*

### II.1.2. Systèmes abstraits d'argumentation

La théorie d'argumentation proposée par Dung [1993; 1995] est basée sur une notion de système d'argumentation définie comme :

**Définition 2.2 :** Une paire d'ensemble d'arguments, et une relation binaire représentant la relation d'attaque entre les arguments. Un argument est une entité abstraite dont le rôle est seulement déterminé par ses relations à d'autres arguments.

Formellement :

**Définition 2.3 :** Un système d'argumentation est une paire  $\mathcal{AF} = (\mathcal{A}, \mathcal{T})$  où :

$\mathcal{A}$  est un ensemble fini d'arguments, et  $\mathcal{T}$  est une relation binaire sur  $\mathcal{A}$ , c.-à-d.  $\mathcal{T} \subseteq \mathcal{A} \times \mathcal{A}$ . Supposant qu'il y on a deux arguments A et B, la relation  $\mathcal{T}(A, B)$  signifie que A représente une attaque contre B. La relation d'attaque peut être écrite :  $A\mathcal{T}B$  ou  $(A, B) \in \mathcal{T}$ . On peut dire aussi qu'un ensemble d'argument  $\mathcal{E} \subset \mathcal{A}$  attaque un argument A ou (A est attaqué par  $\mathcal{E}$ ) si un argument de  $\mathcal{E}$  attaque l'argument A.

**Exemple :** Soit  $\mathcal{AF} = (\mathcal{A}, \mathcal{T})$  un système d'argumentation où :  $\mathcal{A} = \{A, B, C\}$  et  $\mathcal{T} = \{(A, C), (C, A), (B, C)\}$ . Selon la relation  $\mathcal{T}$ , on peut dire que l'ensemble  $\{A, C\}$  attaque l'argument B.

**Définition 2.4 :** Un ensemble d'argument  $\mathcal{E} \subset \mathcal{A}$  est dit *sans-conflit* s'il n'y a aucun argument A et B dans  $\mathcal{E}$  tels que A attaque B., c.-à.-d., si  $\mathcal{E} \subset \mathcal{A}$  alors

$$\forall a, b \in \mathcal{E} \Rightarrow (a, b) \notin \mathcal{T}.$$

Dans l'exemple précédent, on peut dire que l'ensemble  $\{A, C\}$  est *sans-conflit*.

Pour un agent rationnel G, un argument A est *acceptable* si G peut défendre A contre toutes les attaques sur A. De plus, il est raisonnable de supposer qu'un agent rationnel accepte un argument seulement s'il est acceptable. Cela signifie que l'ensemble de tous les arguments acceptés par un agent rationnel est un ensemble d'arguments qui

peuvent se défendre contre toutes les attaques. Le *statut* est une autre notion qui est utilisé dans l'approche de Elvang et Krause [Elvang & Krause, 1993], selon cette notion les arguments sont divisés en trois classes d'arguments. Soit  $\mathcal{AF}=(\mathcal{A}, \mathcal{T})$  un système d'argumentation :

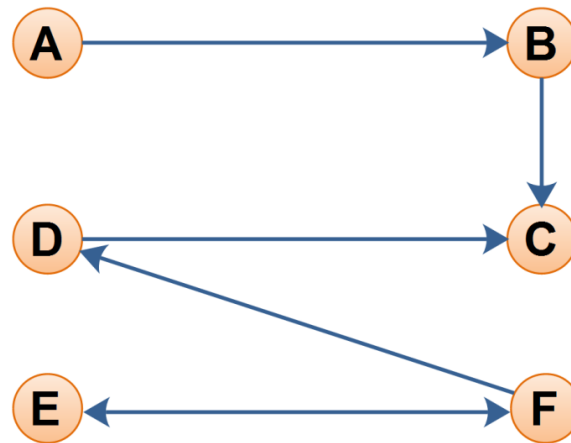
- La classe des *arguments acceptables*  $C_{ACC}=\{ A \in \mathcal{A} \mid \nexists B \in \mathcal{A} \text{ tel que } B \mathcal{T} A\}$ , qui représente les « bons » arguments.
- La classe des *arguments rejetés*  $C_{REJ}=\{ A \in \mathcal{A} \mid \exists B \in C_{ACC} \text{ tel que } B \mathcal{T} A\}$ , ceux sont les arguments défaits par des arguments acceptables.
- La classe des *arguments en suspension*  $C_{SUS} = \mathcal{A} \setminus (C_{ACC} \cup C_{REJ})$ , pour les arguments qui ne sont ni acceptables ni rejetés.

La notion d'*acceptabilité* nous mène à d'autres notions:

**Définition 2.5 :**

- 1) Un argument  $A \in \mathcal{A}$  est *acceptable par rapport* à un ensemble d'arguments  $S$  ssi pour chaque argument  $B \in \mathcal{A}$  : si  $B$  attaque  $A$  alors  $B$  est attaqué par  $S$ .
- 2) Un ensemble d'arguments *sans-conflit*  $S$  est *admissible* ssi chaque argument dans  $S$  est acceptable par rapport à  $S$ .
- 3) Une *extension préférée* d'un système d'argumentation  $\mathcal{AF}$  est un ensemble admissible maximal (par rapport à l'inclusion des autres ensembles), pour un système d'argumentation il existe au moins une *extension préférée*.
- 4) Un ensemble sans-conflit d'arguments  $S$  est appelé une *extension stable* ssi  $S$  attaque chaque argument qui n'appartient pas à  $S$ . Chaque *extension stable* et une *extension préférée*, l'inverse n'est pas vrai.

Un exemple sur un système d'argumentation est illustré dans la figure suivante :



**Figure. II.2 :** Exemple de système abstrait d'argumentation

- Ensembles sans-conflit :  $\{A, D, E\}, \{A, D\}, \{D, E\}, \{A, E\}, \{C, F\}, \{B, D\}, \dots$
- Ensembles admissibles :  $\{\}, \{A\}, \{E\}, \{F\}, \{A, D, E\}, \{A, F\}, \{A, C, F\}, \{D, E\}$ .
- Extension préférée :  $\{A, D, E\}, \{A, C, F\}$ .
- Extension stable :  $\{A, D, E\}, \{A, C, F\}$ .

A cause de l'importance de l'approche proposée par Dung [1993; 1995], plusieurs approches d'argumentation, qui présentent des extensions de l'approche de Dung, sont proposées :

- *Système à base de supposition* [Bondarenko et al., 1993; 1997] : Un système à base de supposition est un pair  $((\mathcal{L}, \mathcal{R}), Ab)$ , où  $(\mathcal{L}, \mathcal{R})$  est un système déductif tel que :  $\mathcal{L}$  est un langage formel avec une formule spéciale  $\perp \in \mathcal{L}$ , dénotant la fausseté, et  $\mathcal{R}$  est un ensemble de règles d'inférence de la forme  $\frac{\alpha_1, \dots, \alpha_n}{\alpha}$ , où  $\alpha, \alpha_1, \dots, \alpha_n \in \mathcal{L}$  et  $n \geq 0$ .

$Ab \in \mathcal{L}$ , est l'ensemble de toutes les suppositions candidates qui peuvent être utilisées pour l'extension d'une *théorie* donnée, sachant qu'une *théorie* est un ensemble de formules  $\mathcal{T} \subseteq \mathcal{L}$ . Dans cette approche les attaques sont définies entre des ensembles de suppositions par rapport à une théorie donnée  $\mathcal{T}$ .

- *Système à base de valeur* [Bench-Capon, 2002; 2003] :  $\mathcal{VAF}=(\mathcal{A}, \mathcal{T}, \mathcal{V}, val, valPref)$ , où  $\mathcal{V}$  est un ensemble non vide de valeurs,  $val$  est une fonction :  $\mathcal{A} \rightarrow \mathcal{V}$  qui associe à un argument  $A \in \mathcal{A}$  une valeur  $val(A) \in \mathcal{V}$ .  $valPref$  est une relation (transitive, irréflexive et asymétrique) de préférence sur  $\mathcal{V} \times \mathcal{V}$ . On dit qu'un argument  $A \in \mathcal{A}$  défasse un argument  $B \in \mathcal{A}$  ssi  $A$  attaque  $B$  et  $non\ valPref(val(B), val(A))$ .

Dans cette approche les notions : *acceptable*, *sans-conflit*, *admissible* et *extension préférée* ont eu des modifications.

- *Système à base de préférence* [Amgoud & Cayrol, 2002] :  $\mathcal{PAF}=(\mathcal{A}, \mathcal{T}, Pref)$ , où  $Pref \subseteq \mathcal{A} \times \mathcal{A}$  est une relation de préférence entre arguments. Pour deux arguments  $A, B \in \mathcal{A}$ ,  $A \gg_{Pref} B$  signifie que  $A$  est strictement préféré de  $B$  et  $A \equiv_{Pref} B$  signifie que  $A$  et  $B$  ont un ordre de préférence équivalent.
- *Système avec des attaques en forces-variées* [Martinez et al., 2008] :  $\mathcal{AFV}=(\mathcal{A}, \mathcal{T}, \mathcal{R})$ , où  $\mathcal{R} \subseteq \mathcal{T} \times \mathcal{T}$  est une relation qui dénote un ordre de force entre les conflits des arguments. L'ordre de force est déterminé par :  $\forall ti, tj \in \mathcal{T}$ , si  $(ti, tj) \in \mathcal{R}$  et  $(tj, ti) \notin \mathcal{R}$  alors on dit que l'attaque  $ti$  est plus fort que l'attaque  $tj$  ou  $ti \gg tj$ .

### II.1.3. Technique d'analyse de champ de force

L'analyse de champ de force (Force Field Analysis, en anglais) est une technique développée par le psychologue Allemand Lewin Kurt [1947] dans les années 50'. Lewin a considéré l'environnement social comme un champ dynamique qui est influencé de manière interactive avec la conscience humaine. Alors, il a proposé la théorie de champ pour l'analyse systématique des facteurs (tel que les personnes, les ressources, les attitudes, les traditions, les règlements, les valeurs, les besoins, les désirs, etc.) trouvés dans des problèmes complexes. La technique d'analyse de champ de force se base sur l'analyse des forces *conduisant* et des forces *maintenant* qui influencent un changement ou une décision suggérée. Les forces *conduisant* sont toutes les forces qui poussent et encouragent le processus de changement. Par contre, les forces *maintenant* sont les forces qui rendent le changement plus difficile. Ces forces contrecarrent les

forces conduisant et mènent à la résistance de changement. Cette technique a été utilisée pour résoudre des différents types de problème, par exemple pour :

- La gestion de changement.
- L'amélioration d'une situation (personnelle, organisationnelle, ...).
- Résoudre les problèmes complexes (recherche de meilleure solution).
- Aide à la prise de décision (Choix stratégique, lancement d'un projet, ...).

L'application de cette technique passe par plusieurs étapes :

- 1) Identifier le changement à analyser.
- 2) Identifier la situation actuelle et la situation désirée.
- 3) Identifier les différents facteurs qui vont être affectés par le changement.
- 4) Identifier les forces conduisant le changement : les conséquences désirées sur les facteurs.
- 5) Identifier les forces maintenant le changement : les conséquences non désirées sur les facteurs.
- 6) Calculer le niveau ou la valeur pour chaque force, (l'intervalle des valeurs est prédéfini).
- 7) Calculer la somme des forces conduisant et la somme des forces maintenant, les forces ayant la valeur supérieure sont ceux qui indiquent si le changement est viable.
- 8) Si le changement est non viable, alors soit le décideur annule le changement, soit il pense à augmenter les valeurs des forces conduisant ou à diminuer les forces maintenant.

L'application de cette technique se fait à l'aide d'un diagramme qui s'appelle le diagramme d'analyse de champ de force. La figure. II.3 illustre un exemple d'utilisation de ce diagramme pour l'étude de l'amélioration de processus de collecte d'information (d'un processus manuel à un processus automatisé) [Swinton].

L'analyse de champ de force est appliquée aussi par Patterson [Patterson, 2005] pour la planification de négociation. L'objectif de cette application était d'aider les négociateurs à préplanifier et à implémenter des stratégies pour réussir une négociation. L'analyse de champ de force permet à un négociateur d'identifier quelles sont les forces qui supportent un point donné et quelles sont les force qui le résistent. Il est très utile de

traiter les points subjectifs qu'on trouve généralement dans une négociation et de déterminer les stratégies appropriées, les tactiques et les informations à employer pour atteindre le résultat désiré. Cependant, un négociateur doit se rendre compte que l'autre partie a aussi ses forces conduisant qui poussent à ses propres résultats privilégiés et a besoins d'anticiper et de préparer pour les atteindre. Selon Patterson, dans une négociation multi-attributs (sur plusieurs points) chaque point négocié peut avoir sa propre analyse de champ de force.

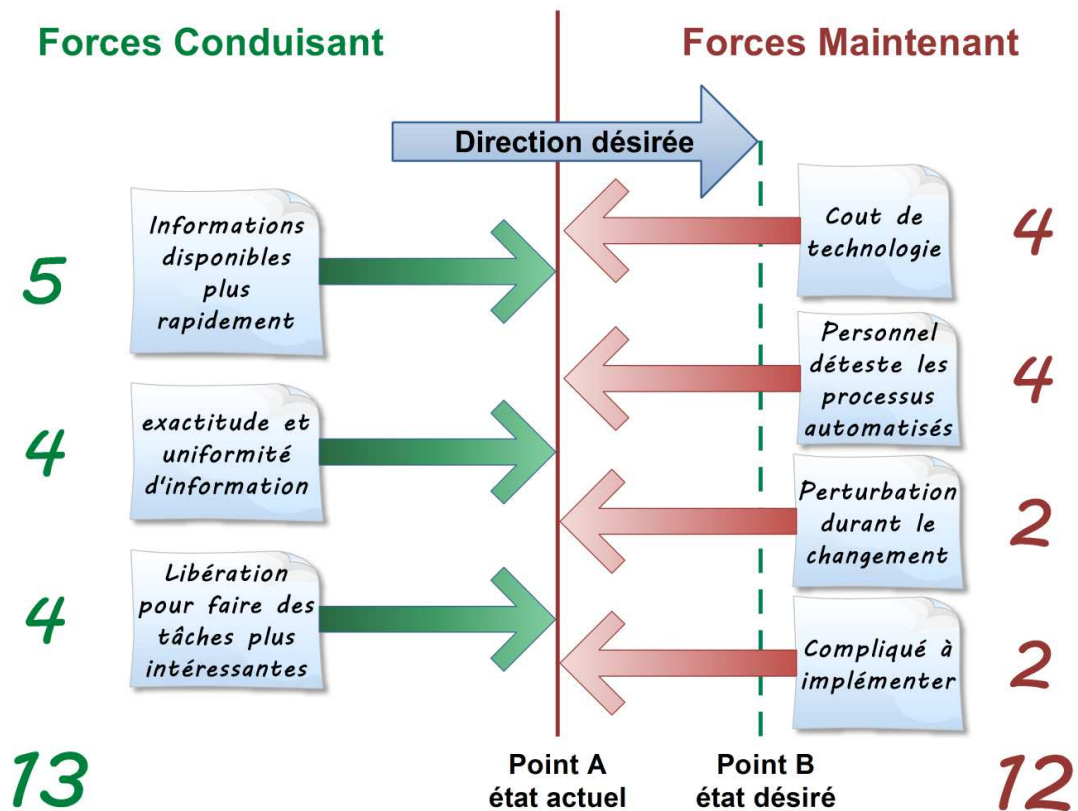


Figure. II.3 : Exemple de diagramme d'analyse de champ de force

## II.2. Frameworks de négociation à base d'argumentation (NA)

Comme nous l'avons mentionné dans le chapitre précédent, le but de la négociation à base d'argumentation est d'enlever les limitations des anciennes approches (basées sur la théorie des jeux et basées sur les heuristiques). Pour cette raison, ce type de négociation est devenu plus populaire, ainsi qu'il a été le sujet de plusieurs travaux de recherche sur différents domaines d'application, tel que l'e-commerce.



Pour développer un système de NA on doit définir d'abord son architecture. Rahwan et al. [2003] ont présentés en détail les *éléments externes* d'un framework de NA, ainsi que les *éléments internes* d'un mécanisme d'agent négociateurs. Les auteurs de Rahwan et al. [2003] sont parmi les chercheurs les plus connus dans le domaine de négociation par argumentation, pour cette raison la composition des frameworks de NA présentée dans cette thèse est influencée par la composition présenté dans [Rahwan et al., 2003].

Un framework de NA peut être vu comme un ensemble d'agents, ayant chacun son propre objectif. Comme l'objectif ; à atteindre par l'agent ; dépend directement de l'accord des autres agents (un ou plusieurs) il doit alors négocier, en suivant sa propre stratégie, avec les autres agents afin de les persuader d'accepter la situation qui convienne avec son objectif. Les éléments d'un framework de NA peuvent être identifiés sur deux espaces. Le premier espace (*inter-agents*) comprend les éléments employés à l'interaction entre les agents. Les éléments de l'espace *inter-agents* sont : le langage de négociation, le protocole de négociation et le stockage et le partage de données et de connaissances. L'espace *intra-agent* comprend les éléments du mécanisme d'argumentation, sachant que le mécanisme est l'ensemble de stratégies de l'agent avec le respect des règles du protocole de négociation. Un mécanisme de NA se devise généralement en trois grandes phases : l'*évaluation* de l'argument reçu, la *génération* des arguments candidats et enfin la *sélection* de l'argument le plus fort.

### **II.2.1. Espace inter-agents**

L'espace *inter-agents* représente l'environnement de négociation entre les agents, dont les agents vus comme des boîtes noires. L'interaction dans cet espace est assurée par l'échange de messages et par le partage de données et de connaissances. Dans un framework de NA, les messages échangés portent généralement des propositions et des arguments générés par les agents négociants. Pour que le dialogue de négociation se déroule correctement chaque agent doit comprendre les messages reçus et doit savoir à chaque tournée quel genre de proposition ou d'argument doit-il générer. Ce nécessite la définition d'un langage de négociation, d'un protocole de négociation et d'un espace de stockage et de partage de données et de connaissances (voir figure. II.4).

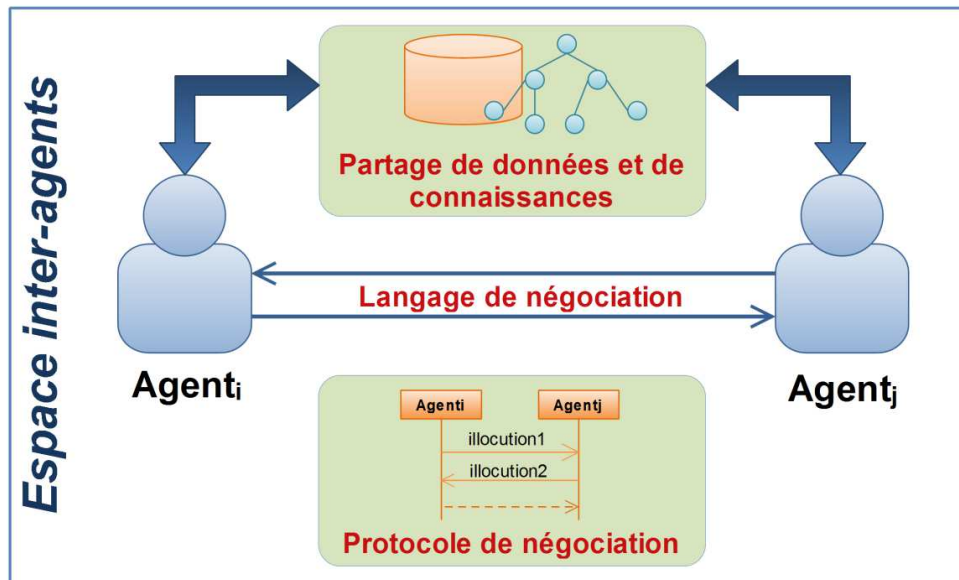


Figure. II.4 : Espace inter-agents dans un framework de NA

#### II.2.1.1. Langage de négociation

Puisque la négociation est une forme d'interaction, alors un framework de négociation exige un tel langage de communication [Labrou et al., 1999]. Les mécanismes de négociation automatisés traditionnels incluent normalement les locutions de base comme "*propose*" pour faire des propositions, "*accept*" pour accepter des propositions, et "*rejet*" pour rejeter des propositions qui sont appelées aussi les actes de langage. En plus un langage du contenu ou (langage du domaine) est nécessaire pour pouvoir partager les concepts de l'environnement.

Dans les systèmes multi-agents, il existe deux langages majeurs de communication des agents : KQML (Knowledge Query and Manipulation Language) [Mayfield et al., 1996] et FIPA ACL (Foundation for Intelligent Physical Agents' Agent Communication Language) [FIPA, 2001]. Malgré que ces langages de communication offrent plusieurs types de locutions (actes du langage), ils sont échoués à exprimer toutes les locutions exigées dans les framework de NA, car ces frameworks nécessitent la définition d'un langage de négociation, qui est souvent plus riche et capable d'exprimer plusieurs types d'informations. Afin de traiter ce problème, les concepteurs des framework de NA ont choisi de fournir leurs propres locutions spécifiques de négociation qui tiennent la sémantique appropriée du message. Par exemple Sierra et al.

[1997] et Ramchurn et al. [2003] fournissent des locutions explicites pour exprimer la menace et la récompense (i. e.  $threaten(i, j, \alpha, \beta)$  et  $promise(i, j, \alpha, \beta)$ ).

Le langage de négociation doit être capable de représenter toutes les informations qui concernent l'objet négocié. La phrase suivante [Sierra et al., 1997] représente une proposition :  $(Price = \text{£}10) \wedge (Quality = \textit{high}) \wedge (Penalty = ?)$ .

Il existe d'autres frameworks qui utilisent des langages capables d'exprimer les attitudes mentales des agents (croyances, désirs, intention, ...). Les frameworks présentés par Kraus et al. [1998] et par Parsons et al. [1998], par exemple, sont basés sur les logiques de BDI (Belief, Desire and Intention) et permettent de représenter les croyances d'un agent sur des croyances, des désirs et des intentions des autres agents. Il existe aussi une approche basée sur les ontologies pour la représentation de la structure des arguments échangés [Munoz & Botia, 2008; Willmott et al., 2006].

### II.2.1.2. Protocole de négociation

Le langage de négociation (ensemble de locutions) d'un framework de NA doit être utilisé avec le respect d'un protocole de négociation. Selon Jennings et al. [2001], un protocole est vu comme un ensemble formel de conventions régissant l'interaction des participants. Le protocole de négociation spécifie dans chaque état de négociation le « qui » et le « quoi », c'est la partie autorisée et l'ensemble de locutions potentielles. Par exemple, quand un agent reçoit une proposition, il doit l'accepter ou la rejeter (protocole : *take it or leave it*).

Les protocoles dans les frameworks de NA sont plus complexes, car ils sont capables d'impliquer un plus grand nombre de locutions et un plus grand nombre de règles. Les protocoles peuvent être représentés de différentes manières:

- Explicite : soit par les machines à états finis [Sierra et al., 1998; Parsons et al., 1998; Heras et al., 2011; Emele et al., 2011; Fabregues & Sierra, 2011], ou par les jeux de dialogue [Amgoud et al., 2000; Amgoud et Parsons, 2001; McBurney et al., 2003; Kok et al., 2011; Bentahar & Labban, 2009].
- Implicite : par l'utilisation des contraintes logiques exprimées sous forme de règles *if-then* incluses dans la spécification de l'agent [Kraus et al., 1998; Sadri et al., 2001].

### **II.2.1.3. Espace de stockage et de partage**

Les frameworks de NA utilisent souvent des espaces de stockage pour garder l'historique des locutions échangées. Ces espaces de stockage peuvent être soit locaux (internes) soit partagés par tous les agents. Les espaces de stockage locaux sont utilisés pour garder l'historique de l'agent sous forme de cas ou sous forme de locutions [Heras et al., 2011; Saha & Sen, 2005] ou bien pour garder l'état mental de l'agent [Bentahar & Labban, 2009]. Les espaces partagés sont utilisés généralement pour le stockage des engagements des agents [Maudet & Chaib-draa, 2003; McBurney & Parsons, 2009]. Par exemple, quand un agent promet de faire une action A, A est ajoutée à ses engagements dans un espace de stockage partagé. Les ontologies sont aussi utilisées pour la représentation et le stockage des connaissances du domaine pour les arguments échangés [Munoz & Botia, 2008].

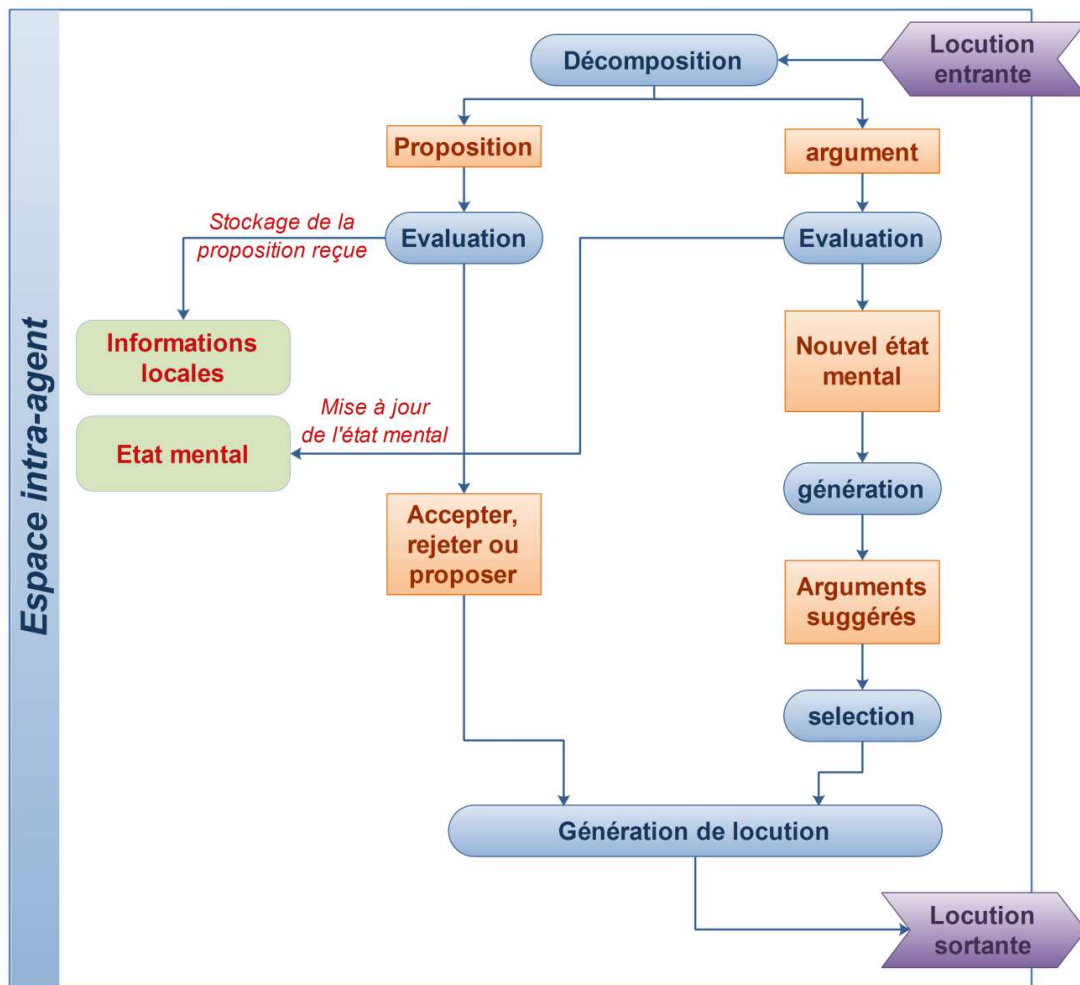
### **II.2.2. Espace intra-agent**

Nous désignons de l'espace intra-agent d'un framework de NA, la représentation interne de l'agent participé au dialogue de négociation par argumentation. Cette représentation est plus compliquée qu'une représentation interne d'un agent participé à un dialogue de négociation classique. Selon Rahwan et al. [2003], un agent capable de participer à une négociation par argumentation doit être équipé des mécanismes d'évaluation, de génération et de sélection des arguments (voir figure. II.5).

#### **II.2.2.1. Mécanisme d'évaluation**

Le mécanisme d'évaluation permet à un agent de comparer le contenu de l'argument et de la proposition reçu avec les éléments de son état mental (croyance, désirs, prérequis, ...). Le résultat de cette évaluation met à jours l'état mental de l'agent et lui permet de formuler une nouvelle proposition et un nouvel argument qui est capable de répondre à l'argument reçu.

L'évaluation des arguments est appliquée dans plusieurs travaux de recherches en argumentation, tel que [Kakas et Moraitis, 2003; Prakken et Sartor, 2001; Fox et Parsons, 1998; Saha & Sen, 2005; Heras et al., 2011].



**Figure. II.5 :** Espace intra-agent dans un framework de NA

Après l'évaluation d'une proposition l'agent doit l'accepter, la rejeter ou donner une nouvelle proposition. L'évaluation d'un argument est effectuée par l'application d'une inférence sur l'attitude interne de l'agent, cette inférence est suivie par une mise à jour de l'état mental de l'agent.

#### II.2.2.2. Mécanisme de génération

La génération des arguments consiste à construire des nouveaux arguments candidats à envoyer, avec la proposition générée, à l'agent de la contrepartie. Ces arguments sont généralement envoyés afin d'inciter la contrepartie à accepter un certain accord proposé.

Dans les frameworks de NA existants, la génération des arguments est faite à la suite de l'évaluation de certaines utilités, par exemple Sierra et al. [1998] et Ramchurn et al. [2003] supposent que les agents ont un moyen de générer des propositions qui augmentent (ou maximisent) leurs utilités. S'ils ne parviennent pas à obtenir l'acceptation immédiate, ils peuvent proposer d'effectuer une action (ou un ensemble d'actions) ou de fournir des ressources en échange pour l'acceptation. Cela peut être fait simplement en donnant ce qu'ils n'ont pas besoin, ou en mesurant les utilités qu'ils perdent à l'échange. Dans quelques frameworks tel que [Kraus et al., 1998; Bentahar & Labban, 2009], les arguments sont générées par l'utilisation des règles explicites, si le corps d'une règle est satisfait alors l'argument est choisi comme argument candidat. Dans autres frameworks tel que Ramchurn et al. [2003], il existe pour chaque argument des pré-conditions pour qu'il devienne un argument candidat. La génération peut être faite aussi à partir des cas d'arguments précédemment générés [Heras et al., 2011].

### **II.2.2.3. Mécanisme de sélection**

La sélection consiste à choisir le meilleur argument, du point de vue de l'agent argumentant, parmi les arguments candidats pour l'envoyer à l'agent opposant. Les arguments dans un framework de NA peuvent avoir un ordre de force, par exemple dans le travail de Kraus et al. [1998] les arguments ont l'ordre suivant : menace, promesse, appel et enfin contre-exemple. Dans le framework de [Amgoud et al., 2000], basé sur le système de Dung [1995], les agents comparent des arguments en basant sur un ordre préférentiel plus de leurs propositions constitutives. Pour Sadri et al. [2001], les agents peuvent comparer les coûts de différents alternatifs pour présenter aux contreparties. Il existe d'autres approches estimatives telles que l'approche probabiliste [Saha & Sen, 2005] basée sur les réseaux baysiens ou l'approche basée sur la logique floue [Hsairi et al., 2008].

Enfin, dans la sélection des arguments on doit prendre en compte les informations sur l'agent de la contrepartie, pour assurer que l'argument envoyé soit capable de l'influencer.

# 3

## **Contribution 1 : Framework proposé pour une négociation**

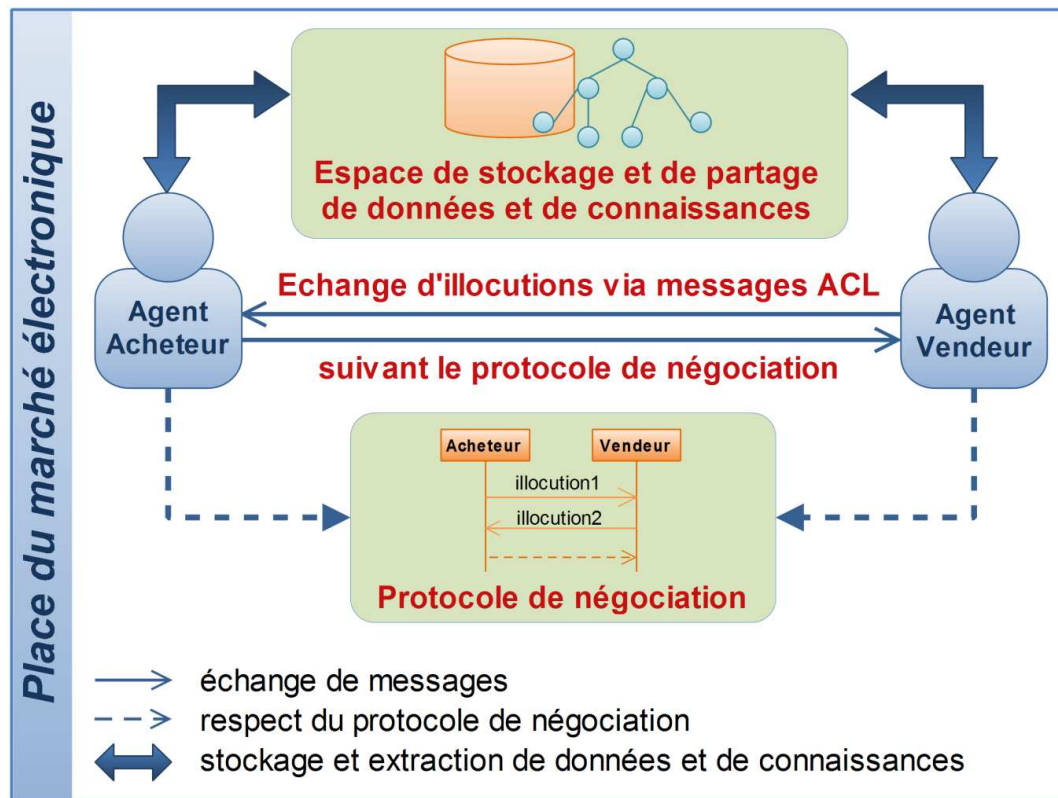
### **III.1. Introduction**

Ce chapitre représente notre première contribution, quelques parties de cette contribution sont publiées dans [Said et al., 2011].

Dans ce chapitre nous présentons le framework proposé pour une négociation à base d'argumentation, qui comprend deux représentations : représentation de l'espace inter-agents (l'environnement d'interaction) et de l'espace intra-agent (la représentation interne de l'agent). Le framework proposé se base sur la description formelle des systèmes abstraits.

### **III.2. Représentation de l'espace de négociation inter-agents**

Dans cette section nous présentons l'architecture générale du marché électronique à base d'agent. Cette architecture représente l'espace inter-agents, dont les agents sont vus comme des boites noirs qui échangent des illocutions.



**Figure. III.1 :** Architecture de l'espace inter-agent du framework de négociation

Dans l'espace inter-agent nous représentons tous ce qui concerne l'interaction entre les agents au cours du dialogue de négociation par argumentation. Le framework de NA est défini comme suite:

$\mathcal{FNA} = \langle \mathcal{G}, \mathcal{L}, \mathcal{PN} \rangle$ , où :

- $\mathcal{G} = \mathcal{A} \cup \mathcal{V}$  : l'ensemble d'agents de négociation.
- $\mathcal{A}$  : l'ensemble d'agents acheteurs.
- $\mathcal{V}$  : l'ensemble d'agents vendeurs.
- $\mathcal{L}$  : le langage de négociation qui représente l'ensemble de locutions utilisées dans le système de négociation.
- $\mathcal{PN} = \langle \mathcal{S}, trans \rangle$  : est une machine d'états finis qui représente le protocole de négociation.



### III.2.1. Langage de négociation

- $\mathcal{L} = \mathcal{L}_n \cup \mathcal{L}_p$ : est le langage de négociation utilisé par les agents négociateurs, il comprend deux sous-langages  $\mathcal{L}_n$  et  $\mathcal{L}_p$ .
- $\mathcal{L}_n$ : représente l'ensemble de locutions non persuasives de type  $\{Accepter, Rejeter, Proposer, Suspendre, Abandonner\}$ , utilisés pour décrire des offres et des contre-offres.

Les locutions non persuasives ont la forme suivante :

$locution(a_1, a_2, \mathcal{P}, t)$ , avec  $(a_1 \in \mathcal{A}, a_2 \in \mathcal{V}$  ou  $a_1 \in \mathcal{V}, a_2 \in \mathcal{A})$ ,  $\mathcal{P}$  : proposition et  $t$  : temps.

Les types des locutions non persuasives sont décrits dans le tableau suivant :

Locution	Description
$Proposer(a_1, a_2, \mathcal{P}, t)$	$a_1$ propose une offre $\mathcal{P}$ à $a_2$ ou une modification sur une offre.
$Accepter(a_1, a_2, \mathcal{P}, t)$	$a_1$ accepte une proposition $\mathcal{P}$ de $a_2$ .
$Rejeter(a_1, a_2, \mathcal{P}, t)$	$a_1$ rejette une proposition (ou une partie de proposition) $\mathcal{P}$ de $a_2$ .
$Suspendre(a_1, a_2, \mathcal{P}, t)$	$a_1$ suspend le dialogue de négociation avec $a_2$ sur $\mathcal{P}$ ( $\mathcal{P}$ est le point de reprise).
$Abandonner(a_1, a_2, \mathcal{P}, t)$	$a_1$ abandonne le dialogue de négociation avec $a_2$ sur $\mathcal{P}$ .

**Tableau III.1.** Description des locutions non persuasives

- $\mathcal{L}_p$  : l'ensemble de locutions persuasives de type  $\{Encourager, Décourager, Promettre, FaireAppel\}$ ; des arguments utilisées par un agent pour persuader l'agent opposant afin qu'il accepte sa proposition. Les locutions non persuasives ont la forme suivante :

$locution(a_1, a_2, \mathcal{P}, \varphi, t)$ , avec  $(a_1 \in \mathcal{A}, a_2 \in \mathcal{V}$  ou  $a_1 \in \mathcal{V}, a_2 \in \mathcal{A})$ ,  $\mathcal{P}$  : proposition,  $\varphi$  : élément persuasif et  $t$  : temps.

Les types des locutions persuasives sont décrits dans le tableau suivant :

Locution	Description
<i>Encourager</i>	Persuader l'agent opposant d'accepter la proposition qu'il à reçu.
<i>Décourager</i>	Persuader l'agent opposant de rétracter la proposition qu'il à envoyé.
<i>Promettre</i>	Donner une promesse à l'agent opposant de lui donner une proposition plus favorisé dans une négociation ultérieure, s'il accepte la proposition actuelle.
<i>FaireAppel</i>	Faire appel à une promesse passé ou à une règle conventionnelle.

**Tableau III.2.** Description des locutions persuasives

- $\mathcal{T}$ : l'ensemble de tous les attributs de l'objet négocié.
- $\mathcal{D}_1, \dots, \mathcal{D}_{|\mathcal{T}|}$ : chaque  $\mathcal{D}_i$  est un domaine d'un attribut  $x_i \in \mathcal{T}$  (ensemble de valeurs possibles).
- $\mathcal{P}$ : collection de composantes du triplet de la forme  $(x_i, val, neg)$ , avec  $x_i \in \mathcal{T}$ ,  $val \in \mathcal{D}_i$  et  $neg \in \{vrai, faux\}$  qui indique si l'attribut  $x_i$  est négociable ou non. La collection  $\mathcal{P}$  représente la proposition d'un agent négociateur.
- $\varphi$ : représente un élément persuasive intercalé dans les locutions persuasives pour influencer l'agent opposant d'accepter la proposition qu'il a reçu ou de modifier la proposition qu'il a envoyé. L'élément  $\varphi$  à une structure non atomique, il se compose lui-même d'un ensemble d'éléments. La structure de  $\varphi$  est inspiré du modèle théorique de l'argument proposé par le philosophe S. Toulmin [1958]. Dans notre approche l'élément  $\varphi$  représente une expression tactique de négociation qui est un triplet de la forme :  $(exp(\tau), \psi, \mu)$ , tel que  $exp$  est une expression (déclaration ou action),  $\tau$ : un pair  $(x_i, val)$ ,  $\psi$ : l'appui qui prend une (des) valeur(s) permettant de supporter l'expression et  $\mu$ : le qualificateur qui spécifie le degré auquel l'appui ï peut supporter l'expression sur  $\tau$ .

Le tableau suivant montre la correspondance entre les éléments de l'argument de Toulmin et les éléments d'une locution persuasive :

Argument de Toulmin	Locution persuasive	Description
Garantie	$exp(\tau)$	Expression tactique (déclaration ou action).
Appui	$\psi$	Une (des) valeur(s) calculé ou obtenue de l'espace de stockage et de partage de données et de connaissances, par une requête.
Qualificateur	$\mu$	Degré de certitude ou d'appui (à quel degré $\psi$ peut supporter $exp(\tau)$ ), $\mu \in [0,1]$ .
Déclaration	Type de locution	Les différents types des locutions persuasives ( <i>Encourager, Décourager, Promettre, FaireAppel</i> ).
Réfutation	Collision	Pré-requis de l'agent opposant ou une locution de contradiction (i.e. <i>Décourager</i> ).

**Tableau III.3.** Les éléments d'une locution persuasive

La forme générale d'une locution persuasive devient:

$$locution(a1, a2, \mathcal{P}, exp(\tau), \psi, \mu, t)$$

Dans une négociation commerciale il existe plusieurs exemples d'expressions tactiques, qui peuvent être employées afin d'influencer l'agent opposant de modifier son état mental et, par conséquent, de modifier sa proposition. Les tableaux III.3 et III.4 présentent quelques expressions tactiques utilisées respectivement par les agents acheteurs et les agents vendeurs, elles sont décrites de manière informelle:

Acheteur	N°	Expression tactiques	Appuis
Encourager	1	Concéder à une partie de la proposition (moins importante) pour garder une partie (plus importante).	Sans appui.
	2	Déclarer une fidélité au vendeur.	Historique du nombre d'achat au passé.
Décourager	3	Alléguer que l'offre proposée n'est pas concurrent.	Nombre d'offres disponibles mieux que l'offre actuel.
	4	Alléguer que les caractéristiques de l'offre actuelle ne sont pas en demande.	Statistiques sur les produits vendus ayant ces caractéristiques.
	5	Avertir d'abandonner la négociation si la proposition ne sera pas changée.	Sans appui.
Promettre	6	Promettre d'être fidèle au vendeur.	Sans appui.
	7	Promettre de concéder à une partie de la proposition (de l'agent acheteur) si l'agent vendeur l'accepte.	Sans appui.
FaireAppel	8	Faire appel à une promesse passée de compensation.	Extrait de l'historique des promesses.
	9	Faire appel à une promesse de concéder à une partie de la proposition.	Extrait de l'historique des promesses.

**Tableau III.4.** Eléments de persuasion utilisés par l'agent acheteur

Vendeur	N°	Expression tactiques	Appuis
Encourager	10	Les caractéristiques dans la proposition sont parmi les plus demandées.	Statistiques sur les produits vendus ayant ces caractéristiques.
	11	Concéder à une partie (moins importante) de la proposition.	Sans appui.
	12	Compenser la proposition dans une partie pour garder une autre partie.	Sans appui.
	13	Déclarer que l'offre propose est limité (quantité ou temps).	Quantité restante ou temps restant.
	14	Déclarer que l'offre est exceptionnelle.	Comparaison avec des offres précédentes.
Décourager	15	Notifier que l'acheteur n'a pas fait le meilleur choix.	Statistiques sur d'autres caractéristiques plus demandés
	16	Notifier qu'une partie de la proposition est non raisonnable.	Comparaison avec des propositions de l'historique.
Promettre	17	Promettre de faire une compensation dans une négociation ultérieure.	Sans appui.
	18	Promettre de concéder à une partie de la proposition (de l'agent vendeur) si l'agent acheteur l'accepte.	Sans appui.
Faire Appel	19	Faire appel à une promesse de fidélité.	Historique des promesses.
	20	Faire appel à une promesse de concéder à une partie de la proposition.	Historique des promesses.

**Tableau III.5.** Eléments de persuasion utilisés par l'agent vendeur

Les expressions tactiques présentées sont inspirées de la négociation commerciale classique (humain-humain), mais elles ne sont pas les seules expressions utilisées, nous avons choisi quelques exemples pour expliquer l'approche que nous avons proposée. Dans ce qui suit, la représentation formelle des locutions persuasives est projetée sur les exemples précédents.

N°	Locutions persuasives d'un agent acheteur
1	<p><math>Encourager(acht_i, vend_j, \mathcal{P}, Conceder(\tau), \psi, \mu, t)</math>, avec :</p> <p><math>\psi</math> non déterminé.</p> <p><math>\mu</math> sera évalué par l'agent <math>vend_j</math> selon de sont état mental.</p>
2	<p><math>Encourager(acht_i, vend_j, \mathcal{P}, Fidele(), \psi, \mu, t)</math>, avec :</p> <p><math>\psi = requete(taille(histor\_achat(acht_i, vend_j)))</math></p> <p><math>\mu</math> sera calculé par l'agent <math>vend_j</math>, il dépend de sont état mental, s'il ne prend pas en considération la fidélité alors <math>\mu = 0</math>.</p>
3	<p><math>Decourager(acht_i, vend_j, \mathcal{P}, \neg Concurrent(\tau), \psi, \mu, t)</math>, avec :</p> <p><math>\tau \subseteq \mathcal{P}</math></p> <p><math>\psi = requete(taille(histor\_local(\mathcal{P}')), pref(\mathcal{P}') \succ pref(\mathcal{P}))</math></p> <p><math>\mu = \psi / requete(taille(histor\_local(\mathcal{P}')), \mathcal{P}' \neq \mathcal{P})</math></p>
4	<p><math>Decourager(acht_i, vend_j, \mathcal{P}, \neg Demande(\tau), \psi, \mu, t)</math>, avec :</p> <p><math>\tau \subseteq \mathcal{P}</math>, <math>taille(\tau) = 1</math> et <math>\tau = \{(att, val)\}</math></p> <p><math>\psi = requete(ordre(histor\_vente(\tau)))</math></p> <p><math>dernier = requete(taille(histor\_vente(att)))</math></p> <p><math>\mu = (\psi - 1) / (dernier - 1)</math>,</p> <p>avec <math>ordre(histor\_vente(\tau))</math> : l'ordre selon la quantité des produits vendu au marché ayant le pair <math>(att, val)</math>.</p> <p><math>histor\_vente(att)</math> : ensemble de différentes valeurs de <math>att</math> dans l'historique de vente.</p> <p><u>Remarque</u> : si <math>\psi = 1</math> alors <math>\mu = 0</math> et si <math>\psi = dernier</math> alors <math>\mu = 1</math>.</p>
5	<p><math>Decourager(acht_i, vend_j, \mathcal{P}, Abandonner(), \psi, \mu, t)</math>, avec :</p> <p><math>\psi</math> non déterminé.</p> <p><math>\mu = 1</math>.</p>

6	<p><math>Promettre(acht_i, vend_j, \mathcal{P}, Fidele(), \psi, \mu, t)</math>, avec :</p> <p><math>\psi</math> non déterminé.</p> <p><math>\mu</math> sera évalué par l'agent <math>vend_j</math>, il dépend de son état mental.</p>
7	<p><math>Promettre(acht_i, vend_j, \mathcal{P}, Conceder(\tau), \psi, \mu, t)</math>, avec :</p> <p><math>\psi</math> non déterminé.</p> <p><math>\mu</math> sera évalué par l'agent <math>vend_j</math>, il dépend de son état mental.</p>
8	<p><math>FaireAppel(acht_i, vend_j, \mathcal{P}, Compenser(\tau), \psi, \mu, t)</math>, avec :</p> <p><math>\tau \subseteq \mathcal{P}</math>, <math>taille(\tau)=1</math> et <math>\tau=\{att, val\}</math>.</p> <p><math>\psi = requete(histor\_promesse(Compenser(\tau), vend_j, acht_i))</math></p> <p><math>\mu=1</math>, l'agent vendeur peut remis <math>\mu</math> à <math>0</math> s'il existe une compensation faite après cette promesse (cas de réfutation).</p>
9	<p><math>FaireAppel(acht_i, vend_j, \mathcal{P}, Conceder(\tau), \psi, \mu, t)</math>, avec :</p> <p><math>\tau \subseteq \mathcal{P}</math>, <math>taille(\tau)=1</math> et <math>\tau=\{att, val\}</math>.</p> <p><math>\psi = requete(histor\_promesse(Conceder(\tau), vend_j, acht_i))</math></p> <p><math>\mu=1</math>, l'agent vendeur peut remis <math>\mu</math> à <math>0</math> s'il existe une concession faite après cette promesse (cas de réfutation).</p>

**Tableau III.6.** Représentation formelle des locutions persuasives d'un agent acheteur

N°	Locutions persuasives d'un agent vendeur
10	<p><math>Encourager(vend_i, acht_j, \mathcal{P}, Demande(\tau), \psi, \mu, t)</math>, où:</p> <p><math>\tau \subseteq \mathcal{P}</math>, <math>taille(\tau)=1</math> et <math>\tau=\{att, val\}</math></p> <p><math>\psi = requete(ordre(histor\_vente(\tau)))</math></p> <p><math>dernier = requete(taille(histor\_vente(att)))</math></p> <p><math>\mu = (dernier - \psi) / (dernier - 1)</math></p> <p>avec <math>ordre(histor\_vente(\tau))</math> : l'ordre selon la quantité des produits vendu au marché ayant le pair <math>(att, val)</math>.</p> <p>dernier: représente l'ordre de dernier élément à la vente selon <math>att</math>.</p> <p><u>Remarque</u> : si <math>\psi=1</math> alors <math>\mu=1</math> et si <math>\psi = dernier</math> alors <math>\mu=0</math>.</p>
11	<p><math>Encourager(vend_i, acht_j, \mathcal{P}, Conceder(\tau), \psi, \mu, t)</math>, où:</p> <p><math>\psi</math> non déterminé.</p> <p><math>\mu</math> sera évalué par l'agent <math>acht_j</math> selon de son état mental.</p>

12	<p><math>Encourager(vend_i, acht_j, \mathcal{P}, Compencer(\tau), \psi, \mu, t)</math>, où:</p> <p><math>\psi</math> non déterminé.</p> <p><math>\mu</math> sera évalué par l'agent <math>acht_j</math> selon de sont état mental.</p>
13	<p><math>Encourager(vend_i, acht_j, \mathcal{P}, Limite(\tau), \psi, \mu, t)</math>, où:</p> <p><math>\tau \subseteq \mathcal{P}</math></p> <p><math>\psi = requete(taille(histor\_offre(vend_i, \tau)))</math></p> <p><math>\mu</math> sera évalué par l'agent <math>acht_j</math>, si <math>requete(taille(histor\_offre(vend_i, \tau))) \neq 0</math></p> <p>si <math>k \neq i</math> alors <math>\mu = 0</math> sinon <math>\mu</math> sera évalué par l'agent <math>acht_j</math> selon de sont état mental.</p>
14	<p><math>Encourager(vend_i, acht_j, \mathcal{P}, Exception(\tau), \psi, \mu, t)</math>, où:</p> <p><math>\tau \subseteq \mathcal{P}</math></p> <p><math>\psi = requete(ordre(histor\_vente(vend_i, \tau)))</math></p> <p><math>\mu = 1</math> si <math>\psi = 1</math> et <math>\mu = 0</math> sinon.</p>
15	<p><math>Decourager(vend_i, acht_j, \mathcal{P}, \neg Meilleur(\tau), \psi, \mu, t)</math>, où:</p> <p><math>\tau \subseteq \mathcal{P}</math></p> <p><math>\psi = requete(ordre(histor\_vente(\tau)))</math></p> <p>si <math>\psi = 1</math> alors <math>\mu = 0</math> sinon <math>\mu</math> sera évalué par l'agent <math>acht_j</math> selon de sont état mental.</p>
16	<p><math>Decourager(vend_i, acht_j, \mathcal{P}, \neg Raisonnable(\tau), \psi, \mu, t)</math>, où:</p> <p><math>\psi = requete(taille(histor\_vente(\tau')), pref(\tau') \prec pref(\tau))</math></p> <p>si <math>\psi \neq 0</math> alors <math>\mu = 0</math> sinon <math>\mu</math> sera évalué par l'agent <math>acht_j</math> par l'application de <math>requete(taille(histor\_vente(\tau')), pref(\tau') \succ pref(\tau))</math></p>
17	<p><math>Promettre(vend_i, acht_j, \mathcal{P}, Compencer(\tau), \psi, \mu, t)</math>, où:</p> <p><math>\psi</math> non déterminé.</p> <p><math>\mu</math> sera évalué par l'agent <math>acht_j</math> selon de sont état mental.</p>
18	<p><math>Promettre(vend_i, acht_j, \mathcal{P}, Conceder(\tau), \psi, \mu, t)</math>, avec :</p> <p><math>\psi</math> non déterminé.</p> <p><math>\mu</math> sera évalué par l'agent <math>acht_j</math> selon de sont état mental.</p>
19	<p><math>FaireAppel(vend_i, acht_j, \mathcal{P}, Fidele(), \psi, \mu, t)</math>, avec :</p> <p><math>\psi = requete(histor\_promesse(Fidele(), acht_i, vend_j))</math></p> <p>si <math>requete(taille(histor\_vente(vend_i, acht_j))) \geq 2</math> alors <math>\mu</math> sera évalué par</p>



	l'agent $acht_j$ selon de sont état mental, sinon $\mu = 0$ .
20	<p><math>FaireAppel(vend_i, acht_j, \mathcal{P}, Conceder(\tau), \psi, \mu, t)</math>, avec :</p> <p><math>\tau \subseteq \mathcal{P}</math>, <math>taille(\tau)=1</math> et <math>\tau = \{(att, val)\}</math>.</p> <p><math>\psi = requete(histor\_promesse(Conceder(\tau), acht_i, vend_j))</math></p> <p><math>\mu=1</math>, l'agent acheteur peut remis <math>\mu</math> à 0 s'il existe une concession faite après cette promesse (cas de réfutation).</p>

**Tableau III.7.** Représentation formelle des locutions persuasives d'un agent vendeur

### III.2.2. Protocole de négociation

- $\mathcal{PN} = \langle \mathcal{S}, trans \rangle$  : est une machine d'états finis qui représente le protocole de négociation
- $\mathcal{S} = \{(Initial, acht), (S_1, vend), (S_2, acht), (S_3, acht), (S_4, acht), (S_5, vend), (Final,)\}$  : est l'ensemble des états, qui contient un état initial et un état final. Un état  $S_i \in \mathcal{S}$  est un pair  $(nom, partie)$ , avec  $nom$  : le nom de l'état  $S_i$  et  $partie \in \{vendeur, acheteur\}$  qui est l'agent ayant la main de générer des locutions.
- $trans : \mathcal{S} \times Types \rightarrow \mathcal{S}$  est une fonction de transition définit de la manière suivante :  
 $\forall S_i \in \mathcal{S}$  et  $\forall loc \in Types$ ,  $trans(S_i, loc) = S_j \in \mathcal{S}$  si la locution de type  $loc$  est potentiel à l'état  $S_i$  et  $trans(S_i, loc) = \emptyset$  sinon.

Le protocole  $\mathcal{PN}$  est représenté par une machine d'états finis d'UML suivante :

Les machines d'états finis sont utilisées dans plusieurs travaux tel que [Sierra et al., 1997; Parson et al., 1998]. Une machine d'états finis est capable de décrire les différents états de négociation et de déterminer pour chaque état les locutions potentielles à envoyer. La machine d'états finis, présentée dans la figure. III.2, ne décrit pas complètement le dialogue de négociation, mais seulement une tournée du dialogue de négociation. Le dialogue complet sera présenté ultérieurement dans la description du processus d'achat.

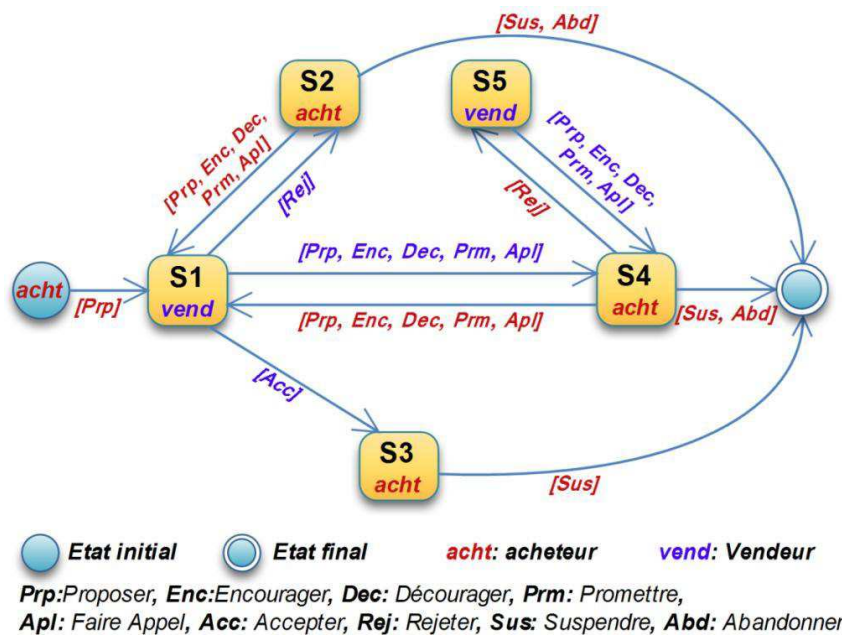


Figure. III.2 : Représentation du protocole de négociation

### III.3. Représentation de l'espace de négociation intra-agent

L'espace intra-agent comprend les différents composants à l'intérieur de l'agent qui assurent l'application de la stratégie de l'agent négociateur en suivant le protocole de négociation. Selon [Rahwan et al, 2003], une locution passe par trois opérations : l'évaluation de la proposition reçue, la génération d'un ensemble d'arguments candidats et enfin la sélection de l'argument le plus fort.

Dans notre approche nous avons adopté le terme locution pour décrire les arguments et les propositions échangés, une locution peut être persuasive (argument) ou non persuasive (proposition ou contre-proposition). Le mécanisme de négociation d'un agent (vendeur ou acheteur) comprend plusieurs opérations, mais les opérations principales sont : l'évaluation de la locution entrante, la génération des locutions candidats et enfin la sélection de la locution la plus puissante. La figure suivante explique la relation entre les éléments du mécanisme de négociation :

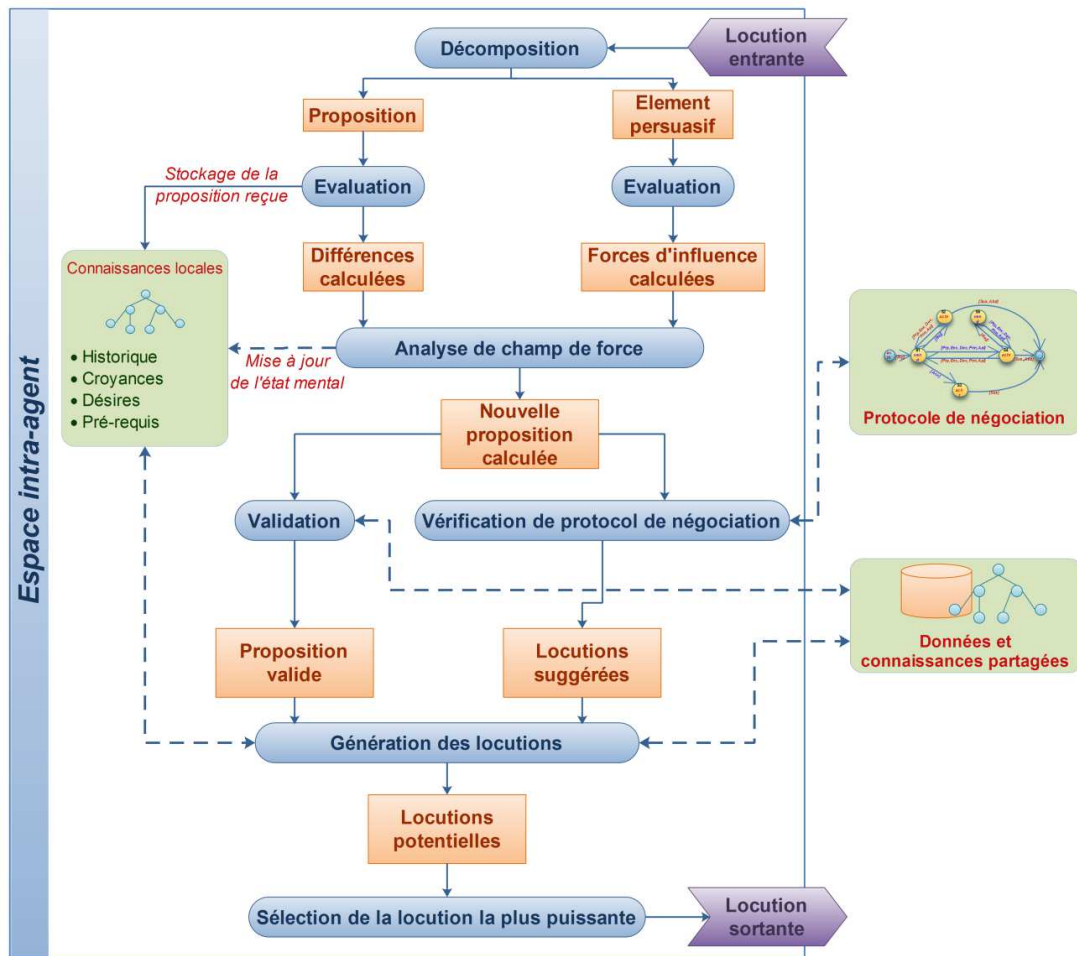


Figure. III.3 : Les éléments de l'espace intra-agent

### III.3.1. Les connaissances locales

Lors d'une négociation, un agent a toujours besoin de stocker, d'exploiter et de mettre à jours (enrichir) des connaissances locales. Ces connaissances ne seront pas exploitables que par l'agent lui-même. Nous distinguons deux types de connaissances locales : les connaissances qui représentent l'état mental de l'agent (croyances, désirs et pré-requis) et les connaissances qui dépendent de l'environnement (historique des locutions échangées, état de négociation, les agents opposants, ...).

L'état mental d'un agent (vendeur / acheteur) se compose de trois différentes familles de connaissances : *croyances*, *désirs* et *pré-requis*.

### III.3.1.1. Les croyances

Les connaissances qui représentent les croyances de l'utilisateur concernant les attributs de l'objet négocié (poids de chaque attribut), les types de locutions persuasives adoptés et les préférences sur les attributs de l'objet négocié.

Nous désignons du poids d'un attribut  $x_i \in \mathcal{T}$ ; son degré d'importance au point de vu de l'utilisateur, il est défini par la fonction *poids*. Pour un objet négocié ayant  $n$  attributs, la somme :  $\sum_{i=1}^n \text{poids}(x_i) = 1$ .

Pour chaque attribut  $x_i \in \mathcal{T}$  il existe un ordre de préférence dans son ensemble de valeurs  $\mathcal{D}_i$ . Les préférences sont définies à l'aide de la fonction *pref* de la manière suivante :

pour  $x_i \in \mathcal{T}$  et  $v_1, v_2 \in \mathcal{D}_i$  on a  $\text{pref}(x_i, v_1) \succ \text{pref}(x_i, v_2)$  si  $v_1$  est plus préférée que  $v_2$  de l'attribut  $x_i$  pour l'agent.

De manière générale, si on a une proposition  $\mathcal{P}$  tel que  $\mathcal{P} = \{(x_1, v_1), \dots, (x_k, v_k)\}$  alors  $\text{pref}(\mathcal{P}) = \text{pref}(x_1, v_1) \cdot \text{poids}(x_1) + \dots + \text{pref}(x_k, v_k) \cdot \text{poids}(x_k)$ .

### III.3.1.2. Les désirs

Elles sont les valeurs désirées pour chaque attribut de l'objet négocié définis par l'utilisateur. La proposition initiale est formée à partir des désirs, qui seront influencés à la suite par les locutions persuasives reçus, ce qui engendre la mise à jour des désirs et par conséquence la modification de la proposition formée par l'agent négociateur.

Le deuxième type de connaissances dépend de l'espace inter-agents et comprend : l'historique des locutions échangées, l'état de négociation et les connaissances, qui concerne les agents opposants, employées dans le dialogue de négociation.

### III.3.1.3. Les pré-requis

Ils définissent l'ensemble d'attributs négociables et pour chaque attribut, elles définissent l'ensemble de valeurs négociables. La plage de négociabilité pour chaque attribut peut être défini par la fonction *pneg* :

$\text{pneg} : \mathcal{T} \rightarrow \mathcal{D}_1 \cup \dots \cup \mathcal{D}_{|\mathcal{T}|}$ , par exemple  $\text{pneg}(x_i) = d_i$ , où  $x_i \in \mathcal{T}$  et  $d_i \subseteq \mathcal{D}_i$ .

Si  $d_i = \emptyset$  alors l'attribut  $x_i$  est non négociable, le paramètre *neg* de l'attribut  $x_i$  devient par conséquence *faux* dans la proposition formées par l'agent ( $\text{neg} = \text{faux}$ ) et dans ce cas, la première valeur proposé par l'utilisateur sera la seule valeur acceptée par l'agent.

### III.3.2. Le mécanisme de négociation

Comme il est montré dans la figure. III.3, le mécanisme de négociation comprend plusieurs phases, qui décrivent le cycle de vie d'une locution. Si l'agent est à l'état intermédiaire de la négociation alors le cycle de vie commence par la réception d'une locution de la provenance de l'agent opposant. Une locution entrante est décomposée, en séparant la proposition et l'élément persuasif. Chaque partie est traitée séparément en calculant d'une part la différence entre la proposition reçue et la dernière proposition envoyée sur chaque attribut et d'autre part les différentes forces qui influencent l'état mental de l'agent. Les différences et les forces d'influence calculées sont employées à l'analyse de champ de force afin de définir la nouvelle proposition qui va être incluse dans la locution sortante. Après la validation de la nouvelle proposition, l'agent génère un ensemble de locutions candidats en respectant le protocole de négociation. Enfin, l'agent choisit la locution ayant la plus grande force d'influence parmi les locutions candidats et l'utilise comme une locution sortante.

#### III.3.2.1. Mécanisme d'évaluation

L'évaluation d'une locution entrante comprend l'évaluation de la proposition incluse dans la locution par la comparaison de cette proposition avec la proposition incluse dans la dernière locution sortante et le calcul des différences entre les valeurs des attributs dans chacune d'eux. L'évaluation comprend, en plus, le calcul des forces qui influence l'état mental de l'agent.

Avant le calcul des différences entre les valeurs des attributs, l'agent vérifie s'il y a une collision avec les pré-requis dans la locution entrante à l'aide des règles suivantes:

- (1)  $\neg adopte(locution) \Rightarrow situation = rejet()$
- (2)  $\exists \tau = (x_i, v_i, neg_i) \in \mathcal{P}, neg_i \wedge v_i \notin pneg(x_i)$   
 $\Rightarrow situation = situation \cup rejet(\tau)$
- (3)  $\exists \tau = (x_i, v_i, neg_i) \in \mathcal{P}, \neg neg_i \wedge v_i \notin pneg(x_i) \Rightarrow situation = aband()$

La variable *situation* est un ensemble d'éléments,  $Card(situation) \geq 1$ , qui représente la situation de l'agent devant la locution entrante, il est utilisé dans les règles précédentes pour définir les cas de collision entre la locution entrante et les pré-requis.

Dans les règles précédentes l'ensemble *situation* peut contenir des différents indicateurs de situation : *rejet()* signifie que la locution est rejetée complètement car son type n'est pas adopté, *rejet( $\tau$ )* signifie que la locution est rejetée à cause de  $\tau$  et *aband()* indique l'abandonnement du dialogue de négociation.

Le calcul des différences s'effectue à l'aide de la règle suivante :

$\mathcal{P}'$  la proposition reçue,  $\mathcal{P}$  la dernière proposition envoyé :

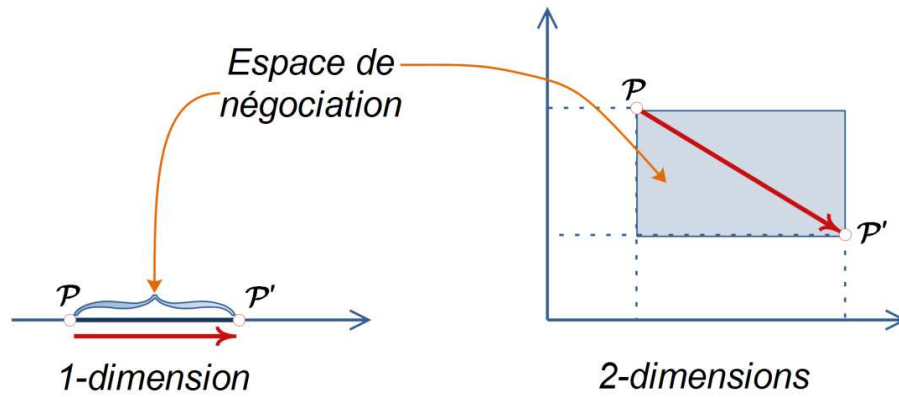
$$(4) \quad \forall (x, v, neg) \in \mathcal{P}, \forall (x', v', neg') \in \mathcal{P}', x = x' \Rightarrow diff(x) = v' \ominus v.$$

L'opération  $\ominus$  est une opération de soustraction spéciale, elle est utilisée dans le calcul de différences. Les attributs de la proposition ne sont pas de même type et quelques attributs n'admettent pas des valeurs numériques. Donc, les différences peuvent être calculées de deux manières : le calcul d'une soustraction entre les valeurs des attributs (i.e.  $x = prix$ ) ou le calcul du degré de distinction selon l'ordre de préférence (i.e.  $x = marque$ ). Le résultat de la différence peut être dans le sens non désiré (cas générale  $diff < 0$ ) et peut être dans le sens désiré (cas de compensation  $diff > 0$  ou concession  $diff = 0$ ). Généralement, la fonction *diff* définit le résultat de différence entre  $\mathcal{P}$  et  $\mathcal{P}'$  pour chaque attribut  $x$ .

La nouvelle proposition sortante est formée à partir de la proposition de la dernière locution sortante et la proposition de la locution entrante par l'application de la technique d'analyse de champ de force proposée par [Lewin, 1947], cette technique est utilisée souvent dans la définition des stratégies, dans la prise de décision et même dans la planification des négociations commerciales. Cette technique est adaptée sur notre problème et formalisée de la manière suivante :

Une proposition est considérée comme un point dans un espace de  $n$ -dimensions, tel que  $n$  est le nombre d'attributs de la proposition. Donc les deux propositions  $\mathcal{P}'$  (la proposition entrante) et  $\mathcal{P}$  (la dernière proposition sortante) sont considérées comme deux points dans un espace de  $n$ -dimensions. Les forces d'influence engendrées par l'élément persuasif associé à  $\mathcal{P}'$  sont appliquées sur  $\mathcal{P}$  dans le sens de  $\mathcal{P}'$ . Par l'application de la technique d'analyse des champs de force, on peut avoir deux nouvelles positions possibles de  $\mathcal{P}$ , soit  $\mathcal{P}$  reste dans la position actuelle (cas de refus), soit elle déplace à la position de  $\mathcal{P}'$  (cas d'acceptation).

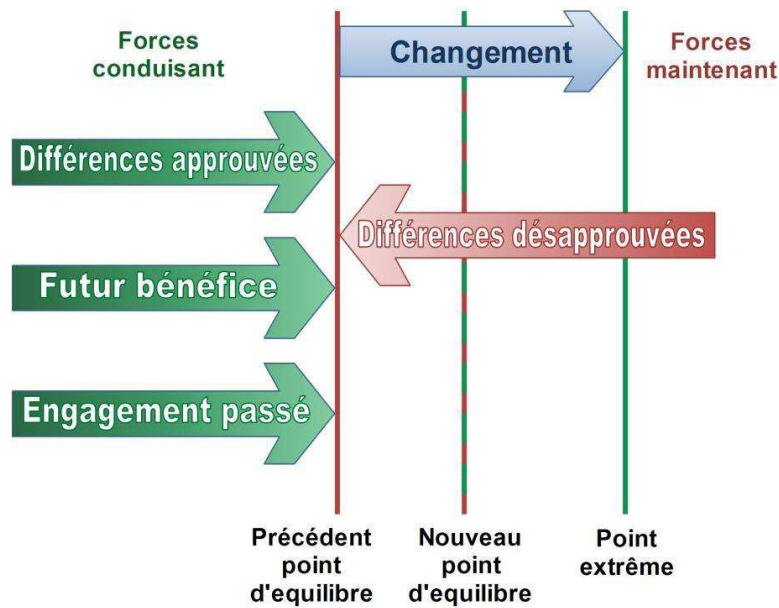
L'adaptation est faite dans le sens de permettre au point correspondant à  $\mathcal{P}$  de prendre une nouvelle position dans l'espace de négociation défini par les deux points  $\mathcal{P}$  et de  $\mathcal{P}'$  comme il est illustré dans la figure suivante :



**Figure. III.4 :** Espace de négociation entre deux propositions

La nouvelle proposition calculée  $\mathcal{P}''$  est définie par les valeurs des attributs qui appartiennent à l'espace de négociation. S'il existe un attribut  $x$  tel que  $x = v''$  pour  $\mathcal{P}''$  et  $x = v'$  pour  $\mathcal{P}'$  et  $v' = v''$  alors on élimine une dimension de l'espace de négociation. De cette manière, si la dimension de l'espace de négociation diminue à 0 alors c'est le cas d'un accord.

Selon le problème de négociation que nous traitons, plusieurs forces sont définies dans le champ de la négociation. L'élément persuasif inclus dans une locution influence ces forces de différentes manières. Des forces sont augmentées, d'autres forces sont diminuées et le reste des forces gardent leurs valeurs. La figure suivante présente les différentes forces dans le champ de la négociation :



**Figure. III.5 :** Les forces définies dans le champ de la négociation

Les forces influencées par l'élément persuasif ne sont pas appliquées séparément sur chaque attribut de la proposition  $\mathcal{P}$ . Donc, le problème est plus compliqué qu'un simple calcul de la somme des forces physiques. Prenons l'exemple suivant :

Acheteur :  $\mathcal{P} = \{(Marque, B), (Prix, 10 DA), (Garantie, 12 mois)\}$

Vendeur :  $\mathcal{P}' = \{(Marque, B), (Prix, 13 DA), (Garantie, 18 mois)\}$

Acheteur :  $\mathcal{P}'' = \{(Marque, B), (Prix, 11 DA), (Garantie, 15 mois)\}$

Cet exemple explique la dépendance entre les attributs et montre que la modification de la valeur d'un attribut peut influencer la modification de valeurs d'autres attributs dans la prochaine proposition. Donc, il existe d'autres facteurs à introduire dans le calcul de la nouvelle proposition qui ont une relation avec l'état mental de l'agent et la stratégie qu'il suit.

Généralement, les forces définies dans la figure. III.5 sont influencées par les locutions persuasives de différentes manières :

Locution persuasif	Force influencée	Façon d'influence
<i>Encourager</i>	Différences Approuvées (DA)	Incrémenter (+ +)
<i>Decourager</i>	Différences Désapprouvées (DD)	Décémenter (- -)
<i>Promettre</i>	Futur Bénéfice(FB)	Avoir une valeur (>0)
<i>FaireAppel</i>	Engagement Passé (EP)	Avoir une valeur (>0)

**Tableau III.8.** Les forces influencées par les locutions persuasives



Pour chaque analyse, les forces dans le champ de la négociation sont initialisées :

- (5)  $FB(x) = 0, \forall (x, v, neg) \in \mathcal{P}$ .
- (6)  $EP(x) = 0, \forall (x, v, neg) \in \mathcal{P}$ .
- (7)  $DA(x) = 1$ , si  $\exists (x, v, vrai) \in \mathcal{P}, \exists (x, v', vrai) \in \mathcal{P}'$  où  $diff(x) \geq 0$   
 $DA(x) = 0$ , sinon.
- (8)  $DD(x) = 1$ , si  $\exists (x, v, vrai) \in \mathcal{P}, \exists (x, v', vrai) \in \mathcal{P}'$  où  $diff(x) < 0$   
 $DD(x) = 0$ , sinon.

L'élément persuasif  $(exp(\tau), \psi, \mu)$ , associé à la proposition  $\mathcal{P}'$  de la locution entrante, va influencer les forces ( $DA, DD, FB$  et  $EP$ ) qui définissent la nouvelle proposition  $\mathcal{P}''$ .

- Pour chaque attribut  $(x, v) \in \tau$  :
- (9)  $FB(x) = \mu$ , si *locution* = *Promettre*.
- (10)  $EP(x) = \mu$ , si *locution* = *FaireAppel*.
- (11)  $DA(x) = \max(DA(x), \mu)$ , si *locution* = *Encourager*.
- (12)  $DD(x) = \min(DD(x) - \mu, 0)$ , si *locution* = *Decourager*.
- Ensuite, pour chaque attribut  $(x, v, vrai) \in \mathcal{P}$  on a :
- (13)  $Influence(x) = FB(x) + EP(x) + DA(x) - DD(x)$

Nous rappelons que le qualificateur  $\mu$  est calculé de différentes manières (voir les tableaux III.6 et III.7). Dans la règle (11), si la valeur initiale de  $DA(x)$  était 0 alors elle devient  $\mu$ , si elle était 1 elle ne se change pas. La règle (12) est différente, si la valeur initiale de  $DA(x)$  était 1 alors elle diminue par  $\mu$ , si elle était 0 elle ne change pas. Enfin, la règle (13) détermine le degré et le sens d'influence pour chaque attribut  $x \in \mathcal{P}$ , si  $influence(x) > 0$  alors l'agent est influencé sur l'attribut  $(x, v)$ , sinon il n'est pas influencé.

Dans cette phase les croyances de l'agent peuvent être mises à jour si la locution entrante est de type *Encourager* (une valeur d'attribut monté) ou *Decourager*.

### III.3.2.2. Mécanismes de génération et de validation

Dans cette phase l'agent (ayant la main) forme la nouvelle proposition qui représente pour lui le nouveau point d'équilibre. Puisque il existe une dépendance entre les attributs, la nouvelle valeur  $v''$  d'un attribut  $x$  n'est pas calculée séparément par

l'application de la force d'influence obtenue par la règle (13) sur chaque attribut. Les forces d'influences sont alors distribuées sur tous les attributs négociables afin d'avoir un gain maximal et une perte minimale. Donc, la règle du calcul de la nouvelle proposition fait toujours une concession ou une compensation sur l'attribut négociable ayant le poids le plus faible et gardent les valeurs des attributs non négociables et les attributs ayant un poids important.

$$(14) \text{ Influence}(\mathcal{P}) = \text{Moyenne}_{x \in \mathcal{P}}(\text{influence}(x))$$

$$(15) \text{ Si } \text{Influence}(\mathcal{P}) \leq 0 \text{ alors } \mathcal{P}'' = \mathcal{P} \text{ // la locution sera de type } \textit{Proposer}$$

$$\text{Sinon, } v'' = v \oplus \text{diff}(x) * \text{Influence}(\mathcal{P}).$$

Le symbole  $\oplus$  représente l'opération inverse de  $\ominus$ , il est utilisé pour rendre  $(x, v'') \in \mathcal{P}''$  plus proche de  $(x, v') \in \mathcal{P}'$  que  $(x, v) \in \mathcal{P}$ , avec ( $\mathcal{P}$ : la dernière proposition sortante,  $\mathcal{P}'$ : la dernière proposition entrante et  $\mathcal{P}''$ : la nouvelle proposition formée).

Ensuite il génère de nouvelles locutions selon le protocole de négociation qui spécifie les locutions potentielles dans un état de négociation  $S_i$ . La règle suivante définie pour un état  $S_i \in \mathcal{S}$  l'ensemble des types de locutions potentiels ( $\mathcal{Pots}$ ):

$$(16) \text{ Pots}(S_i) = \{loc \in \text{Types} \mid \text{trans}(S_i, loc) \neq \emptyset\}$$

Pour chaque type de locution  $loc \in \mathcal{Pots}(S_i)$  l'agent correspondant à l'état  $S_i$  doit choisir les attributs à inclure dans l'expression tactique  $exp(\tau)$  pour les locutions persuasives:

- Pour  $loc \in \{\textit{Encourager}, \textit{Permettre}, \textit{FaireAppel}\}$  l'agent cherche à renforcer sa nouvelle proposition formée.
- Pour  $loc = \textit{Decourager}$  l'agent cherche à critiquer la dernière proposition qu'il a reçue de l'agent opposant.

### III.3.2.3. Mécanisme de sélection

Dans cette phase l'agent choisit la locution la plus forte, c.-à-d. qui a l'élément  $\mu$  le plus grand. S'il existe plus qu'une locution, alors l'agent choisit la locution la plus préférée selon ces croyances.

# 4

## Contribution 2 : Architecture du système

### IV.1. Introduction

Dans ce chapitre nous présentons l'architecture du système de négociation d'un marché électronique. Les technologies du web sémantique sont utilisées dans la représentation des éléments ainsi que des mécanismes de négociation. Ce chapitre est la deuxième contribution de cette thèse, qui est présentée dans [Said et al., 2011; 2012].

### IV.2. Le web sémantique en tant qu'outil de représentation et de raisonnement

Le Web sémantique [Berners-Lee et al., 2001] est une extension du Web actuel. Il a comme objectif principal d'apporter la sémantique à l'information manipulée en visant à faciliter la communication entre les agents. Tim Bernes-Lee et ces collègues [Berners-Lee et al., 2001] ont pointé au fait que le Web courant est compréhensible par des humains seulement. Le but des technologies du web sémantique est de rendre le contenu des ressources de Web accessible et utilisable par des programmes et des agents logiciel par des systèmes de métadonnées formelles, en utilisant en particulier la famille des langues développées par [W3C].

Les langues du Web sémantiques les plus récentes sont OWL (Ontology Web Language) [McGuinness & Van Harmelen, 2004] pour définir des ontologies de Web et SWRL (Semantic Web Rule Language) [Horrocks et al., 2004] pour définir des règles dans les ontologies OWL. SWRL est la combinaison du langage OWL (comme un langage d'Ontologie Web) et le langage RuleML [Horrocks et al., 2004] (comme un langage de règle). L'extension de règle est matérialisée par l'ajout des axiomes de règle à l'ensemble d'axiomes de OWL. Cette proposition étend la syntaxe abstraite de OWL pour inclure la syntaxe de ces règles et la sémantique de OWL pour fournir une signification formelle pour les ontologies qui incluent des règles écrites en cette

syntaxe. L'extension est strictement syntactique et sémantique, et a par conséquent une intégration forte à OWL.

Le langage OWL offre trois sous-langages avec des puissances croissantes d'expression [McGuinness et Van Harmelen, 2002] :

- Le sous-langage OWL-Lite est capable de représenter une hiérarchie de classifications et de mécanismes de contraintes simples. Par exemple, OWL-Lite gère des contraintes de cardinalité, mais il ne permet que des valeurs de cardinalité de 0 ou 1.
- Le sous-langage OWL-DL permet une expressivité maximum sans sacrifier la complétude de calcul (toutes les inférences sont sûres d'être prises en compte) et la décidabilité (tous les calculs seront terminés dans un intervalle de temps fini) des systèmes de raisonnement. Le langage OWL-DL comprend toutes les structures de langage de OWL avec des restrictions comme la séparation des types (une classe ne peut pas être en même temps un individu ou une propriété, une propriété être un individu ou une classe). OWL DL se nomme ainsi pour sa correspondance avec la logique de description, un champ de la recherche portant sur un fragment décidable particulier de la logique de premier ordre. Le langage OWL DL, conçu pour gérer le secteur existant de la logique de description, offre les propriétés de calcul souhaitées pour les systèmes de raisonnement.
- Le langage OWL-Full permet une expressivité maximum et une liberté syntaxique de RDF, mais sans garantie de calcul. Par exemple, dans OWL-Full, on peut simultanément traiter une classe comme une collection d'individus et comme un individu à part entière. Le sous-langage OWL-Full permet à une ontologie d'augmenter la signification du vocabulaire prédéfini (RDF ou OWL). Mais un système de raisonnement ne pourra probablement pas mettre en œuvre toutes les caractéristiques de OWL-Full.

Dans notre travail nous avons adopté le sous-langage OWL-DL car il permet le raisonnement sur tous ce qu'il est capable d'exprimer.

Les règles SWRL sont de la forme d'une implication entre un antécédent (corps) et un conséquent (entête). La signification prévue peut être lue comme : toute fois que les conditions spécifiées dans l'antécédent sont tenues, alors les conditions spécifiées dans le conséquent doivent également se tenir.

A cause que la syntaxe originale de SWRL est compliquée et difficile à lire, une forme plus simple est utilisée dans la plupart des travaux publiés. Dans cette syntaxe la règle SWRL a la forme : *antécédent*  $\Rightarrow$  *conséquent*, les atomes d'une règle SWRL peuvent être de la forme :

- $C(x)$ , où  $C$  est une classe OWL ou une plage de données et  $x$  est soit un variable, individu OWL (instance de classe) ou une valeur de données OWL.
- $P(x,y)$ , où  $P$  est une propriété OWL (une propriété objet ou une propriété de type donné),  $x$  est soit un variable ou un individu OWL et  $y$  est soit un variable, un individu OWL ou une valeur de donnée OWL.
- $sameAs(x,y)$ ,  $differentFrom(x,y)$ , où  $x, y$  sont des variables ou des individus OWL.
- $BuiltIn(r,x,...)$ , où  $r$  est une relation *built-in* et  $x,...$  sont des valeurs de données OWL, les *built-ins* sont prédéfinis dans d'autres ontologies qui doivent être importées pour qu'on puisse utiliser ces *built-ins*.

Un exemple simple de règle SWRL peut être représenté comme suite :

$$parent(?x, ?y) \wedge frere(?y, ?z) \Rightarrow oncle(?x, ?z)$$

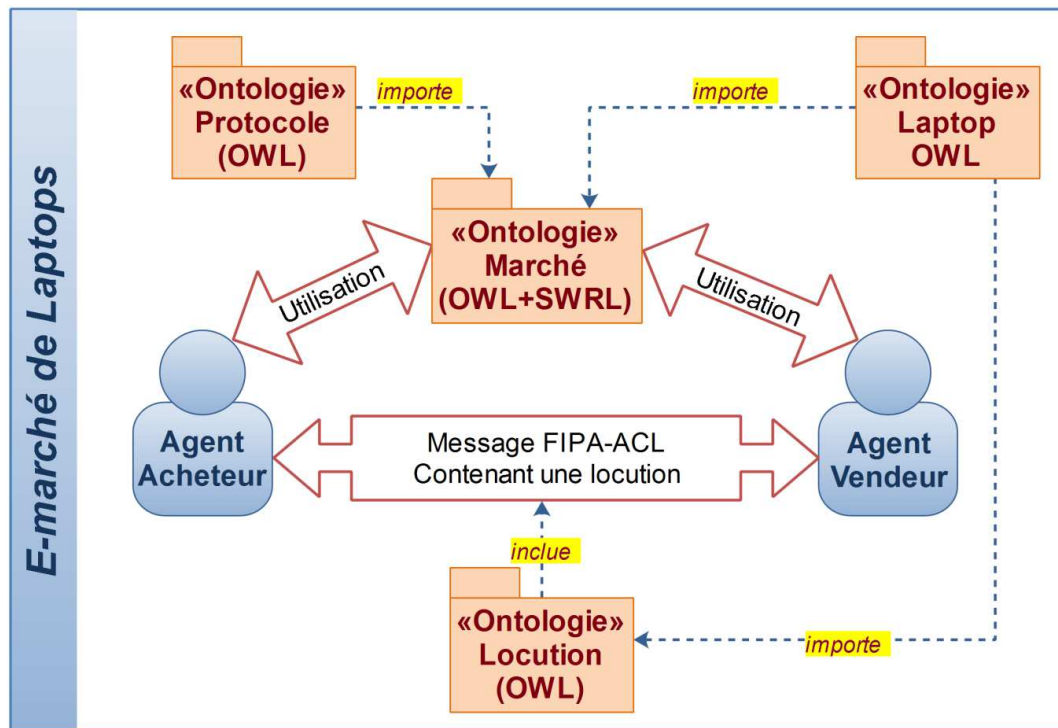
La capacité d'expression du langage SWRL est augmentée par la définition de nombreuse *built-ins* regroupées en familles selon leur utilisation. Les *built-ins* les plus connues sont :

Built-ins	Description
<i>swrlb</i>	<i>Built-ins</i> de base, utilisées pour effectuer les opérations les plus connues (arithmétiques, logiques, chaînes de caractères,...).
<i>swrlm</i>	<i>Built-ins</i> pour effectuer les fonctions mathématiques les plus connues.
<i>swrlx</i>	<i>Built-ins</i> d'extension, utilisées pour la création des classes ou des

	individus lors de l'exécution des règles.
<i>sqwrl</i>	<i>Built-ins</i> de requêtes, utilisées pour la formulation des requêtes sur les ontologies ces une combinaison entre SWRL et SPARQL. Il existe un sous-ensemble de <i>sqwrl</i> pour la gestion des collections (ensemble) qui est utilisé dans règles présentées ultérieurement. Les <i>Built-ins</i> de la gestion des collections sont précédées par ‘°’.

### IV.3. Architecture de l'espace de négociation inter-agents

Dans cette section, nous présentons l'architecture de l'espace de négociation inter-agent d'une place du marché électronique (e-marché). La représentation de plusieurs éléments dans cette espace est basée sur la technologie du web sémantique. Dans le domaine d'e-commerce l'objet négocié peut être généralement un produit ou un service. Puisque chaque type de produit ou de service est un ensemble d'attributs différent, nous avons choisi l'ordinateur portable (Laptop) à titre d'exemple du produit pour l'approche de négociation proposée. Ce choix est motivé par le fait qu'il existe aujourd'hui plusieurs marchés et boutiques électroniques de vente de laptops. En plus, les laptops peuvent être décrits et comparés par plusieurs caractéristiques de différents types. Le schéma suivant décrit les éléments de l'espace inter-agent et les relations entre ces éléments :



**Figure IV.1 :** Architecture de l'espace inter-agent d'un e-marché de Laptops

Dans cette architecture nous distinguons deux types d'agent : *Agent Acheteur* et *Agent Vendeur*. La négociation se déroule toujours entre deux différents types d'agent. Les agents communiquent entre eux par l'échange de messages, en utilisant le langage de communication des agents FIPA-ACL [FIPA].

Dans cette architecture, quatre différentes ontologies sont utilisées. Les ontologies sont définies en langage OWL-DL, mais il existe une représentation visuelle de OWL-DL par UML présenté dans [Brockmans et al., 2004]. Donc, toutes les ontologies seront représentées dans ce chapitre en UML.

- *Ontologie Laptop (OWL)* : cette ontologie définit les différentes valeurs des caractéristiques du Laptop (marque, processeur, disque dure, ram, lecteur de disque et affichage). Elle importée dans l'ontologie *Marché* pour la description des Laptops existants à l'e-marché. La figure IV.2 présente une description UML de l'ontologie *Laptop*.
- *Ontologie Protocole (OWL)* : c'est l'ontologie qui décrit les différents état et transitions du protocole de négociation respecté par les agents (vendeurs et acheteurs)

pendant le dialogue de négociation. L'ontologie *Protocole* contient deux classes principales *Etat* et *Transition*, les instances de la classe *Etat* représentent les différents états dans le protocole négociation et les instances de la classe *Transition* représentent toutes les transitions possibles dans le protocole de négociation. La figure IV.3 présente une description UML de l'ontologie *Protocole*.

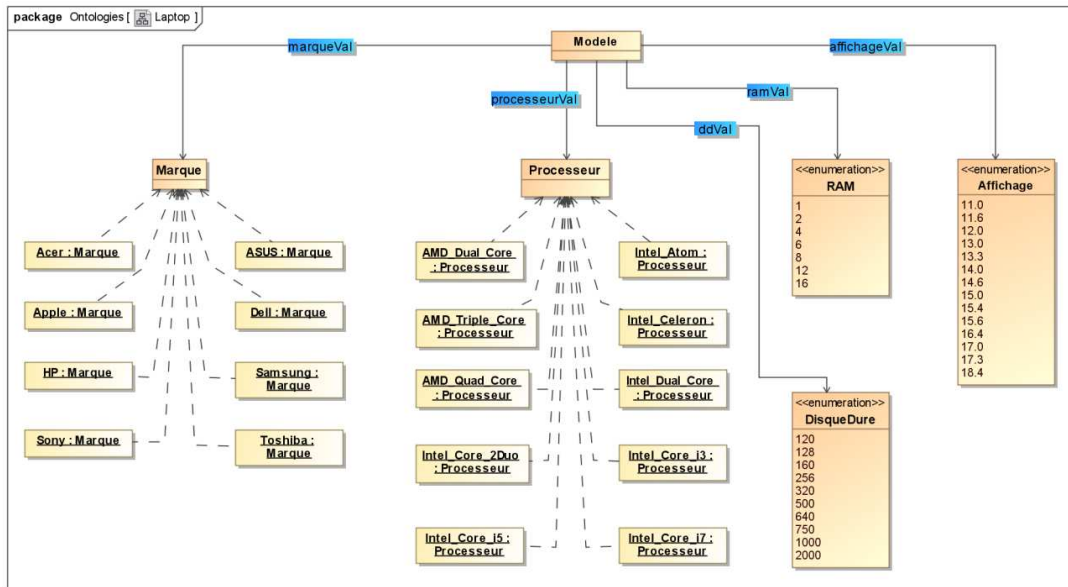


Figure IV.2 : Description UML de l'ontologie *Laptop*

- *Ontologie Locution (OWL)* : elle décrit les locutions échangées entre les agents. C'est une représentation ontologique du langage de négociation  $\mathcal{L}$  défini dans le chapitre précédent. Les locutions générées par les agents (vendeurs et acheteurs) sont héritées des classe *LocPers* (locution persuasive) et *LocNonPers* (locution non persuasive). L'ontologie *Laptop* est importée dans cette ontologie et utilisée pour définir les différentes valeurs des attributs de la proposition incluse dans une locution. La figure IV.4 présente une description UML de l'ontologie *Locution*.

- *Ontologie Marché (OWL + SWRL)* : elle représente les connaissances et elle permet de stocker les données qui concernent tout l'e-marché. Dans cette ontologie, deux autres ontologies sont importées *Laptop* et *Protocol*. L'ontologie *Marché* comprend :
  - Les identifiants de tous les utilisateurs enregistrés (acheteurs et vendeurs).
  - Tous les modèles de laptops présentés sur l'e-marché.



- Les dialogues de négociation en cours de déroulement et l'historique de tous les dialogues qui sont terminés, plus l'état dans lequel chaque dialogue est terminé et les promesses données.
- Les règles de conformité (en SWRL) ou les règles du marché qui dirigent les dialogues de négociation.

La figure IV.5 présente une description UML de l'ontologie *Marché*.

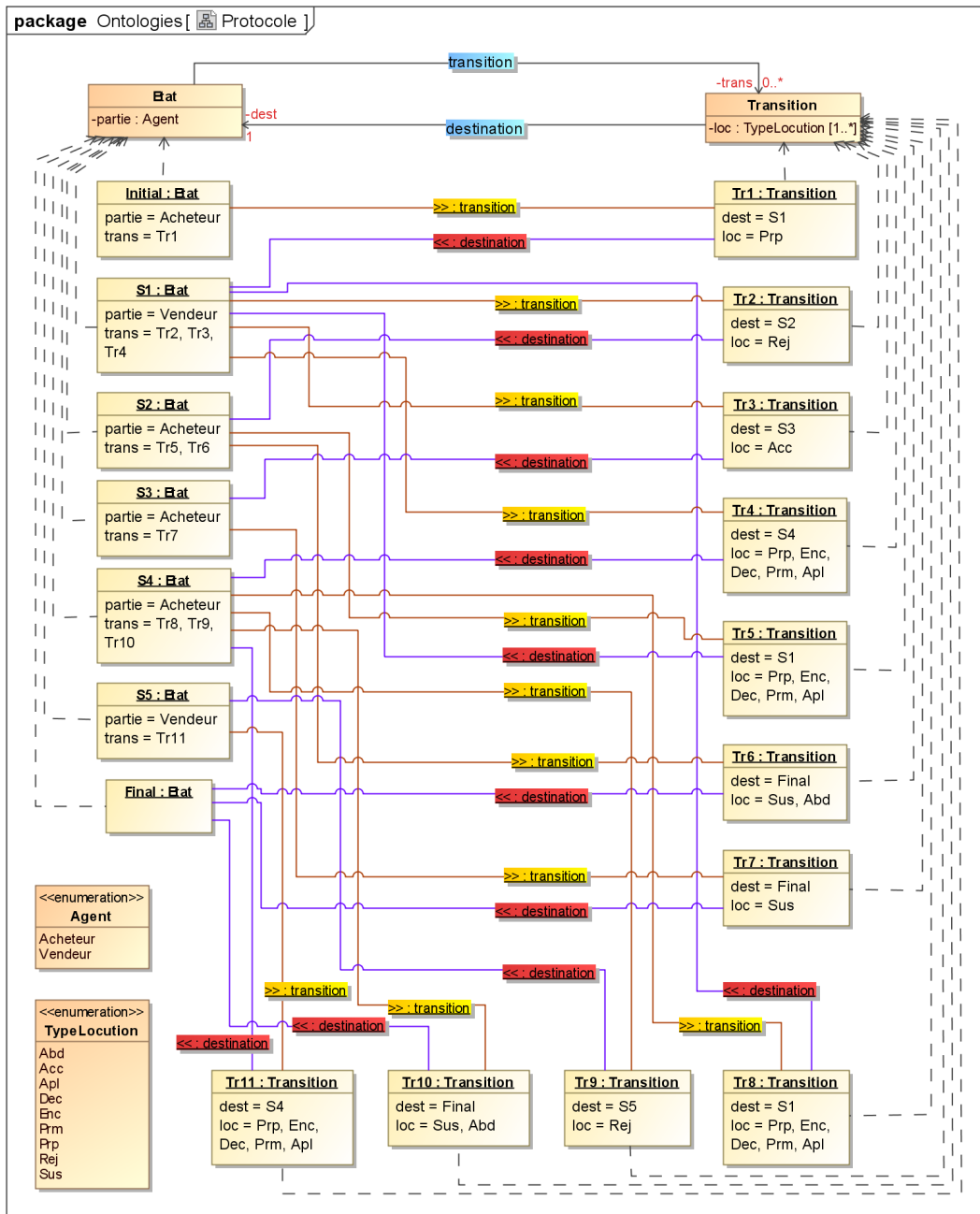


Figure IV.3 : Description UML de l'ontologie *Protocole*

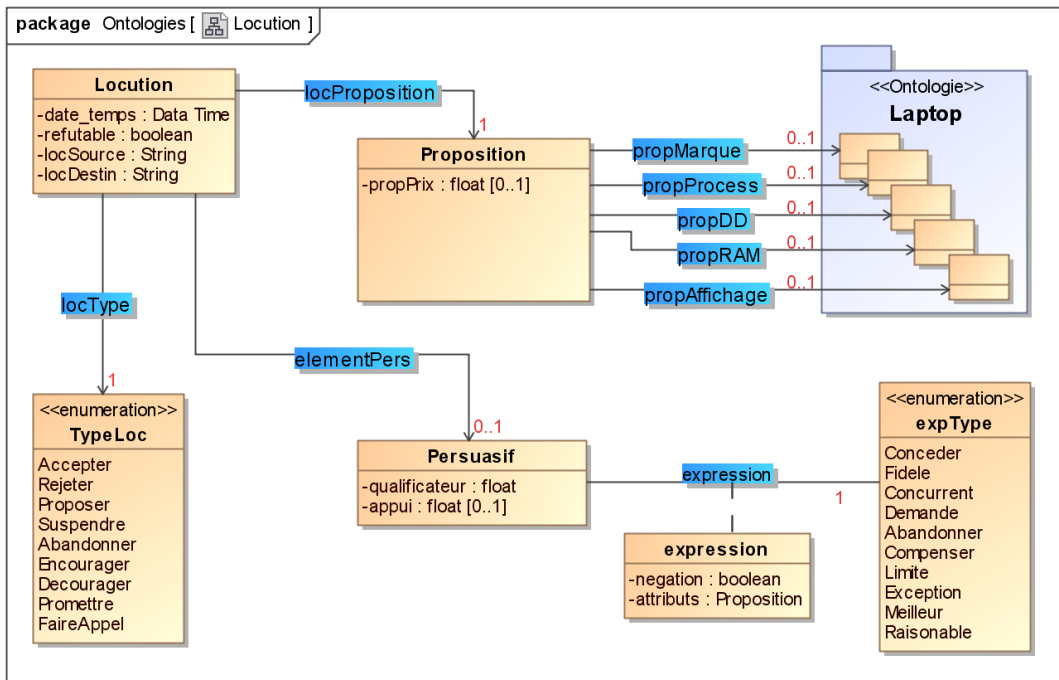


Figure IV.4 : Description UML de l'ontologie *Locution*

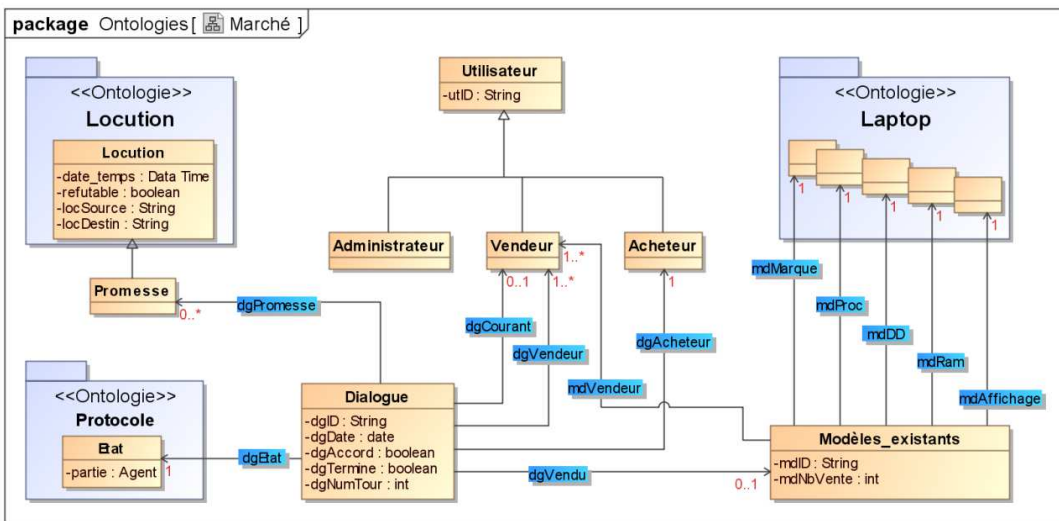


Figure IV.5 : Description UML de l'ontologie *Marché*

#### IV.4. Architecture interne des agents (Vendeurs et Acheteurs)

Dans le chapitre précédent, nous avons présenté les différents éléments de l'espace intra-agent et les différentes phases comprises dans le mécanisme de négociation. Les deux types d'agent (Vendeur et Acheteur) ont presque la même architecture qui comprend essentiellement une ontologie. L'ontologie *Vendeur* ou *Acheteur* se compose d'un ensemble d'éléments *OWL* (classes, relations, individus, ...) et d'un ensemble de règles *SWRL* qui décrivent le mécanisme de négociation. La figure IV.6 représente l'architecture interne d'un agent *Vendeur* ou *Acheteur*.

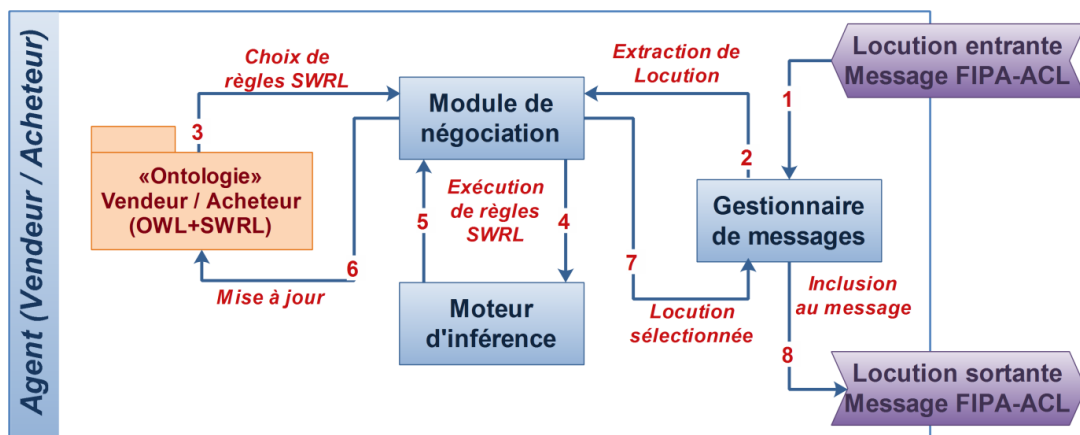


Figure IV.6 : Architecture interne d'un agent (Vendeur / Acheteur)

A l'intérieur d'un agent (Vendeur / Acheteur) le message reçu, qui contient la locution entrante, est traité par le gestionnaire de message. Ce dernier retire la locution incluse dans le message et le transmet au module de négociation. Le traitement de la locution entrante est fait par le module de négociation, en choisissant pour chaque phase du mécanisme de négociation les règles *SWRL* correspondantes à partir de l'ontologie. Les règles choisies sont exécutées par un moteur d'inférence qui retourne les résultats au module de négociation. Ensuite, l'ontologie est enrichie par la mise à jour de l'état mental. Enfin, la locution sélectionnée est envoyée au gestionnaire de messages, qui l'inclut dans le message de réponse. Les locutions échangées entre les agents sont des instances de l'ontologie *Locution*, elles sont alors définies en langage *OWL-DL*. La question posée est : comment inclure une instance d'ontologie dans un message *FIPA-ACL*? Schiemann et Schreiber [2006] ont discuté la possibilité d'utiliser le langage

OWL-DL autant qu'un langage de contenu de FIPA-ACL. L'utilisation de cette solution sera discutée en détail au chapitre suivant.

Nous avons défini deux ontologies : ontologie *AcheteurOWL* et ontologie *VendeurOWL*. Chaque agent Acheteur du marché électronique utilise sa propre ontologie *AcheteurOWL* et chaque agent Vendeur du marché électronique utilise sa propre ontologie *VendeurOWL*. Les figures IV.7 et IV.8 présentent respectivement une description UML des ontologies *AcheteurOWL* et *VendeurOWL*.

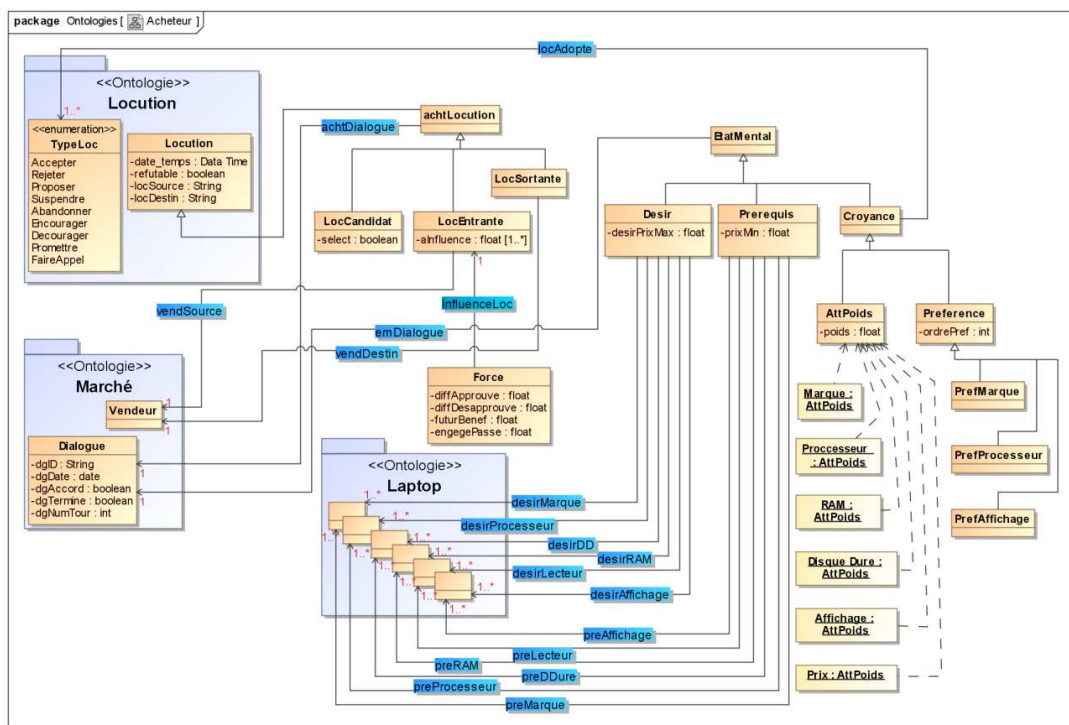


Figure IV.7 : Description UML de l'ontologie *Acheteur*

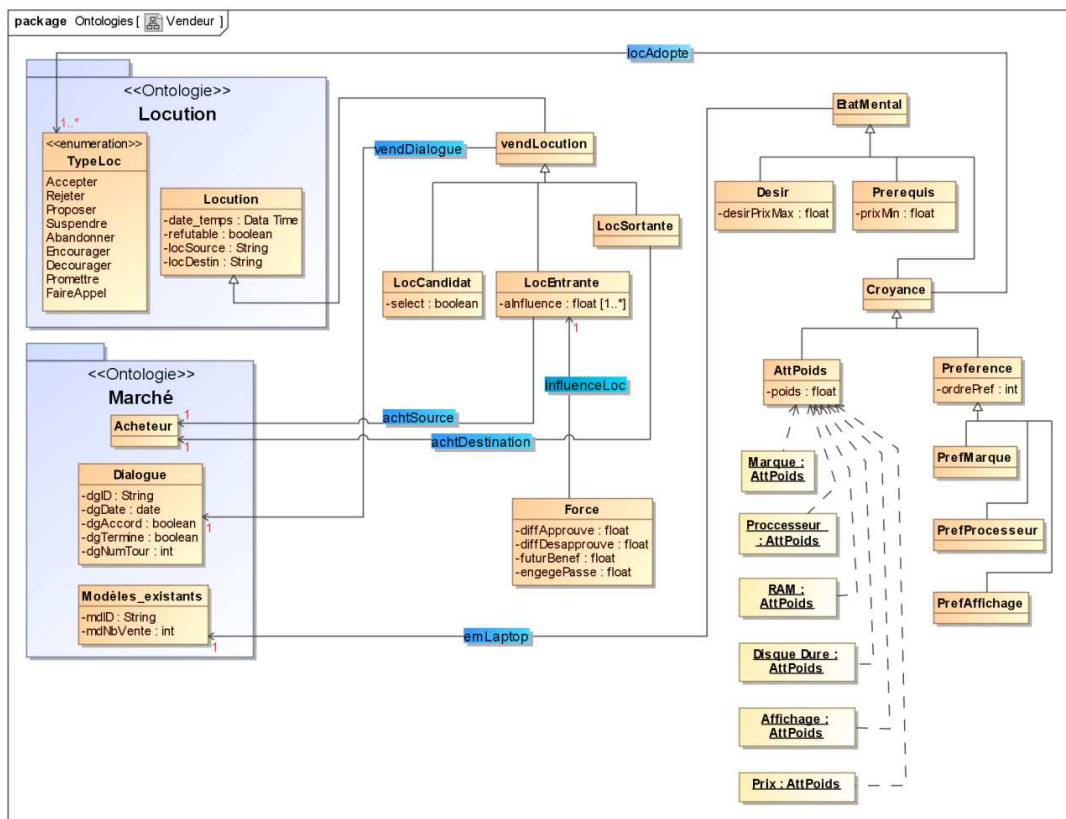


Figure IV.8 : Description UML de l'ontologie *Vendeur*

## IV.5. La représentation des mécanismes de négociation par les règles SWRL

Dans cette section, les règles décrivant les différents mécanismes de négociation, présentées dans le chapitre précédent, seront représentées en langage SWRL. Ce langage est puissant et expressive, mais à cause de sa grande complexité, nous allons paraphraser chaque règle SWRL par une règle de type IF-THEN plus simple et plus lisible en gardant les mêmes caractéristiques du langage SWRL. Pour montrer le niveau de complexité de la description en SWRL nous allons présenter la première règle en SWRL et en IF-THEN et le reste sera présenté seulement en IF-THEN.

Avant de commencer la description des règles nous présentons la syntaxe des règles (IF-THEN) :

**Syntaxe d'une règle (IF-THEN):**

```
regle → IF antecedent THEN consequent
antecedent → atome AND antecedent
              | atome WHERE antecedent
              | atome
consequent → terme AND consequent
              | terme
terme → atome
        | fonction
        | [ consequent ] FOR IDENT : NOMBRE .. IDENT ( IDENT )
        | [ consequent ] FOR IDENT ( IDENT , CONST )
atome → IDENT ( IDENT ) // une classe
        | IDENT ( IDENT , IDENT ) // une propriété ou une relation
        | IDENT ( IDENT , CONST ) // une propriété
        | [ antecedent ] FOR IDENT : NOMBRE .. IDENT ( IDENT )
        | [ antecedent ] FOR IDENT ( IDENT , CONST )
fonction → IDENT ( args , IDENT ) // 2ème IDENT reçoit le résultat de la fonction
args → IDENT , args
        | CONST , args
        | IDENT
        | CONST
```

**IV.5.1. Les règles d'évaluation de la locution entrante**

Les règles d'évaluation permettent de:

- spécifier si la locution entrante est persuasive ou non persuasive;
- vérifier s'il existe une collision entre la proposition reçue et les pré-requis;
- calculer la différence pour chaque attribut entre sa valeur dans la proposition reçue et sa valeur dans la dernière proposition envoyée;
- calculer le degré d'influence de la locution entrante selon (le type de locution et l'élément persuasif);

- déterminer les forces qui contribuent à la prise de décision influencées par la locution reçue;
- mettre à jour l'ontologie *AcheteurOWL* / *VendeurOWL*.

La première étape d'évaluation consiste à déterminer si la locution entrante est persuasive ou non persuasive. Ensuite, vérifier qu'il n'y a plus de collision avec les pré-requis, par la comparaison des valeurs des attributs dans la proposition reçue avec les valeurs pré-requis. En cas de collision, la locution sera directement rejetée.

La règle1 permet d'effectuer la vérification de collision:

**Règle1 (IF-THEN):**

**IF** LocEntrante(loc) **AND** aProposition(loc, prop) **WHERE**  
aAttribut(prop, attribs) **WHERE** [estPrerequis(attribs[i], true)] **FOR**  
i:1..TAILLE(attribs)  
**THEN** refutable(loc,false)

**Règle1 (SWRL):**

LocEntrante(?loc\_ent)  $\wedge$  locution:estDernier(?loc\_ent, true)  $\wedge$   
laptop:ddVal(?loc\_ent, ?dd\_val)  $\wedge$  laptop:ecranVal(?loc\_ent, ?ecran\_val)  $\wedge$   
laptop:marqueVal(?loc\_ent, ?marq\_val)  $\wedge$  laptop:prixVal(?loc\_ent, ?prix\_val)  $\wedge$   
laptop:ramVal(?loc\_ent, ?ram\_val)  $\wedge$  laptop:processVal(?loc\_ent, ?process\_val)  $\wedge$   
prereqMarque(AchPrereq, ?pre\_marque)  $\wedge$   
prereqProcessor(AchPrereq, ?pre\_process)  $\wedge$  prereqDD(AchPrereq, ?pre\_dd)  $\wedge$   
prereqEcran(AchPrereq, ?pre\_ecran)  $\wedge$  prereqPrixMin(AchPrereq, ?pre\_prix)  $\wedge$   
prereqRam(AchPrereq, ?pre\_ram)  $\circ$  sqwrl:makeSet(?set\_marque, ?pre\_marque)  $\wedge$   
sqwrl:makeSet(?set\_process, ?pre\_process)  $\wedge$  sqwrl:makeSet(?set\_dd, ?pre\_dd)  $\wedge$   
sqwrl:makeSet(?set\_ecran, ?pre\_ecran)  $\wedge$  sqwrl:makeSet(?set\_ram, ?pre\_ram)  $\circ$   
sqwrl:element(?dd\_val, ?set\_dd)  $\circ$  sqwrl:element(?ecran\_val, ?set\_ecran)  $\circ$   
sqwrl:element(?marq\_val, ?set\_marque)  $\circ$  sqwrl:element(?ram\_val, ?set\_ram)  $\circ$   
sqwrl:element(?process\_val, ?set\_process)  $\wedge$   
swrlb:lessThanOrEqual(?prix\_val, ?pre\_prix)  $\rightarrow$  locution:refutable(?loc\_ent, false)

Le calcul de la différence pour chaque attribut se fait à l'aide de la règle 2 :

**Règle2 (IF-THEN):**

**IF** LocEntrante(loc\_ent) **AND** locProposition(loc\_ent, prop\_ent) **AND**  
aAttribut(prop\_ent, attribs\_ent) **AND** LocSortante(loc\_sort) **AND**  
aProposition(loc\_sort, prop\_sort) **AND** aAttribut(prop\_ent, attribs\_ent) **AND**  
aAttribut(prop\_sort, attribs\_sort) **AND** [preference(attribs\_ent[i], pref\_ent[i])  
**AND** preference(attribs\_sort[i], pref\_sort[i])] **FOR** i:1.. TAILLE(attribs\_ent)  
**THEN** [SOUST(pref\_sort[i], pref\_ent[i], difference[i])] **FOR** i:1.. TAILLE(pref\_ent)  
**AND** aDifference(loc\_ent, difference)

Dans cette règle la différence est calculée selon l'ordre de préférence, spécifié dans les croyances de l'agent pour tous les attributs ayant des valeurs numériques ou non numériques. La propriété *preference* utilisée dans cette règle représente une évaluation de préférence unifié pour tous les attributs quelque soit leurs.

L'étape suivante de l'évaluation consiste à calculer le degré d'influence dans le cas de locution persuasive. Le degré d'influence est calculé en fonction du qualificateur, du type de locution et des poids des attributs de l'élément persuasif:

$influence(loc\_ent) = qualificateur(loc\_ent) \times \sum_{i \in U} poids(attribut_i)$ , où U représente le nombre d'attributs dans l'expression de l'élément persuasif.

**Règle3 (IF-THEN):**

**IF** LocEntrante(loc\_ent) **AND** locProposition(loc\_ent, prop\_ent) **AND**  
aAttribut(prop\_ent, attribs\_ent) **WHERE** [ attPoids(attribs\_ent [i], poids[i]) ]  
**FOR** i:1..TAILLE(attribs\_ent) **AND** elementPers(loc\_ent, pers\_ent) **WHERE**  
qualificateur(loc\_ent, qualif\_ent) **AND** expression(pers\_ent, exp\_ent) **WHERE**  
attributs(exp\_ent, exp\_attribs)  
**THEN** [ MULT(poids[i], qualify\_ent[i], influence[i]) ] **FOR** i:DANS(exp\_attribs)  
**AND** aInfluence(loc\_ent, influence)

Après le calcul de degré d'influence, les forces qui participent à la prise de décision et qui sont influencées par la locution entrante, sont déterminées en utilisant la technique d'analyse de champ de force. Les différents types de forces et leur calcul sont



expliqués dans le chapitre précédent (tableau III.8 et règles 5, 6, 7, 8). Les règles qui déterminent ces forces sont définies dans la section suivante.

#### IV.5.2. Les règles de génération de locutions candidats

Les règles de génération de locutions candidats permettent à l'agent de générer des locutions candidats pour répondre à la locution entrante. Les types de locutions candidates sont déterminés dans la règle 4 selon l'état de négociation et suivant le protocole défini dans l'ontologie *Protocole*.

**Règle4 (IF-THEN):**

**IF** LocEntrante(loc\_ent) **AND** aDialogue(loc\_ent, dialogue) **WHERE**  
dgEtat(dialogue, etat) **WHERE** transition(etat, trans) **WHERE**  
loc(trans, loc\_pots)  
**THEN** [NOUVEAU(candidats[i]) **AND** locType(candidats [i], loc\_pots [i]) **AND**  
LocCandidat(candidats[i])] **FOR** i:1..TAILLE(candidats)

Ensuite, une nouvelle proposition est déterminée par le calcul d'un nouveau point d'équilibre en appliquant la technique d'analyse de champ de force présenté dans le chapitre précédent. Les règles suivantes permettent de calculer les différentes forces utilisées pour la détermination de la nouvelle proposition.

**Règle5 (IF-THEN):**

**IF** LocEntrante(loc\_ent) **AND** aDifference(loc\_ent, difference)  
**THEN** [ NOUVEAU(force[i]) **AND** Force(force[i]) **AND**  
influenceLoc(force[i], loc\_ent) [ diffApprouve(forces[i],1) **AND**  
diffDesapprouve(force[i],0) ] **FOR** SUPEREG(difference[i], 0) **AND**  
[ diffApprouve(force[i],0) **AND** diffDesapprouve(force[i],1) ] **FOR**  
INFER(difference[i], 0) ] **FOR** i:1..TAILLE(difference)

**Règle6 (IF-THEN):**

**IF** LocEntrante(loc\_ent) **AND** locType(loc\_ent, Encourager) **AND**  
aInfluence(loc\_ent, influence) **AND** elementPers(loc\_ent, pers\_ent) **WHERE**  
expression(pers\_ent, exp\_ent) **WHERE** attributs(exp\_ent, exp\_attribs) **AND**  
Force(force) **AND** influenceLoc(force, loc\_ent) **AND**  
diffApprouve(force, approuve)

**THEN** [ diffApprouve(force[i],MIN(approuve[i]+influence[i],1)) ]

**FOR** i: DANS(exp\_attribs)

---

/\* Locution= Encourager => diffApprouve est incrémentées \*/

**Règle7 (IF-THEN):**

**IF** LocEntrante(loc\_ent) **AND** locType(loc\_ent, Decourager) **AND**  
aInfluence(loc\_ent, influence) **AND** elementPers(loc\_ent, pers\_ent) **WHERE**  
expression(pers\_ent, exp\_ent) **WHERE** attributs(exp\_ent, exp\_attribs) **AND**  
Force(force) **AND** influenceLoc(force, loc\_ent) **AND**  
diffDesapprouve(force, desapprouve)

**THEN** [ diffDesapprouve(force[i], MAX(desapprouve[i]- influence[i],0)) ]

**FOR** i: DANS(exp\_attribs)

---

/\* Locution= Decourager => diffDesapprouve sont décrémentées \*/

**Règle8 (IF-THEN):**

**IF** LocEntrante(loc\_ent) **AND** locType(loc\_ent, Promettre) **AND**  
aInfluence(loc\_ent, influence) **AND** elementPers(loc\_ent, pers\_ent) **WHERE**  
expression(pers\_ent, exp\_ent) **WHERE** attributs(exp\_ent, exp\_attribs) **AND**  
Force(force) **AND** influenceLoc(force, loc\_ent)

**THEN** [ futurBenef{force[i], influence[i]} ] **FOR** i: DANS(exp\_attribs)

```
/* Locution=Promettre => FuturBenef = Degré d'influence*/
```

**Règle9 (IF-THEN):**

```
IF LocEntrante(loc_ent) AND locType(loc_ent, FaireAppel) AND  
aInfluence(loc_ent, influence) AND elementPers(loc_ent, pers_ent) WHERE  
expression(pers_ent, exp_ent) WHERE attributs(exp_ent, exp_attribs) AND  
Force(force) AND influenceLoc(force, loc_ent)  
THEN [ engegePasse(force[i], influence[i]) ] FOR i: DANS(exp_attribs)
```

```
/* Locution=FaireAppel => engegePasse = Degré d'influence*/
```

Après la définition des différentes forces, la nouvelle proposition est calculée par l'application de ces forces sur les attributs concernés. Il est important de considérer que les valeurs des attributs, qui déterminent les caractéristiques des laptops, doivent être validées par la recherche d'un laptop parmi les laptops présentés par l'agent vendeur ayant les caractéristiques les plus proches des valeurs des attributs calculés.

**Règle10 (IF-THEN):**

```
IF LocCandidat(loc_cand) AND locProposition(loc_cand, prop_cand) AND  
aAttribut(prop_cand, attribs_cand) AND vendDestin(loc_cand, vendeur) AND  
ModeleExist(laptop) AND modelVendeur(laptop, vendeur) AND  
aCaracteristique(laptop, caracter)  
THEN REMPLACER(caracter, attribs_cand) WHERE [MIN(DIFF(attribs_cand[i],  
caracter[i]))] FOR i:1..TAILLE(caracter)
```

```
/* Pour chaque LocCandidat, on choisie le Laptop ayant les valeurs de caractéristiques  
les plus proches des attributs de la proposition */
```

La dernière étape dans la phase de génération de locution est la création de l'élément persuasif. Prenons l'exemple d'un élément persuasif d'encouragement d'un agent vendeur : *"La marque (HP) est parmi les marques les plus demandées au*

*marché*". L'appui de cette locution doit être présenté sous forme d'une statistique sur les laptops de la marque "HP" vendus au marché.

**Règle11 (IF-THEN):**

**IF** LocCandidat(loc\_cand) **AND** locProposition(loc\_cand, prop\_cand) **AND**  
elementPers(loc\_cand, pers\_cand) **AND** expression(pers\_cand, express\_cand)  
**WHERE** attributs(express\_cand, attrib\_cand) **AND** aVal(attrib\_cand, val\_cand)  
**WHERE** ordreVente(val\_cand, ordre\_vente)  
**THEN** appui(pers\_cand, ordre\_vente)

**Règle12 (IF-THEN):**

**IF** LocCandidat(loc\_cand) **AND** locProposition(loc\_cand, prop\_cand) **AND**  
elementPers(loc\_cand, pers\_cand) **AND** expression(pers\_cand, express\_cand)  
**WHERE** appui(express\_cand, appui\_cand) **AND**  
attributs(express\_cand, attrib\_cand) **WHERE**  
ensembleVal(attrib\_cand, ensble\_val)  
**THEN** TAILLE(ensble\_val, taille) **AND** SOUST(taille, appui\_cand, val1) **AND**  
SOUST(taille, 1, val2) **AND** DIVIS(val1, val2, val3) **AND**  
qualificateur(express\_cand, val3)

---

/\* taille = taille de l'ensemble de valeur pour l'attribut utilisé dans l'expression

ordre = ordre de vente de l'attribut utilisé dans l'expression

qualificateur = (taille – ordre) / (taille – 1) \*/

**IV.5.3. La règle de sélection**

Dans la phase de sélection l'agent choisie la locution la plus forte, en comparant les qualificateurs de toutes les locutions candidats.

**Règle14 (IF-THEN):**

**IF** [ LocCandidat(loc\_cand[i]) **AND** elementPers(loc\_cand[i], pers\_cand[i])  
**WHERE** qualificateur(loc\_cand[i], qualif\_cand[i]) ] **FOR** i:1..length(loc\_cand)  
**THEN** select(loc\_cand[j], true) **WHERE** MAX(qualif\_cand[j])

Dans le cas où il existe plus qu'une locution sélectionnée, l'agent choisit la locution ayant le type le plus préféré (état mental).

#### IV.6. Le processus d'achat

Le processus d'achat décrit les différentes étapes du dialogue de négociation pour l'achat d'un laptop. Dans le processus d'achat la négociation entre un agent vendeur et un agent acheteur ne décrit qu'une seule tour du dialogue complet de négociation, qui se déroule entre un agent acheteur et un ensemble d'agents vendeurs. Donc, le dialogue de négociation complet se déroule de la manière suivante :

- 1) *L'acheteur remplit les informations concernant le produit demandé **Laptop** (caractéristiques, croyances, proposition désirée et préconditions), ensuite l'agent **Acheteur** sera lancé par le système après l'application d'un contrôle de conformité avec la politique générale de marché (règles de marché).*
- 2) *L'agent **Acheteur** fait une requête dans l'ontologie de marché pour connaître les vendeurs de **Laptop** désiré, ensuite il reçoit une liste de tous les vendeurs concernés.*
- 3) *L'agent **Acheteur** envoie un message **CFP** (Call For Proposal) aux agents **Vendeur<sub>i</sub>** concernés par la négociation.*
- 4) *Les agents **Vendeur<sub>i</sub>** répondent à l'agent **Acheteur** avec des propositions initiales **Proposition<sub>i</sub>**.*
- 5) *L'agent **Acheteur** évalue les propositions **Proposition<sub>i</sub>**, ensuite il ajoute les agents **Seller<sub>i</sub>** dans une liste triée selon un ordre de préférences.*
- 6) *Si la liste est non vide alors l'agent **Acheteur** fait une négociation avec chacun des agents **Vendeur<sub>i</sub>** de la liste, puis il suspend la négociation à un état final représentant la meilleure offre proposée par **Vendeur<sub>i</sub>**.*

- 7) L'agent **Acheteur** retire la liste des vendeurs selon les offres auxquelles les négociations sont suspendues. S'il y a des changements dans la liste de vendeurs alors aller à 6. Sinon aller à 8.
- 8) S'il existe des propositions dans la liste qui vérifient les prérequis de l'agent **Acheteur**, alors il accepte la meilleure proposition de la liste.  
Sinon l'agent **Acheteur** rejette toutes les propositions.

Durant le déroulement du processus d'achat les étapes 6 et 7 se répètent plusieurs fois, jusqu'à ce que l'agent **Acheteur** sera satisfait.

## Contribution 3 : Expérimentation

### V.1. Introduction

Dans ce chapitre, nous présentons le prototype développé à base de l'approche proposée dans les chapitres précédents. Nous présentons aussi les outils de développement, utilisés dans l'implémentation de différents composants du prototype (agents, ontologies, modules et interfaces).

Dans ce qui suit nous expliquons et motivons la stratégie suivie dans la phase d'expérimentation, la nature des données utilisées et nous discutons les résultats obtenus en déterminant les points forts (avantages) et les points faibles (limites) de l'approche proposée dans cette thèse.

Ce chapitre représente la troisième contribution de la thèse qui est présentée dans [Said et al., 2012].

### V.2. Les outils et les plateformes utilisés

Comme un outil de développement, nous avons utilisé l'IDE NetBeans7.0 basé sur le langage Java. Nous avons utilisé JADE (version 4.1) comme plate-forme de développement des SMA, JADE est basé sur un langage de communication des agents appelé FIPA-ACL. L'éditeur Protégé-OWL (version 3.4.4) est utilisé pour la création des ontologies. L'API du moteur d'inférence JESS (Java Expert System Shell) [JESS] est utilisée pour l'exécution des règles SWRL définis dans les ontologies. Pour un bon affichage de résultats sous forme de diagramme de barres empilées au cours d'exécution, nous avons utilisé le framework JavaFX 2.2 [JavaFX].

#### V.2.1. Présentation de l'éditeur d'ontologie Protégé

Protégé [Protégé] est un outil utilisé pour créer des ontologies à plusieurs formats, le format le plus connu et le plus utilisé est OWL. On peut utiliser l'éditeur

Protégé OWL pour définir les concepts d'un domaine ainsi que les relations entre ces concepts, la définition se fait de manière visuelle, facile et souple. A l'aide de Protégé il est possible de définir des règles en langage SWRL (Semantic Web Rule Language) en utilisant le plugin SWRLTab.

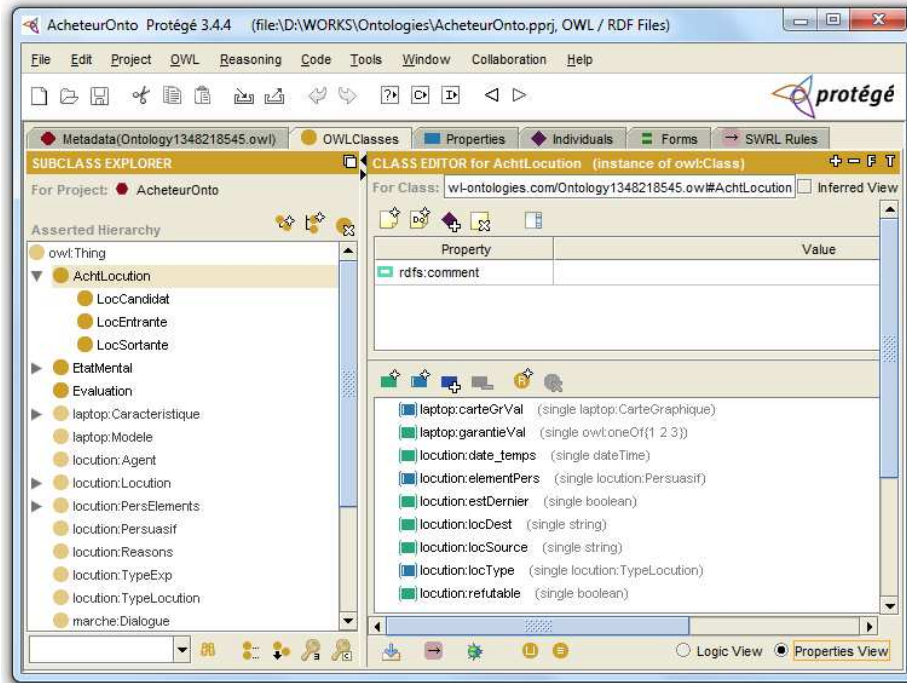


Figure V.1: L'éditeur d'ontologie Protégé version 3.4.4

## V.2.2. La plate-forme JADE

Le meilleur moyen pour construire un système multi-agents (SMA) est d'utiliser une plate-forme multi-agent. Une plate-forme multi-agents est un ensemble d'outils nécessaire à la construction et à la mise en service d'agents au sein d'un environnement spécifique. Ces outils peuvent servir également à l'analyse et au test du SMA ainsi créé. Ces outils peuvent être sous la forme d'environnement de programmation (API) et d'applications permettant d'aider le développeur. Nous allons présenter dans cette partie la plate-forme JADE [JADE].

### V.2.2.1. Description

JADE est un Framework entièrement implémentée en Java. Il est utilisé pour le développement des systèmes multi-agents et des applications conformes aux standards de FIPA [FIPA]. La plate-forme JADE peut être distribuée sur plusieurs machines



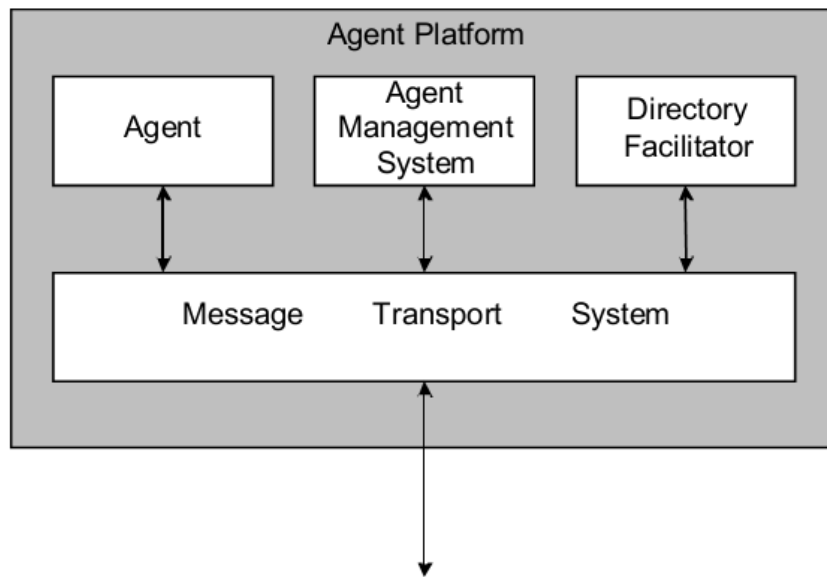
ayant des systèmes d'exploitation différents (Windows, Linux, ...). Cette plateforme possède trois modules principaux (nécessaires aux normes FIPA) :

- 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.

### V.2.2.2. Architecture logiciel de la plate-forme JADE

JADE reprend donc l'architecture « Agent Management Reference Model » proposé par [FIPA]. Les différents modules présentés dans la figure suivante sont présentés sous forme de services. Les services de base proposés sont le *Directory Facilitator* (DF) et *Agent Management System* (AMS). Il est possible de lui demander de tenir en plus le service de Message Transport Service (MTS) pour communiquer entre plusieurs plates-formes.

L'agent est l'acteur fondamental de la plate-forme JADE, un *Agent Identifier* (AID) identifie un agent de manière unique. Le DF est un composant qui fait office d'annuaire. C'est un service de « pages jaunes » qui permet de mettre en relation les agents avec leurs compétences. Un agent peut enregistrer ses compétences dans le DF ou interroger le DF pour connaître les compétences proposées par les autres agents. L'AMS est un autre composant important car il contrôle l'accès et l'utilisation de la plate-forme et maintient un répertoire contenant les adresses de transport des agents de la plate forme. Ce service est plus un service de type « pages blanches » qui effectuent la correspondance entre l'agent et l'AID.



**Figure V.2:** Architecture logiciel de la plate-forme JADE [JADE]

### V.2.2.3. Langage de communication de la plate-forme JADE (FIPA-ACL)

Le langage de Communication de la plate-forme JADE est FIPA-ACL (Agent Communication Language). La classe `ACLMessage` représente les messages qui peuvent être échangés par les agents. La communication de messages se fait en mode asynchrone. Lorsqu'un agent souhaite envoyer un message, il doit créer un nouvel objet `ACLMessage`, compléter ces champs avec des valeurs appropriées et enfin appeler la méthode `send()`. Lorsqu'un agent souhaite recevoir un message, il doit employer la méthode `receive()` ou la méthode `blockingReceive()`.

## V.3. Présentation du prototype

### V.3.1. Architecture globale

Cette architecture représente la structure du prototype implémenté qui se compose d'un ensemble de modules (Voir la figure V.3).

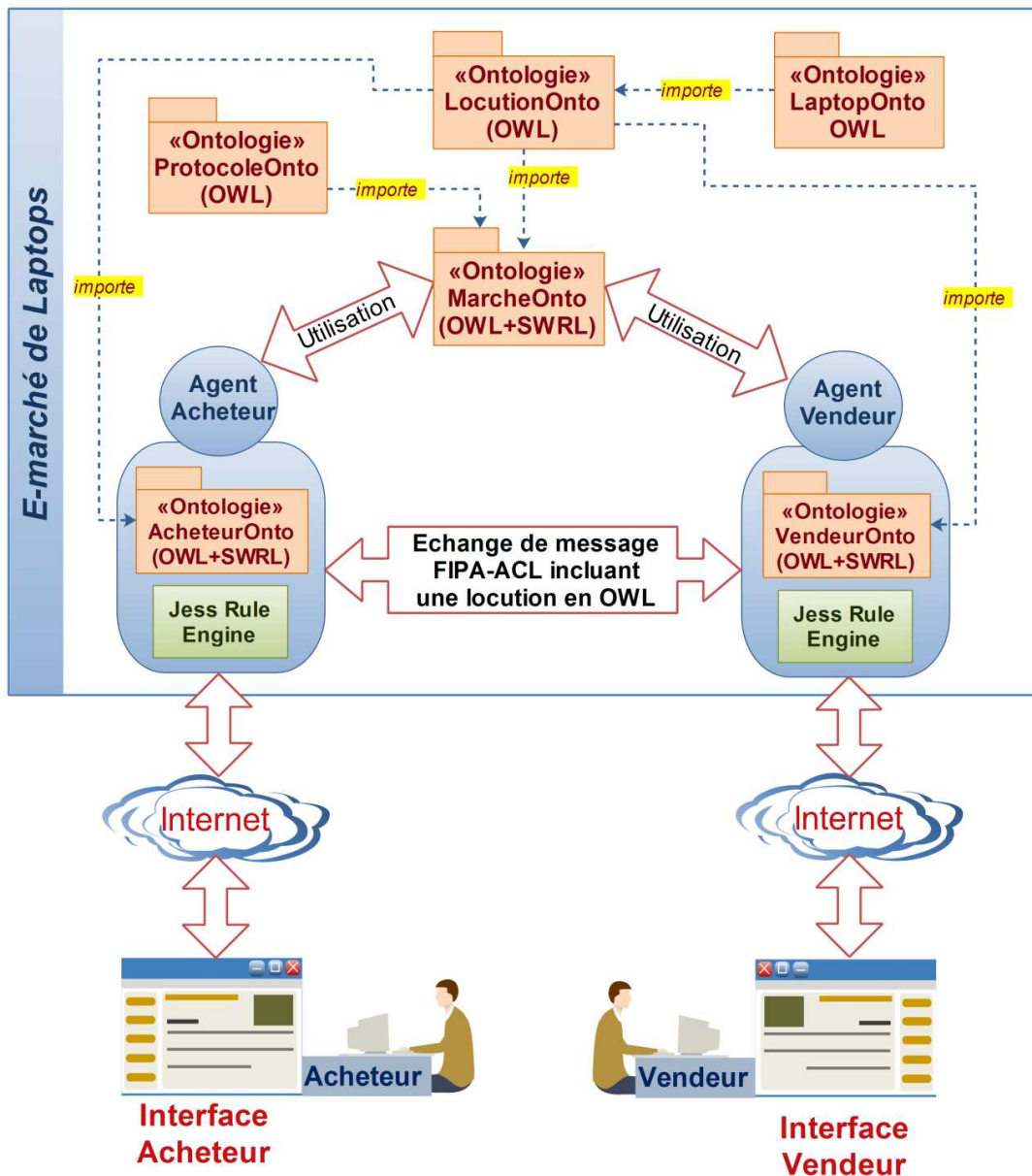


Figure V.3: L'architecteur globale du prototype

Le prototype présenté dans la figure précédente est constitué de trois types de composants : les interfaces utilisateur, les ontologies et les agents.

### V.3.2. Présentation des interfaces utilisateur

Dans cette section nous présentons quelques interfaces utilisateur du marché électronique pour les vendeurs et les acheteurs.

### V.3.2.1. L'interface d'accueil

L'interface d'accueil permet à l'utilisateur (vendeur / acheteur) d'authentifier (par la saisie du nom d'utilisateur et du mot de passe), ou d'inscrire dans le cas d'un nouveau utilisateur. Après l'authentification l'utilisateur peut accéder à l'interface correspondante (AcheteurUI / VendeurUI).

### V.3.2.2. L'interface du vendeur (VendeurUI)

L'interface du vendeur comprend quatre grands onglets (*Configuration, Voir les modèles et Résultats de ventes*):

- Onglet *Configuration* : permet à un vendeur de configurer son magasin électronique, par la spécification de l'importance des attributs du Laptop, la spécification de l'ordre de préférence pour les caractéristiques des Laptops et les locutions, la mise à jour des modèles existants et l'ajout d'un nouveau modèle de Laptop. Ces fonctionnalités peuvent être accédées à travers quatre sous-onglets :
  - Sous-onglet *Importance* : le concept importance des attributs est différent pour le vendeur que pour l'acheteur, car le gain d'un vendeur augmente quand le prix augmente. Donc les caractéristiques qui influent l'augmentation du prix plus vite sont les plus importants.
  - Sous-onglet *Préférences* : la même règle en ce qui concerne l'ordre de préférence sur chaque attribut. Par exemple le vendeur préfère une marque qui est disponible à son magasin et qui lui ramène le meilleur bénéfice. A travers cet onglet le vendeur peut spécifier l'ordre de préférence pour les attributs *Marque, Processeur et Ecran*, ainsi que pour les locutions persuasives. Les attributs qui restent sont ordonnés proportionnellement avec leurs valeurs.
  - Sous-onglet *Modèles* : permet au vendeur de mettre à jour les modèles exposés à son magasin électronique.
  - Sous-onglet *Nouveau modèle* : utilisé pour l'ajout d'un nouveau modèle au magasin (voir figure V.5).



Figure V.4: La fenêtre d'authentification

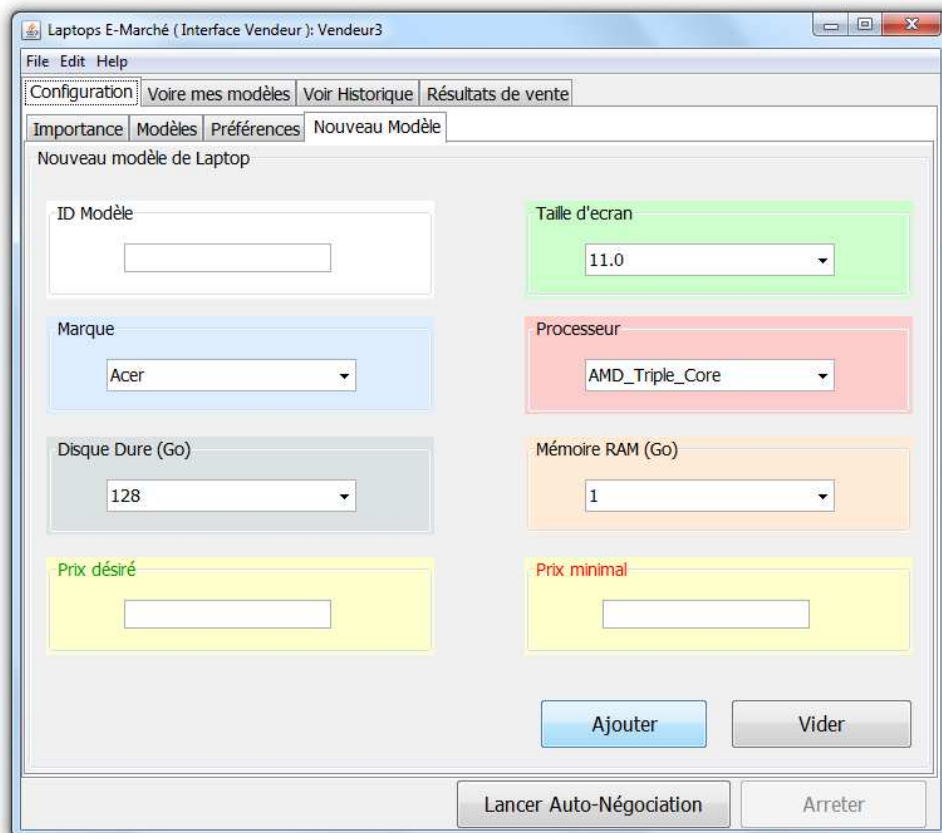


Figure V.5: L'interface du vendeur

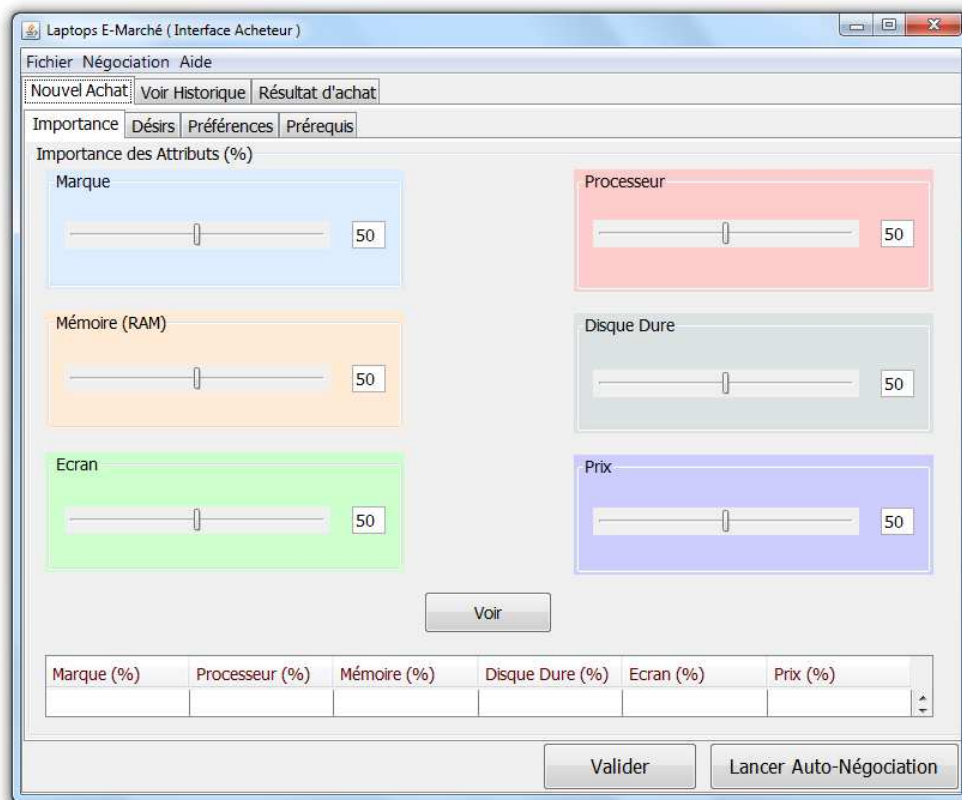
- Onglet *Voir mes modèles* : donne au vendeur une vue globale sur les modèles de Laptop exposés à son magasin électronique. Si le vendeur désire faire une modification sur un modèle le sous-onglet *Modèles* s'affiche automatiquement.
- Onglet *Voir Historique* : avec cet onglet le vendeur peut voir l'historique de toutes les négociations passées pour pouvoir améliorer sa stratégie de vente.
- Onglet *Résultats de vente* : à travers cet onglet le vendeur peut voir des statistiques sur les modèles vendues.

### V.3.2.3. L'interface de l'acheteur (AcheteurUI)

L'interface de l'acheteur comprend trois grands onglets (*Nouvel Achat*, *Voir Historique* et *Résultat d'achat*):

- Onglet *Nouvel Achat* : utilisé par l'acheteur pour fournir les connaissances nécessaires à qualifier l'agent *Acheteur* de faire une négociation avec les agents *Vendeur* pour l'achat d'un Laptop. Avant le lancement de l'auto-négociation l'acheteur doit visiter les quatre sous-onglets suivants :
  - Sous-onglet *Importance* : à travers duquel l'acheteur spécifie l'importance (poids) des attributs à négocier. Le poids des attributs est utilisé ensuite par l'agent Acheteur pour l'évaluation des propositions reçues durant la négociation.
  - Sous-onglet *Désirs* : utilisé par l'acheteur pour choisir, sans classement, l'ensemble (ou l'intervalle) de valeurs désirées pour chaque attribut. Par exemple l'acheteur peut spécifier une ou plusieurs valeurs désirées pour l'attribut *Marque*.
  - Sous-onglet *Préférences* : permet à l'acheteur de spécifier l'ordre de préférence pour les attributs *Marque*, *Processeur* et *Ecran*, ainsi que pour les locutions persuasives. Les attributs qui restent sont ordonnés automatiquement, où l'ordre de préférence des attributs *Mémoire* et *Disque Dur* est directement proportionnel avec leurs valeurs, par contre l'ordre de préférence de l'attribut *Prix* est inversement proportionnel avec ses valeurs.
  - Sous-onglet *Pré requis* : les pré requis sont les valeurs négociables (non réfutables) d'un attribut, donc l'acheteur peut spécifier pour chaque attribut l'ensemble (ou l'intervalle) de valeurs négociables. Une proposition ayant une valeur d'attribut qui n'appartient pas à des valeurs pré requises, sera refusée par l'agent Acheteur.

- Onglet *Voir Historique* : utilisé pour voir l'historique des négociations faites par l'agent Acheteur.
- Onglet *Résultat d'achat* : permet à l'acheteur de voir l'évaluation finale effectuée par l'agent Acheteur sur les dernières propositions de tous les agents Vendeur. Le résultat est affiché sous forme d'un diagramme de barres empilées et sous forme d'un tableau trié selon le résultat de l'évaluation.



**Figure V.6:** L'interface de l'acheteur

### V.3.3. Les ontologies

Le schéma de figure V.3 montre six différentes ontologies : *LaptopOnto*, *LocutionOnto*, *ProtocoleOnto* et *MarcheOnto* sont utilisées dans l'espace inter-agents, les ontologies *VendeurOnto* et *AcheteurOnto* sont utilisées respectivement à l'intérieur de l'agent Vendeur et l'agent Acheteur.

Les éléments de l'ontologie *LaptopOnto* sont importés et utilisés dans l'ontologie *LocutionOnto* qui est elle-même importé dans les ontologies : *MarcheOnto*,

*VendeurOnto* et *AcheteurOnto*. L'ontologie *ProtocoleOnto* est importée dans *MarcheOnto*. Durant le dialogue de négociation l'ontologie *MarcheOnto* est utilisée par les agents *Vendeur*, les agents *Acheteur* et par les interfaces utilisateur.

Les ontologies sont créées par de l'éditeur d'ontologie Protégé-OWL [Protégé], description UML de ces ontologies est présentée dans le chapitre précédent. La manipulation des ontologies en java se fait à l'aide de l'API Java fourni par [Protégé]. Nous présentons quelques exemples de la manipulation en Java des éléments d'une ontologie OWL:

### *Ouverture et lecture d'une ontologie*

```
String uri = "http://www.emarchelap.com/MarcheOnto.owl";
OWLModel owlModel = ProtegeOWL.createJenaOWLModelFromURI(uri);
```

### *Lecture d'une classe de l'ontologie*

```
OWLNamedClass exist = owlModel.getOWLNamedClass("Modeles_existants");
```

### *Lecture des individus d'une classe*

```
String cls = "Modeles_existants";
Collection inst= owlModel.getOWLNamedClass(cls).getInstances(false);
for (Iterator it = inst.iterator(); it.hasNext();) {
    OWLIndividual individual = (OWLIndividual) it.next();
    . . .
}
```

### *Lecture d'une propriété d'un individu*

```
OWLIndividual laptop = owlModel.getOWLIndividual("Dell_Inspiron_15R");
OWLDatatypeProperty property=
    owlModel.getOWLDatatypeProperty("ramVal");
int laptopRam = (Integer) laptop.getPropertyValue(property);
```



### *Enregistrement d'une ontologie*

```
FileWriter fWriter= new FileWriter("C:/Travail/MarcheOnto.owl");  
owlModel.getJenaModel().write(fWriter,"RDF/XML-ABBREV");  
w.close();
```

### **V.3.4. Les agents (Vendeur et Acheteur)**

Dans l'environnement de négociation il existe deux types d'agents, agent *Vendeur* et agent *Acheteur*. Un agent *Vendeur* négocie pour chaque vendeur sur le marché avec la prise en compte des connaissances remplies à travers l'interface *VendeurUI* et stockées dans l'ontologie *VendeurOnto*. Un agent *Acheteur* négocie pour chaque acheteur sur le marché avec la prise en compte des connaissances remplies à travers l'interface *AcheteurUI* et stockées dans l'ontologie *AcheteurOnto*. Pour chaque agent *Vendeur* une instance d'ontologie *VendeurOnto* est associée et pour chaque agent *Acheteur* une instance d'ontologie *AcheteurOnto* est associée.

Les règles de négociation par argumentation ne sont pas codées directement dans les agents, mais elles sont définies en langage SWRL à l'intérieur des ontologies *VendeurOnto* et *AcheteurOnto*. Avec cette solution les agents deviendront plus légers ce qui facilite la modification et la mise à jour des règles de négociation. Pour l'exécution d'une règle SWRL l'agent fait appel au moteur d'inférence JESS qui exécute la règle et insert les résultats directement à l'ontologie.

La négociation entre un agent *Vendeur* et un agent *Acheteur* se fait par l'échange de messages FIPA-ACL en respectant le protocole de négociation défini dans le chapitre 3 par la machine à états finis (Finite State Machine). La définition du protocole de négociation se fait au niveau des agents ainsi qu'au niveau de l'ontologie *ProtocoleOnto* qui sera importé par la suite dans l'ontologie *MarcheOnto*. Au niveau des agents *Vendeur* et *Acheteur* le protocole de négociation est implémenté sous forme d'un comportement d'agent de la classe *FSMBehaviour* comme suite :

### *Protocole de négociation de l'agent Vendeur*

```
protected void setup() {  
    // création d'un comportement FSM de l'agent Vendeur  
    FSMBehaviour agentV_beh= new FSMBehaviour();
```

```
// les états de FSM
agentV_beh.registerFirstState(new Attendre(), "Initial");
agentV_beh.registerState(new Reception(), "Recep");
agentV_beh.registerState(new Generer(), "S1");
agentV_beh.registerState(new Attendre(), "S2");
agentV_beh.registerState(new Attendre(), "S3");
agentV_beh.registerState(new Attendre(), "S4");
agentV_beh.registerState(new Generer(), "S5");
agentV_beh.registerLastState(new fin(), "Final");

// les transitions de FSM
agentV_beh.registerDefaultTransition("Initial", "Recep");
agentV_beh.registerDefaultTransition("S2", "Recep");
agentV_beh.registerDefaultTransition("S3", "Recep");
agentV_beh.registerDefaultTransition("S4", "Recep");
agentV_beh.registerTransition("S1", "S2", 2);
agentV_beh.registerTransition("S1", "S3", 3);
agentV_beh.registerTransition("S1", "S4", 4);
agentV_beh.registerTransition("Recep", "S5", 5);
agentV_beh.registerTransition("Recep", "S1", 1);
agentV_beh.registerTransition("Recep", "Final", 6);
agentV_beh.registerTransition("S5", "S4", 4);

// appliquer le comportement
addBehaviour(agentV_beh);
}
```

### *Protocole de négociation de l'agent Acheteur*

```
protected void setup(){
    // création d'un comportement FSM de l'agent Acheteur
    FSMBehaviour agentA_beh= new FSMBehaviour();

    // les états de FSM
    agentA_beh.registerFirstState(new Proposer(), "Initial");
    agentA_beh.registerState(new Reception(), "Recep");
    agentA_beh.registerState(new Attendre(), "S1");
    agentA_beh.registerState(new Generer(), "S2");
    agentA_beh.registerState(new Generer(), "S3");
    agentA_beh.registerState(new Generer(), "S4");
    agentA_beh.registerState(new Attendre(), "S5");
    agentA_beh.registerLastState(new fin(), "Final");

    // les transitions de FSM
```

```
agentA_beh.registerDefaultTransition("Initial", "S1");
agentA_beh.registerDefaultTransition("S1", "Recep");
agentA_beh.registerDefaultTransition("S5", "Recep");
agentA_beh.registerTransition("Recep", "S2", 2);
agentA_beh.registerTransition("Recep", "S3", 3);
agentA_beh.registerTransition("Recep", "S4", 4);
agentA_beh.registerTransition("S2", "S1", 1);
agentA_beh.registerTransition("S2", "Final", 6);
//agentA_beh.registerDefaultTransition("S3", "Final");
agentA_beh.registerTransition("S3", "Final", 6);
agentA_beh.registerTransition("S4", "S1", 1);
agentA_beh.registerTransition("S4", "S5", 5);
agentA_beh.registerTransition("S4", "Final", 6);
    // appliquer le comportement
addBehaviour(agentA_beh);
}
```

Les constructeurs *Proposer()*, *Reception()*, *Attendre()*, *Generer()* et *Fin()* sont définis dans des classes héritées de la classe *OneShotBehaviour* qui représente un comportement simple, par exemple les classes *Attendre* et *Reception* sont implémentées de la manière suivante :

```
private class Attendre extends OneShotBehaviour{

    public void action() {

        block(); // l'agent est bloqué en attente d'un message

    }

}
```

```
private class Reception extends OneShotBehaviour{

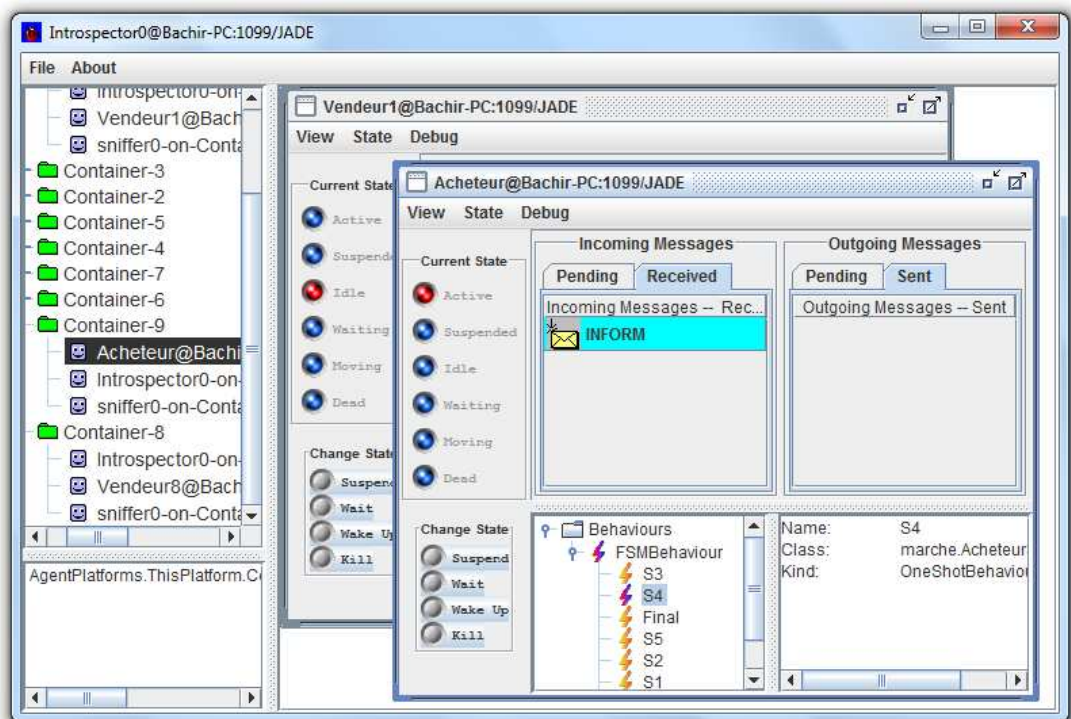
    public void action() {
        msgIn = receive();
        locId = msgIn.getConversationId();
        locContenu = msgIn.getContent();

        if(locId!=null){
```

```

        if(locId.equals("Rej")) etat=2;
        else if(locId.equals("Acc")) etat=3;
        else etat=4;
        } // le nouveau état est défini selon la locution reçue
    }
    public int onEnd(){
        return etat;
    }
}

```



**Figure V.7:** FSMBehaviour de l'agent Acheteur sur l'inspecteur de JADE

Le protocole de négociation est défini aussi dans l'ontologie *ProtocoleOnto*, cette ontologie peut être utilisée dans les règles SWRL pour la génération des locutions candidats convenables à un certain état de négociation.

Comme nous avons mentionné précédemment les locutions échangées sont définies en langage OWL-DL par l'ontologie *LocutionOnto* et sont incluses dans le contenu des messages FIPA-ACL. Les codes suivants sont utilisés pour l'envoi et la réception d'une locution via un message FIPA-ACL :

### *Envoie d'une locution (OWL-DL) via un message*

```
. . .
String uri = " http://www.emarchelap.com/LocutionOnto.owl";
OWLModel locOWL = ProtegeOWL.createJenaOWLModelFromURI(uri);
chargerLocOut(locOWL);
StringWriter writer = new StringWriter();
locOWL.getJenaModel().write(writer, FileUtils.langXMLAbbrev);
String tmp = writer.getBuffer().toString();
ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
msg.setContent(tmp);
msg.addReceiver(new AID( agentRec , AID.ISLOCALNAME) );
send(msg);
. . .
```

### *Réception et interprétation d'une locution (OWL) d'un message*

```
. . .
ACLMessage msgIn = receive();
if (msgIn!=null) {
JenaOWLModel locOWL= ProtegeOWL.createJenaOWLModelFromInputStream(
    newByteArrayInputStream(msgIn.getContent().getBytes("UTF-8")));
. . .
```

Les règle de négociation définit en SWRL peuvent être exécuté par un agent à l'aide du moteur d'inférence JESS [JESS]. Le code suivant est inclus dans l'agent pour l'exécution d'une règle SWRL :

### *Exécution d'une règle SWRL par un agent*

```
. . .
String uri = "http://www.emarchelap.com/AchetueuOnto.owl";
try {
OWLModel owlModel = ProtegeOWL.createJenaOWLModelFromURI(uri);
SWRLRuleEngineBridge bridge
    =BridgeFactory.createBridge("SWRLJessBridge", owlModel);
SWRLFactory factory = new SWRLFactory(owlModel);
factory.disableAll();
factory.getImp("Rule-4").enable();
```

```
bridge.reset();
bridge.importSWRLRulesAndOWLKnowledge(); // OWL+SWRL -> Jess
bridge.run(); // Execute Jess
bridge.writeInferredKnowledge2OWL(); // Jess -> OWL
}
catch(Exception e){
    e.printStackTrace();
}
. . .
```

### V.4. Exemple d'expérimentation

#### V.4.1. Description du scénario

Dans cette expérimentation, cent cinq modèles de Laptop sont définis dans l'ontologie *MarcheOnto*. Ces modèles sont relevés de cinq boutiques électroniques en France [Amazone1; Darty; Pixmania; Surcouf; Fnac]. Dans cette expérimentation, huit agents Vendeur sont lancés.

L'achat d'un Laptop s'effectue en suivant le processus d'achat décrit dans la section (IV.6). Dans la première étape de ce processus l'acheteur entre les connaissances nécessaires sur ses croyances, ses désirs et ses pré requis à travers l'interface *AcheteurUI*. Les figures suivantes présentent les connaissances nécessaires pour la négociation à base d'agent définis par l'acheteur:

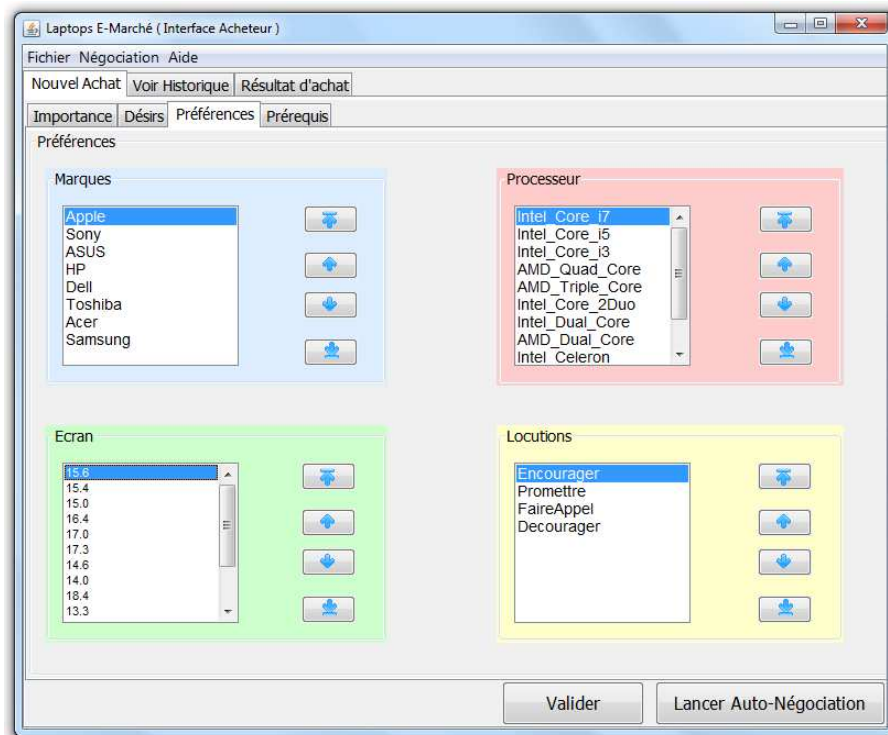


Figure V.8: Spécification des ordres de préférence de l'acheteur

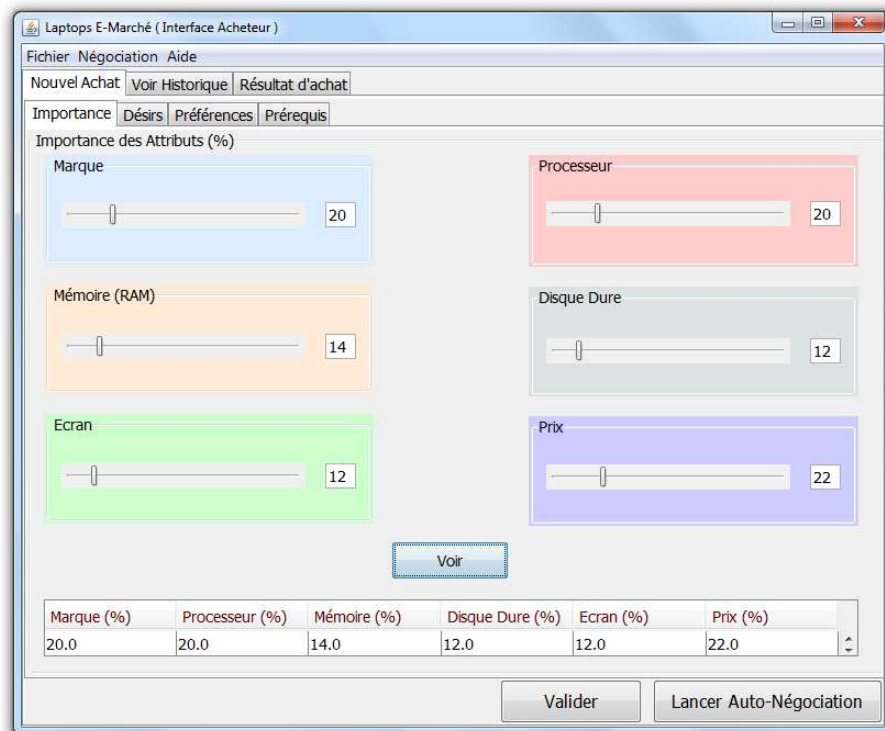


Figure V.9: Spécification des poids des attributs pour l'acheteur

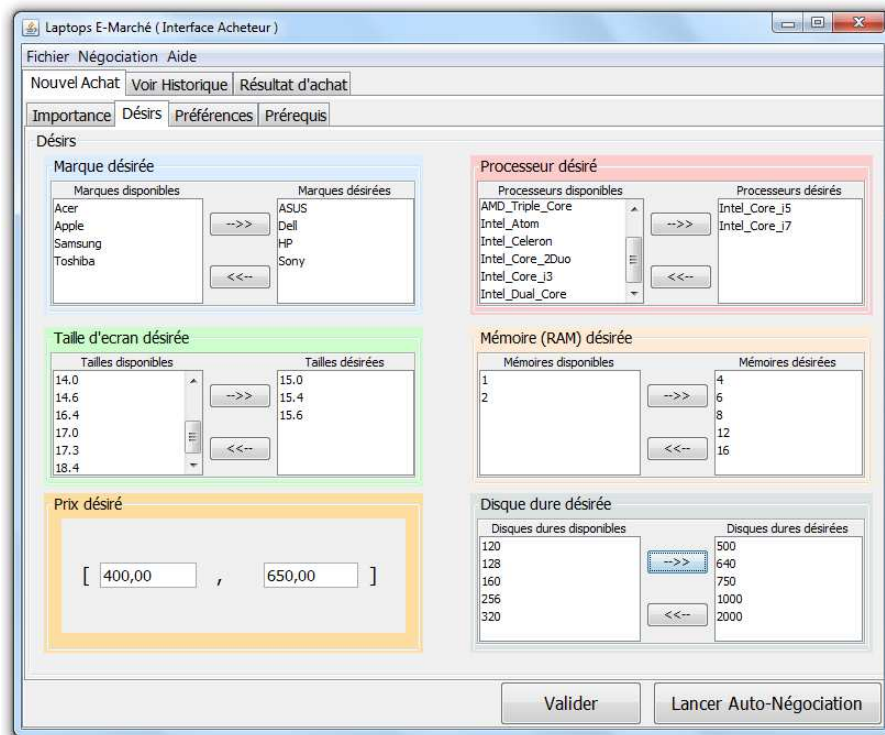


Figure V.10: Spécification des désirs de l'acheteur

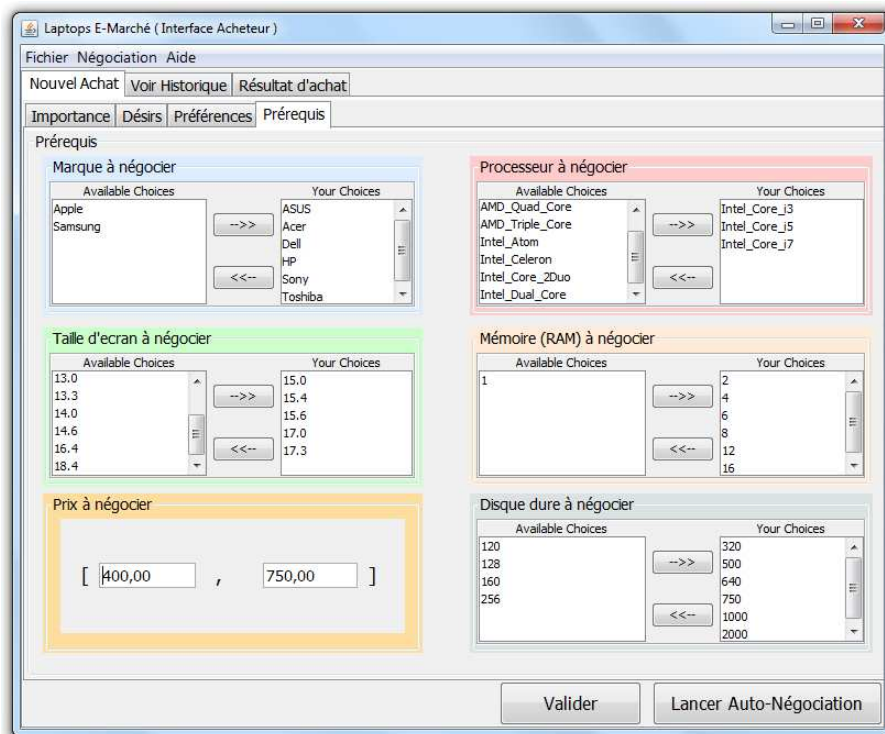


Figure V.11: Spécification des pré requis de l'acheteur



Le tableau suivant décrit le résumé de toutes les connaissances entrées par l'acheteur dans la première étape du processus d'achat :

	<b>Croyances sur l'ordre de préférence</b>	<b>Croyances sur les poids des attributs</b>	<b>Désirs</b>	<b>Pré requis</b>
Marque	Apple > Sony > Asus > HP > Dell > Toshiba > Acer > Samsung	0.2	HP / Asus / Dell / Sony	Sony / HP / Asus / Dell / Toshiba / Acer
Processeur	Intel > AMD	0.2	≥ Intel Core i5.	≥ Intel Core i3.
Mémoire (Ram)	Directement proportionnelle	0.14	4 Go ou plus.	2 Go ou plus.
Disque Dure	Directement proportionnelle	0.12	500 Go ou plus.	320 Go ou plus.
Ecran	15" > 17" > 14" > 13" > 12" > 11" > 10"	0.12	15" / 15.4" / 15.6"	15" / 15.4" / 15.6" / 17" / 17.3"
Prix	Inversement proportionnelle	0.22	≥ 400 € ≤ 650 €	≥ 400 € ≤ 750 €
Locutions	Encourager > FaireAppel > Promettre > Décourager	/	Toutes	Toutes

**Tableau V.1.** Croyances, désirs et pré requis sur les attributs négociés

L'ordre de popularité supposé sur les caractéristiques des Laptops dans le marché électronique est présenté dans le tableau suivant :

	<b>Marque</b>	<b>Processeur</b>	<b>Ecran</b>
L'ordre de popularité supposé	HP > Acer > Dell > Sony > Asus > Apple > Toshiba > Samsung	Intel Core i5 > Intel Core i7 > Intel Core i3 > Intel Core 2 duo > ...	15" > 14" > 17" > 13" > 12" > 11" > 10"

**Tableau V.2.** Ordre de popularité supposé sur les caractéristiques des Laptops

Après cette étape l'agent Acheteur commence la négociation comme il est décrit dans la section IV.6 par le processus d'achat. Durant le dialogue de négociation l'échange de locutions se fait séparément à chaque tour avec tous les agents *Vendeur* concernés. La figure suivante représente la capture de l'échange de messages entre l'agent *Acheteur* et l'agent *Vendeur1* dans le premier tour du dialogue de négociation par *sniffer* de JADE :

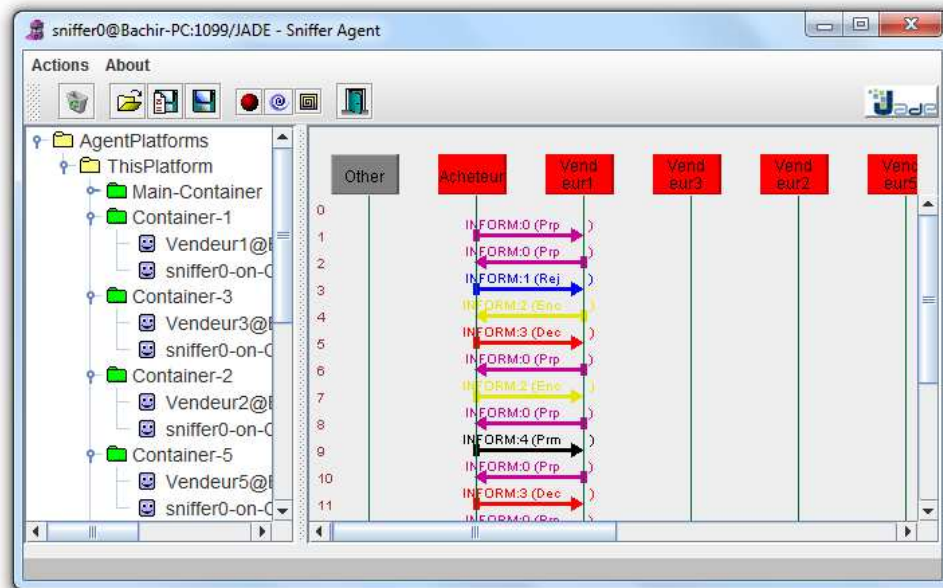


Figure V.12: Capture de l'échange de messages par Sniffer de JADE

#### V.4.2. Résultats

Les résultats de négociation s'affichent sur l'onglet *Résultat d'achat* de l'interface *AcheteurUI* sous forme d'un histogramme empilé et d'un tableau trié selon le facteur de préférence calculé pour chaque agent *Vendeur* (figure V.13). Le tableau montre les propositions finales sur les caractéristiques proposées par les huit agents *Vendeur*. Toutes les propositions sont évaluées et triées par l'agent *Acheteur* selon ces croyances sur les préférences et les poids des attributs.

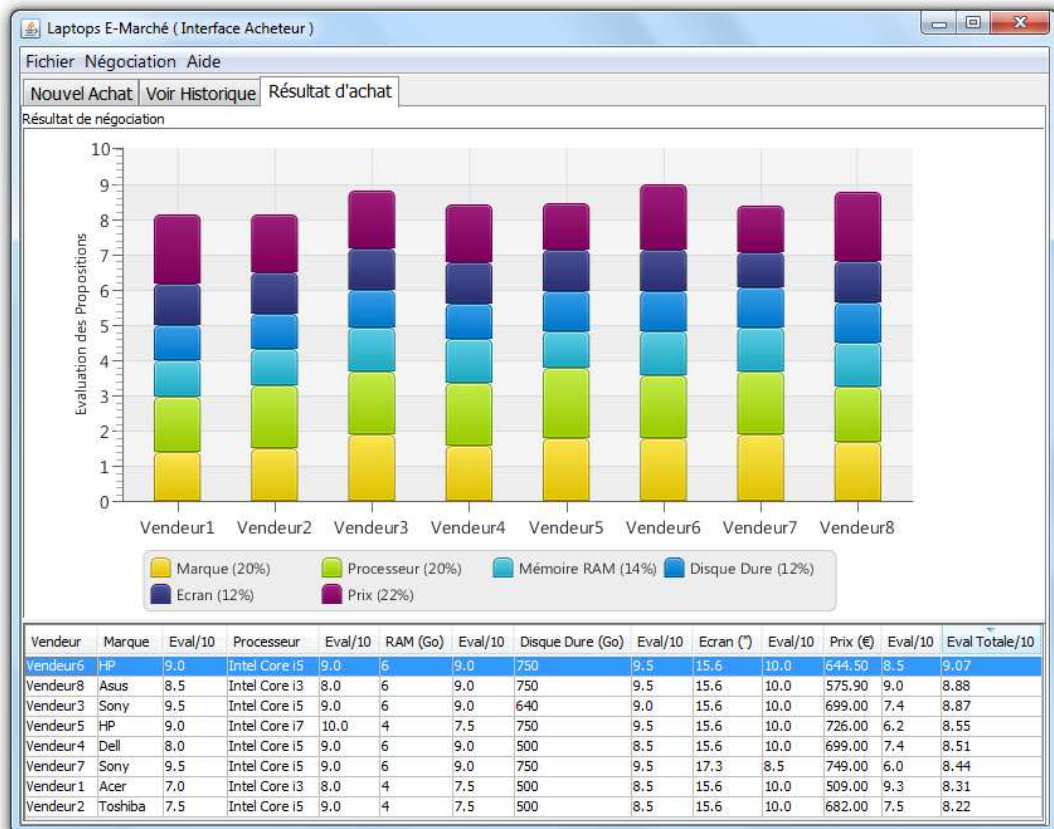


Figure V.13. Evaluation des propositions des agents Vendeur

### V.4.3. Discussion

Les propositions finales données par tous les agents Vendeur (Tableau A.4) sont les meilleures propositions obtenues à partir du dialogue de négociation. Pendant le dialogue de négociation, plusieurs locutions (persuasive et non persuasive) sont échangées entre l'agent Acheteur et les agents Vendeur. Par exemple :

- Encourager (Vendeur6, Acheteur, Proposition, Persuasive, Qualificateur),  
où Persuasive = (Mon, Prix=644, Exceptionnel, Appui=699), Qualificateur=1.
- Encourager (Vendeur1, Acheteur, Proposition, Persuasive, Qualificateur),  
Où Persuasive = (Mon, Marque, Populaire, Appui=2ème), Qualificateur =0.85.

Après chaque tour de négociation un agent (Vendeur ou Acheteur) peut changer ses croyances et ses désirs sur les attributs négociés, s'il est influencé par l'agent opposant. Mais, s'il n'y a aucune influence, l'agent Vendeur et l'agent Acheteur essaient de trouver le modèle de Laptop qui satisfait leurs désirs. Par exemple, si le prix d'un

Laptop proposé est cher, l'agent Acheteur demande à l'agent Vendeur de réduire le prix, si l'agent Vendeur ne peut pas faire une remise, il lui propose un autre modèle de Laptop moins cher.

Après l'arrêt du dialogue de négociation, la liste de propositions finales sera triée. L'agent Acheteur évalue les propositions des agents Vendeur selon ses croyances sur les poids des attributs négociés et l'ordre de préférence. L'évaluation des propositions finales est montrée comme histogramme empilé dans la figure V.10. Chaque barre de l'histogramme représente l'évaluation de la proposition d'un agent Vendeur qui est obtenue par le calcul des facteurs de préférence. Un facteur de préférence  $fp \in [0, 10]$ , est calculé pour chaque attribut négocié. Alors, le facteur total de préférence pour chaque agent Vendeur est obtenu comme suit :

$$fp(\text{proposition}) = \sum_{i=1}^n fp(\text{Attribut}_i) \times Poids(\text{Attribut}_i).$$

La figure V.10 montre que l'agent Vendeur 6 soit le gagnant puisqu'il a le plus grand facteur de préférence (9.05/10).

Cet exemple expérimental prouve que l'utilisation de l'argumentation dans le mécanisme de négociation permet à un agent d'influencer les agents opposants pour changer leurs croyances et leurs désirs et permet à un agent d'être influencé par les agents opposants. Dans l'exemple précédent, on peut remarquer le changement des désirs initiaux dans les propositions finales, pour chaque agent Vendeur. Par exemple, l'acheteur n'a pas introduit la marque Acer dans ses désirs initiaux (Tableau V.1), mais l'agent Vendeur1 est arrivé de le persuader qu'Acer est parmi les marques les plus populaires sur le marché (Tableau V.2).

### V.5. Conclusion

Dans ce chapitre, nous avons présenté les outils utilisés dans l'implémentation du prototype. L'architecture du prototype a montré les différents composants développés, ainsi que la relation entre ces composants. Ensuite, le scénario d'exécution de l'application est expliqué étape par étape en utilisant un exemple de vendeur et d'acheteur. Enfin, des listings importants de code Java pour la manipulation des ontologies, la communication des agents et l'exécution des règles.

L'expérimentation faite sur le prototype nous a permis de valider l'approche proposée dans cette étude. On peut énumérer quelques points forts de l'implémentation présenté dans ce chapitre :

- Utilisation d'un langage d'ontologie (OWL) récent et très expressif.
- Utilisation d'un langage de règle (SWRL) puissant et portable pour exprimer les mécanismes de négociation, au lieu de les coder à l'intérieur des agents. Ce qui a permis aux agents d'être plus légers.
- Utilisation du langage OWL-DL pour la représentation textuelle des locutions aux contenus des messages FIPA-ACL, ce qui a permis l'échange des locutions sans perdre la structure du contenu.

## Conclusion et perspectives

La modélisation d'une négociation à base d'agent pour un domaine spécifique tel que le commerce électronique était un défi pour plusieurs chercheurs. Beaucoup de difficultés apparaissent quand on désire simuler le comportement humain lors d'une négociation commerciale. Car la communication entre les humains se caractérise par le sens et la sémantique et la situation mentale et morale de l'humain s'influence à l'effet d'une interprétation spécifique très compliquée de tous ce qui est reçu.

Donc, pour arriver à résoudre les problèmes existants à la modélisation de la négociation à base d'agent dans le domaine du commerce électronique, nous avons essayé d'étudier les travaux connexes et d'employer les techniques les plus pertinentes. La négociation à base d'argumentation est supposée la plus convenable au problème de négociation commerciale. De plus, les langages du web sémantique (OWL et SWRL) sont choisis pour la représentation du langage de négociation ainsi que l'expression des différents mécanismes et tactiques. Ce choix nous a permis de mieux exprimer les locutions échangées entre les agents lors d'un dialogue de négociation.

Dans l'approche proposée, le modèle théorique de l'argument de Toulmin et la technique psychologique d'analyse de champ de force sont adaptés, en inspirant tous leurs points forts. La prise en compte de l'aspect philosophique et psychologique de la négociation humaine (par l'utilisation des techniques théoriques) nous a permis de modéliser une négociation plus réaliste.

Le processus d'achat proposé a destiné d'impliquer plusieurs agents vendeurs dans une négociation avec le même agent acheteur, ce qui ajoute la gestion de concurrence à l'environnement d'e-commerce.

Malgré les avantages énumérés sur l'approche proposée, la modélisation d'une négociation à base d'agent reste à notre avis un peu loin de la négociation humaine. La négociation humaine est un art, qui nécessite l'emploi et la synthèse de plusieurs techniques et astuces. En plus, elle nécessite une analyse d'un grand nombre de paramètres avec plusieurs valeurs.

Dans les travaux futurs nous visons à utiliser d'autres techniques pour améliorer l'approche proposée, tel que la logique floue (fuzzy logic). La logique floue peut être utilisée pour la représentation des paramètres ayant des valeurs incertaines ou inexacts (qualificateur, influence, force, ...etc.). Même dans les langages du web sémantique, il y a la possibilité de définir et d'utiliser des concepts en basant sur la logique floue (FuzzyOWL).

Il est possible aussi, de modéliser une négociation agent-humain. Dans cette négociation, un agent vendeur négocie avec des acheteurs humains selon un apprentissage initial fait par le vendeur. Cette approche est plus proche d'être mise en service, car la décision de l'acheteur est plus critique et il est difficile qu'elle soit prise par un agent logiciel. Pour faciliter la négociation des acheteurs humains, il est souhaitable de concevoir un agent assistant. L'agent assistant effectue des tâches spécifiques tels que la recherche, le calcul et l'organisation des résultats partiels.

## Bibliographie

Allen, J.F., Perrault, C.R., 1980. Analyzing intention in dialogues. *Artificial Intelligence* 15(3).

Amazone: <http://www.amazon.com>. visité: 08 Mars 2012.

Amazonel: [http://www.amazon.fr/s/ref=sr\\_hi\\_1?rh=n%3A340858031&bbn=340858031&ie=UTF8&qid=1341224778](http://www.amazon.fr/s/ref=sr_hi_1?rh=n%3A340858031&bbn=340858031&ie=UTF8&qid=1341224778). visité: 08 Mars 2012.

Amgoud, L., Cayrol, C., 2002. A reasoning model based on the production of acceptable arguments *Annals of Mathematics and Artificial Intelligence* 34, 197–215.

Amgoud, L., Dimopoulos, Y., Moraitis, P., 2007. A Unified and General Framework for Argumentation-based Negotiation, AAMAS.

Amgoud, L., Parsons, S., 2001. Agent dialogues with conflicting preferences, in: Meyer, J.J., Tambe, M. (Eds.), *8th International Workshop on Agent Theories Architectures and Languages ATAL*, Seattle, pp. 190–205.

Amgoud, L., Parsons, S., Maudet, N., 2000. Arguments, dialogue, and negotiation, in: Horn, W. (Ed.), *European Conference on Artificial Intelligence*. IOS Press, Amsterdam, Netherlands, pp. 338-342.

Auction Bot : <http://ai.eecs.umich.edu/people/wellman/pubs/agents98www.html>.

Babin, G., Maamar, Z., Chaib-draa, B., 1997. Metadatabase meets distributed AI, *First Int. Workshop on Cooperative Information Agents (CIA-97)*.

Bartolini, C., Preist, C., Jennings, N.R., 2004. A Software Framework for Automated Negotiation, *SELMAS'2004*. Springer Verlag, LNCS 3390, pp. 213-235.

Bench-Capon, T.J.M., 2002. Value-based argumentation frameworks, *Proceedings of Non Monotonic Reasoning*, pp. 444-453.



- Bench-Capon, T.J.M., 2003. Persuasion in Practical Argument Using Value-based Argumentation Frameworks. *Logic Computation* 13(3).
- Bentahar, J., Labban, J., 2009. An Argumentation-Driven Model for Flexible and Efficient Persuasive Negotiation. *Group Decision and Negotiation*.
- Berners-Lee, T., Hendler, J., Lassila, O., 2001. The Semantic Web, *Scientific American*, pp. 35-43.
- Bondarenko, A., Dung, P., Kowalski, R., Toni, F., 1997. An abstract argumentation-theoretic approach to default reasoning. *Artificial Intelligence* 93(1-2), 63–101.
- Bondarenko, A., Toni, F., Kowalski, R., 1993. An assumption-based framework for nonmonotonic reasoning, *Proc. LPRNR'93*. MIT Press, pp. 171–189.
- Bourron, T., 2001. Application des systèmes multi-agents dans les télécommunications : États de l'art, enjeux et perspectives. *Agent et systèmes multiagents*.
- Brockmans, S., Volz, R., Eberhart, A., Löffler, P., 2004. Visual Modeling of OWL DL Ontologies Using UML, in: al., S.A.M.e. (Ed.), *ISWC 2004*. LNCS 3298, pp. 198–213.
- Chaib-draa, B., 1995. Industrial applications of distributed AI. *Communications of the ACM* 38(11), 49-53.
- Chavez, A., Maes, P., 1996. 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*, pp. 75-90.
- Cohen, P.R., Levesque, H.J., 1990. Rational interaction as the basis for communication. *Intentions in Communication*, 221-256.
- Darty: [http://www.darty.com/nav/achat/informatique/ordinateur\\_portable/index.html](http://www.darty.com/nav/achat/informatique/ordinateur_portable/index.html), visité: 08 Mars 2012.
- Dash, R.K., Parkes, D.C., Jennings, N.R., 2003. Computational Mechanism Design: A Call to Arms. *IEEE Intelligent Systems* 18(6), 40-47.

Dung, P.M., 1993. On the acceptability of arguments and its fundamental role in non-monotonic reasoning and logic programming, Proceedings of the 13th International Joint Conference on Artificial Intelligence, IJCAI'93, pp. 852–857.

Dung, P.M., 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence 77, 321–357.

E-Bay: <http://www.ebay.com/> .

E-CAP: <http://e-cap.sourceforge.net/>.

Eemeren, F.H.v., Grootendorst, R.F., Henkemans, F.S., 1996. Fundamentals of Argumentation Theory: A Handbook of Historical Backgrounds and Contemporary Applications, Mahwah NJ, USA.

Eichmann, D., 1994. Ethical Web Agents, Proceedings of the 2nd International WWW Conference, Chicago.

Elvang, M., Fox, J., Krause, P., 1993. Dialectic reasoning with inconsistent information, Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence, pp. 114–121.

Emele, C.D., Norman, T.J., Parsons, S., 2011. Argumentation strategies for plan resourcing, in: Yolum, Tumer, Stone, Sonenberg (Eds.), Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AA-MAS 2011), Taipei, Taiwan.

EShopBooster: <http://www.eshopbooster.co.uk> .

Fabregues, A., Sierra, C., 2011. DipGame: a challenging negotiation testbed. Engineering Applications of Artificial Intelligence (EAAI).

Fagin, R., Halpern, I.Y., Moses, Y., Vardi, M.Y., 1995. Reasoning about Knowledge. The MIT Press: Cambridge.

Faratin, P., Sierra, C., Jennings, N.R., 1998. Negotiation Decision Functions for Autonomous Agents. *International Journal of Robotics and Autonomous Systems* 24(3-4), 159-182.

Ferber, J., 1995. *Les systèmes multi-agents, vers une intelligence collective*. InterEditions.

FIPA, 2001. *Communicative Act Library Specification*., Foundation for Intelligent Physical Agents.

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

Fnac: <http://www.fnac.com/Ordinateur-portable/shi48967/w-4#bl=MMmic>, visité: 08 Mars 2012.

Fox, J., Parsons, S., 1998. Arguing about beliefs and actions, in: Hunter, A., Parsons, S. (Eds.), *Applications of Uncertainty Formalisms*. Springer Verlag, Berlin, Germany, pp. 266–302.

Franklin, S., Graesser, A., 1997. Is it an agent, or just a program?: A taxonomy for autonomous agents, in: Mueller, J.P., Wooldridge, M., Jennings, N.R. (Eds.), *Intelligent Agents III: Theories, Architectures, and Languages*. Springer-Verlag, Heidelberg, Germany, pp. 21-35.

Freeley, A.J., Steinberg, D.L., 2008. *Argumentation and Debate: Critical Thinking for Reasoned Decision Making*, Cengage.

Freeman, J.B., 1991. *Dialectics and the Macrostructure of Arguments*. Walter de Gruyter.

Galliers, J.R., 1988. *A Theoretical Framework for Computer Models of Cooperative Dialogue, Acknowledging Multi-Agent Conflict*. PhD thesis, Open University, UK.

Glaser, N., 1996. Contributing to Knowledge Modeling in a Multi-Agent Framework : the CoMoMAS Approach. PhD thesis, Université Henri Poincaré, Nancy 1, France.

Govier, T., 2010. A Practical Study of Argument, Wadsworth.

Guttman, R., Moukas, A., Maes, P., 1998. Agent-mediated Electronic Commerce: A survey. Knowledge Engineering Review 13(2), 147-160.

Guttman, R.H., Maes, P., 1998. Agent-Mediated Integrative Negotiation for Retail Electronic Commerce, in: Noriega, P., Sierra, C. (Eds.), In Selected Papers from the First International Workshop on Agent Mediated Electronic Trading on Agent Mediated Electronic Commerce (AMET '98). Springer-Verlag, London, UK, pp. 70-90.

Hage, J.C., 1997. Reasoning With Rules: An Essay on Legal Reasoning. Springer.

Halpern, J.Y., 1995. Reasoning about knowledge: A survey, in: Gabbay, D.M., Hogger, C.J., Robinson, J.A. (Eds.), Handbook of Logic in Artificial Intelligence and Logic Programming. Oxford University Press, Oxford, England, pp. 1-34.

Heras, S., Botti, V., Julian, V., 2011. Case-based Argumentation Framework: Dialogue Protocol, Technical Report. Polytechnic university of Valencia.

Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosz, B., Dean, M., 2004. SWRL: A Semantic Web Rule Language Combining OWL and RuleML, W3C Member Submission.

Hsairi, L., Ghédira, K., Alimi, A.M., BenAbdelhafid, A., 2008. R2-IBN: Argumentation Based Negotiation Framework for the Extended Enterprise, in: Corchado, E., Abraham, A., Pedrycz, W. (Eds.), HAIS 2008, pp. 533–542.

Iglesias, C.A., Garijo, M., González, J.C., Velasco, J.R., 1997. Analysis and design of multi-agent systems using mas-commonkads, AAI'97 Workshop on Agent Theories, Architectures and Languages, Providence, RI.

JADE: <http://jade.csel.it/>.

JavaFX : <http://www.oracle.com/technetwork/java/javafx/overview/index.html>

Jennings, N.R., 2000. Agent methodology for software engineering. *Communication of ACM*.

Jennings, N.R., Faratin, P., Lomuscio, A.R., Parsons, S., Sierra, C., Wooldridge, M., 2001. Automated negotiation: prospects, methods and challenges. *International Journal of Group Decision and Negotiation* 10(2), 199–215.

Jennings, N.R., Parsons, S., Noriega, P., Sierra, C., 1998. On argumentation-based negotiation, *Proceedings of the International Workshop on Multi-Agent Systems, Boston, USA*, pp. 1-7.

Jennings, N.R., Wooldridge, M., Sycara, K., 1998. A roadmap of agent research and development. *Int Journal of Autonomous Agents and Multi-Agent Systems* 1(1), 7- 38.

Jennings, N.R., Wooldridge, M.J., 1998. *Agent Technology: Foundations, applications, and Markets*. Springer-Verlag: , Heidelberg, Germany.

JESS. Java Expert System Shell: <http://www.jessrules.com/jess/academic/>

Kakas, A., Moraitis, P., 2003. Argumentation based decision making for autonomous agents, *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2003), Melbourne, Australia*, pp. 883–890.

Kaplan, F., 1998. A new approach to class formation in multi-agent simulation of language evolution, *Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS-2), Paris, France*, pp. 158-165.

Kephart, J.O., 2002. Software agents and the route to the information economy, *Proc. National Academy of Science*, pp. 7207-7213.

Klügl, F., 2009. Multi-agent simulation model design strategies, *MAS & S Workshop at MALLOW, CEUR Workshop Proceedings, Turin, Italy*.

Kobsa, A., 1989. *User Models in Dialog Systems*. Springer-Verlag, Heidelberg, Germany.

Kok, E.M., Meyer, J.J.C., Prakken, H., Vreeswijk, G.A.W., 2011. A Formal Argumentation Framework for Deliberation Dialogues, in: McBurney, P., Rahwan, I., Parsons, S. (Eds.), ArgMAS 2010, pp. 31-48.

Kraus, S., Sycara, K., Evenchik, A., 1998. Reaching agreements through argumentation: A logical model and implementation. *Artificial Intelligence* 104(1-2), 1-69.

Labrou, Y., Finin, T., Peng, Y., 1999. Agent communication languages: The current landscape. *Intelligent Systems* 14(2), 45-52.

Lakemeyer, G., Levesque, H., 1999. Query evaluation and progression in AOL knowledge bases, *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, Stockholm, Sweden, pp. 124-131.

Lespérance, Y., 1991. A Formal Theory of Indexical Knowledge and Action, PhD thesis, Toronto.

Lespérance, Y., Levesque, H.J., Lin, F., Marcu, D., Reiter, R., Scherl, R.B., 1994. A logical approach to high-level programming- A progress report, in: Kuipers, B. (Ed.), *Control of the Physical World by Intelligent Systems, Papers from the 1994 AAAI fall Symposium*, New Orleans, LA, pp. 78-85.

Lewin, K., 1947. Frontiers in group dynamics: concept, method and reality in social science; social equilibria and social change. *Human Relations* 1 (1), 5-41.

Lomuscio, A., Wooldridge, M., Jennings, N.R., 2001. A Classification Scheme for Negotiation in Electronic Commerce, *Agent-Mediated Electronic Commerce: A European Perspective*. Springer-Verlag, pp. 19-33.

Magnet : <http://airvl.cs.umn.edu/magnet/>.

Martinez, D.C., Garcia, A.J., Simari, G.R., 2008. An Abstract Argumentation Framework with Varied-Strength Attacks, *Proceedings, Eleventh International Conference on Principles of Knowledge Representation and Reasoning*.

- Maudet, N., Chaib-draa, B., 2003. Commitment-based and dialogue-game based protocols—news trends in agent communication language. *Knowledge Engineering Review* 17(2), 157–179.
- Mayfield, J., Labrou, Y., Finin, T., 1996. Evaluating kqml as an agent communication language, in: Wooldridge, M.J., Muller, J.P., Tambe, M. (Eds.), *Intelligent Agents II, Lecture Notes in Computer Science*. Springer Verlag, Berlin, Germany, pp. 347-360.
- McBurney, P., Eijk, R.M.v., Parsons, S., Amgoud, L., 2003. A dialogue-game protocol for agent purchase negotiations. *Journal of Autonomous Agents and Multi-Agent Systems* 7(3), 235–273.
- McBurney, P., Parsons, S., 2009. Dialogue Games for Agent Argumentation, in: Rahwan, I., Simari, G.R. (Eds.), *Argumentation in Artificial Intelligence*.
- McGuinness, D.L., Harmelen, F.V., 2004. OWL web ontology language: Overview. *W3C Recommendation*.
- Meany, J., Shuster, K., 2002. *Art, Argument, and Advocacy*. IDEA.
- Munoz, A., Botia, J.A., 2008. ASBO: Argumentation system based on ontologies, in: Klusch, M., Pechoucek, M., Polleres, A. (Eds.), *CIA 2008, LNCS (LNAI)*. Springer, Heidelberg, pp. 191-205.
- Negroponte, A., 1995. *Being Digital*. Hodder and Stoughton.
- Neumann, J.V., Morgenstern, O., 1944. *Theory of Games and Economic Behaviour*. Princeton University Press, Princeton, NJ.
- Noriega, P., 1998. *Agent mediated auctions: The Fishmarket Metaphor*, PhD thesis. University of Barcelona.
- Noriega, P., Sierra, C., 1999. *Agent Mediated Electronic Commerce*, LNAI. Springer, Berlin.
- Nwana, H.S., Ndumu, D.T., 1999. Agents of change in future communication systems. *Applied AI Journal* 13(1).

Parsons, S., Sierra, C., Jennings, N.R., 1998. Agents that reason and negotiate by arguing. *Journal of Logic and Computation* 8(3), 261-292.

Patterson, J.L., 2005. Using Force Field Analysis in Negotiation Planning, 90th Annual International Supply Management Conference.

Pixmania: <http://www.pixmania.com/ft/ft/home.html>, visité: 08 Mars 2012.

Prakken, H., Sartor, G., 2001. The role of logic in computational models of legal argument: a critical survey, in: Kakas, A., Sadri, F. (Eds.), *Computational Logic: From Logic Programming into the Future*. Springer Verlag, Berlin, Germany, pp. 342-349.

Price-Minister : <http://www.priceminister.com/> .

Protégé: <http://protege.stanford.edu/> .

Pruitt, D.G., 1981. *Negotiation Behaviour*. Academic Press.

Rahwan, I., Ramchurn, S.D., Jennings, N.R., McBurney, P., Parsons, S., Sonenberg, L., 2003. Argumentation-based negotiation. *The Knowledge Engineering Review* 18, 343-375.

Raiffa, H., 1982. *The Art and Science of Negotiation*. Harvard University Press.

Ramchurn, S.D., Jennings, N.R., Sierra, C., 2003. Persuasive negotiation for autonomous agents: a rhetorical approach, in: Reed, C., Grasso, F., Carenini, G. (Eds.), *Proceedings of the IJCAI Workshop on Computational Models of Natural Argument*. AAAI Press, pp. 9-17.

RoboCup : <http://www.robocup.org/>.

Rosenschein, J.S., Zlotkin, G., 1994. Designing Conventions for Automated Negotiation, *AI Magazine (AAAI)*.

Sadri, F., Toni, F., Torroni, P., 2001. Logic agents, dialogues and negotiation: an abductive approach, in: Stathis, K., Schroeder, M. (Eds.), *Proceedings of the AISB 2001 Symposium on Information Agents for E-Commerce*.



- Saha, S., Sen, S., 2005. A Bayes Net Approach to Argumentation Based Negotiation, in: al., I.R.e. (Ed.), ArgMAS 2004. LNAI 3366, pp. 208-222.
- Said, B., Kazar, O., Benharkat, A.N., 2011. An Argumentation-Based Negotiation Approach in Electronic Marketplace based on Semantic Web. *Journal of Artificial Intelligence* 4 (3), 170-186.
- Said, B., Kazar, O., Benharkat, A.N., 2012. Semantic Web Based Formalization of Argumentation Mechanism in E-Commerce Negotiation. *Journal of Intelligent Computing* 3(1), 23-39.
- Schiemann, B., Schreiber, U., 2006. OWL DL as a FIPA ACL content language, Proceedings of the 18th European Summer School of Language, Logic and Information, Workshop on Formal Ontology for Communicating Agents (FOCA), University of Malaga, Spain, 31 July – 11 Aug 2006., pp. 1-84.
- Sierra, C., Faratin, P., Jennings, N.R., 1997. A service-oriented negotiation model between autonomous agents, Proceedings of the 8th European Workshop on Modeling Autonomous Agents in a Multi-Agent World (MAAMAW-1997), Ronneby, Sweden, pp. 17-35.
- Sierra, C., Jennings, N.R., Noriega, P., Parsons, S., 1997. A framework for argumentation-based negotiation, 4 th International Workshop on Agent Theories, Architectures and Languages, Rhode Island, USA, pp. 167-182.
- Smith, R.G., 1980. The Contract Net Protocol: high-level communication and control in a distributed problem solver. *IEEE Transactions on computers* C-29(12), 1104-1113.
- Surcouf: <http://www.surcouf.com/F-10459-ordinateur-portable>, visité : 08 Mars 2012.
- Swinton, L., Kurt Lewin's Force Field Analysis: Decision Making Made Easy, Management for the Rest of Us.
- Sycara, K., 1990. Persuasive argumentation in negotiation. *Theory Decision Journal*, Springer Netherlands 28, 203-242.

Sycara, K., 1992. The Persuader, in: Shapiro, D. (Ed.), The Encyclopedia of Artificial Intelligence. John Wiley & Sons, New York.

Toulmin, S., 1958. The Uses of Argument, eBook updated 2003 ed. Cambridge University Press, Cambridge.

Treuil, J.P., Drogoul, A., Zucker, J.D., 2008. Modélisation et Simulation à base d'agents. Dunod.

W3C. World Wide Web Consortium : <http://www.w3.org/>.

Willmott, S., Vreeswijk, G., Chesnevar, C., South, M., McGinis, J., Modgil, S., Rahwan, I., Reed, C., Simari, G., 2006. Towards an argument interchange format for multiagent systems, in: Maudet, N., Parsons, S., Rahwan, I. (Eds.), Proceedings of the 3rd International Workshop on Argumentation in Multi-Agent Systems (ArgMAS), Hakodate, Japan.

Wooldridge, M., 2002. An introduction to multi agent systems. John Wiley & Sons, Inc.

Wurman, P.R., Wellman, M.P., Walsh, W.E., 1998. The Michigan Internet AuctionBot: A Configurable Auction Server for Human and Software Agents, Proceedings of the Second International Conference on Autonomous Agents (Agents-98), Minneapolis, MN, USA, May 1998, pp. 301-308.