

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mohamed Khider Biskra
Faculté des Sciences et des Sciences de l'ingénieur
Département d'Informatique

N°d'ordre :.....

Série :.....

Mémoire

Présenté en vue de l'obtention du diplôme de Magistère en Informatique

Option: **Intelligence Artificielle**

:

Une approche formelle pour la négociation automatique entre agents (application e-commerce)

Par :

M^{elle} Sahli Siham

Soutenu le : / /2010

Devant le jury :

Mr. Nouredine Djeddi,	Pr. Université de Biskra	Président
Mr. Kazar Okba,	M.C. Université de Biskra	Rapporteur
Mr. Boufaida Mahmoud	Pr. Université de Constantine	Examineur
Mr. Zarour Nacereddine,	Pr. Université de Constantine	Examineur

Remerciement

Je commence, avant toute chose, par remercier ALLAH LE TOUT PUISSANT pour son aide et de m'avoir doté de beaucoup de force physique et morale et du courage nécessaire pour l'élaboration de ce modeste travail.

Ensuite, je tiens à remercier particulièrement mon encadreur, Dr. *Kazar Okba*, pour ses judicieux conseils, ses orientations claires et précises et surtout sa patience et sa confiance.

Finalement, je remercie les membres du jury. Tout d'abord Pr. *Nouredine Djeddi*, Pr. *Boufaïda Mahmoud* et Pr. *Zarour Nacereddine* de m'avoir fait l'honneur d'accepter d'examiner ce modeste manuscrit.

ملخص

تمثل حاليا المفاوضات الآلية مجالا نشطا للبحث حيث يحاول الباحثون في هذا المجال تصميم برامج بإمكانها محاكاة سلوك الأشخاص المفاوضين مما يسمح للمستعمل أن يطلق مفاوضات آلية عبر الويب من دون أن ينتقل و يلتقي بالطرف الآخر . و بظهور الأنظمة متعددة الأعوان أو الوكلاء شهدت المفاوضات الآلية تطورا ملحوظا أين يعوض الشخص المفاوض بوكيل اصطناعي ذكي ومستقل بذاته مما يعطي انطبعا بان المفاوضات عبارة عن أعوان اصطناعيين . من جهتنا فإننا نقترح نموذج مولد مكون من أعوان مفاوضين وبروتوكول جديد للتفاوض متعدد الخاصيات ومصادقة هذا الأخير بنموذج شكلي رياضي باستعمال شبكات بتري الملونة.

كلمات مفاتيح:

المفاوضات الآلية وبروتوكولات المفاوضات، النظم متعددة الأعوان، النظم الرياضية الشكلية، شبكات بتري الملونة.

Résumé

La négociation automatique présente un domaine de recherche très actif où les chercheurs essayent de simuler le comportement des négociateurs humains par des logiciels artificiels qui permettent au négociateur humain de lancer une négociation automatique sur le web sans déplacer et rencontrer l'autre côté de négociation. Et avec l'apparition des systèmes multi-agent la négociation automatique a évolué d'une manière remarquable où le négociateur humain se remplace par un agent artificiel qui se caractérise par l'intelligence et l'autonomie qui donne l'impression que les négociateurs sont des humains et non pas des agents artificiels.

Dans notre travail, nous allons proposer une architecture générique composé d'agents négociateurs et d'établir un nouveau protocole de négociation multi-attributs avec la validation de ce dernier par une modélisation formelle en utilisant les réseaux de Petri colorés.

Mots clés :

La négociation automatique et protocole de négociation, systèmes multi-agent, système formel, RdPC.

Table des matières

Université Mohamed Khider Biskra	1
Département d'Informatique	1
Table des matières	4
Liste des figures	9
Liste des tableaux	10
Chapitre 1	14
L'interaction entre Agents	14
1. Introduction	14
2. L'interaction entre les agents	14
<i>2.1. C'est quoi l'interaction entre les agents?</i>	<i>15</i>
Pour illustrer la vision beaucoup plus claire, nous ajoutons qu'il y a une différence entre communication et interaction ; la communication permet aux agents de s'envoyer des messages et de les comprendre. L'interaction va plus loin car elle permet des conversations entre agents structurées sur la communication (échange de messages) [4].	20
3. La Communication	20
3.1 Définitions de la communication	21
3.2. Actes de langage	22
3.3 Langages de communication inter-agents	23
3.3.1 Langage de communication KQML	24
3.3.2. Langage de communication FIPA-ACL	25
5. Conclusion	29
Chapitre 2	31
La négociation entre Agents	31
1. Introduction	31
2. Définitions et formes de négociation automatique	31
2.2. Les formes de négociation automatique	33

3.3.3. Le protocole Kasbah.....	43
4. Conclusion	44
1. Introduction	46
2. Les approches de spécification de négociation.....	46
❖ Structuration.....	55
❖ Communication	55

La logique temporelle est utilisée lors de la validation du protocole d'interaction par le biais de la vérification de modèles. La vérification de modèle commence par générer un graphe des états accessibles pour le protocole considéré. Le concepteur définit ensuite la propriété à vérifier par l'intermédiaire d'une formule en logique temporelle. La vérification consiste alors à s'assurer que le graphe dispose de la propriété. 60

3. Synthèse 60

Nous avons essayé le long de ce chapitre de présenter les différents formalismes existants qui nous permettent de spécifier et décrire les interactions (négociation) dans un système multi-agents, notons que dans notre modélisation nous devons utiliser un formalisme formel. De ce fait dans cette synthèse nous allons focaliser notre effort beaucoup plus sur les formalismes formels :..... 60

Cette synthèse se présente par une discussion des différents avantages et inconvénients de chaque formalisme. Absolument cette synthèse nous amène à découvrir qui est le formalisme le plus adéquat dans notre cas. Commençons par les Modèles de spécification formelles pour les SMA qui se présentent par les AEF et les RdP : 60

↳ Comparons les AEF au reste des formalismes, ils présentent le moyen le plus simple pour décrire les protocoles d'interaction (négociation) car ils sont capables de mémoriser l'état de l'agent ce qui permet de décrire facilement son comportement. Le deuxième avantage des AEF est qu'ils facilitent la tâche des concepteurs du protocole d'interaction car ils possèdent une représentation graphique qui donne aux concepteurs une vue globale et évidente du protocole. Passons au troisième avantage des AEF qui concerne la validation du protocole. Nous pouvons noter que l'aspect de validation des protocoles présente le point le plus fort qui caractérise les AEF par rapport aux autres formalismes car la majorité des algorithmes de validation nécessitent d'abord une traduction du protocole dans un formalisme donné vers les AEF avant validation. Malgré tous les avantages que nous avons cité pour les

AEF, ils présentent aussi des inconvénients surtout pour la modélisation de l'évolution d'un système multi-agents où le comportement de ce système s'avère plus complexes que les autres. D'une part nous avons dit que parmi les avantages des AEF est qu'ils permettent de décrire un protocole d'interaction de façon simple où chaque état de l'automate correspond à une étape de la conversation, cet avantage conduit à un grand problème ou inconvénient dont le nombre d'état est combinatoire avec le nombre d'agent impliqués dans l'interaction et puisque aussi ce nombre d'états d'automate doit être fini malgré qu'il peut être énorme, la capacité de calcul des AEF est limitée. Alors les AEF ne peuvent donc se souvenir d'une suite d'événements arbitrairement longue. d'autre part, les AEF ne peuvent décrire que des processus séquentiels, de ce fait, tous les calculs, toutes les actions parallèles, la synchronisation leur sont pratiquement interdits ; sachons que le parallélisme et la synchronisation sont des caractéristiques importantes dans un protocole de négociation, donc ce formalisme (AEF) n'est pas adapté lorsqu'on veut représenter un système d'interaction (négociation) malgré tous ses avantages décrits précédemment..... 60

↳ Généralement les RdP possèdent presque les mêmes avantages des AEF et permettent aussi d'éviter les inconvénients de ces derniers. Au niveau de la conception du protocole : les RdP comme les AEF possèdent une représentation graphique qui facilite la tâche de conception. A l'encontre des AEF les RdP possèdent la notion de modularité par exemple dans les RdP hiérarchique où un RdP est constitué d'un ensemble des RdP. Les critères de synchronisation et parallélismes qui sont absents dans les AEF existent dans les RdP où nous trouvons que ces derniers ont été conçus initialement afin de permettre la synchronisation et la communication des processus concurrents. Maintenant au niveau de la validation de la conception de protocole : les RdP peuvent être facilement validés car il existe une conversion possible de protocole vers les AEF, de plus il existe un ensemble de propriétés qui peuvent être directement validés sur les RdP, comme la propriété qui vérifie qu'un marquage donné est accessible à partir de l'état initial. 61

Pour le deuxième formalisme qui est la logique, les notions de réutilisabilité, modularité et synchronisation n'existent pas. La validation est présente par l'intermédiaire de vérification de modèle. La conception n'est pas facile avec la logique temporelle car il n'existe pas de représentation graphique du protocole et la conception nécessite de travailler avec des concepts abstraits, ajoutons aussi qu'il n'existe pas d'outils pour la conception mais des outils pour la validation, tous ces inconvénients rend la spécification avec la logique une tâche difficile. 62

Passons au troisième formalisme qui est les langages formels ; ils présentent un moyen formel pour spécifier des comportements des systèmes distribués et des SMA mais, est-ce qu'ils présentent le bon moyen ou non ! 62

↳ Pour le langage Z, il a l'avantage de l'existence des notions de réutilisabilité et modularité qui facilitent la tâche de conception puisqu'il est possible de décomposer le protocole en un ensemble de modules (composants) qu'il peut les spécifier, valider et vérifier formellement à condition que ces composants sont pris isolément. Cependant, le langage Z ne possède ni une représentation graphique ni d'outils pour la conception et la validation des protocoles, et en plus de ça, il n'offre pas un modèle de communication et de synchronisation ce qui constitue une des faiblesses majeure, dans la modélisation des protocoles. 62

↳ Pour le langage SDL il se distingue par une représentation graphique qui facilite la conception des protocoles et qui présente aussi une raison à son développement dans l'industrie car il facilite la lecture même par des personnes non spécialistes du formalisme. SDL aussi permet la réutilisabilité et il possède aussi des outils pour la conception et la validation de protocole. Mais, malgré tout ça, dans SDL il n'y a pas de notion de synchronisation ce qui élimine l'aspect des rendez-vous entre agents. 62

↳ Malgré que Estelle est basé sur la notion de module qui permet au concepteur du protocole de réutiliser et d'avoir une architecture modulaire, il l'oblige de concevoir les protocoles d'une manière textuelle car il ne possède pas d'une représentation graphique mais seulement d'une représentation textuelle, cet inconvénient dégrade énormément les capacités de ce langage et c'est sans doute une des raisons pour lesquelles ce formalisme a très peu de notoriété dans le monde industriel, il est essentiellement utilisé dans le milieu universitaire sachons aussi que ce langage ne dispose pas de mécanisme de synchronisation. 62

↳ LOTOS est un langage qui garantit les notions de modularité et réutilisabilité car il y a la possibilité de décomposer un processus en sous-processus, aussi le mode de coopération des processus est obtenu à l'aide d'un jeu d'opérateurs : parallélisme, synchronisation, choix, séquence et préemption le modèle de communication entre le système et son environnement et entre les processus d'un même système est synchrone pour LOTOS. Il existe aussi des outils et des algorithmes pour la validation des protocoles. Cependant, la représentation textuelle de LOTOS rend la conception du protocole difficile. Alors toutes les caractéristiques générales de LOTOS incluent la puissance d'expression des éléments de services, de protocole et des

interfaces, mais sa notation abstraite, difficile à comprendre, et le style algébrique ne sont pas appréciés dans le monde industriel.....	63
4. Conclusion.....	63
Chapitre 4	64
La conception du protocole de négociation.....	64
1. Introduction	64
3.3. Architecture d'agent de négociation	75
5. Conclusion	84
Chapitre 5	86
Étude de cas et implémentation.....	86
3.1. <i>Le premier groupe</i>	91
<i>Classification des applications agents du premier groupe :</i>	91
3.2. <i>Le deuxième groupe</i>	92
4.1. Choix du langage de programmation JAVA	92
Conclusion générale et perspectives	102
Bibliographie	

Liste des figures

Département d'Informatique	1
Table des matières	4
Liste des figures	9
Liste des tableaux	13
Chapitre 1	17
L'interaction entre Agents	17
Chapitre 2	34
La négociation entre Agents.....	34
La logique temporelle est utilisée lors de la validation du protocole d'interaction par le biais de la vérification de modèles. La vérification de modèle commence par générer un graphe des états accessibles pour le protocole considéré. Le concepteur définit ensuite la propriété à vérifier par l'intermédiaire d'une formule en logique temporelle. La vérification consiste alors à s'assurer que le graphe dispose de la propriété.	63
3. Synthèse	63

Nous avons essayé le long de ce chapitre de présenter les différents formalismes existants qui nous permettent de spécifier et décrire les interactions (négociation) dans un système multi-agents, notons que dans notre modélisation nous devons utiliser un formalisme formel. De ce fait dans cette synthèse nous allons focaliser notre effort beaucoup plus sur les formalismes formels :..... 63

Cette synthèse se présente par une discussion des différents avantages et inconvénients de chaque formalisme. Absolument cette synthèse nous amène à découvrir qui est le formalisme le plus adéquat dans notre cas. Commençons par les Modèles de spécification formelles pour les SMA qui se présentent par les AEF et les Rdp : 63

↳ Comparons les AEF au reste des formalismes, ils présentent le moyen le plus simple pour décrire les protocoles d'interaction (négociation) car ils sont capables de mémoriser l'état de l'agent ce qui permet de décrire facilement son comportement. Le deuxième avantage des AEF est qu'ils facilitent la tâche des concepteurs du protocole d'interaction car ils possèdent une représentation graphique qui donne aux concepteurs une vue globale et évidente du protocole. Passons au troisième avantage des AEF qui concerne la validation du protocole. Nous pouvons noter que l'aspect de validation des protocoles présente le point le plus fort qui caractérise les AEF par rapport aux autres formalismes car la majorité des algorithmes de validation nécessitent d'abord une traduction du protocole dans un formalisme donné vers les AEF avant validation. Malgré tous les avantages que nous avons cité pour les AEF, ils présentent aussi des inconvénients surtout pour la modélisation de l'évolution d'un système multi-agents où le comportement de ce système s'avère plus complexes que les autres. D'une part nous avons dit que parmi les avantages des AEF est qu'ils permettent de décrire un protocole d'interaction de façon simple où chaque état de l'automate correspond à une étape de la conversation, cet avantage conduit à un grand problème ou inconvénient dont le nombre d'état est combinatoire avec le nombre d'agent impliqués dans l'interaction et puisque aussi ce nombre d'états d'automate doit être fini malgré qu'il peut être énorme, la capacité de calcul des AEF est limitée. Alors les AEF ne peuvent donc se souvenir d'une suite d'événements arbitrairement longue. d'autre part, les AEF ne peuvent décrire que des processus séquentiels, de ce fait, tous les calculs, toutes les actions parallèles, la synchronisation leur sont pratiquement interdits ; sachons que le parallélisme et la synchronisation sont des caractéristiques importantes dans un protocole de négociation, donc ce formalisme (AEF) n'est pas adapté lorsqu'on veut représenter un système d'interaction (négociation) malgré tous ses avantages décrits précédemment..... 63

↳ Généralement les Rdp possèdent presque les mêmes avantages des AEF et permettent aussi d'éviter les inconvénients de ces derniers. Au niveau de la conception du protocole : les Rdp comme les AEF possèdent une représentation graphique qui facilite la tâche de conception. A l'encontre des AEF les Rdp possèdent la notion de modularité par exemple dans les Rdp hiérarchique où un Rdp est constitué d'un ensemble des Rdp. Les critères de synchronisation et parallélismes qui sont absents dans les AEF existent dans les Rdp où nous trouvons que ces derniers ont été conçus initialement afin de permettre la synchronisation et la communication des processus concurrents. Maintenant au niveau de la validation de la conception de protocole : les Rdp peuvent être facilement validés car il existe une conversion possible de protocole vers les AEF, de plus il existe un ensemble de propriétés qui peuvent être directement validés sur les Rdp, comme la propriété qui vérifie qu'un marquage donné est accessible à partir de l'état initial. 64

Pour le deuxième formalisme qui est la logique, les notions de réutilisabilité, modularité et synchronisation n'existent pas. La validation est présente par l'intermédiaire de vérification de modèle. La conception n'est pas facile avec la logique temporelle car il n'existe pas de représentation graphique du protocole et la conception nécessite de travailler avec des concepts abstraits, ajoutons aussi qu'il n'existe pas d'outils pour la conception mais des outils pour la validation, tous ces inconvénients rend la spécification avec la logique une tâche difficile. 65

Passons au troisième formalisme qui est les langages formels ; ils présentent un moyen formel pour spécifier des comportements des systèmes distribués et des SMA mais, est ce qu'ils présentent le bon moyen ou non ! 65

↳ Pour le langage Z, il a l'avantage de l'existence des notions de réutilisabilité et modularité qui facilitent la tâche de conception puisqu'il est possible de décomposer le protocole en un ensemble de modules (composants) qu'il peut les spécifier, valider et vérifier formellement à condition que ces composants sont pris isolément. Cependant, le langage Z ne possède ni une représentation graphique ni d'outils pour la conception et la validation des protocoles, et en plus de ça, il n'offre pas un modèle de communication et de synchronisation ce qui constitue une des faiblesses majeure, dans la modélisation des protocoles. 65

↳ Pour le langage SDL il se distingue par une représentation graphique qui facilite la conception des protocoles et qui présente aussi une raison à son développement dans l'industrie car il facilite la lecture même par des personnes non spécialistes du formalisme.

SDL aussi permet la réutilisabilité et il possède aussi des outils pour la conception et la validation de protocole. Mais, malgré tous ça, dans SDL il n'y a pas de notion de synchronisation ce qui élimine l'aspect des rendez vous entre agent. 65

↳ Malgré que Estelle est basé sur la notion de module qui permet au concepteur du protocole de réutiliser et d'avoir une architecture modulaire, il l'oblige de concevoir les protocoles d'une manière textuelle car il ne possède pas d'une représentation graphique mais seulement d'une représentation textuelle, cet inconvénient dégrade énormément les capacités de ce langage et c'est sans doute une des raison pour lesquelles ce formalisme a très peu de notoriété dans le monde industriel, il est essentiellement utilisé dans le milieu universitaire sachons aussi que ce langage ne dispose pas de mécanisme de synchronisation. 65

↳ LOTOS est un langage qui garantie les notions de modularité et réutilisabilité car il y a la possibilité de décomposer un processus en sous processus, aussi le mode de coopération des processus est obtenu a l'aide d'un jeu d'opérateurs : parallélisme, synchronisation, choix, séquence et préemption le modèle de communication entre le système et son environnement et entre les processus d'un même système est synchrone pour LOTOS. Il existe aussi des outils et des algorithmes pour la validation des protocoles. Cependant, la représentation textuelle de LOTOS rend la conception du protocole difficile. Alors toutes les caractéristiques générales de LOTOS incluent la puissance d'expression des éléments de services, de protocole et des interfaces, mais sa notation abstraite, difficile à comprendre, et le style algébrique ne sont pas appréciés dans le monde industriel..... 66

4. Conclusion..... 66

Chapitre 4 67

La conception du protocole de négociation..... 67

3.3. Architecture d'agent de négociation 78

Chapitre 5 89

Étude de cas et implémentation..... 89

Conclusion générale et perspectives 105

Liste des tableaux

Chapitre 1 : L'interaction entre Agents

Table 1.1 : Les différents types de situation d'interaction05

Chapitre 4 : La conception du protocole de négociation

Table 4.1 : Comparatif des formalismes par rapport aux critères.....53

Table 4.2 : Liste des performatives de communication FIPA ACL.....60

Introduction générale

Négocier est un art du quotidien multi facettes : négociation sur le prix du bien, négociations sur le temps de travail, négociations du régime des retraites, négociations des conditions d'embauche Que ce soit à l'échelle de l'individu ou de l'organisation, dans les domaines du social, du politique ou bien du commercial, la négociation est partout et à tous les niveaux. La négociation se retrouve à tous les moments de la vie que l'on parle de vie personnelle ou professionnelle.

L'automatisation de la négociation est la tâche la plus difficile jusqu'à nos jours. Elle fait l'objet de multiples recherches et théories avec un manque remarquable de pratique. L'automatisation commence de la forme la plus simple comme les enchères jusqu'à la plus complexe comme la négociation à base d'argumentation.

Grâce au progrès des technologies de l'information et des systèmes multi-agents (SMA), nous trouvons que la plus part des travaux sur la négociation sont traités, en général, dans le cadre des systèmes multi-agents. Alors le besoin d'agents logiciels capables de négocier avec les autres à la place de leur utilisateur devient de plus en plus important.

Il faut noter que la négociation dont nous traitons dans ce mémoire est la négociation compétitive, puisque nous nous intéressons à la négociation entre agents essayant d'obtenir la meilleure solution possible pour eux.

Au niveau des systèmes multi-agent la négociation apparait comme une technique de résolution de conflits. En effet une situation de conflit peut être résolue au sein d'un système multi-agents par le suivi de règles témoignant d'un arbitrage (dans le cas d'agents réactifs) ou bien l'obtention d'un accord résultant d'un processus de négociation (dans le cas d'agents cognitifs) c'est à dire de respecter les règles d'un protocole de négociation et c'est ce que nous avons essayé de le traiter dans notre mémoire où nous allons essayé de proposer une approche formelle pour la négociation automatique entre agents par la proposition d'une architecture générique composé de deux types d'agent et d'établir un nouveau protocole de négociation et de le valider par une modélisation formelle en utilisant les réseaux de Petri colorés.

Dans la réalité nous trouvons que la grande partie des négociations qui se déroulent entre les humains ou celles qui se passent aujourd'hui sur le web concerne le domaine du commerce

Introduction générale

Négocier est un art du quotidien multi facettes : négociation sur le prix du bien, négociations sur le temps de travail, négociations du régime des retraites, négociations des conditions d'embauche Que ce soit à l'échelle de l'individu ou de l'organisation, dans les domaines du social, du politique ou bien du commercial, la négociation est partout et à tous les niveaux. La négociation se retrouve à tous les moments de la vie que l'on parle de vie personnelle ou professionnelle.

L'automatisation de la négociation est la tâche la plus difficile jusqu'à nos jours. Elle fait l'objet de multiples recherches et théories avec un manque remarquable de pratique. L'automatisation commence de la forme la plus simple comme les enchères jusqu'à la plus complexe comme la négociation à base d'argumentation.

Grâce au progrès des technologies de l'information et des systèmes multi-agents (SMA), nous trouvons que la plus part des travaux sur la négociation sont traités, en général, dans le cadre des systèmes multi-agents. Alors le besoin d'agents logiciels capables de négocier avec les autres à la place de leur utilisateur devient de plus en plus important.

Il faut noter que la négociation dont nous traitons dans ce mémoire est la négociation compétitive, puisque nous nous intéressons à la négociation entre agents essayant d'obtenir la meilleure solution possible pour eux.

Au niveau des systèmes multi-agent la négociation apparait comme une technique de résolution de conflits. En effet une situation de conflit peut être résolue au sein d'un système multi-agents par le suivi de règles témoignant d'un arbitrage (dans le cas d'agents réactifs) ou bien l'obtention d'un accord résultant d'un processus de négociation (dans le cas d'agents cognitifs) c'est à dire de respecter les règles d'un protocole de négociation et c'est ce que nous avons essayé de le traiter dans notre mémoire où nous allons essayé de proposer une approche formelle pour la négociation automatique entre agents par la proposition d'une architecture générique composé de deux types d'agent et d'établir un nouveau protocole de négociation et de le valider par une modélisation formelle en utilisant les réseaux de Petri colorés.

Dans la réalité nous trouvons que la grande partie des négociations qui se déroulent entre les humains ou celles qui se passent aujourd'hui sur le web concerne le domaine du commerce électronique, alors nous allons essayer d'appliquer notre proposition sur un exemple très intéressant et très actif qui est la location des logements.

Pour atteindre ces différents objectifs, nous avons organisé notre mémoire en cinq chapitres :

Le premier chapitre représente l'état de l'art concernant l'interaction entre les agents intelligents où on présente une généralité sur l'interaction et les types d'interactions, la communication et les actes de langages et les langages de communication, finalement les protocoles d'interaction et leurs types.

Le deuxième chapitre représente l'état de l'art concernant la négociation automatique tout en rassemblant tout ce qui est en relation avec la négociation et sa mise en place. Ce sujet est traité, en général, dans le cadre des négociations dans les systèmes multi-agents SMA.

Le troisième chapitre représente premièrement l'état de l'art concernant les approches formelles de spécification des protocoles de négociations qui se présentent par les techniques semi-formelles de spécification, les modèles de spécification formelles pour les SMA, les langages formels de description et la logique et deuxièmement une synthèse générale sur ces approches.

Le quatrième chapitre représente une conception de la négociation entre nos deux agents intelligents *Initiateur* et *Participant* en détaillant leurs architectures avec la description du protocole de négociation qu'ils doivent le respecter pendant le processus de négociation et finalement la spécification et la validation de ce protocole par un réseau de Petri coloré . Nous allons décrire dans ce chapitre un modèle de négociation multi-attributs intelligent et dynamique.

Le cinquième chapitre va traiter l'implémentation de notre travail où nous allons donner un aperçu général sur la manière dont on peut implémenter un système de location de logements en précisant tous les moyens et outils qui doivent être utilisés.

Chapitre 1

L'interaction entre Agents

1. Introduction

Un agent sans interaction avec d'autres agents n'est plus qu'un corps isolé [11], ce qu'indique que l'interaction est une des plus importantes briques d'un système multi-Agents (SMA). Ceci justifie l'intérêt que lui portent les chercheurs de la communauté des SMA. Cette vision permet vraiment de comprendre l'importance de l'interaction dans les systèmes multi-agents, de la même façon on trouve qu'un agent humain dans son travail ; quelque soit ce dernier il doit coordonner, coopérer, négocier ... quelque soit l'opération qu'il fait il doit interagir avec les autres personnes ou agents humains.

Cette prise en compte de l'interaction comme élément essentiel et fondateur dans les SMA poussait beaucoup les chercheurs de travailler et créer dans ce domaine qui est vraiment attractif.

Ce chapitre est composé de trois sections, la première présente une généralité sur l'interaction ; dans cette section nous trouvons la définition détaillée de l'interaction et les types d'interaction (situations d'interactions). La deuxième section concerne la communication, les actes de langage et les langages de communication et enfin la troisième section qui introduit les protocoles d'interaction, dans cette section nous allons définir les protocoles et donnons les types de ces derniers.

2. L'interaction entre les agents

Avant de définir l'interaction entre les agents, nous devons noter que nous allons travailler sur des agents cognitifs, alors les caractéristiques qui distinguent ces agents aux autres sont [26] :

Capacités d'action et de perception : ces capacités, présentes dans tous les modèles d'agents permettent aux agents de percevoir et d'agir sur leur environnement. On retrouve ici la boucle d'interaction avec l'environnement introduite par la cybernétique et élaborée depuis.

Capacités cognitives : représentation explicite des connaissances (incluant généralement le maintien d'un modèle des autres aussi complet que possible) et raisonnement formel,

apprentissage (ou au moins actualisation des connaissances : mise à jour et révision des connaissances), planification garantissant la pro-activité et l'autonomie.

Capacités sociales : gestion des engagements (cela suppose le raisonnement temporel), capacités de communication (manipulation d'un cadre interactionnel).

2.1. C'est quoi l'interaction entre les agents?

Dans ce qui suit nous allons nous concentrer seulement sur l'interaction directe entre les agents pas l'interaction au sens général. Citons par exemple l'interaction par l'intermédiaire d'un autre agent ou même en agissant sur leur environnement, et puisque l'objectif de notre travail aussi est de trouver une approche formelle pour la négociation automatique entre agents nous étions obligés d'étudier seulement l'interaction entre les agents cognitifs qui est basée sur la communication intentionnel (que nous allons voir après), à raison que comment un agent peut négocier sans qu'il a une connaissance au préalable sur les autres agents ou qu'il a la volonté directe d'obtenir un effet du récepteur.

Dans ce sens nous trouvons qu'il n'y a pas une seule définition pour l'interaction entre les agents, mais il y a plusieurs qui présentent la multiplicité des vues des différents chercheurs et qui arrivent au même sens. J. FERBER dans son ouvrage [11] a donné une vision sur les SMA qui est vraiment évidente et intéressante, il a aussi essayer de définir l'interaction entre les agents comme suit : « *Une interaction est une mise en relation dynamique de deux ou plusieurs agents par le biais d'un ensemble d'actions réciproques. Les interactions s'expriment ainsi à partir d'une série d'actions dont les conséquences exercent en retour une influence sur le comportement futur des agents. Les agents interagissent le long d'une suite d'évènements pendant lesquels les agents sont d'une certaine manière en contact les uns avec les autres...* »

M. BARBUCEANU et M S. FOX ajoutent dans [3] que L'interaction entre les agents est plus qu'un simple échange de messages: un des aspects d'interaction montre une conversation basé sur un échange partagé et conventionné de messages.

2.2. Conditions d'apparition d'interaction

On ne peut pas comprendre la notion d'interaction sans savoir les conditions intrinsèques qui poussent les agents à interagir. J. FERBER indique les conditions suivantes [11] : la présence d'agents capables d'agir et/ou de communiquer.

des situations susceptibles de servir de point de rencontre entre agents: collaboration, déplacement de véhicules amenant à une collision, utilisation de ressources limitées, régulation de la cohésion d'un groupe.

des éléments dynamiques permettant des relations locales et temporaires entre agents: communication, choc, champ attractif ou répulsif, etc.

un certain "jeu" dans les relations entre les agents leur permettant à la fois d'être en relation, mais aussi de pouvoir se séparer de cette relation, c'est-à-dire de disposer d'une certaine autonomie. Si des agents sont totalement liés par un couplage fixe, leur interaction devient rigide et ils n'interagissent plus au sens plein du terme.

2. 3. Composantes des interactions

Les principales composantes d'interactions sont [11]:

❖ Buts compatibles et incompatibles

Les objectifs et interactions des agents sont évidemment au centre des problématiques de l'interaction : les agents ont-ils des buts concordants, c'est à dire allant dans le même sens, ou bien leurs objectifs sont-ils contradictoires, voire opposés? D'une manière générale, on dira que des buts sont incompatibles si la satisfaction de l'un entraîne l'insatisfaction de l'autre, la réciproque étant automatiquement vraie.

❖ Relations aux ressources

Une autre composante importante des interactions est relative aux ressources extérieures dont disposent les agents. On entend par ressources tous les éléments environnementaux et matériels utiles à la réalisation d'une action. Il peut s'agir de valeurs énergétiques, de possibilité financière, d'outils, de matières premières, mais aussi de la prise en compte de notions plus générales telles que l'espace et le temps disponibles pour l'accomplissement d'une tâche.

❖ Capacités des agents par rapport aux tâches

Le rapport entre les capacités des agents et les tâches à accomplir constitue la troisième composante fondamentale des interactions entre agents. Est-ce qu'un agent peut réaliser seul une tâche ou bien a-t-il besoin des autres pour parvenir à son but?

Par exemple une maison ce n'est pas simplement "construire des murs plus mettre des conduites d'eau plus mettre de l'électricité", un avion ne peut se résumer à la juxtaposition d'un fuselage et de moteurs. Dans tous ces cas, la chose produite est un plus par rapport à la simple somme des compétences de chacun des agents et cette somme apparaît dans l'interaction de chaque élément.

2. 4. Définition de situation d'interaction

Selon FERBER [11] on appellera situation d'interaction un ensemble de comportements résultant du regroupement d'agents qui doivent agir pour satisfaire leurs objectifs en tenant compte des contraintes provenant des ressources plus ou moins limitées dont ils disposent et de leurs compétences individuelles.

En combinant les trois critères (composantes) d'interaction, nous aurons huit types de situation d'interaction organisés en trois catégories [11] (voire Table 1.1) :

Buts	Ressources	Compétences	Types de situation	Catégories
Compatibles	Suffisantes	Suffisantes	Indépendance	Indifférence
Compatibles	Suffisantes	Insuffisantes	Collaboration simple	Coopération
Compatibles	Insuffisantes	Suffisantes	Encombrement	
Compatibles	Insuffisantes	Insuffisantes	Collaboration coordonnée	
Incompatibles	Suffisantes	Suffisantes	Compétition individuelle pure	Antagonisme
Incompatibles	Insuffisantes	Suffisantes	Compétition collective pure	
Incompatibles	Suffisantes	Insuffisantes	Compétition individuelle sur les ressources	
Incompatibles	Insuffisantes	Insuffisantes	Compétition collective sur les ressources.	

Table 1.1 : Les différents types de situation d'interaction [11].

2.4.1. L'Indifférence

Dans le cas, où les agents disposent de capacités suffisantes (aucun agent n'a besoin des autres agents pour atteindre son but), des ressources suffisantes (chaque agent dispose des ressources nécessaires pour atteindre son but), et si les buts des agents sont complètement indépendants, les agents sont en situation d'indépendance. Dans ce cas, il n'y a aucune interaction car il n'y a aucun point de rencontre. Par exemple : deux voitures qui se déplacent sur deux routes différentes, deux programmes informatiques qui s'exécutent sur deux machines différentes, ...etc.

2.4.2. La Coopération

C'est dans le cas, où les buts des agents sont compatibles (dans le sens où satisfaire le but d'un agent participe et améliore la satisfaction des buts des autres agents), qu'on parle de coopération. On distingue trois situations possibles :

1. Collaboration simple

Dans ce cas, les buts des agents sont compatibles et les ressources sont suffisantes. Mais les agents veulent résoudre un problème qui nécessite des capacités non disponibles dans un seul agent. Cette insuffisance de capacité pousse les agents à interagir pour partager les connaissances du problème, diviser le problème en sous tâches et allouer ces sous tâches aux agents. Les agents sont en collaboration simple. Par exemple deux programmes qui s'exécutent sur deux machines différentes (chaque programme dispose de toutes les ressources nécessaires) interconnectées, et qui travaillent ensemble sur une même tâche pour l'accomplir en moins de temps.

2. Encombrement

Les buts des agents sont compatibles, les agents ont des capacités individuelles suffisantes (aucun agent n'a besoin de l'aide des autres) mais les ressources nécessaires aux agents sont insuffisantes. Cette insuffisance de ressource rend les agents en situation d'encombrement. Pour vaincre l'encombrement, les agents doivent suivre certains mécanismes (la coordination, la négociation) déjà mentionnés. Par exemple, les habitants d'un immeuble qui ont des buts compatibles (chacun exerce un métier différent), des compétences suffisantes et qui partagent l'espace, l'escalier, l'ascenseur, ... de leur immeuble. Le trafic routier, le trafic aérien, ... etc. où plusieurs entités partagent un seul espace, comportent aussi des situations d'encombrement.

3. Collaboration Coordonnée

Dans ce cas, en plus de l'insuffisance des ressources, les agents souffrent d'une insuffisance de capacités (les agents veulent réaliser une tâche qui nécessite des capacités distribuées sur différents agents). Les agents doivent collaborer pour gérer l'insuffisance des capacités et coordonner leurs actions ou négocier pour contourner la limite des ressources. On dit que les agents sont en situation de collaboration coordonnée. Par exemple le cas des robots des minerais, qui ont un but commun et

qui doivent partager un ensemble de ressources limitées : environnement, énergie, ...etc.

2.4.3. L'Antagonisme

C'est dans le cas où les buts des agents sont incompatibles. On distingue quatre situations possibles :

1. Compétition individuelle pure

Quand les agents ont des capacités suffisantes (aucun agent n'a besoin des autres), des ressources suffisantes et des buts incompatibles, ils se luttent pour atteindre leurs objectifs individuels. Dans ce cas les agents sont en situation de compétition individuelle pure. Un combat entre deux individus (deux animaux, deux personnes, ...etc.) où chaque individu a à sa disposition les armes qui lui sont nécessaires pour continuer à combattre, est une situation de compétition individuelle pure.

2. Compétition individuelle sur les ressources

En plus de la compétition due à l'incompatibilité des buts, les agents se luttent aussi sur les ressources limitées. On dit que les agents sont en compétition individuelle sur les ressources. Un combat entre deux individus (deux animaux, deux personnes, ...etc.), où les ressources (les armes, les aliments, ...etc.), nécessaires pour les deux, pour continuer à combattre, sont limitées et critiques, est un exemple de compétition individuelle sur les ressources.

3. Compétition collective pure

Si les agents n'ont pas de capacités suffisantes, ils vont se regrouper en des collections d'agents. Chaque collection aura un but global. Ces collections d'agents, qui ont des buts globaux incompatibles, entrent en compétition pour atteindre leurs buts. On dit que c'est une compétition collective pure. Par exemple, dans un match de football, les joueurs de chacune des deux équipes travaillent ensemble pour gagner le match. Chaque joueur participe selon ses compétences et son rôle, pour que son équipe puisse vaincre l'adversaire.

4. Compétition collective sur les ressources

Dans le cas, où les collections d'agents n'ont pas des ressources suffisantes. Ces collections, en plus de leur compétition due à l'incompatibilité de leurs buts, se luttent sur Les ressources limitées. La guerre représente un exemple de compétition collective sur les ressources : Les ennemies sont des collections d'agents, qui se luttent pour atteindre des buts incompatibles (par exemple : occuper un certain territoire), et cette compétition peut aussi toucher des ressources nécessaires pour survivre et vaincre la guerre : l'eau, les armes, ... etc.

Finalement, pour mieux comprendre les choses il faut savoir sur quoi elles sont basées. Nous trouvons que Les communications dans les systèmes multi-agents comme chez les humains, sont à la base des interactions et de l'organisation sociale [11]; alors nous ne pouvons pas parler sur l'interaction sans passer aussi par la communication, et précisément la communication intentionnelle qui est basée sur les messages intentionnelles (comme dans notre cas ou les agents sont cognitifs).

L'interaction constitue donc un niveau d'abstraction supérieur à la notion de communication et d'action. Le problème est alors de savoir combiner ces éléments (communications et actions) afin de coordonner et de contrôler les échanges entre plusieurs agents pour avoir un comportement collectif cohérent du système.

Pour illustrer la vision beaucoup plus claire, nous ajoutons qu'il y a une différence entre communication et interaction ; la communication permet aux agents de s'envoyer des messages et de les comprendre. L'interaction va plus loin car elle permet des conversations entre agents structurées sur la communication (échange de messages) [4].

3. La Communication

Avant de passer à la définition nous devons préciser quelle forme de communication utilisent les agents pendant la négociation. Selon FERBER dans [12], Il existe deux formes de communications :

1. les communications intentionnelles qui mettent en contact des agents cognitifs par le biais d'envois de messages. Cette forme de communication est surtout utilisée par des agents organisés en réseaux.

2. les communications réactives qui prennent la forme de signaux transmis entre agents réactifs. On rencontre ces formes de communication surtout dans la robotique collective mobile et la simulation de sociétés animales, beaucoup moins dans les réseaux.

A partir de cette définition on peut tirer la forme de communication utilisée par les agents pendant la négociation, évidemment c'est la communication intentionnelle (que nous allons détailler après) car la négociation exige la présence des agents capables de raisonner et de prendre des décisions, c'est-à-dire des agents cognitifs

3.1 Définitions de la communication

L'idée de communiquer n'est pas récente, elle est apparue même avant l'apparition des SMA et elle est basée sur la théorie de communication développée par *Shannon et Weaver*, durant les années 40. Ils supposaient que la communication est une transmission d'information d'un émetteur vers un récepteur. Cette information doit être codée dans un langage et elle sera décodée, à son arrivée, par le récepteur. Le code utilisé est explicite et il doit respecter une syntaxe, bien définie, reconnue par les deux communicants, et tout ça à travers un canal (medium).

La même chose dans les SMA, la communication peut être définie comme une forme d'action locale d'un agent vers d'autres agents [11]. Nous distinguons essentiellement deux modes de communication : la communication indirecte qui est une communication par signaux via l'environnement, et la communication directe qui procède à un échange de messages entre les agents [19]. Le mode de communication par échange de messages entre les agents permet à un agent de planifier un acte de communication envers un autre agent, on parle alors de communication intentionnelle. Cette approche s'est inspirée de l'interaction sociale telle que nous la trouvons dans d'autres contextes comme la communication entre les humains.

Pour mieux comprendre le modèle de communication dans les SMA nous devons voir les questions abordées par un modèle de communication qui sont proposées par MAZOUZI [23], ces questions peuvent être résumées par l'interrogation suivante : qui communique quoi, à qui, quand, pourquoi, et comment ?

1. ***Pourquoi les agents communiquent-ils ?***, la communication doit permettre la mise en œuvre de l'interaction et par conséquent la coopération et la coordination d'actions.

2. ***Quand les agents communiquent-ils ?***, les agents sont souvent confrontés à des situations où ils ont besoin d'interagir avec d'autres agents pour atteindre leurs buts locaux ou globaux. La difficulté réside dans l'identification de ces situations. Par exemple, une communication peut être sollicitée suite à une demande explicite par un autre agent.
3. ***Avec qui les agents communiquent-ils ?***, les communications peuvent être sélectives sur un nombre restreint d'agents ou diffusées à l'ensemble des agents. Le choix de l'interlocuteur dépend essentiellement des accointances de l'agent (connaissances qu'a l'agent sur les autres agents).
4. ***Comment les agents communiquent-ils ?***, la mise en œuvre de la communication nécessite un langage de communication compréhensible et commun à tous les agents. Il faut identifier les différents types de communication et définir les moyens permettant non seulement l'envoi et la réception de données mais aussi le transfert de connaissances avec une sémantique appropriée à chaque type de message.

3.2. Actes de langage

La théorie des actes de langage est issue de la philosophie du langage. Initialement pensée pour le langage naturel, sa nature formelle la rend utilisable pour les modèles computationnels. L'idée maîtresse de cette théorie est qu'une instance d'utilisation de la langue est une action comme les autres : « dire c'est faire » [1].

Selon AUSTIN dans [1], pour chaque acte de langage « primitif », on distingue quatre composantes qui peuvent être vues comme quatre actes :

- *Enonciation* : le locuteur fournit l'énoncé dans le contexte par transmission ou prononciation du message ; c'est le niveau physique.
- *Acte locutoire ou locution* : l'interlocuteur (ou les interlocuteurs, le cas échéant) aperçu l'énonciation. Il lui faut interpréter le sens de l'énoncé en termes de son contenu propositionnel. Si le contenu propositionnel interprété est celui que le locuteur voulait transmettre, on dira que l'aspect locutoire de l'acte est accompli avec succès. Par exemple : « il pleut », « it's raining » et « es regnet » sont trois énoncés différents qui correspondent à un seul acte locutoire de contenu propositionnel : il pleut. Dans la suite, on notera p ce contenu propositionnel qui correspond à ce qui est dit.
- *Acte illocutoire ou illocution* : cet acte traduit les intentions du locuteur envers son (ses) interlocuteur(s). Une fois que l'interlocuteur a perçu et interprété le sens propositionnel de l'énoncé, il doit inférer ce que le locuteur a voulu exprimer par cet

énoncé. « On va être riche » peut être interprétée selon le contexte comme une information, une prédiction, une promesse ou une blague ironique et cela change tout. Si l'interlocuteur saisit le sens que le locuteur a voulu donner à son énoncé on dira que l'acte illocutoire a réussi.

- *Acte perlocutoire ou perlocution* : un tel acte porte sur les effets du message sur le destinataire : action, modification de croyance, modification de ses attitudes propositionnelles (AP). L'effet perlocutoire concerne la réaction du destinataire, les effets de son interprétation sémantique du message.

Le terme « acte de langage » est souvent employé pour désigner un acte illocutoire, et pour Austin c'est l'acte le plus intéressant. Après Austin d'autres chercheurs ont poursuivi leurs recherches en se basant sur les travaux d'Austin ; parmi ces chercheurs on trouve Searle qui a défini cinq catégories et chaque acte illocutoire doit appartenir à l'une parmi ces dernières. Searle a bien détaillé ces catégories dans [29] :

- a. *Acte assertif* : le locuteur exprime comment les objets auxquels il se réfère sont dans le monde. Il s'agit des assertions, des informations, des témoignages, des démentis, etc.
- b. *Acte commissif* : l'auteur s'engage à accomplir une action. Il s'agit des promesses, des vœux, des menaces, etc.
- c. *Acte directif* : le locuteur fait en sorte que l'interlocuteur accomplisse une action. Il s'agit des demandes, des questions, des ordres, des conseils, etc.
- d. *Acte expressif* : le locuteur manifeste son état mental face à un état de chose. Il s'agit des excuses, remerciements, félicitations, récriminations, etc.
- e. *Acte déclaratif* : le locuteur accomplit au moment de l'énonciation l'action qu'il dit accomplir.

3.3 Langages de communication inter-agents

Ces langages se focalisent essentiellement sur la manière de décrire exhaustivement des actes de communication d'un point de vue syntaxique et sémantique supportant un langage de représentation des connaissances. Toutefois, l'aspect ontologique et l'utilisation de conventions garantissant un comportement collectif cohérent du système et l'aspect conversationnel n'est pas facile à garantir. Plusieurs tentatives de normalisation de la

communication inter-agents ont été effectuées au sein de la communauté multi-agents ces dernières années. Selon Mazouzi dans [23], nous allons présenter les plus intéressants.

3.3.1 Langage de communication KQML

Aux Etats-Unis, un standard de langage de communication de haut niveau appelé KQML “ Knowledge Query and Manipulation Language ” (Labrou et Finin, 1997) a été développé. Ce dernier est fondé sur la théorie des actes de langage dans le but de permettre aux agents cognitifs de coopérer. Il est basé sur le fait de pouvoir coder explicitement dans les messages des actes illocutoires en termes de type de message ou “ performatives ” et repose sur les états mentaux des agents. Le contenu du message échangé est une expression spécifiée en KIF (Knowledge Interchange Format) qui utilise le formalisme de la logique de premier ordre.

KQML est un langage de communication et un protocole de haut niveau pour l'échange de l'information, orienté messages et indépendant de la syntaxe du contenu et de l'ontologie applicable. Une ontologie est une spécification ou une vue simplifiée et abstraite du domaine qui sera représenté. En d'autres termes, une ontologie définit le vocabulaire dans un domaine donné pour que les agents puissent se comprendre. En plus, KQML est indépendant du mécanisme de transport (TCP/IP, SMTP, IIOP ou autre), indépendant du langage du contenu échangé (KIF, SQL, Prolog ou autre) et indépendant de l'ontologie utilisé. Conceptuellement, nous pouvons identifier trois couches dans un message de KQML : contenu, communication et message.

- La couche “ contenu ” comporte la teneur réelle du message utilisant un langage de représentation propre au système. KQML peut supporter n'importe quel langage de représentation, y compris des langages exprimés en ASCII et ceux exprimés en utilisant la numération binaire.
- La couche de “communication ” code un ensemble de dispositifs au message qui décrivent les paramètres de communication les plus bas, tels que l'identité de l'expéditeur et du destinataire et un identifiant unique associé à la communication.
- La couche “ message ”, qui code un message qu'une application voudrait transmettre à une autre, est le noyau de KQML. Cette couche détermine les genres d'interactions au sein des agents dialoguant via KQML. La fonction de la couche message est d'identifier l'acte du langage ou la performative que l'expéditeur

attache au contenu. Cet acte de langage indique si le message est une affirmation, une question, une commande ou tout autre d'un ensemble de performatifs connus (types de messages primitifs). En outre, puisque le contenu est opaque à KQML, le message inclut également les options qui décrivent le langage du contenu, l'ontologie qu'il suppose utiliser et une certaine description du contenu, tel qu'un descripteur appelant une matière dans l'ontologie. Ces dispositifs permettent une analyse correcte des messages sans avoir à accéder au contenu.

Les messages de KQML ne concernent pas seulement des phrases dans un langage quelconque, mais sont enrichis d'une attitude sur le contenu (affirmation, désengagement, requête, question, etc.). Ce langage propose une description d'un nombre important de performatives permettant les conversations entre agents mais manque de spécifications et de formalisation.

3.3.2. Langage de communication FIPA-ACL

Récemment, la collaboration internationale des membres d'organisations universitaires et industrielles regroupées au sein de FIPA (*Foundation for Intelligent Physical Agents*) a permis de spécifier des standards dans la technologie agent et vise à favoriser l'interopérabilité des applications, des services et des équipements informatiques basés sur le paradigme agent. Ils ont défini un certain nombre de spécifications principales d'agents. Notamment, un standard de langage de communication agent ACL (*Agent Communication Language*) a été proposé et spécifié. Comme KQML, ce dernier est basé sur la théorie des actes de langage : les messages sont des actions ou des actes communicatifs, car ils sont prévus pour effectuer une certaine action en vertu de l'envoi. Les spécifications de FIPA-ACL se composent d'un ensemble de types de message et de la description de leur pragmatique que sont, des effets sur les attitudes mentales des agents (expéditeur et récepteur). Les spécifications décrivent chaque acte communicatif avec une forme narrative et une sémantique formelle basée sur la logique modale.

Elles fournissent également la description normative d'un ensemble de protocoles d'interaction de haut niveau, y compris la demande d'action, l'établissement de contrat et plusieurs genres de ventes aux enchères. FIPA-ACL est superficiellement semblable à KQML. Sa syntaxe est identique à celle de KQML excepté différents noms pour quelques primitifs réservés. Ainsi, il maintient l'approche de KQML de distinguer le langage externe

du langage interne. Le langage externe définit la signification prévue du message; le langage interne ou le contenu, dénote l'expression à laquelle s'appliquent les croyances, les désirs, et les intentions des interlocuteurs.

Les différentes approches développées à ce jour respectent partiellement les bases de la théorie des actes de langage (Searle, 1969). Elles considèrent que l'interaction est basée sur des actions communicatives mutuellement échangées entre les agents et qui sont initiées par des intentions, ce qui justifie l'utilisation d'une logique pour raisonner sur les intentions.

L'enchaînement des actes de communication de telle sorte à produire une "discussion" cohérente entre les agents nécessite une formalisation non seulement au niveau de ses composantes, c'est-à-dire les actes de communication. La logique et la sémantique des actes doivent alors être étendues pour la problématique de l'enchaînement par la notion de protocole. Les documents de référence de FIPA-ACL et de KQML ne spécifient aucune implantation par l'utilisation d'un langage de programmation ni de plate-forme particulière. Ce champ de développement est particulièrement propre à chaque équipe désirant implanter un tel langage. La seule obligation est que l'implantation doit être conforme aux spécifications du langage.

4. Les protocoles d'interaction

Les protocoles d'interaction ne sont pas les protocoles de communication que nous trouvons dans les systèmes distribués. En effet, les premières recherches dans le domaine des systèmes multi-agents ont adapté les protocoles de communication. On parle d'adaptation car il n'est pas possible d'utiliser directement les protocoles de communication. De nombreuses différences existent entre les processus ou objets et les agents. En particulier, les agents sont autonomes et ont la possibilité de se réunir afin de coopérer.

Un protocole d'interaction est une spécification qui guide la communication entre agents. Il précise qui peut dire quoi à qui et les réactions possibles à ce qui est dit par l'autre partie. La fonction de base d'un protocole d'interaction est de permettre aux agents de communiquer sans avoir besoin de planifier explicitement chaque acte du langage en se basant sur ses effets et actions attendus. Le protocole englobe tous les messages qui peuvent être échangés entre les rôles communicants. Généralement l'agent Initiateur est celui qui commence la communication dans le but d'accomplir une tâche précise en contactant des agents participants [10].

Une autre définition est celle introduite par M. MAXIME dans [25], On appelle protocole d'interaction un ensemble de règles plus ou moins génériques qui sont partagées par les agents du système et auxquelles les interactions doivent se conformer, Les protocoles permettent de guider l'interaction des agents. Ce sont des plans définis a priori pour coordonner leurs interactions.

Il y a quatre notions fondamentales qui caractérisent un protocole d'interaction [19] :

- Un protocole d'interaction est un pattern d'interaction. Ce qui explique le besoin de le représenter d'une manière générique, indépendamment du contexte d'application.
- Chaque protocole d'interaction a un but.
- Un protocole d'interaction fait intervenir deux ou plusieurs agents. Chacun de ces agents joue un rôle qui permet de l'identifier au cours de l'interaction.
- Le protocole d'interaction définit les règles qui permettent de régir une interaction. Ces règles définissent l'ordonnancement des messages, ainsi que les actions auxquelles le protocole fait appel.

Nous trouvons dans la littérature que la négociation est la forme d'interaction qui nécessite le plus de communication entre les agents [19], alors la majorité des protocoles d'interaction sont des protocoles de négociation.

Cependant, nous pouvons classier les protocoles d'interaction selon MAZOUZI dans [23], ou il a classifié les protocoles d'interaction en quatre principales classes : les protocoles de coordination, les protocoles de coopération, les protocoles de négociation et les protocoles de vente aux enchères.

4.1. Les protocoles de coordination

Dans des environnements à ressources limitées, la coordination se traduit par un comportement individuel visant à servir ses propres intérêts tout en essayant de satisfaire le but global du système. JENINGS dans [20] caractérise la coordination par deux aspects étroitement liés, à savoir les engagements et les conventions :

- ✓ Les engagements fournissent la structure nécessaire pour des interactions prévisibles. Pendant que les situations changent, les agents doivent évaluer si les engagements existants sont encore valides.
- ✓ Les conventions fournissent des moyens pour contrôler les engagements dans des circonstances changeantes. Un des exemples de protocole pour la coordination est celui du réseau contractuel (Contract-Net). L'intérêt d'un tel

protocole est qu'il réalise la coordination de tâches parmi les agents en assurant l'allocation la plus optimale possible.

4.2. Les protocoles de coopération

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

4.3. Les protocoles de négociation

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

4.4. Les mécanismes du commerce électronique

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

Une vente aux enchères est un mécanisme économique permettant de déterminer le prix d'un article. Elle exige une méthodologie prédéfinie, où se réunissent un ou plusieurs contractants qui souhaitent acheter l'article. Habituellement l'article est vendu d'une façon publique au plus offrant.

Nous constatons que MAZOUZI a limité les protocoles de négociation seulement sur les agents compétitifs, mais il y a d'autres chercheurs qu'ont un autre point de vue ; où ils voient que la négociation peut être compétitive comme peut être coopérative comme Chaib-draa dans [7].

5. Conclusion

Nous avons essayé dans ce chapitre d'illustrer la notion d'interaction entre les agents qui présente un élément intrinsèque dans les SMA, et puisque nous travaillons sur des agents cognitifs nous pouvons conclure directement que l'interaction que nous considérons est l'interaction langagière c'est-à-dire celle qui est basée sur les actes de langage. Dans le même sens nous pouvons dire aussi que les agents sur lesquels nous travaillons doivent négocier au sein d'un système multi-agents mais qui n'ont pas un objectif global et commun c'est-à-dire des agents compétitifs et ne sont pas des agents coopératifs.

Chapitre 2

La négociation entre Agents

1. Introduction

Selon Brahim CHAIB-DRAA dans [7], en intelligence artificielle distribuée (IAD), il existe deux tendances principales : la résolution distribuée de problèmes et les systèmes d'agents autonomes.

- ❖ La première se caractérise par plusieurs agents conçus pour résoudre un problème global. Ces agents sont prédisposés à la coopération. En effet, leur but premier est de résoudre une tâche globale et, sans coopération, cela est impossible. Implicitement, il y a une notion d'utilité globale chez les agents. Dans ce cadre, la négociation sert à la distribution des tâches et des ressources entre les agents et ceci est fait, en ne perdant pas de vue qu'il s'agit de la résolution d'un seul et unique problème.
- ❖ La deuxième tendance fait cohabiter des agents autonomes dans un même environnement. Un agent autonome est un agent qui cherche à maximiser son utilité personnelle en fonction de ses buts. Dans un tel contexte, les agents négocient afin de partager le travail qu'ils ont à accomplir ou de résoudre des conflits lors de l'utilisation de ressources limitées. Les agents en viennent à une entente lorsque c'est avantageux pour eux.

Dans notre travail nous allons nous concentrer surtout sur la négociation compétitive où les agents cherchent à maximiser leurs utilités personnelles en fonction de leurs buts c'est-à-dire, des agents égoïstes. Nous avons divisé ce chapitre en trois sections, la première contient quelques définitions sur la négociation, la deuxième présente les formes (types) de négociation et la troisième concerne les protocoles de négociation où nous avons spécifié les types de ces derniers et nous avons donné aussi des exemples sur ces protocoles.

2. Définitions et formes de négociation automatique

La négociation était et reste toujours un thème très intéressant dans les SMA. Nous pouvons dire aussi que sa valeur est augmentée grâce à la propagation et le développement du commerce électronique.

En général les chercheurs donnent des définitions de la négociation mais qui sont similaires, nous pouvons poser la question : pourquoi similaires ?

La réponse est simple car la notion de négociation n'est pas récente, elle est apparue même avant l'apparition de l'informatique et des systèmes multi-agents. Les chercheurs n'ont rien apporté de nouveau, ils décrivent seulement la réalité

2.1. Définition

Selon le point de vue de MH.VERRON dans [34], il y a négociation lorsqu'il y a une discussion, des propositions entre les protagonistes et lorsque l'accord final satisfait au mieux tous les participants. VERRON a donné aussi une autre définition dans [34]. Une négociation met en jeu des ressources, qui seront rassemblées afin d'être négociées dans un contrat et un ensemble de personnes qui participent à cette négociation. Il y a toujours un ou plusieurs manageurs (vendeur ou autre) et un ou plusieurs contractants (acheteurs ou autre).

Une autre définition dans [22] ; « La négociation s'effectue sur un contrat pour obtenir des ressources communes et à la demande d'un initiateur ». Elle réunit un ensemble de participants et un initiateur et se déroule jusqu'à ce qu'un accord satisfaisant un pourcentage de participants soit trouvé. Les participants cherchent également à obtenir la meilleure solution possible pour eux tout en donnant le minimum d'informations aux autres.

Mais c'est quoi un contrat ?, un contrat est l'entité qui sera négociée. Il comprend l'initiateur de la négociation, les ressources impliquées, le délai d'attente des réponses et une réponse par défaut pour le cas où un participant ne répondrait pas à temps.

En outre, la mise en œuvre de la négociation entre agents doit respecter les caractéristiques suivantes :

- a. les intervenants doivent pouvoir être indifféremment humains ou logiques.
- b. ils doivent tous utiliser le même langage pour communiquer.
- c. la communication doit être asynchrone et banalisée.
- d. un agent ne doit pas avoir accès à la totalité de la connaissance des autres agents.

- e. si une proposition de contrat n'est pas acceptée, l'agent doit être capable de modifier d'autres contrats et de négocier la modification de contrats des autres interlocuteurs.
- f. plusieurs négociations simultanées doivent être possibles.
- g. certains contrats doivent être plus prioritaires que d'autres.
- h. certains participants doivent être plus prioritaires que d'autres.
- i. l'humain peut commencer la négociation, laisser son agent continuer un certain temps puis reprendre la négociation sans que les autres intervenants ne se rendent compte du changement.
- j. le système ne doit pas se bloquer.

2.2. Les formes de négociation automatique

Selon MH.VERRON dans [34], les types (formes) de négociation sont:

2.2.1. Les systèmes de vote

Les systèmes de vote sont utilisés pour élire une alternative parmi les différentes alternatives possibles. Les plus simples concernent un choix entre une alternative et le statu quo. Cela revient à proposer l'alternative et à recueillir les votes pour et les votes contre cette alternative. Les systèmes plus complexes impliquent un nombre d'alternatives supérieur à deux, les votants devant alors choisir l'alternative qu'ils préfèrent. Le terme de choix social est utilisé afin de représenter l'alternative qui satisfait au mieux la population.

Les systèmes de vote sont des systèmes de négociation à un seul tour de parole (les participants envoient leur liste de préférences et l'initiateur réalise la procédure choisie) qui permettent de choisir parmi toutes les alternatives possibles. Il faut donc que chacun évalue l'ensemble des alternatives et les classe par ordre de préférence avant que le vote n'ait lieu. Cela est alors très coûteux en temps et en espace.

2.2.2. Les enchères

Les enchères se sont démocratisées sur internet grâce aux sites comme eBay, Yahoo enchères, ou Sale . . . C'est la forme de négociation la plus connue, bien que peu de systèmes d'enchères offrent de réelles possibilités de négociation. Elles sont utilisées pour la vente d'objets d'art, d'antiquités, de produits périssables, mais également pour l'immobilier et les voitures, et elles sont même utilisées dans certains jeux, il y a

différentes formes d'enchères, nous allons seulement présenter quatre formes d'enchères :

❖ **Les enchères ascendantes (anglaise) :** ces enchères sont dites ascendantes car le prix proposé pour le bien mis en vente augmente avec le temps. Elles sont aussi appelées enchères ouvertes, orales ou anglaises. Ces enchères se déroulent sur plusieurs tours. Lors d'enchères ascendantes, le prix est successivement augmenté jusqu'à ce qu'il ne reste qu'un seul acheteur, qui gagne le bien au prix final. Dans le modèle le plus couramment utilisé, le prix est continuellement augmenté par le vendeur et les acheteurs quittent les enchères au fur et à mesure que le prix dépasse leur budget. Les autres acheteurs observent les départs et une fois qu'un acheteur a quitté l'enchère, il ne peut plus y revenir. Ces enchères se déroulent sur plusieurs tours, il y a donc discussion entre le vendeur et les acheteurs, ce qui est bien, à notre sens, de la négociation. Le protocole de négociation de ces enchères peut se formaliser ainsi : le vendeur propose son article et le prix qu'il veut en obtenir à un ensemble d'acheteurs. Ceux-ci acceptent ou refusent la proposition. S'ils refusent, ils quittent la négociation. S'ils acceptent, ils restent en course pour l'obtention de l'article. S'il ne reste qu'un seul acheteur, il gagne le contrat. Sinon, le vendeur propose à nouveau son article avec un prix plus élevé. Le processus se termine lorsqu'il ne reste plus qu'un acheteur. Comme ce sont des enchères, il n'est pas possible pour un acheteur de se rétracter : une fois qu'il a acquis l'objet, il ne peut pas le rendre.

❖ **Les enchères descendantes (hollandaises) :** ces enchères sont dites descendantes car le prix proposé pour le bien mis en vente diminue avec le temps. Elles sont utilisées notamment aux Pays-Bas pour la vente des fleurs, c'est pourquoi les économistes les appellent aussi enchères hollandaises. Ces enchères fonctionnent de façon exactement opposée aux précédentes : le vendeur propose un très haut prix pour son bien et le diminue jusqu'à ce qu'un acheteur se manifeste pour acquérir le bien au prix alors mentionné. Ces enchères sont aussi une négociation, puisqu'elles se déroulent sur plusieurs tours. Le protocole se définit comme suit : le vendeur propose son article à un prix très élevé à un ensemble d'acheteurs. Si un acheteur accepte la proposition, il gagne le contrat. Si tous les acheteurs refusent, le vendeur leur propose à nouveau son article à un prix moins élevé. Le processus se termine lorsqu'un acheteur accepte la proposition ou lorsque le prix de réserve est atteint et que personne n'accepte la proposition.

❖ **Les offres scellées au meilleur prix** : lors de ces enchères, qui se déroulent sur un seul tour, chaque acheteur propose un prix unique, sans connaître le prix proposé par les autres (d'où le terme offres scellées). Le vainqueur est celui qui a proposé le prix le plus élevé et paie son prix, d'où leur nom. Comme il n'y a pas de seconde proposition possible, ces enchères ne sont pas une négociation à proprement parler. Le protocole de ces enchères est très simple : le vendeur propose un article à un ensemble d'acheteurs. Ceux-ci peuvent soit accepter la proposition, auquel cas ils proposent un prix, soit refuser la proposition. Le gagnant de l'enchère est celui qui a proposé le prix le plus élevé (supérieur au prix de réserve). L'une des différences avec les précédentes enchères est qu'il n'y a pas de contreproposition possible.

❖ **Les offres scellées au second meilleur prix (Vickrey)** : ces enchères sont également appelées enchères de Vickrey. Le déroulement de ces enchères est exactement le même que celui des enchères aux offres scellées au meilleur prix, mis à part que le vainqueur paie le second meilleur prix proposé. Ici aussi, il n'y a pas de seconde proposition possible et donc pas de négociation à proprement parler. Le protocole est identique à celui des offres scellées au meilleur prix, la seule différence étant le prix à payer par le gagnant du contrat.

2.2.3. La négociation à base d'argumentation

La négociation à base d'argumentation est utilisée chez les agents logiques qui possèdent une base de connaissances avec des prédicats et des règles d'inférence. L'argumentation a alors pour but de modifier les croyances des autres agents afin qu'ils adoptent le même point de vue, les mêmes croyances et intentions que l'agent argumentant. Dans la négociation, les agents peuvent utiliser différents types d'arguments. Chaque type d'argument est défini par des pré-conditions pour son utilisation. Si ces conditions sont satisfaites, alors l'agent peut utiliser l'argument. Évidemment, parmi les arguments possibles dans une certaine situation, l'agent doit choisir le bon argument ; par conséquent l'agent a besoin d'une stratégie pour décider quel argument utiliser.

Les types d'arguments sont les suivants :

- ✓ **Appels à une promesse passée** : le négociateur A rappelle à B une promesse passée concernant l'objet de négociation, autrement dit l'agent B a promis dans une négociation antérieure à l'agent A d'offrir ou effectuer un objet de négociation.

Pré-conditions : l'agent A doit vérifier si une promesse concernant un objet de négociation a été reçue dans le passé à l'occasion d'une négociation conclue avec succès.

- ✓ **Promesse d'une récompense future** : le négociateur A promet de faire l'objet de négociation pour un autre agent B à un moment futur.

Pré-conditions : l'agent A doit trouver un désir de l'agent B pour un moment futur, si possible un désir qui peut être satisfait par une action (service) que A peut effectuer mais que B ne peut pas effectuer.

- ✓ **Appels au propre intérêt** : l'agent A croit que la conclusion d'un accord sur l'objet de négociation est dans l'intérêt de B et essaye de convaincre B de cela.

Pré-conditions : l'agent A doit trouver (ou déduire) un des désirs de B qui sera satisfait si l'agent B a l'objet de négociation ou A doit trouver un autre objet de négociation ON^f qui a été offert auparavant sur le marché et démontrer que l'objet de négociation proposé est plus intéressant que ON^f .

- ✓ **Appel à une pratique fréquente** : l'agent A croit que B a refusé la proposition parce que B croit que la proposition contredit un de ses buts. Dans ce cas, l'agent A donne à B l'exemple d'une pratique fréquente qui démontre que l'acceptation de la proposition ne contredit pas le but de B.

Pré-condition : l'agent A doit trouver un autre agent ayant le même but que B, qui a été déjà d'accord avec une proposition semblable et qui a vu que l'accord n'a pas nui à ses buts.

- ✓ **Menace** : le négociateur menace de refuser à faire/offrir quelque chose à B ou il menace de faire quelque chose qui contredit les désirs de B.

Pré-conditions : l'agent A doit trouver un des désirs de B directement satisfiable par un objet de négociation que l'agent A peut offrir ou A doit trouver une action qui est contradictoire avec ce qu'il croit être un des désirs de B.

2.2.4. Les autres formes de négociation

Il y a d'autres formes de négociation moins connues du grand public, ces négociations sont tout aussi nombreuses et variées. Parmi ces négociations, on trouve *le take it or leave it offer*, *les négociations multi attributs*, *les négociations multi-niveaux*, *les négociations combinées*.

- ***Le take it or leave it offer***

Cette forme de négociation est très primaire, puisqu'elle consiste à formuler une proposition qui est à prendre ou à laisser par le ou les participants. Cette négociation se déroule sur un seul tour, sans contre-proposition ni renégociation. C'est celle que l'on rencontre tous les jours pour acheter son pain, par exemple. Elle ressemble aux offres scellées au meilleur et au second meilleur prix, la différence vient du fait que dans les enchères, le prix est proposé par les acheteurs, tandis que dans le *take it or leave it offer* c'est le vendeur qui propose un prix ferme. Le protocole est donc très simple : le vendeur propose sa ressource (bien, service, etc.) à un prix ferme à un acheteur qui soit accepte, soit refuse. S'il accepte, l'acheteur paie le prix pour obtenir la ressource. Le take it or leave it offer ne présente pas pour beaucoup de chercheurs une véritable négociation, et nous sommes d'accord avec eux.

- ***Les négociations multi-attributs***

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

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

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

Où la négociation comprend n attributs, w_i le poids de l' i ème attribut, u_i la fonction d'utilité d' i ème attribut et σ_j^i la valeur de l' i ème attribut dans l'offre σ_j . Le but est de maximiser la valeur de cette fonction d'utilité.

- ***Les négociations multi-niveaux***

C'est un type de négociation présenté par Meyer et ses collègues en 1988 où le contrat se décompose en plusieurs "*sous contrats*", dépendants les uns des autres. La négociation s'effectue séquentiellement pour chaque sous-contrat, la réussite globale est atteinte lorsque tous les sous-contrats ont été négociés avec succès. Lorsqu'un sous-contrat est en cours de négociation et qu'aucune solution n'est possible, on effectue un backtrack pour la négociation du sous-contrat précédent. Par exemple, pour la prise de rendez-vous, on peut décomposer le contrat en recherche d'un jour, puis d'une heure dans ce jour.

Conry et ses collègues utilisent la négociation multi niveaux pour la satisfaction de contraintes distribuées et plus particulièrement pour le cas d'agents ayant des buts locaux qui nécessitent l'obtention de ressources communes.

La négociation est alors utilisée pour que tous les agents puissent accomplir leurs buts locaux. Ils utilisent également la négociation multi-niveaux pour la restauration de canaux de communication dans un système de communications complexe.

- ***Les négociations combinées***

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

Chaque forme de négociation possède son propre protocole, qui définit le déroulement du processus de négociation, c'est à dire les actes de langage utilisés et leur séquençement [15], alors nous allons maintenant passer à la définition de ces protocoles de négociation et donner les différents types de ces protocoles avec bien sur des exemples.

3. Les protocoles de négociation

3.1. Définition d'un protocole de négociation

Nous trouvons que la définition d'un protocole de négociation n'est pas très loin à celle d'un protocole d'interaction (que nous avons déjà fait dans le chapitre précédent), seulement le point différent est que la négociation exige l'apparition d'un conflit entre les agents:

Alors nous adoptons la définition suivante : « Chaque forme de négociation possède son propre protocole, qui définit le déroulement du processus de négociation, c'est-à-dire les actes de langage utilisés et leur séquençement » [34].

3.2. Classification des protocoles de négociation

Cette section présente quelques propositions de classification des protocoles de négociation [19].

3.2.1 Première classe : Les protocoles compétitifs vs coopératifs

Selon P. Maes, il identifie deux classes de protocoles.

- ✓ La première classe regroupe les protocoles distributifs qui se basent sur le principe du partage des gains entre les participants. Chacun des participants cherche à maximiser sa part de profit ; la réussite de l'acheteur signifie la défaite du vendeur, et vice versa. Dans la théorie de jeu, on les appelle des jeux à somme nulle. P. Maes les considère comme des protocoles compétitifs, car c'est l'esprit qui domine dans ce genre de protocoles. Les protocoles de vente aux enchères sont considérés comme des protocoles compétitifs.
- ✓ La deuxième classe regroupe les protocoles intégratifs, dont l'objectif de ses participants est d'essayer d'élargir l'espace d'entente, et de chercher de nouvelles solutions qui maximisent les profits mutuels des différents participants. Dans la théorie de jeu, on les appelle des jeux à somme non nulle, ils sont considérés comme des protocoles coopératifs en intégrant plusieurs paramètres de négociation.

3.2.2. Deuxième classification : Les protocoles unidirectionnels vs bidirectionnels

- ✓ Dans le protocole de négociation unidirectionnel, on se limite à l'acceptation ou le rejet d'une proposition. On trouve dans cette catégorie les protocoles de vente aux enchères et le protocole Contract Net.
- ✓ Le protocole de négociation bidirectionnel, dont le scénario est plus complexe, se base sur un échange de messages (offre et contre-offre). Ce type de protocoles offre plus de flexibilité à la négociation soit en changeant la valeur d'un ou de plusieurs attributs négociables, via une contre-proposition, soit en changeant la structure de la proposition, en ajoutant par exemple un nouveaux attribut négociable.

3.2.3. Autres classifications

Autres classifications des protocoles d'interaction ont été proposées :

- ❖ Wurman et al, proposent une classification basée sur les critères suivants :
 - Simple ou double* : un protocole est dit simple si le type des enchérisseurs est soit des agents acheteurs ou soit des agents vendeurs. Dans la double enchère on trouve des agents acheteurs et des agents vendeurs qui participent à la même enchère.
 - Privé ou public* : dans le premier cas, les propositions soumises par les participants ne sont connues qu'après la clôture de l'enchère. Dans le deuxième cas, les propositions sont publiques, connues par tous les participants.
 - Conditionné ou inconditionné* : Un protocole est conditionné si les propositions sont régies par des règles, par exemple, le prix proposé doit être ascendant ou descendant.
- ❖ *London Classification* est un modèle de classification des systèmes de négociation, défini à l'issue du workshop *Negotiations in Electronic Markets*. Il traite l'aspect participant, l'aspect produit, l'aspect décisionnel et l'aspect processus de négociation. Ce dernier aspect est défini par un ensemble d'attributs, parmi lesquels :
 1. Le type de l'agent initiateur de la négociation qui peut être l'agent vendeur ou l'agent acheteur.
 2. Le nombre de phases : Un protocole est dit à simple phase si les règles de négociations ne changent pas tout au long du processus de négociation, dans le cas contraire, on le considère comme un protocole multi-phases.
 3. L'ordonnancement dans la négociation multi-attributs peut être en une seule étape si tous les attributs sont négociés en même temps ou multi-étapes si on négocie chaque attribut à part.

4. La publication des offres ou des propositions, si la proposition est entendue par tous les participants le protocole est dit public (ou *open cry* pour les protocoles de ventes aux enchères). Dans le cas contraire on l'appelle protocole privé ou *sealed bid*.
5. Le prix que paie le gagnant dans l'enchère dépend des règles du protocole utilisé : dans les protocoles discriminatifs le gagnant paie le prix de la proposition qu'il a soumis (par exemple l'enchère anglaise). Dans les protocoles non discriminatifs le gagnant paie un prix plus bas que celui qu'il avait proposé. Par exemple, dans le protocole Vickrey le gagnant paie le prix de la deuxième proposition.
6. Le désistement d'un participant sur un engagement qu'il avait pris au cours d'une négociation peut être permis ou non par le protocole de négociation.

❖ Sandholm aussi identifie cinq types de protocoles de négociation : les systèmes de vote, les enchères, marchandage, systèmes contractuels et la formation de coalition.

3.3. Quelques protocoles de négociation

3.3.1. Le protocole Contract Net

Le Protocole *Contract Net* [28] proposé par Smith en 1980 est un mécanisme de négociation entre deux types d'agents : contractant et gestionnaire. Le Contract Net permet à un gestionnaire, suite à quelques échanges avec un groupe d'agents, de retenir les services d'un agent appelé contractant pour l'exécution d'une tâche (contrat). Ce protocole est qualifié de type 'sélection mutuelle' puisque, pour 'signer un contrat', l'agent choisi doit s'engager envers le gestionnaire pour l'exécution de la tâche et le gestionnaire ne sélectionne que l'agent ayant fourni la proposition la plus avantageuse. La version originale du protocole comporte trois étapes principales :

1. L'appel d'offre
2. La soumission de proposition
3. L'attribution de contrat.

L'algorithme 3.1 présente le protocole *Contract Net*. Sa description détaillée en sept étapes, regroupe les traitements du gestionnaire et des contractants, ainsi que les échanges entre eux.

Pour appliquer ce protocole, le concepteur doit détailler le contenu des messages échangés, le temps d'expiration ainsi que les fonctions « évalue-annonce » et « évalue-soumission ».

Etant donné une tâche, un gestionnaire, un groupe de (n-1) soumissionnaire :

1. le gestionnaire envoie un message de type « annonce- tâche » à un groupe d'agents (ou fait un 'broadcast').
2. Chaque agent évalue la tâche annoncée à l'aide d'une fonction locale 'évalue- annonce'.
3. L'évaluation précédente permet à certains agents de soumettre une proposition à l'aide d'une 'soumission- tâche' au gestionnaire.
4. Si une proposition est jugée satisfaisante (l'aide de la fonction 'évalue- soumission'), alors le gestionnaire envoie un message de type 'acceptation' à celui dont la proposition est retenue. Il envoie également un message de type 'refus' aux autres agents dont les propositions n'ont pas été retenues.
5. Le gestionnaire peut mettre fin à la période d'acceptation de proposition si le temps d'expiration est dépassé.
6. C'est aucune proposition n'a été retenue, alors le gestionnaire fait parvenir à tous les agents non retenus un message de type 'refus' pour indiquer le rejet de chacune des propositions.
7. Il peut alors se retirer de la négociation, retenir la proposition la plus acceptable redémarrer un nouvel appel d'offre (nouveau 'annonce- tâche') ou prolonger le temps d'expiration de la période d'acceptation de proposition.
8. L'agent ayant obtenu le contrat, remet un rapport d'exécution lorsque la tâche est complétée.

Figure 2.1 : Algorithme du Contract Net

3.3.2. Le protocole Contract Net étendu

Le modèle Contract Net étendu est basé sur une décomposition décentralisée de tâches. L'agent gestionnaire fractionne une tâche en plusieurs sous-tâches et les annonce à un agent ou à un groupe d'agents soumissionnaires. Mais contrairement au *Contract Net* original où une tâche ne peut être qu'accordée ou rejetée, dans le *Contract Net étendu*, une tâche peut être accordée temporairement, rejetée temporairement, accordée définitivement et rejetée définitivement par les agents.

Dans le Contract Net étendu, l'agent gestionnaire annonce une commande à un groupe d'agents. Il sélectionne le meilleur agent soumissionnaire à qui il envoie un accord temporaire. Tous les autres agents reçoivent un rejet temporaire. Si la meilleure soumission ne couvre pas la totalité de la commande, la partie restante de la commande est ré-annoncée par le gestionnaire jusqu'à couvrir sa totalité. Au terme de ce processus les accords et des rejets temporaires sont transformés par des accords et des rejets définitifs. Quand un agent reçoit un accord temporaire, il crée une copie de son ancien plan, ainsi il sera en mesure de le récupérer en cas de rejet définitif.

3.3.3. Le protocole Kasbah

Kasbah [34] a été développé au MIT Media Lab par Pattie Maes en 1996. C'est un système où les utilisateurs créent des agents pour négocier la vente et l'achat de biens pour leur compte sur Internet. Ces biens sont classifiés, reprenant ainsi l'idée des petites annonces classées par type.

Lors de la création d'un agent, pour la vente ou l'achat, l'utilisateur spécifie le type de bien à négocier, la date à laquelle il souhaite que la transaction soit effectuée, le prix désiré, le plus petit (resp. grand) prix acceptable et la stratégie de négociation à choisir parmi les 3 proposées qui correspondent aux fonctions linéaire, quadratique et exponentielle pour le calcul de l'évolution du prix selon le temps. L'utilisateur précise également si l'agent doit demander son accord avant de conclure la vente et s'il veut être averti par mail lorsqu'un accord est trouvé. Une fois l'accord atteint, la transaction physique peut avoir lieu, ce qui doit être réalisé par les agents humains.

Du point de vue de l'implémentation, Kasbah met en relation les agents ayant des buts communs, les communications entre agents se font de 1 vers 1. Le fonctionnement en parallèle des agents est simulé en accordant un time-slice à chacun à tour de rôle. Durant ce time-slice, l'agent détermine le prix courant désiré, décide avec quel agent communiquer et enfin communique avec celui-ci. Les agents communiquent via des actes de langages spécifiques à Kasbah. Si une évolution vers KQML a été prévue, ce système n'en reste pas moins axé sur l'e-commerce, qui est un aspect très spécifique de la négociation.

3.3.4. Le protocole AuctionBot

C'est un serveur d'enchères expérimental développé et opérationnel au laboratoire d'intelligence artificielle de l'université du Michigan par Michael P. Wellman et Peter R en

1998. Wurman [34]. Son but est de permettre à n'importe quel internaute de participer aux enchères sur le net.

AuctionBot est un framework utile à la fois pour le commerce et pour la recherche au sens où il propose une large variété de types d'enchères et une API pour créer ses propres agents qui participeront à la place de marché d'*AuctionBot*. Son architecture est asynchrone, il stocke les enchères dans une base de données et il peut en gérer plusieurs simultanément. Afin de participer aux enchères, il faut s'enregistrer. Les utilisateurs humains peuvent consulter leurs comptes via une page web ou choisir d'être avertis de l'avancement des enchères par mail. Comme *AuctionBot* répertorie les enchères proposées dans un catalogue organisé de façon hiérarchique, un vendeur peut placer son enchère n'importe où dans le catalogue existant ou étendre celui-ci. Il peut également choisir de mettre son enchère dans le catalogue public ou de la proposer à un groupe privé.

Du point de vue de l'implémentation, les agents placent les enchères dans la base de données, tandis qu'*AuctionBot* collecte les enchères, détermine une allocation d'après un ensemble de règles d'enchères bien défini et avertit les participants. En revanche, il n'exécute pas les transactions, il n'impose pas les échanges ni ne vérifie la crédibilité des participants. Ce n'est donc pas l'agent initiateur du contrat qui gère l'enchère mais un programme d'*AuctionBot*. *AuctionBot* ne propose un support que pour les mécanismes d'enchères, tandis que notre objectif est de fournir une plateforme générique de négociation qui permet d'effectuer des enchères mais aussi d'autres types de négociation tels que la prise de rendez-vous, le choix d'un restaurant, etc. De plus, nous laissons aux agents l'autonomie de la gestion de la négociation, chaque agent initiateur (vendeur dans le cas des enchères) gère le déroulement de sa négociation, il n'y a pas de centralisateur d'offres de vente et d'achat qui effectue l'appariement selon les règles d'enchères définies comme dans *AuctionBot*.

Le serveur d'*AuctionBot* est utilisé pour TAC (Trading Agent Competition) une compétition qui met en concurrence des agents chargés de négocier les composants d'un voyage (vol aller, nuits d'hôtel, sorties et vol retour) pour un ensemble de 8 clients, créée elle aussi par Michael P. Wellman et Peter R. Wurman. Les vols, les nuits d'hôtel et les sorties sont négociés par des enchères de différents types. Chaque client a des préférences pour le jour d'arrivée, le jour de départ, le type d'hôtel (luxueux ou standard) et les sorties possibles. Chaque agent est ensuite noté en fonction de ses dépenses, de ses revenus et de la satisfaction de ses clients.

4. Conclusion

La négociation joue un rôle fondamental dans les cas les personnes doivent résoudre des conflits qui pourraient mettre en péril des comportements coopératifs. De ce fait nous avons essayé au long de ce chapitre de donner la plus part des principes qui permettent de bien comprendre la négociation entre les agents et sans oublier les concepts sur lesquels elle est basée. Nous pouvons dire qu'avec la négociation nous pouvons résoudre des conflits sans toucher l'aspect d'autonomie des agents car les agents sont capables de continuer ou arrêter la négociation.

Chapitre 3

Les approches de spécifications de négociation

1. Introduction

En analysant le domaine de la négociation dans les systèmes Multi-agents, nous pouvons conclure qu'il y a trois thèmes de recherches:

1. *Les protocoles de négociation* : c'est l'ensemble des règles qui gouvernent l'interaction. Cela couvre les types de participants autorisés, les états de la négociation, les événements causant le changement d'état de la négociation et les actions valides pour un participant selon l'état courant de la négociation.
2. *Les objectifs de la négociation* : c'est l'intervalle de critères pour lesquels une solution doit être atteinte. L'objectif peut contenir un seul critère (comme un prix), mais il peut également contenir des centaines de critères (relatifs au prix, à la qualité, au temps, aux pénalités, aux termes et conditions, etc.).
3. *Les modèles décisionnels des agents* : c'est le moyen par lequel un agent atteint ses objectifs de négociation tout en suivant les règles de la négociation. La sophistication du modèle tout comme l'intervalle des décisions qui doivent être prises, sont influencés par le protocole utilisé, par la nature de l'objectif de la négociation et par l'intervalle d'opérations qui peuvent lui être appliquées.

Pour notre travail, nous pouvons dire qu'il est consacré aux protocoles de négociation, et selon Mazouzi dans [23] un protocole de négociation est un type de protocoles d'interaction. De ce fait nous allons modéliser un protocole de négociation en utilisant un formalisme formel qui permet à la fois la spécification et la validation de ce protocole.

Ce chapitre contient deux sections ; la première présente un état de l'art sur les approches de spécification existantes et le deuxième donne une synthèse générale sur ces approches.

2. Les approches de spécification de négociation

Nous avons parlé dès le début que notre principal objectif est de trouver une approche de spécification formelle pour la négociation automatique entre agents, mais pour que la vision soit plus claire nous allons citer et expliquer toutes les approches de spécification de négociation existantes sur terrain pas seulement les approches formelles ; et nous allons beaucoup plus détailler la partie formelle. Nous avons noté aussi que pour notre travail le plus intéressant c'est le protocole de négociation ni les objectifs de la négociation ni les modèles décisionnels des agents nous intéressent.

Selon M. MORGE [25] le comportement d'un protocole décrit la séquence des messages qui peuvent être échangés, notre objectif maintenant est de modéliser le comportement d'un protocole de négociation. Ce comportement est spécifié par les quatre approches de spécification suivantes :

1. Techniques semi-formelles de spécification
2. Modèles de spécification formelles pour les SMA
3. Les langages formels de description
4. La logique

2.1 Techniques semi-formelles de spécification (UML)

Le langage de modélisation unifié UML (Unified Modelling Language) est une notation universelle pour la modélisation d'objets. Toutefois, il ne permet pas de modéliser un système multi-agents car il y a une grande différence entre l'objet et l'agent : les agents sont en activité permanente car ils agissent selon des buts qui émergent d'eux même. C'est dans cette perspective que le langage de modélisation unifié à base d'agents AUML (Agent-based Unified Modelling Language) a été proposé.

Le langage AUML étend la notion de diagramme de séquence utilisé dans UML 2.0 à celle de diagramme d'interaction qui permet de représenter les protocoles d'interaction (protocoles de négociation) et leurs instances. Un diagramme d'interaction est un graphique constitué d'un cadre qui permet d'identifier le protocole (nom, paramètres, etc. . .). Ce cadre délimite la représentation graphique du flux de messages entre les différents rôles. Un diagramme d'interaction est une représentation graphique en deux dimensions. Le temps est représenté selon un axe vertical. Les différents rôles joués par les participants sont représentés selon un axe horizontal.

Puisque c'est le diagramme d'interaction qui permet de représenter les protocoles d'interaction dans les SMA, nous allons maintenant résumer les différences entre ce diagramme et le diagramme de séquence de l'UML dans les points suivants :

- En UML, une ligne de vie dans un diagramme de séquence est associée à une instance unique. En AUML, une ligne de vie dans un diagramme d'interaction est associée à un rôle. Une ligne de vie est représentée par un trait vertical pointillé surmonté d'un rectangle qui contient le nom du rôle associé.
- Les messages FIPA-ACL sont représentés dans un diagramme d'interaction à l'aide d'une flèche qui lie la ligne de vie du rôle joué par l'agent émetteur et la (ou les) ligne(s) de vie du (ou des) rôle(s) joué(s) par les agents destinataires.
- Dans le diagramme d'interaction d'un protocole, les messages ne sont pas totalement spécifiés. Seule une partie des champs est informée. En d'autres termes, les messages représentés dans un diagramme d'interaction sont plus ou moins génériques. Dans la plupart des cas, les agents sont libres d'en choisir le contenu. Étant donné que les lignes de vie sont associées à des rôles, la cardinalité des messages doit être précisée. La cardinalité est une expression qui indique le nombre d'agents qui émettent et/ou reçoivent le type de message envisagé. Les agents sont libres de choisir parmi une liste de messages génériques autorisés. Ce choix peut être représenté de manière compacte et concise à l'aide d'opérateur. Par exemple, l'opérateur *or* dont le symbole est 3 permet de sélectionner au moins un des messages génériques proposés.

2.2. Modèles de spécification formelles pour les SMA

2.2.1. Les automates à états finis (AEF)

Grâce à sa simplicité le formalisme des automates à états finis peut être considéré comme l'un des premiers et l'un des plus employés pour représenter un protocole d'interaction ou de communication. La figure 3.1 montre que l'automate se représente sous la forme d'un graphe orienté dont les nœuds sont les états de l'automate et les arcs ses transitions. Sur les arcs, on indique à la fois les événements qui le font passer d'un état à un autre et les actions qui doivent être entreprises lors de cette transition ; alors il s'agit d'une succession d'état ou le changement d'un état à un autre correspond à l'émission ou la réception d'un message.

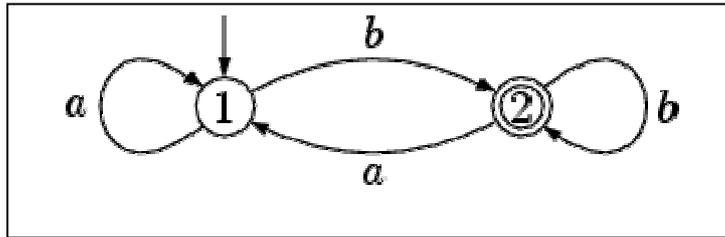


Figure 3.1 : Exemple d'un automate d'états

La définition de base d'un automate à état fini est donnée formellement comme un sextuplet [11]:

$$A = \langle E, S, C, \psi, \phi, s_0 \rangle$$

- ✓ E est un ensemble fini d'évènements (décrits par des mots comprenant k caractères)
- ✓ S est un ensemble fini d'états,
- ✓ C est l'ensemble des commandes égal à B^p (B est le domaine 0,1),
- ✓ s_0 est un élément particulier de S que l'on appelle état initial,
- ✓ ψ est la fonction de transitions, c'est-à-dire une fonction de la forme

$$\psi: S \times E \rightarrow S$$

- ✓ Et ϕ est la fonction d'activité, c'est-à-dire la fonction qui lance une commande et applique une opération lors d'une transition:

$$\phi: S \times E \rightarrow C$$

La sémantique pour le franchissement d'une transition est la suivante : une transition $\psi(S_1, e)=S_2$ est dite tirable, si étant dans l'état d'entrée S_1 , il y a réception (ou émission) d'un événement e , alors il est possible de tirer la transition et de passer dans l'état de sortie $\psi(S_1, e) = S_2$.

Spécifier un système par automate d'états finis est fait par une composition d'automates décrivant le comportement local de chacun de ses composants c'est à dire Chaque entité communicante est définie par un automate. Les automates communiquent entre eux en échangeant des messages. Il est de même possible que les entités communicantes utilisent le même protocole.

Comme exemple de spécification d'un système, l'ensemble des comportements possibles de ce système est l'ensemble des mots finis et infinis reconnu par l'automate modélisant ce dernier.

2.2.2. Les réseaux de Petri (RdP)

L'un des modèles les plus formels mais aussi les plus développés pour représenter des systèmes multiprocessus est celui des réseaux de Petri. Ces derniers peuvent être considérés comme des outils de modélisation de processus essentiellement asynchrones fondés sur une représentation à la fois graphique et mathématique. L'aspect graphique permet de visualiser simplement les processus et de faciliter la conception et la communication de modèles de comportement, tandis que la formulation mathématique assure à ce formalisme des fondements rigoureux et solides.

Un réseau de Petri est défini comme un graphe orienté comprenant deux sortes de nœuds: des places et des transitions. Comme le montre la figure 3.2, ce graphe est constitué de telle sorte que les arcs du graphe ne peuvent relier que des places aux transitions ou des transitions aux places. On représente graphiquement les places par des cercles et les transitions par des barres.

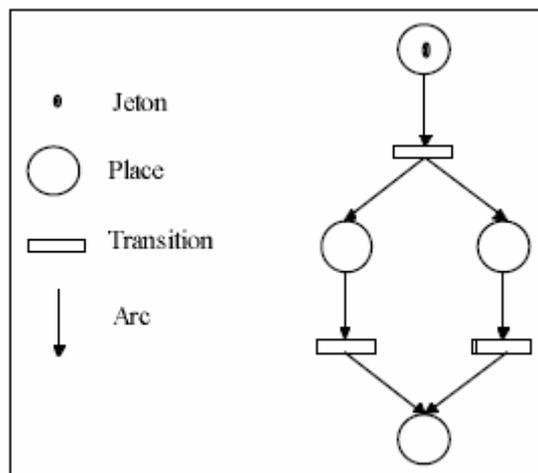


Figure 3.2 : Exemple de réseau de Petri ordinaire

Nous donnons maintenant la définition formelle de base d'un réseau de Petri [23], notons que cette définition concerne tous les réseaux de Pétri ordinaires ou classiques, c'est à dire ceux qui sont à la base de toute la famille des réseaux de Petri:

Un réseau de Petri est défini par $R = (B, M, E, Pre, Post)$, où :

- ✓ B est l'ensemble des conditions (places),
- ✓ M est un sous-ensemble non vide de B (le marquage initial),
- ✓ E est l'ensemble des événements (transitions),
- ✓ $Pre : E \rightarrow \rho(B)$ est une application des pré-conditions telle que pour tout $e \in E, Pre(e)$ est non vide,
- ✓ $Post : E \rightarrow \rho(B)$ est une application des post-conditions telle que pour tout $e \in E, Post(e)$ est non vide,

Ils sont utilisés dans de nombreux domaines où il est nécessaire de représenter des mécanismes qui s'exécutent en parallèle: commandes de procédés industriels, protocoles de communications dans des réseaux, sémantique de programmes et de langages, prototypage de systèmes parallèles, etc. Enfin, il existe non pas un modèle de réseau de Pétri, mais toute une famille permettant de modéliser des mécanismes aussi complexes que cela s'avère nécessaire.

❖ Comment franchir une transition ?

Les réseaux de Petri, à l'encontre des automates, permettent la représentation des aspects dynamiques des processus en déplaçant des marques de places en places. Dans les réseaux de Petri ordinaires, ceux qui sont à la base de toute la famille des réseaux de Petri, ces marques ne sont que des jetons, c'est-à-dire des éléments simples et indiscernables, alors que dans d'autres types de réseaux, tels que les réseaux colorés que nous verrons plus loin, ces marques sont des structures nettement plus complexes qui s'apparentent à des "records" à la Pascal. Les marques évoluent de place en place en suivant ce que l'on appelle des règles de franchissement.

Alors, on dit qu'une transition T peut être franchie, ou encore qu'une transition est validée, si toutes ses places d'entrées P pour lesquelles il existe un arc orienté de P vers T possèdent une marque. Parmi toutes les places validées on en prend une au hasard qui sera effectivement déclenchée, et on dira que les marques franchissent cette transition. Franchir une transition revient à enlever une marque sur chaque place d'entrée et à en ajouter une sur chaque place de sortie (les places P pour lesquelles il existe un arc de T vers P).

❖ Exemple du franchissement d'une transition

La figure 3.3 montre un exemple de franchissement de transition [11].

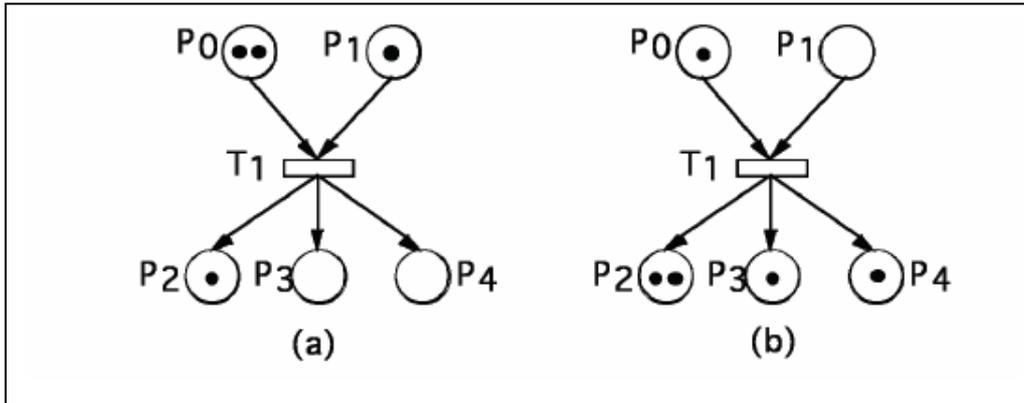


Figure 3.3: Exemple de franchissement de transition, avant (a) et après (b)

Lorsqu'il y a plusieurs transitions possibles en sortie d'une place, la règle de tirage au sort permet de déterminer laquelle des transitions doit être franchie, comme le montre la figure 3.4.

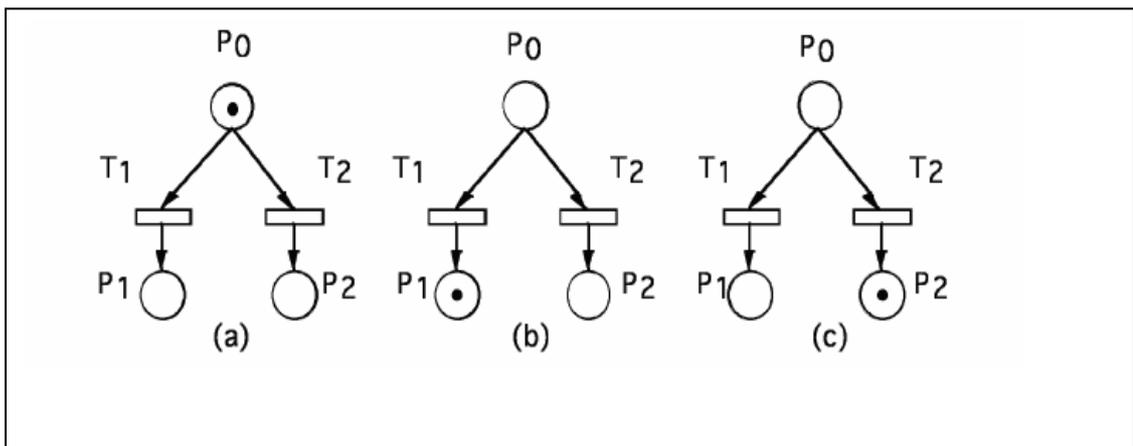


Figure 3.4 : Les transitions T_1 et T_2 étant validées en (a), le tirage au sort des transitions à franchir produit le marquage (b) ou (c)

On appelle marquage une distribution de jetons sur les places, et marquage initial la distribution initiale de jetons dans le réseau.

Enfin, il existe non pas un modèle de réseau de Petri, mais toute une famille permettant de modéliser des mécanismes aussi complexes que cela s'avère nécessaire : RdP colorés, RdP à prédicats, RdP temporels, RdP temporisés, RdP stochastiques, RdP objets, RdP hiérarchiques, RdP récursifs. Par exemple si nous souhaitons distinguer entre les jetons dans un RdP, il est possible d'utiliser des réseaux de Petri colorés. Dans ce cas, nous pouvons associer une évaluation dans un domaine fini, que l'on appelle « couleur » du jeton.

2.3. Les langages formels de description

2.3.1 Langage Z

Le langage Z a été développé à l'Université d'Oxford à la suite des travaux de Jean René Abrial. Selon Davis dans [9], Le langage Z est un langage de spécifications formelles sert à la représentation abstraite de certains aspects d'un produit logiciel, principalement quant à la manipulation des données (*data manipulation design*). Ce que nous pouvons comprendre de la définition de Davis c'est que le langage Z permet de donner une représentation parfaitement découplée de toute implémentation et c'est exactement ce qu'il veut dire par une représentation abstraite.

Dans [31] Spivey a défini le langage Z comme des Méthodes formelles Techniques, basées sur les mathématiques, pouvant être utilisées à tous les stades du développement logiciel. L'idée de Spivey c'est que le langage Z a une sémantique formelle qui permet de rendre précises des idées de développement auparavant vagues ou simplement intuitives et ça en réduisant toute expression du langage à une expression logique.

Au point de vue sémantique, le Z peut être subdivisé en trois parties [14] :

1. Un langage mathématique utilisant propositions, logique prédicative, ensembles et relations.
2. Un langage schématique représentant des manipulations algorithmiques de données ou des objets.
3. Une théorie du raffinement entre des types abstraits de données.

En ce qui nous concerne ce n'est pas la notation mathématique mais celle schématique car selon Spivey dans [31] c'est la notation schématique qui permet de faire le lien entre les mathématiques pures et une abstraction informatique, n'est basée sur aucune formalisation. Alors pour spécifier un système avec le langage Z nous devons utiliser les définitions et les schémas suivants [23]:

- Définition des ensembles de base.

- Définition des types.
- Schéma d'état (espace d'état).
- Schéma d'initialisation.
- Schéma d'opération.

A partir des éléments constitutants de la spécification Z décrites au dessus, nous concluons qu'elle est structurée en schémas ; cependant nous pouvons tirer deux composantes principales d'une spécification formelle en Z qui sont :

- a) Des déclarations globales : noms de types, d'ensembles, abréviations (équivalences syntaxiques et définitions génériques).
- b) Un ou plusieurs types de schémas.

Nous avons vu que la seconde composante d'une spécification formelle en langage Z c'est le schéma. Les schémas viennent toujours en trois parties [14] :

- ✓ *Un nom* qui identifie le schéma en portée globale sur la spécification.
- ✓ *Une partie déclarative* (peut être vide) qui définit les différentes données membres du schéma. Les données définies pour la première fois sont de portée locale. Son contenu s'appelle signature.
- ✓ *Une partie prédicative* (peut être vide) qui énumère les contraintes que doivent respecter les données accessibles dans la portée courante, localement ou globalement, pour que le schéma soit suivi.

Généralement nous pouvons distinguer aussi deux types de schémas qui sont :

- 1) *Orientés données* : ils regroupent ou décrivent alors un ensemble de données. Ils donnent possiblement leur état initial.
- 2) *Orientés opérations* : ils donnent alors des manipulations abstraites des données définies ailleurs, possiblement dans des schémas orientés données.

Finalement nous pouvons dire que le langage Z est utilisé pour composer et structurer une série de descriptions mathématiques c'est-à-dire qu'il permet de grouper ensemble des informations (données ou opérations), de les encapsuler et de les nommer pour réemploi.

2.3.2. Langage SDL

Le langage SDL (Specification and Description Language) est un langage formel défini par la recommandation Z-100 de l'Union Internationale des Télécommunications

(UIT/ITU). Ce langage est destiné à la modélisation de systèmes complexes, événementiels, interactifs et temps réel pouvant comprendre plusieurs activités simultanées et communiquant par le biais de signaux discrets. Le langage SDL existe en deux versions l'une graphique et l'autre textuelle. Il est basé sur le principe suivant (voir fig3.5) [2]:

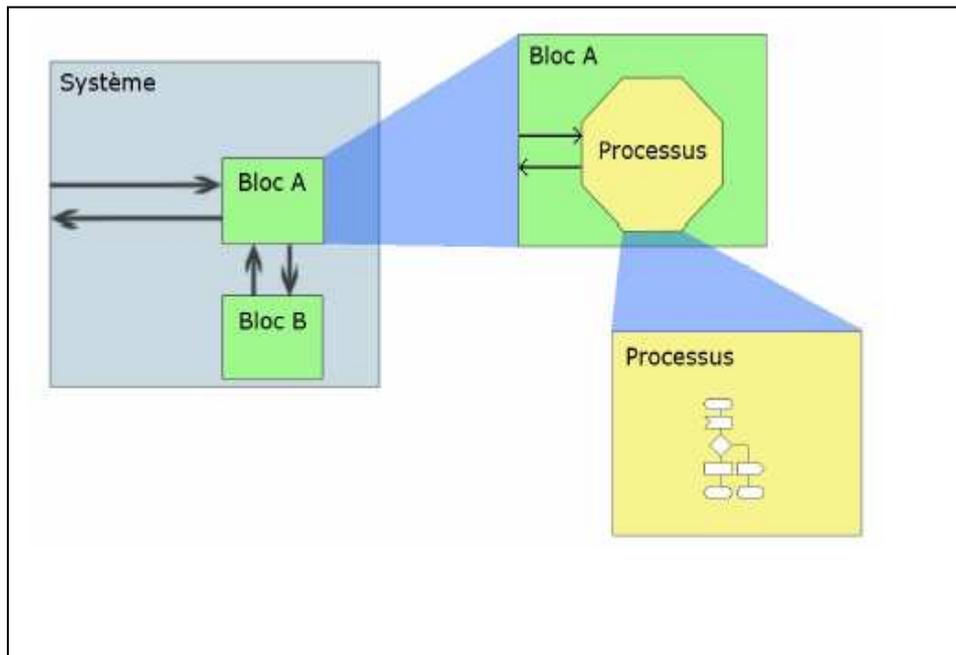


Figure 3.5 : La hiérarchie imbriquée de SDL

❖ **Structuration**

En SDL, dans un premier niveau, le système modélisé est composé de blocs liés les uns aux autres et à la frontière du système au moyen de canaux. Les blocs sont composés de sous-bloc et de processus. Un processus, au sens SDL, est une machine à états finis communicante. Les instances de processus sont créées statiquement ou dynamiquement. Chaque instance possède son identifiant ou PID (Processus Identifier).

❖ **Communication**

Les processus communiquent via des signaux de manière asynchrone (pas d'attente de l'émetteur). Les signaux sont véhiculés par les canaux. Quand un signal est émis, il est transmis, via son canal, au bloc de destination. Sur un canal, l'ordre d'émission entre signaux est conservé. Par contre, si deux signaux arrivent en même temps ils sont traités arbitrairement. Quand un signal arrive sur un processus, il est mis en attente dans une file gérée selon le principe FIFO (premier arrivé/premier servi). Si le

destinataire du signal n'est pas explicité via son PID, le signal est consommé par la première des instances du processus pouvant le consommer.

❖ **Comportement**

Un processus SDL est une machine à états caractérisée par une file d'attente FIFO, des données, des signaux entrants ou sortants et un ensemble d'états reliés par des transitions. Un état particulier du processus est l'état de départ où se place l'instance de processus à sa création. Une transition est caractérisée par une condition de tir et un corps. La fin de la transition aboutit à un nouvel état (pouvant être le même que l'état précédent). Une transition est tirée soit sur réception d'un signal, soit de manière aléatoire (transition spontanée). Ce dernier aspect est utile pour modéliser des systèmes non déterministes.

❖ **Les données**

La structuration des données se fait via des types (prédéfinis ou construits) déclarés au niveau des blocs ou du système, selon la syntaxe ADT (Abstract Data Types). Les données sont, elles, exclusivement déclarées au sein d'un processus. Les données et les types ne sont pas visibles à l'extérieur de leur zone de déclaration. L'échange de données entre processus, se fait soit par signal soit par partage des données

❖ **Concurrence, protection et synchronisation des données**

Les instances de processus s'exécutent en parallèle. Pour une instance, une seule transition est exécutée à la fois : il n'existe pas de concurrence intra-« instance de processus » (c'est l'hypothèse RTC « Run To Completion »). Par contre, si deux transitions (de 2 instances de processus) sont tirables au même instant, aucune relation d'ordre n'est donnée a priori. En SDL, les données étant locales aux processus, elles sont inaccessibles de l'extérieur. Comme leur modification a lieu dans une transition et que cette dernière ne peut être interrompue, les données sont implicitement protégées car manipulées en exclusion mutuelle. Enfin, les opérations de partage se font en mode atomique pour assurer la protection des données.

2.3.3. Langage Estelle

Estelle (Extended State Transition Language) est un formalisme défini par l'ISO (ISO 9074) qui s'est développé en concurrence avec SDL pour la description des protocoles de communication. Le langage Estelle peut être décrit comme fondé sur un modèle d'automate étendu avec le langage de programmation Pascal. Il inclut tous le langage Pascal, mais

l'encapsule dans des éléments qui en font un véritable langage pour l'expression de comportement parallèles. Trois caractéristiques principales sont à noter [8]:

1. Une spécification est composée de plusieurs modules. Estelle permet de bien spécifier les interfaces de chacun d'entre eux.
2. La description de comportement de chaque module est assez fine et précise, afin que le comportement de la spécification ne soit pas ambigu, ou que ses ambiguïtés (non déterminisme) soient explicites.
3. La notion de typage (définition de type, puis création d'exemplaires) du langage Pascal est étendue aux objets parallèles (comme les modules, canaux...), dont certains sont même paramétrables.

Nous pouvons résumer le principe de base du langage Estelle selon Courtia dans [8] comme suit :

❖ **La notion de module**

Le comportement d'un module (tel qu'il est perçu à travers ses points d'interaction) est donné par un modèle à transition d'état (automate). Un modèle à transition d'état est défini par un ensemble d'entrées, de sorties, d'états et de transition. Néanmoins, ce modèle est étendu pour lui donner la puissance et la concision d'un langage de programmation. L'état interne est prolongé par un vecteur de variables PASCAL. Dans le principe l'état principal (celui de l'automate d'état fini sous-jacent) détermine les phases du protocole décrit, c'est-à-dire l'aspect contrôle, le langage Estelle définit une syntaxe originale. Pour l'aspect donné, Estelle reprend intégralement les constructions Pascal.

❖ **L'interconnexion**

L'architecture du système est définie par un ensemble de modules communicants. Ces modules s'échangent des messages (interaction via des points) d'interaction (ou ports). Chacun de ces points est typé par un canal. La spécification d'un canal contient une énumération de toutes les interactions possibles sur les ports se référant à ce type. Pour chaque type d'interaction, on donne une liste de paramètres transportés par l'interaction et leurs types. Les concepts de canaux et points d'interaction servent à partitionner les échanges de signaux dans le système. A chaque point d'interaction est associée une file dans laquelle sont placées les interactions reçues. Toutes les files sont non bornées et de discipline FIFO. Deux modules peuvent être interconnectés en reliant deux points

d'interaction. Seules les interactions spécifiées par le canal associé au point d'interaction sont autorisées. Une liaison est à priori bidirectionnelle.

❖ **La structuration**

Une spécification est constituée de plusieurs modules. Chaque module est composé d'un en-tête et d'un corps. Le corps d'un module contient une partie d'initialisation et un ensemble de transitions. Il peut se suffire à lui-même ou être structuré en sous modules. Les exemplaires de sous modules sont créés uniquement à l'intérieur du module immédiatement englobant dans les parties initialisation ou transition. On parle de configuration statique lorsque la création n'a lieu que dans la partie initialisation. A un niveau donné de la hiérarchie statique, deux types de synchronisation entre modules peuvent être considérés, couvrant un nombre varié de problèmes. Il s'agit du mode asynchrone (La synchronisation ne se fait que par échange d'interaction) et du mode synchrone (les transitions tirables au même moment sont tirées simultanément). Le partage de variables est autorisé entre un module parent et ses enfants (dans la hiérarchie). Pour cette raison, une transition d'un module père ne peut s'exécuter simultanément avec une de celles de ses fils. Enfin un module peut créer dynamiquement (dans les actions de transition) des sous modules fils, puis les détruire. La partie d'initialisation d'un module est activée lors de sa création.

2.3.4. Le langage LOTOS

LOTOS est un langage parallèle qui s'inspire des algèbres de processus: le contrôle des programmes est décrit par des expressions algébriques appelées comportements. La synchronisation et la communication s'exécutent exclusivement par rendez-vous, sans partage de mémoire.

Comme la plupart des langages de programmation, LOTOS permet de définir et de manipuler des structures de données. Mais, à la différence des langages classiques, LOTOS utilise le formalisme des types abstraits algébriques (Abstract Data Types, ADT). Dans une large mesure, le choix des types abstraits pour la description des structures de données est conforme aux objectifs d'un langage de spécification [16]: les spécifications algébriques constituent un modèle mathématique exprimant les propriétés que doit vérifier toute réalisation, sans imposer de contraintes d'implémentation superflues les propriétés des données et des opérations sont complètement décrites. En choisissant les types abstraits,

LOTOS évite les difficultés rencontrées en Estelle où les données sont spécifiées au moyen des types du langage PASCAL, ce qui pose des problèmes bien connus.

❖ **Les spécifications LOTOS**

Selon Garavel dans [17], une spécification LOTOS est un texte ASCII qui regroupe un ensemble de définitions de processus (encadrées par les mots-clés `\process` et `\endproc`) et de définitions de types (délimitées par les mots-clés `\type` et `\endtype`).

LOTOS possède une structure de blocs imbriqués : chaque définition de processus peut contenir des définitions de processus ou de types qui lui sont locales ; en revanche une définition de type ne peut pas englober d'autres définitions de types ni de processus. Au plus haut niveau, une spécification LOTOS se comporte comme une définition de processus, bien que la syntaxe soit légèrement différente de celle des processus ordinaires (les mots-clés `\spécification` et `\endspec` sont utilisés).

Les définitions de processus décrivent le contrôle et les définitions de types décrivent les données. Cette dichotomie qui apparaît au niveau syntaxique rend bien compte du fait que le langage LOTOS se compose de deux parties tout à fait orthogonales qui seront, pour cette raison, présentées séparément : les structures de contrôle et les structures de données.

Les identificateurs utilisés dans une spécification LOTOS se repartissent en six classes (processus, portes, variables, types, sortes, opérations) qui seront définies tour à tour. Dans tous les exemples, les mots clés seront écrits en minuscules et les identificateurs en majuscules.

2.4. La logique

Seule la logique temporelle a fait l'objet d'une utilisation pour décrire les protocoles d'interactions. La logique temporelle est type de logique modale où la relation d'accessibilité entre les mondes est interprétée comme une relation temporelle. La temporalité de cette logique intervient par l'intermédiaire des opérateurs. La figure 3.6 donne un exemple de ces opérateurs :

$\circ\phi$	ϕ est vrai dans le prochain état
$\odot\phi$	il y avait un précédent état et dans celui-ci, ϕ était vrai
$\bullet\phi$	s'il y avait un précédent état alors ϕ y était vrai
$\diamond\phi$	ϕ sera vrai dans certains futurs états
$\blacklozenge\phi$	ϕ était vrai dans certains précédents états
$\square\phi$	ϕ sera vrai dans tous les futurs états
$\blacksquare\phi$	ϕ était vrai dans tous les précédents états

Figure 3.6 : Exemple des opérateurs de logique

La logique temporelle est utilisée lors de la validation du protocole d'interaction par le biais de la vérification de modèles. La vérification de modèle commence par générer un graphe des états accessibles pour le protocole considéré. Le concepteur définit ensuite la propriété à vérifier par l'intermédiaire d'une formule en logique temporelle. La vérification consiste alors à s'assurer que le graphe dispose de la propriété.

3. Synthèse

Nous avons essayé le long de ce chapitre de présenter les différents formalismes existants qui nous permettent de spécifier et décrire les interactions (négociation) dans un système multi-agents, notons que dans notre modélisation nous devons utiliser un formalisme formel. De ce fait dans cette synthèse nous allons focaliser notre effort beaucoup plus sur les formalismes formels :

- Modèles de spécification formelles pour les SMA
- Les langages formels de description

Cette synthèse se présente par une discussion des différents avantages et inconvénients de chaque formalisme. Absolument cette synthèse nous amène à découvrir qui est le formalisme le plus adéquat dans notre cas. Commençons par les Modèles de spécification formelles pour les SMA qui se présentent par les AEF et les RdP :

↳ Comparons les AEF au reste des formalismes, ils présentent le moyen le plus simple pour décrire les protocoles d'interaction (négociation) car ils sont capables de mémoriser l'état de l'agent ce qui permet de décrire facilement son comportement. Le deuxième avantage des AEF est qu'ils facilitent la tâche des concepteurs du protocole d'interaction car ils possèdent

une représentation graphique qui donne aux concepteurs une vue globale et évidente du protocole. Passons au troisième avantage des AEF qui concerne la validation du protocole. Nous pouvons noter que l'aspect de validation des protocoles présente le point le plus fort qui caractérise les AEF par rapport aux autres formalismes car la majorité des algorithmes de validation nécessitent d'abord une traduction du protocole dans un formalisme donné vers les AEF avant validation. Malgré tous les avantages que nous avons cités pour les AEF, ils présentent aussi des inconvénients surtout pour la modélisation de l'évolution d'un système multi-agents où le comportement de ce système s'avère plus complexe que les autres. D'une part nous avons dit que parmi les avantages des AEF est qu'ils permettent de décrire un protocole d'interaction de façon simple où chaque état de l'automate correspond à une étape de la conversation, cet avantage conduit à un grand problème ou inconvénient dont le nombre d'état est combinatoire avec le nombre d'agent impliqués dans l'interaction et puisque aussi ce nombre d'états d'automate doit être fini malgré qu'il peut être énorme, la capacité de calcul des AEF est limitée. Alors les AEF ne peuvent donc se souvenir d'une suite d'événements arbitrairement longue. D'autre part, les AEF ne peuvent décrire que des processus séquentiels, de ce fait, tous les calculs, toutes les actions parallèles, la synchronisation leur sont pratiquement interdits ; sachons que le parallélisme et la synchronisation sont des caractéristiques importantes dans un protocole de négociation, donc ce formalisme (AEF) n'est pas adapté lorsqu'on veut représenter un système d'interaction (négociation) malgré tous ses avantages décrits précédemment.

↳ Généralement les RdP possèdent presque les mêmes avantages des AEF et permettent aussi d'éviter les inconvénients de ces derniers. Au niveau de la conception du protocole : les RdP comme les AEF possèdent une représentation graphique qui facilite la tâche de conception. À l'encontre des AEF les RdP possèdent la notion de modularité par exemple dans les RdP hiérarchique où un RdP est constitué d'un ensemble des RdP. Les critères de synchronisation et parallélismes qui sont absents dans les AEF existent dans les RdP où nous trouvons que ces derniers ont été conçus initialement afin de permettre la synchronisation et la communication des processus concurrents. Maintenant au niveau de la validation de la conception de protocole : les RdP peuvent être facilement validés car il existe une conversion possible de protocole vers les AEF, de plus il existe un ensemble de propriétés qui peuvent être directement validés sur les RdP, comme la propriété qui vérifie qu'un marquage donné est accessible à partir de l'état initial.

Pour le deuxième formalisme qui est la logique, les notions de réutilisabilité, modularité et synchronisation n'existent pas. La validation est présente par l'intermédiaire de vérification de modèle. La conception n'est pas facile avec la logique temporelle car il n'existe pas de représentation graphique du protocole et la conception nécessite de travailler avec des concepts abstraits, ajoutons aussi qu'il n'existe pas d'outils pour la conception mais des outils pour la validation, tous ces inconvénients rend la spécification avec la logique une tâche difficile.

Passons au troisième formalisme qui est les langages formels ; ils présentent un moyen formel pour spécifier des comportements des systèmes distribués et des SMA mais, est-ce qu'ils présentent le bon moyen ou non !

↳ Pour le langage Z, il a l'avantage de l'existence des notions de réutilisabilité et modularité qui facilitent la tâche de conception puisqu'il est possible de décomposer le protocole en un ensemble de modules (composants) qu'il peut les spécifier, valider et vérifier formellement à condition que ces composants sont pris isolément. Cependant, le langage Z ne possède ni une représentation graphique ni d'outils pour la conception et la validation des protocoles, et en plus de ça, il n'offre pas un modèle de communication et de synchronisation ce qui constitue une des faiblesses majeure, dans la modélisation des protocoles.

↳ Pour le langage SDL il se distingue par une représentation graphique qui facilite la conception des protocoles et qui présente aussi une raison à son développement dans l'industrie car il facilite la lecture même par des personnes non spécialistes du formalisme. SDL aussi permet la réutilisabilité et il possède aussi des outils pour la conception et la validation de protocole. Mais, malgré tout ça, dans SDL il n'y a pas de notion de synchronisation ce qui élimine l'aspect des rendez-vous entre agents.

↳ Malgré que Estelle est basé sur la notion de module qui permet au concepteur du protocole de réutiliser et d'avoir une architecture modulaire, il l'oblige de concevoir les protocoles d'une manière textuelle car il ne possède pas d'une représentation graphique mais seulement d'une représentation textuelle, cet inconvénient dégrade énormément les capacités de ce langage et c'est sans doute une des raisons pour lesquelles ce formalisme a très peu de notoriété dans le monde industriel, il est essentiellement utilisé dans le milieu universitaire sachons aussi que ce langage ne dispose pas de mécanisme de synchronisation.

↳ LOTOS est un langage qui garantit les notions de modularité et réutilisabilité car il y a la possibilité de décomposer un processus en sous processus, aussi le mode de coopération des processus est obtenu à l'aide d'un jeu d'opérateurs : parallélisme, synchronisation, choix, séquence et préemption le modèle de communication entre le système et son environnement et entre les processus d'un même système est synchrone pour LOTOS. Il existe aussi des outils et des algorithmes pour la validation des protocoles. Cependant, la représentation textuelle de LOTOS rend la conception du protocole difficile. Alors toutes les caractéristiques générales de LOTOS incluent la puissance d'expression des éléments de services, de protocole et des interfaces, mais sa notation abstraite, difficile à comprendre, et le style algébrique ne sont pas appréciés dans le monde industriel.

4. Conclusion

Tous les différents formalismes que nous avons cités dans ce chapitre ont été appliqués aux systèmes multi-agents pour une prise en compte des actions, de l'activité des agents et de l'évolution des SMA. Ils modélisent généralement l'action comme transformation d'un état global du système. L'exécution des agents s'effectuant en parallèle présente des comportements non déterministes et difficiles à appréhender. Il est donc nécessaire d'utiliser un formalisme ayant un grand pouvoir d'expression tout en étant capable d'offrir des possibilités de vérification formelle sur des propriétés des SMA, c'est ce que nous allons le voir dans le chapitre suivant.

Chapitre 4

La conception du protocole de négociation

1. Introduction

Dans ce chapitre nous allons décrire notre conception qui est la proposition d'une approche formelle pour la négociation automatique entre agents. Notre contribution se présente par un nouveau protocole de négociation et la modélisation et la validation formelle de ce protocole par un outil de modélisation formel.

D'abord nous allons justifier le choix du formalisme que nous allons utiliser pour valider formellement notre protocole de négociation, sachant que cette justification sera basée sur la synthèse faite dans le chapitre précédent.

Ensuite et avant de présenter notre protocole nous commençons la description d'une architecture basée agents supportant ce protocole. Après, nous présentons la description des structures internes des différents agents, ainsi que leurs rôles dans notre architecture.

Finalement nous allons présenter la description formelle de notre protocole de négociation qui permet de valider ce dernier où nous allons essayer d'éviter les inconvénients existants dans les autres travaux.

2. Justification du choix

Comme nous avons noté dans le chapitre précédent ; les différents formalismes habituellement utilisés pour concevoir des protocoles de communication et des protocoles d'interaction sont les même utilisés pour décrire les protocoles de négociation. Dans cette partie nous allons faire une comparaison entre ces derniers pour extraire le formalisme le plus adéquat pour spécifier notre protocole de négociation. Notre comparaison est basée sur un ensemble de caractéristiques que nous recherchons.

Mais avant de proposer notre ensemble de critères nous devons présenter d'autres études faites par d'autres chercheurs pour bien illustrer la vision :

- ❖ Selon Huet [18] : il y a cinq critères qui doivent être respectés et qui sont :
 - ✓ C1 : *Réutilisabilité* : le concepteur doit être en mesure de pouvoir réutiliser des protocoles précédemment créés ou des portions de protocole.

- ✓ C2 : *synchronisation* : les agents dans un système multi-agent agissent de manière concurrente. il est nécessaire que le formalisme utilisé dispose de la capacité de synchroniser les agents en un point donné de l'interaction.
 - ✓ C3 : *Validation* : il est nécessaire que le formalisme dispose d'algorithmes pour la validation des propriétés.
 - ✓ C4 : *facilité de conception* : le concepteur doit disposer d'un langage de représentation graphique du protocole.
 - ✓ C5 : Outils : le concepteur doit disposer d'outils pour la conception et la validation du protocole.
- ❖ Lai & jirachietanna[18] présentent aussi un ensemble de critères que doit satisfaire le formalisme désiré :
- ✓ *Pouvoir d'expression* : le formalisme doit être en mesure d'exprimer une large gamme de propriétés.
 - ✓ *Pouvoir d'analyse* : le formalisme doit être fondé sur un modèle formel qui permet la vérification.
 - ✓ *Modularité* : le formalisme doit supporter une spécification modulaire.
 - ✓ *Niveau d'abstraction* : le formalisme doit être en mesure de cacher certains détails qui ne sont pas importants.

Avant de présenter les critères désirés et attendus du formalisme que nous allons utiliser, nous devons noter que le choix du formalisme dépend de la fonctionnalité du protocole, donc elle ne trouve pas une réponse en dehors du contexte applicatif [7]. En effet, par exemple le concepteur n'utilisera pas le formalisme des AEF s'il désire pouvoir gérer le caractère concurrentiel des interactions entre les agents.

En se basant sur la synthèse faite dans le chapitre précédent (*les approches formels*) et les critères proposés par *huget* et *Lai & jirachietanna* nous tirons des critères qui expriment bien notre besoin essentiel qui est la spécification d'un protocole de négociation et liés en même temps aux différents étapes du cycle de vie d'un logiciel, commençant par la conception du protocole, la validation en arrivant à la négociation elle même. Notons que l'analyse des besoins dépend de l'analyseur ou concepteur qui interagit avec l'utilisateur pour arriver à respecter les besoins de ce dernier. Nous avons classé les critères selon trois niveaux et qui sont:

A. Le niveau de la conception

La tâche de la conception devient simple si les conditions suivantes sont existantes :

C1 : Avoir une représentation graphique

C2 : Modularité et réutilisabilité

C3 : Existence d'outils de conception

B. Le niveau de la validation

C4 : Pouvoir de validation du protocole

C5 : Existence d'outils de validation

C. Le niveau de la négociation elle même

Ce point signifie les besoins essentiels qui caractérisent notre protocole

C6 : Synchronisation et parallélisme

La table 4.1 présente une comparaison entre les différents formalismes, le signe plus (+) indique que le critère est satisfait dans le formalisme considéré, Le signe (-) indique qu'il n'est pas satisfait. L'insatisfaction ne dénote pas forcément une absence du critère mais aussi une implémentation incomplète.

Formalisme	C1	C2	C3	C4	C5	C6	
AEF		+	-	+	+	+	-
Rdp		+	+	+	+	+	+
Logique temporelle		-	-	-	+	+	-
Langage Z		-	+	-	-	+	-
LOTOS		-	+	+	+	+	+
Estelle		-	+	+	+	-	+
SDL		+	+	+	+	+	-

Table 4.1: Comparatif des formalismes par rapport aux critères

A partir de cette comparaison nous pouvons conclure que le formalisme Rdp est le plus adéquat pour spécifier notre protocole sachant qu'il est le seul qui répond à la pluparts des critères désirés de notre part.

3. L'architecture proposée

3.1. La proposition d'une architecture générique

L'architecture que nous allons proposer dans ce travail est une architecture basée agent pour la négociation automatique entre agents conçue pour aider les deux cotés de la négociation soit initiateur ou participants d'établir des transactions dans des points de négociation virtuels utilisant les dernières technologies de traitements de l'information et de la télécommunication sans avoir besoin de beaucoup de déplacements coûteux en matière de temps et de l'argent. Elle définit l'ensemble des composants et modules fonctionnels décrits en termes de leurs comportements et constituants, ainsi que la façon d'interaction de ces composants afin d'accomplir correctement l'ensemble des tâches du système de négociation. Dans le contexte de notre étude, les composants fonctionnels correspondent aux différents agents constituant le système, et leurs modules internes.

Nous ajoutons aussi que parmi les points forts de notre architecture est qu'elle est générique : la même architecture peut être utilisée pour modéliser différentes applications de négociations (selon le domaine de négociation suivie); et n'est pas propre à un seul projet. Dans ce cas, notre architecture sera essentiellement un cadre de développement.

3.1.1. La description de l'architecture du système

Dans cette partie nous allons présenter une architecture d'un système qui permet de réaliser une négociation entre deux agents intelligents négociateurs sachant que chacun négocie à la place d'un utilisateur initiateur ou participant. Nous proposons cette architecture pour permettre d'appliquer notre protocole de négociation, en se concentrant toujours à notre objectif qui est la spécification formelle de ce protocole en utilisant les réseaux de Petri.

L'objectif de notre protocole est de définir les messages que les agents pourront s'envoyer avec la dynamique opérationnelle associée. Le protocole de négociation que nous proposons est caractérisé par une suite de messages échangés entre un initiateur et un participant en suivant la forme de négociation multi-attributs.

La figure 4.1 présente notre architecture contenant deux types d'agents : initiateur et participant et d'un protocole de négociation. Nous ajoutons que cette architecture est conçue d'une manière permettant une généralisation de la négociation que signifie que la négociation peut être réalisée entre un agent initiateur et plusieurs agents participants.

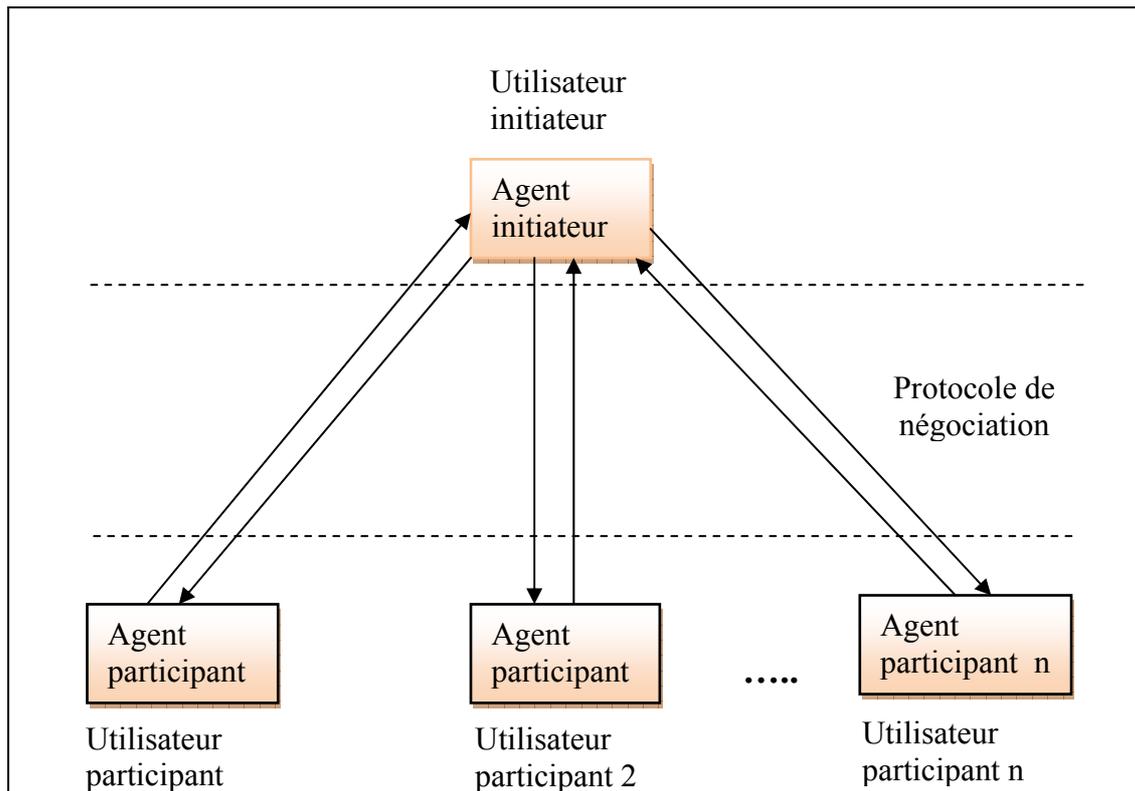


Figure 4.1 : architecture de notre système

3.1.2. Types d'utilisateurs

Nous devons spécifier les types d'utilisateurs qui vont être remplacés par les agents négociateurs initiateur et participant c'est à dire les utilisateurs de ce système. Nous signalons qu'il y a deux types :

1. *Utilisateur initiateur* : ce type concerne la personne qui publie une offre (proposition) pour une ou plusieurs autres personnes sur internet (par exemple un vendeur),
2. *Utilisateur participant* : les personnes à qui la proposition est faite et qui cherchent à négocier des offres s'appellent les participants à la négociation (par exemple un acheteur).

3.2. Le protocole de négociation

3.2.1 Description du protocole de négociation

Avant de commencer la description de notre protocole nous devons noter ces différentes propositions qui présentent les bases de notre protocole :

✓ l'objectif de notre protocole est de réaliser une négociation automatique entre deux types d'utilisateurs participant et initiateur en utilisant les agents intelligents où chaque utilisateur est remplacé par un agent négociateur artificiel. Donc il nous résulte deux types d'agents :

- *Agent initiateur* : qui négocie à la place d'utilisateur initiateur.
- *Agent participant* : qui négocie à la place d'utilisateur participant.

Un utilisateur initiateur publie une offre (qui spécifie toutes les conditions que l'initiateur souhaite) sur internet et attend les demandes des utilisateurs participants, et pour faire la négociation il délègue un agent initiateur à sa place.

✓ Un utilisateur participant cherche sur internet, (lui-même ou en utilisant un agent de recherche) et trouve qu'il y a une offre qui le satisfait mais avec un peu de changement, il doit alors déléguer un agent participant qui négocie à sa place. Donc l'agent participant reçoit des renseignements concernant l'adresse d'agent initiateur et la contre offre de son utilisateur participant. Notre protocole de négociation doit être précédé par une phase de recherche pour trouver quel est l'agent (initiateur) avec qui on doit négocier et ce n'est pas le rôle de notre agent participant.

La forme de négociation suivie dans ce protocole est la négociation multi-attributs. L'agent initiateur utilise une liste de pourcentage triée où le pourcentage signifie la distance entre la contre offre du participant et l'offre d'initiateur. Si un agent trouve qu'une offre est convenable, il doit retourner à son utilisateur avant de décider de l'accepter.

Par la suite nous allons décrire notre protocole, qui est défini par trois phases lesquels sont :

1 La phase de proposition

- L'agent participant essaye de contacter un agent initiateur qui présente le résultat d'une phase de recherche soit par l'utilisateur lui-même ou d'un autre agent de recherche.
- Alors l'agent participant envoie une demande de négociations à l'agent initiateur, cette demande se présente seulement par une contre offre car s'il envoie un accord cela ne présente pas une opération de négociation.

- Si la réponse d'agent initiateur ne présente pas un accord, alors on passe à la phase de conversation et cette fois-ci on peut dire que la négociation est déclenchée. Autrement on passe directement à la phase de décision finale (les critères qui décident le rejet et l'acceptation seront décrits dans la section de processus de décision).

2 **La phase de conversation**

Nous avons dit précédemment dans la phase de proposition que la négociation se déclenche si et seulement si la réponse d'agent initiateur est différente d'un accord, donc nous sommes dans la phase de conversation.

Selon MAZOUZI dans [23] une conversation est un protocole d'interaction en cours d'exécution composé d'un ensemble totalement ou partiellement ordonné de messages qui respectent le protocole.

Pour notre protocole nous passons de la phase de proposition à celle de conversation si la réponse d'initiateur présente soit une liste de pourcentage ou bien une nouvelle offre ; ici aussi si la réponse est un accord la conversation se termine et nous passons directement à la phase de décision finale.

Alors il nous résulte deux cas :

- A. Arrêt de conversation : l'arrêt de conversation qui présente une terminaison du processus de négociation apparaît selon les points suivants :
 - ✓ Si un agent soit initiateur ou participant trouve qu'il a trop concédé et que l'autre côté exagère dans ses ambitions il peut arrêter la négociation, et cela en envoyant un rejet à l'autre agent négociateur.
 - ✓ Si l'utilisateur lui-même décide d'arrêter la négociation son agent négociateur doit envoyer un rejet vers l'autre agent antagoniste.
- B. Prolongation (extension) de la conversation : la conversation continue dans les cas suivants :
 - ✓ Si la réponse d'initiateur présente une liste de pourcentage ça signifie qu'il veut plus de concessions, alors l'agent participant essaie d'analyser la liste et finalement décidera s'il peut continuer dans cette négociation ou non et en cas de oui l'agent participant envoie un message à l'initiateur contenant un contre offre.

- ✓ L'agent initiateur essaye d'analyser le contre offre du participant et il le trie dans la liste de pourcentage et finalement lui répond par cette liste ou bien par une nouvelle offre s'il trouve que tous les contres offres sont très loin de son offre.

3 La phase de décision finale

- Nous arrivons à la phase de décision lorsqu'un agent accepte une offre (contre offre) ou un agent termine le processus de négociation. Notre architecture permet aussi la rétraction en cas où la négociation est terminée avec un succès et que le participant gagnant de cette négociation n'a pas pu payer par exemple, alors il ne doit pas compenser les pertes d'initiateur.

3.2.2. La cardinalité de la négociation

La notion de la cardinalité de la négociation signifie le nombre des agents participants dans un processus de négociation soit initiateur ou participant. Généralement il y a plusieurs types de cardinalités de négociation commençant par le plus simple de 1 vers 1, passant par le type 1 vers n jusqu' à le type le plus complexe de m vers n.

Pour notre protocole ; la négociation se déroule entre un initiateur et un participant, ce qui donne l'aspect que la cardinalité est de type 1 vers 1 mais en réalité l'initiateur peut recevoir plusieurs demandes de négociations en parallèle (cardinalité 1 vers n), mais ce qui nous intéresse c'est de modéliser la négociation entre un initiateur et un participant seulement.

Ce protocole sera implémenté au dessus d'un autre protocole d'une couche inférieure qui sera l'un des protocoles adoptés par les spécifications de FIPA ACL.

3.2.3. Choix du langage de communication utilisé : *FIPA ACL*

En plus du protocole, on a besoin d'un langage de communication pour exprimer les propositions de négociation et tout autre message de telle sorte qu'il soit compréhensible par tous les agents de la même manière (langage standard).

Nous avons vu dans le chapitre1 *interaction entre agents* que dans le champ des SMAs [Systèmes Multi-Agents] à base d'agents cognitifs, le besoin d'un modèle de communication standard se fait sentir et les efforts se multiplient dans ce sens. Les technologies agents et multi-agents permettent de concevoir et de développer des applications complexes. La caractéristique fondamentale de celles-ci dans le paradigme actuel de l'informatique répartie est l'habileté des agents à communiquer entre eux de

manière utile à leurs objectifs tant individuels que collectifs. Les spécifications de FIPA ACL, que nous allons utiliser dans notre projet se composent d'un ensemble de types de message et de la description de leur pragmatique. Les spécifications décrivent chaque acte communicatif avec une forma narrative et une sémantique formelle basée sur la logique modale. Elles fournissent également la description normative d'un ensemble de protocole d'interaction de haut niveau, y compris la demande d'action, l'établissement de contrat (contract net) et plusieurs genres de ventes aux enchères. C'est pour cela qu'elle convient à notre protocole de négociation.

Les agents communiquent entre eux en utilisant des messages ce qui peut représenter des actes de langages, et qui sont encodés dans le langage de communication d'agent ACL (Agent Communication Language)

3.2.4. Liste de performatives de communication FIPA ACL

Nous donnons toutes les communications possibles qu'on aura besoin et les regroupons selon leurs actions: [13]

Actions	Syntaxe	Définition – Sens
Accept Proposal	accept-proposal	Communication de l'accord de l'expéditeur d'effectuer une action qui lui a été préalablement soumise.
Agree	Agree	Communication de l'accord de l'expéditeur pour effectuer une action, sans doute dans le futur.
Cancel	Cancel	Communication de l'annulation de l'accord donnée préalablement par l'expéditeur pour effectuer une action.
Call for Proposal	Cfp	Communication par l'expéditeur d'une demande d'effectuer une certaine action.
Confirm	Confirm	Communication par l'expéditeur de la confirmation de la validité (selon les règles de l'agent) de la proposition préalablement reçue.
Disconfirm	Disconfirm	Communication par l'expéditeur de la confirmation de la non validité (selon les règles de l'agent) de la proposition préalablement reçue.
Failure	Failure	Communication par l'expéditeur de l'échec d'une action essayée.
Inform	Inform	Communication par l'expéditeur d'une proposition, pensée vrai par celui-ci.
Inform If	inform-if	Communication par l'expéditeur d'une proposition (pensée vrai par celui-ci), et demande au receveur une confirmation ou une non-confirmation. Macro-action impliquant l'usage de "request".
Inform Ref	inform-ref	Communication par l'expéditeur d'une demande de l'objet qui correspond à une description envoyée. Macro-action impliquant l'usage de "request".
Not Understood	not-understood	Communication par l'expéditeur d'une non compréhension d'une action effectuée par le destinataire.
Propagate	Propagate	Communication par l'expéditeur d'un message à propager à des agents dont la description est fournie. Le destinataire du message traite le sous-message à propager comme s'il lui était directement destiné et envoie le message "propagate" au agent qu'il a identifié
Propose	Propose	Communication par l'expéditeur d'une proposition d'action conditionnée à certaines préconditions données.
Proxy	Proxy	Communication par l'expéditeur d'une demande d'une transmission d'un message à des agents dont la description est donnée.
Query Ref	query-ref	Communication par l'expéditeur d'une demande par l'expéditeur de l'objet référencé par une expression.
Refuse	Refuse	Communication par l'expéditeur de son refus d'effectuer une action donnée, et en donne les raisons.
Reject Proposal	reject-proposal	Communication, pendant une négociation, par l'expéditeur de son refus d'effectuer des actions.
Request	Request	Communication par l'expéditeur d'une demande au destinataire d'effectuer une action.
Request When	request-when	Communication par l'expéditeur d'une demande, au destinataire, d'effectuer une action quand une proposition donnée devient vrai.
Request	request-	Communication par l'expéditeur d'une demande, au destinataire,

Whenever	whenever	d'effectuer une action dès qu'une proposition donnée devient vrai, et à chaque fois que celle-ci redevient vrai.
Subscribe	Subscribe	Communication par l'expéditeur d'une demande d'un objet donnée par une référence envoyé par l'expéditeur, et de renotifier l'agent ayant souscrit dès que l'objet en question change.

3.2.5. Structure des messages de communication simple

Le message minimum type (syntaxe de ce message) du FIPA ACL contient tout d'abord:

- ✓ L'expéditeur du message,
- ✓ Le destinataire du message,
- ✓ Le contenu du message.

Cependant, ces messages minimums ne suffisent pas toujours pour communiquer: on peut avoir besoin, pour la compréhension du message et pour la rapidité de celle-ci ainsi que la rapidité de traitement du message, d'indiquer d'autres informations telles que:

- Le langage utilisé dans le contenu du message ("language ...") : Plusieurs langages peuvent être utilisés pour la description du contenu des messages échangés tels que :
 - ✓ Le langage KIF;
 - ✓ Le langage sémantique (SL);
 - ✓ Prologue;
 - ✓ Le langage XPDL (XML Process Markup Language);
 - ✓ Le langage XML, ...
- le protocole utilisé,
- l'ontologie auquel le message se rattache ("ontology ..."),
- la référence d'un message antérieur auquel le message actuel se rattache ("in-reply-to ..."), ou la référence d'un message ultérieur attendu en retour ("reply-with ...").
- la référence de la conversation.

Exemple:[13]

L'agent A veut informer l'agent B du temps qu'il fera demain, selon ses prévisions:

(inform **Table 4.2** : Liste des performatives de communication FIPA ACL
:sender (agent-identifiant : name_A)
:receiver (set (agent-identifiant : name_B))

:content

"weather (tomorrow, raining)"

:language Prolog)

3.2.6. Evaluation de notre protocole

Pour évaluer et juger l'efficacité de notre protocole nous nous sommes basé sur les propriétés et les attributs spécifiés par Rosenschein et Zlotkin pour le jugement des protocoles de négociation [33] :

1. *Temps de négociation* : les négociations qui se terminent sans délai sont préférées. En effet, le type de négociation qui utilise un délai pour trouver un accord durant la négociation augmente le coût de communication et le temps de calcul d'une façon considérable.
2. *Simplicité* : le protocole doit faire de faibles calculs pour chaque demande émanant des agents ; il utilise le moins possible de communications entre eux,
3. *Distribution* : les règles d'interactions ne doivent pas avoir besoin d'une prise de décision centralisée,
4. *Symétrie* : aucun processus de négociation ne doit être appliqué arbitrairement et partialement contre certains agents.

Pour le premier caractère notre protocole n'exige pas un délai pour trouver un accord, où la négociation se déroule jusqu'à l'arrivée à un accord ou un refus (rejet d'une proposition = arrêt de la négociation) ou bien par l'utilisateur lui-même s'il trouve que la négociation a pris beaucoup de temps sans utilité il peut arrêter la négociation. Pour les autres critères notre protocole est simple, même si nous voyons nous trouvons qu'il y a seulement cinq types de messages : *Propose (contre offre)*, *CFP (nouvelle offre)*, *Inform (liste de pourcentage)*, *Reject-proposal*, *Accept-proposal*, et pour le critère de distribution nous avons conçu un modèle de négociation qui contient un protocole et des agents négociateurs intelligents et le plus important c'est qu'ils se caractérisent par l'autonomie, et pour le critère de symétrie, les règles de notre protocole s'appliquent de la même manière aux agents sans distinction.

3.3. Architecture d'agent de négociation

Avant de présenter notre architecture proposée pour l'agent négociateur nous devons noter que cette architecture est presque la même soit pour l'agent initiateur ou participant avec quelques changements au niveau du module d'évaluation et exactement la fonction d'utilité et aussi au niveau du module de génération de l'offre.

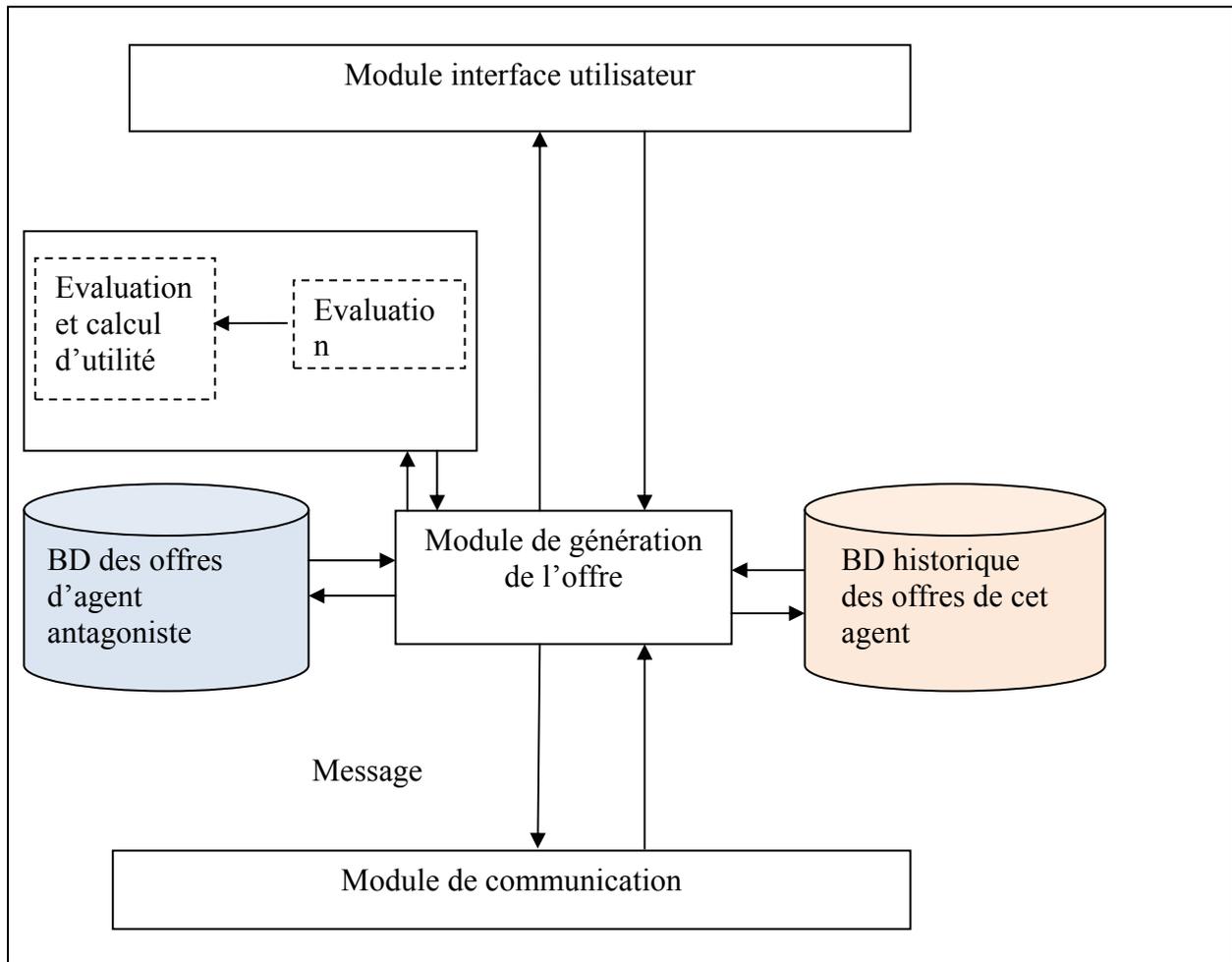


Figure 4.2 : Schéma représentant l'architecture de l'agent négociateur

Le schéma précédent présente l'architecture d'un agent négociateur en générale, ce qui donne l'impression que les deux types d'agent soit initiateur ou participant ont la même architecture mais en réalité nous avons utilisé seulement le même schéma et les noms des modules pour les représenter sachant que nous allons détailler la constitution des deux agents et la différence entre eux. Le schéma illustre que l'agent est composé des quatre modules suivants :

1. Module interface utilisateur

Ce module est responsable de la présentation de ce qui est disponible dans le système et ses différentes fonctionnalités, sous forme d'une interface utilisateur. Alors il peut être considéré comme la fenêtre du système vers l'extérieur. Mais le rôle essentiel de ce module est de permettre la configuration de la négociation en retournant toujours à l'utilisateur car c'est lui qui peut décider d'accepter, rejeter ou proposer une offre. C'est un élément décisif du succès ou l'échec du système, car s'il ne présente pas les fonctionnalités offertes par le système d'une manière simple, intuitive et conviviale pour les utilisateurs qui sont beaucoup plus conscients et exigeants qu'autrefois et avec la concurrence qui existe sur le Web, ils se sauveront vite vers un système qui est plus accueillant et simple à utiliser. D'où, le module d'interface est un composant critique du système, et nécessite pour sa conception et son développement de consacrer beaucoup d'attention aux besoins de l'utilisateur. Il doit avoir au moins les fonctionnalités suivantes :

- Identification de l'utilisateur.
- Transmettre les besoins des utilisateurs au système pour les traiter.
- Configuration de la négociation.
- Présenter les résultats d'une manière intégrée et adaptée aux préférences des utilisateurs.

2. Module d'évaluation

Le module d'évaluation est constitué lui-même de deux sous module qui sont :

- a) *Sous module d'évaluation des attributs* : responsable de l'évaluation des offres réceptionnées auprès des autres agents. La fonction principale de l'évaluation des attributs est de transformer les valeurs des attributs d'une offre donnée aux valeurs numériques. Et puisque c'est le cas de négociation multi-attributs il y a plusieurs attributs qu'il faut les évaluer : temps et prix, prix et qualité...
- b) *Le sous module de calcul de l'utilité* : il calcule l'utilité d'une offre donnée en se basant sur les valeurs des attributs. Si l'offre est en dehors du rang d'utilité spécifié par l'utilisateur il génère une nouvelle offre avec une nouvelle utilité. Si l'offre est dans l'intervalle de l'utilité assigné par l'utilisateur, alors un agrément est abouti et l'exécution est atteinte.

Pour la fonction d'utilité nous allons utiliser celle défini dans [21] où la négociation sera effectués sur plusieurs attributs où chaque attribut a une valeur x_i qui appartient à un intervalle des valeurs $I_i = [min_i, max_i]$ qui est déterminé par l'utilisateur de l'agent. Chaque agent a des fonction d'utilité $V_i(x[i])$ pour évaluer la valeur x de l'attribut i dans l'intervalle d'acceptation $[min_i, max_i]$ tel que :

$$V_i(x[i]) \rightarrow [0,1]$$

$$x_i \rightarrow I_i = [min_i, max_i]$$

Toute valeur de V_i en d'hors de l'intervalle d'acceptation engendrera le rejet de l'offre. Donc la partition d'utilité reste dans l'intervalle $[0,1]$.

L'utilité de tous les attributs est calculée par la fonction linéaire additive des valeurs avec poids suivante :

$$U(O) = \sum_{i=1}^m w_i \cdot V_i(x_i) \quad \text{sachant que} \quad \sum_{i=1}^m w_i = 1$$

Le poids w_i est l'importance relative que l'agent affecte à l'attribut i . Les poids doivent rester dans l'intervalle $[0,1]$. Le symbole m est le nombre des attributs, O est l'offre proposé.

3. Module de génération d'offre

Il est responsable de la génération des offres et contre-offres qui vont être envoyés aux autres agents sous la forme d'un message. Le message peut avoir plusieurs formes qui dépendent selon la décision et l'offre générée d'agent lui même et qui dépendent aussi de la nature d'agent c.-à-d. s'il est initiateur ou participant. Donc il nous résulte deux cas :

1. Cas d'agent initiateur : pour ce cas il y a trois types de message :

- ✓ *CFP(nouvelle offre)* : Au début de la négociation l'agent initiateur n'envoie aucun message, c'est seulement l'agent participant qui doit commencer la négociation par l'envoi d'un message à l'initiateur. Mais après une période de négociation et si l'agent initiateur trouve que son offre dépasse la capacité et la pouvoir des autres participants, il

peut le modifier et il doit générer une nouvelle offre *CFP*(*call for proposal*) en se basant sur ses précédents offres qui se trouvent dans la base de données historique de cet agent initiateur et il l'envoie à tous les participants qui sont présents pour la négociation.

- ✓ *Accord* : l'agent initiateur génère un message d'accord dans le cas où il trouve que le contre offre du participant est convenable. Alors il doit arrêter la négociation avec les autres participants.
- ✓ *Liste de pourcentage* : l'agent initiateur peut recevoir plusieurs demandes de négociation (contres offres), il essaye de trier ces derniers dans une liste selon leur convenance à son offre ; alors comment l'agent initiateur peut mesurer cette convenance ? la solution c'est d'évaluer chaque contre offre à part, et essayer de calculer la distance entre ce dernier et l'offre d'agent initiateur. Le résultat de cette évaluation se présente par un pourcentage qui permet d'illustrer qui est le contre offre le plus proche de son offre.
- ✓ *Rejet* : quand l'agent initiateur trouve que le contre offre du participant est très loin de ses ambitions où bien il se sent que l'agent participant exagère dans ses contres offres et ne montre pas de bonne foi, il peut décider d'arrêter la négociation et d'envoyer un rejet du contre offre ce qui signifie un rejet totale de négociation et pas seulement d'une contre offre. Si l'agent initiateur décide d'accepter une offre d'un participant il doit rejeter les autres offres. De même, si l'utilisateur initiateur décide d'arrêter la négociation l'agent initiateur doit informer tous les agents participants par un rejet signifiant un rejet des contres offres et un arrêt de la négociation.

2. Cas d'agent participant : Il y a deux types de messages :

- ✓ *Propose* : un agent participant envoie une demande de négociation vers l'agent initiateur qui se présente par un contre offre et il attend la réponse pour un délai connu, sachant que l'agent participant génère la contre offre en se basant sur ses précédentes offres et les contre offres d'initiateur.

- ✓ *Accord* : si l'agent participant trouve qu'une offre est convenable, il envoie un accord vers l'initiateur pour l'informer qu'il accepte son offre.
- ✓ *Rejet* : au moment où l'agent participant trouve qu'une offre est imaginaire ou très loin de son ambition, il doit envoyer un rejet pour arrêter la négociation. Cela est possible par une décision par l'utilisateur participant lui-même d'arrêter la négociation, ce qui oblige l'agent participant d'envoyer un rejet vers l'agent initiateur qui signifie qu'il rejete l'offre et qu'il arrête aussi la négociation.

4. Module de communication

Ce module permet aux agents d'échanger des messages entre eux et avec les éléments constituant le système. A travers ce module, l'agent négociateur communique avec les autres agents en envoyant et recevant les contres offre.

4. Aspect formel de notre protocole

4.1. Diagramme AUML

Avant de passer à la modélisation de notre protocole par les réseaux de Petri nous devons illustrer ce protocole par un mécanisme semi formelle pour qu'il soit clair et compréhensible par tout le monde. Nous avons choisit d'utiliser le langage AUML car (nous l'avons déjà noté dans le précédent chapitre) il étend la notion de diagramme de séquence utilisé dans UML 2.0 à celle de diagramme d'interaction qui permet de représenter les protocoles d'interaction (protocoles de négociation) et leurs instances. Un diagramme d'interaction est un graphique constitué d'un cadre qui permet d'identifier le protocole (nom, paramètres, etc. . .). Ce cadre délimite la représentation graphique du flux de messages entre les différents rôles. Un diagramme d'interaction est une représentation graphique en deux dimensions. Le temps est représenté selon un axe vertical. Les différents rôles joués par les participants sont représentés selon un axe horizontal. La figure 4.3 illustre en détails notre protocole en AUML

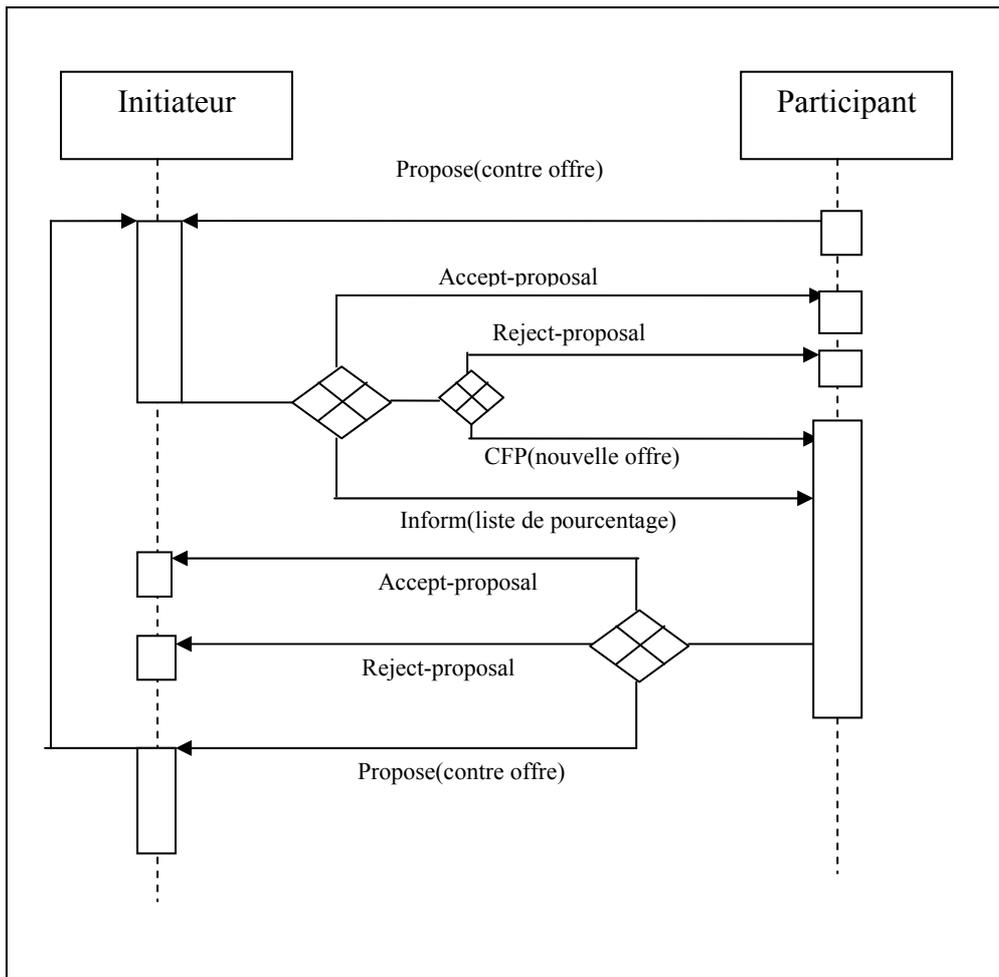


Figure 4.3: Le diagramme AUML de notre protocole

Nous commençons maintenant à décrire les différents messages envoyés entre les deux agents initiateur et participant :

- ✓ Propose (contre offre) : il s'agit de la proposition du participant en réponse à l'appel d'offre d'initiateur.
- ✓ CFP (nouvelle offre) : lorsque l'initiateur veut envoyer une demande à des participants enregistrés auprès de lui, il envoie un tel acte Cfp (Call for proposal) à ses participants.
- ✓ Inform (liste de pourcentage) : si l'agent initiateur veut demander plus de concessions de la part d'agent participant il l'informe avec cet acte fourni avec le paramètre liste de pourcentage pour lui stimuler à concéder.
- ✓ Reject-proposal : Cet acte est généré lorsque l'agent décide d'arrêter la négociation.

✓ Accept-proposal:

- Coté participant: après avoir reçu un appel à propositions de type Cfp ou une liste de pourcentage, le participant vérifie s'il peut accepter l'offre d'initiateur. Dans le cas positif, il génère cet acte qui signifie l'acceptation de l'offre, sinon il répond par un Reject-proposal ou Propose(contre offre) .
- Coté initiateur: l'initiateur choisit la meilleure proposition, et contacte l'agent participant choisi comme gagnant par un tel acte d'acceptation de proposition.

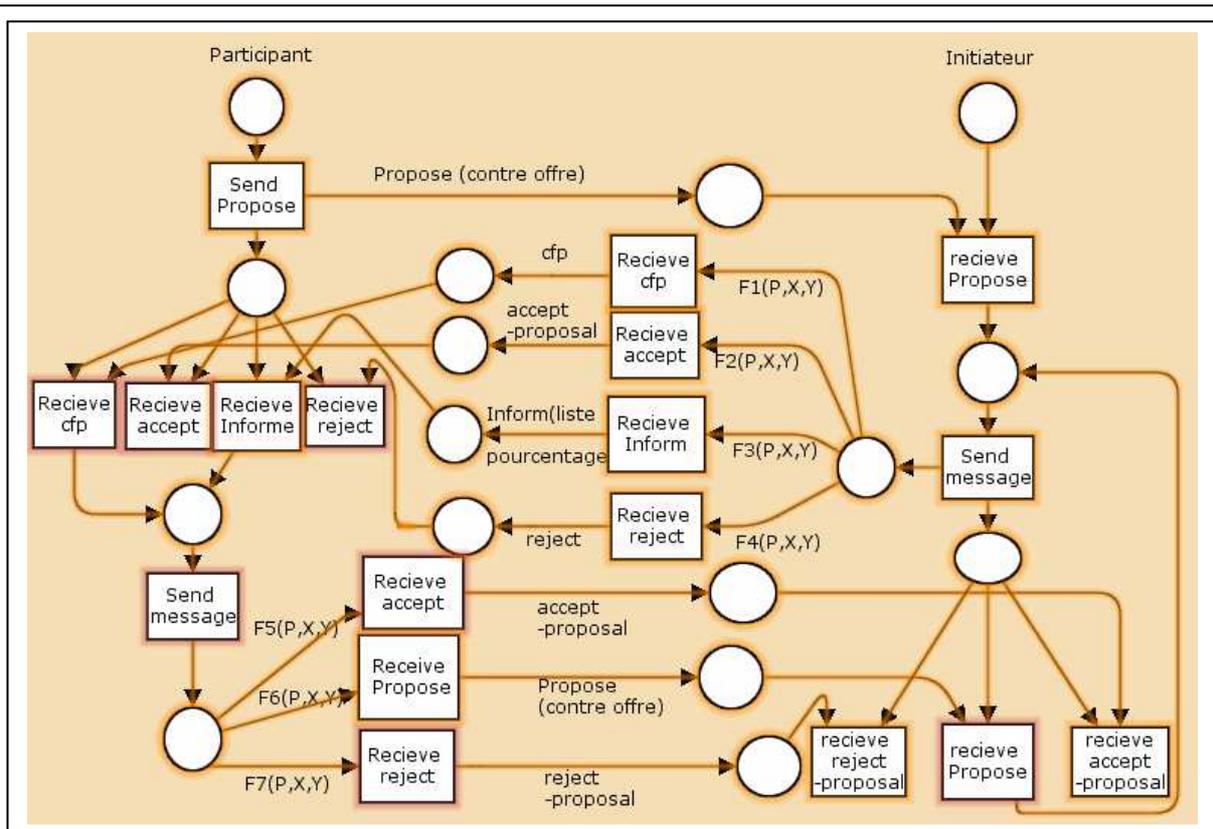
4.2. La modélisation de notre protocole par un RdP

La représentation graphique des RdP permet de visualiser les activités dynamiques des systèmes complexes. Cette visualisation est réalisée à l'aide des jetons qui sont introduits dans le réseau. À partir de ce réseau graphique, nous pouvons décrire le comportement du système. Nous ajoutons que nous étions obligés d'utiliser un type d'RdP qui permet la distinction entre les jetons puisque ces derniers représentent les différents types des messages échangés dans notre protocole et les RdP ordinaires ne permettent pas de modéliser cette distinction. Alors les RdPC présentent un bon formalisme pour spécifier notre protocole de négociation.

Dans les RdP colorés un identificateur (couleur) est associé à chaque jeton d'une place. Chaque transition peut être franchie de différentes manières représentées par les différentes couleurs de franchissement associées à la transition.

Alors pour la modélisation de notre protocole de négociation, nous utilisons une représentation basée sur un réseau de Petri coloré (RdPC).

En se basant sur les règles de traduction de la notation AUML des diagrammes de protocoles aux RdPC [23], donc nous allons modéliser notre protocole de négociation avec les RdPC. Pour mieux comprendre les RdPC, leurs structures et la modélisation des protocoles par ces derniers, le lecteur peut consulter [23].



Ensemble d'agents : {agent initiateur, agent participant}
X : agent initiateur / **Y** : agent participant
P : {Inform (liste de pourcentage), accept-proposal, cfp, Reject-proposal, Propos(contre offre) }
 /* **P** présente l'ensemble des couleurs sachant que les couleurs présentent les différents messages échangés entre les deux types d'agents*/
F : présente la fonction de condition de garde (F est de type vrai ou faux)
 ✓ F1 (P, Y, X) = <P, X, Y> SSI P = <Inform(liste de pourcentage) >
 ✓ F2 (P, Y, X) = <P, X, Y> SSI P = < accept-proposal>
 ✓ F3 (P, Y, X) = <P, X, Y> SSI P = < cfp >
 ✓ F4 (P, Y, X) = <P, X, Y> SSI P = < Reject-proposal >
 ✓ F5 (P, X, Y) = <P, X, Y> SSI P = < accept-proposal>
 ✓ F6 (P, X, Y) = <P, X, Y> SSI P = < Proposa(contre offre)>
 ✓ F7 (P, X, Y) = <P, X, Y> SSI P = < Reject-proposal>

Figure 4.4: la modélisation par RdPC de notre protocole

Pour évaluer la qualité de notre RdPC nous allons le critiquer selon quelques propriétés de base qui sont des conditions fortes de la validité des spécifications des protocoles et qui doivent être contrôlées. Ces propriétés sont essentiellement [23]:

- *L'absence d'interblocage* : qui signifie que les agents impliqués dans une conversation atteignent tous leurs états de fin respectifs. Ce critère est assuré dans notre protocole où les états de fin sont : la réception d'un message de type reject-proposal ou accept-proposal.

- *La vivacité* : qui signifie que tous les états sont atteignables à partir de l'état initial et l'utilisation de tous les messages du protocole est assurée. Ce RdPC présente une spécification formelle de notre protocole de négociation où la négociation se déclenche par le premier contre offre du participant que signifie que tout commence par cet état initiale.

- *Le caractère borné du modèle* : qui signifie que le protocole ne peut pas engendrer un processus avec un nombre infini d'états, ce critère existe aussi dans notre RdPC car dans notre protocole la négociation ne se déroule pas à l'infini et les utilisateurs ont l'autorité d'arrêter la négociation.

5. Conclusion

Nous avons présenté dans ce chapitre notre proposition d'une approche formelle pour la négociation automatique entre agents où nous avons présenté une architecture générique

à base d'agent intelligents pour bénéficier de l'autonomie et l'intelligence des agents dans le domaine de négociation automatique.

Pour les agents constituant cette architecture qui sont l'agent initiateur et l'autre participant nous avons proposé une architecture générique capable de mémoriser les différents cycles de négociation en commençant par la première offre d'initiateur jusqu'aux contres-offres échangées pendant la conversation et tout cela en se basant sur la négociation multi-attributs qui présente le cadre le plus réaliste et le plus utilisé dans la négociation entre les humains. Tous ces indices rendent notre modèle de négociation plus proche à celui qui se déroule entre les humains.

Pour valider notre proposition et plus exactement notre protocole de négociation, nous avons modélisé ce dernier par un outil de spécification formel qui se présente par les réseaux de Petri coloré.

Finalement pour comprendre le fonctionnement de notre architecture sur un système particulier, nous présenterons dans le chapitre suivant un exemple d'une négociation automatique dans le domaine du commerce électronique. En exposant quelques outils nécessaires pour l'implémentation d'un tel système.

Chapitre 5

Étude de cas et implémentation

1. Introduction

Dans le chapitre précédent de ce mémoire, nous avons proposé une architecture basée agents pour la négociation automatique entre agents. Afin d'illustrer les différentes idées et concepts inclus dans l'architecture proposée, nous allons utiliser cette architecture comme base pour une étude de cas dans un environnement réel. Le but est de dérouler les principaux aspects de notre architecture sur un exemple concret afin de montrer la faisabilité et la mise en évidence de nos idées.

Pour cela, nous procédons comme suit : nous commençons par limiter le cadre de notre application qui est le commerce électronique, en justifiant l'utilisation de la technologie agent. Par la suite, nous décrivons brièvement la plate-forme que nous avons adoptée pour l'implémentation de notre architecture, pour montrer par la suite comment nous l'exploitons dans le cadre de notre travail. Les résultats obtenus à partir de l'implémentation de notre étude de cas sont présentés à la fin.

2. Le commerce électronique

2.1 Du commerce traditionnel vers le commerce électronique

Le commerce est un mécanisme qui permet à deux parties, le vendeur et le client de mener des activités de vente d'achats et/ou d'échange de produits (bien, services ou information) sur la base de la théorie de l'offre et la demande.

Contrairement à sa forme traditionnelle dans laquelle le vendeur, le client et éventuellement un ou plusieurs intermédiaires effectuent physiquement une transaction de vente ou d'achat. Le commerce s'est transporté depuis quelques décennies, dans un monde virtuel dominé par les réseaux informatiques, l'internet en l'occurrence pour donner naissance au commerce électronique ou e-commerce [32]. Dans ce contexte virtuel, les deux entités peuvent ne jamais se rencontrer et/ou reconnaître avoir effectué des transactions électroniques. Après l'achat, la livraison peut être physique ou électronique, tout dépend du bien, ou du service acheté.

2.2 Définition du commerce électronique

L'évolution du commerce électronique '**e-commerce**' fait que sa définition tente chaque jours de s'adapter au nouveau contexte qui se présente. Selon [13] , le commerce électronique est un processus d'achat, de vente, de transfert et d'échange de bien, de service et/ou d'information à travers un réseau d'ordinateurs, en l'occurrence d'internet, ou tout autre moyen de communication (exemple : le réseau mobile).

Ebaye.com, amazon.com, entrust.com, sont des exemples de sites web qui pratiquent le commerce électroniques. De façon plus restreinte la définition du e-commerce est fonction du contexte mis en jeu (communication, affaires ou services) [24].

2.3 Entités

Les facteurs du **e-commerce** font partie d'un espace de marché (marketspace) virtuel qui est le lieu de rencontre des vendeurs et des clients pour échanger des produits contre l'argent ou d'autre produits. Un espace de marché se compose des éléments suivants [13]:

- les vendeurs
- les clients,
- les produits,
- l'infrastructure,
- l'interface client,
- l'interface vendeur,
- les intermédiaires,
- les partenaires d'affaires
- et les services de support.

Nous allons maintenant expliquer chaque entité à part :

- ✓ **Le client** est un individu ou groupe d'individus (une organisation par exemple) à la recherche d'un produit. Parfois, il connaît juste son besoin et n'a pas nécessairement une idée précise du produit qui pourrait le satisfaire. Dans tous les cas, le client va interroger le catalogue du vendeur pour rechercher un produit précis (s'il le connaît) ou alors pour décrire son besoin et recevoir des propositions de produits, par l'intermédiaire d'un système de recommandation.
- ✓ **Le vendeur** est un individu ou groupe d'individus (une organisation par exemple) détenteurs de produit (bien, service et information) à vendre. La description de ces produits ainsi que leurs prix et condition de vente sont mis ensemble pour former le catalogue électronique du vendeur.

- ✓ le terme **produit** renvoie à un bien, un service ou une information que le client achète auprès du vendeur. Evidemment, il faut tenir compte du contexte dans lequel il est utilisé pour lui donner le sens que nous venons de mentionner.
- ✓ **L'infrastructure client** ou *front-end* est un ensemble de programmes dite d'interfaces à travers lesquels le client interagit avec le système du vendeur. Le portail électronique, le catalogue électronique, le panier d'achats, le moteur de recherche et les passerelles vers les systèmes de paiements sont des exemples de programmes que le vendeur peut mettre dans l'interface client.
- ✓ **L'interface vendeur** ou back-end met en relief les différentes activités qui permettent de gérer en ligne la passation de commandes de produits. Cette composante inclut donc, entre autre, la gestion d'inventaires, les achats auprès des fournisseurs, le traitement des paiements, l'emballage et la livraison.
- ✓ **Un intermédiaire** en commerce électronique est une entité physique ou virtuelle qui assiste le client et le vendeur à finaliser un processus de vente/achat. Les banques (pour le paiement électronique), les compagnies de livraison de produits, les agents mobiles, entre autre, constituent des exemples d'intermédiaires en e-commerce. De façon générale, les intermédiaires fournissent des services à valeur ajoutée aux clients. Le terme *infomédiaire* désigne les intermédiaires électronique, ceux-là même qui cherchent à contrôler la circulation de l'information sur Internet, et qui peuvent en créer des agrégats à vendre. Les boutiques du coin (convenient stores) et les compagnies de publicité sont d'autres exemples d'intermédiaires. Les premiers constituent parfois des points de livraison des produits physiques achetés en commerce électronique. Quand aux secondes, elles se chargent de promouvoir des produits à la demande des vendeurs (ex. :doubleclick.com). Un dernier exemple d'intermédiaire est relatif à des compagnies organisatrices des conférences. De telles compagnies font tout ce qui est nécessaire à la réussite des conférences, à l'exception de l'évaluation des articles qui y sont soumis.
- ✓ **Un partenaire d'affaire** d'une compagnie est une personne physique ou morale qui est impliquée dans sa gestion, en tant qu'associé. Le partenaire d'affaire peut par exemple disposer des parts dans la compagnie. Dans le cadre du commerce électronique, le vendeur peut s'associer à des agences de livraison pour faciliter l'acheminement des produits à ses clients. Il peut aussi signer des contrats de

responsabilité dans le souci de couvrir les zones qui lui sont difficiles d'accès, à cause par exemple de l'éloignement.

- ✓ **Un service de support** est généralement apporté au client, sous l'appellation support technique, pour gérer l'après-achat. Plus précisément, cette composante permet au vendeur d'apporter une assistance continue à ses clients pour ce qui est de la maintenance et des éventuelles mesures correctives applicables aux produits vendus. Le service de support est un autre exemple d'élément pouvant faire l'objet de partenariat ; c'est à dire, le vendeur confie la gestion de son service après vente à une compagnie tierce.

2.4 Les catégories (types) du commerce électronique

Il est devenu courant d'établir une distinction entre les applications professionnelles du commerce électronique et les applications grand public, nous allons citer quelques unes [5] :

- ❖ **Le B2C pour Business to Consumers** renvoie à la vente au grand public de biens et de services qui peuvent être délivrés en ligne ou non. Amazon ou Dell constituent les archétypes de ce type de service sur le Web. Mais une grande partie des services Minitel relève également de cette logique. Les services B2C sont rendus le plus souvent par des intermédiaires commerciaux, qu'ils soient de nouveaux entrants comme Amazon ou des intermédiaires traditionnels, comme arrefour (Brousseau, 1999).

On observe cependant quelques sites où les industriels vendent directement leurs prestations aux consommateurs. Ils ne sont pas nombreux car, d'une part, industriels et prestataires de services n'ont pas toujours les compétences nécessaires pour vendre leurs produits et gérer leurs clientèles ; d'autre part, ils ne souhaitent pas nécessairement concurrencer les réseaux commerciaux avec lesquels ils coopèrent pour l'essentiel de leurs volumes d'affaires.

- ❖ **Le B2B pour Business to Business** renvoie aux transactions interentreprises. Il résulte soit d'accords bilatéraux entre deux partenaires commerciaux qui décident de s'échanger leurs informations via des médias électroniques, soit de services fournis par des prestataires spécialisés. Ces derniers offrent en général des prestations ciblées destinées à un secteur d'activité ou une industrie, compte tenu des spécificités de la coordination dans chaque segment de l'industrie, dans chaque pays ou zone économique. Ces prestataires sont la plupart du temps des émanations d'entreprises des secteurs où ils interviennent : filiales des opérateurs majeurs du secteur, comme dans

le cas du transport aérien ; joint-ventures entre les différents intervenants comme dans le cas du secteur de l'assurance automobile en France, etc. Parfois, des prestataires spécialisés dans la fourniture de services de communications « à valeur ajoutée » sont mandatés par des associations professionnelles ou des organes de standardisation pour développer des services ad hoc. Ainsi aux États-Unis, GEIS, filiale de General Electric, a développé dès la fin des années 70 des services pour faciliter le courtage entre compagnies pétrolières. Moins connus du grand public que les services de B2C, les services de B2B sont beaucoup plus anciens et représentent des volumes d'échanges beaucoup plus importants.

- ❖ **Le C2C pour Consumers to Consumers** renvoie aux systèmes destinés à supporter les échanges de particulier à particulier. On y trouve des services qui reproduisent le principe des petites annonces. L'interactivité de l'Internet a récemment conduit à mettre au point de véritables marchés électroniques reposant sur le principe des enchères. Ebay constitue l'archétype de ce type de service. Le C2C désigne désormais tous les services d'intermédiation entre particuliers, qu'il s'agisse de la vente d'occasion, de systèmes de trocs ressemblant aux SEL (Système d'Échanges Locaux), ou de services de mise en relation. Dans le passé, de telles applications étaient déjà apparues sur le Minitel. Dans la mesure où nombre de ces services ne génèrent pas à proprement parler d'activités marchandes, il n'est sans doute pas pertinent d'inclure l'ensemble de ces services dans le périmètre du commerce électronique.

En revanche, l'exemple de **Napster** ce système qui permet à tous les utilisateurs du logiciel de partager les données stockées sur les disques durs de chacun montre que de tels systèmes peuvent avoir des incidences importantes sur les activités commerciales.

- ❖ **Le C2B pour Consumers to Business** représente une tentative de renversement de la logique des rapports entre demande et offre. Son principe de base est de s'appuyer sur les réseaux électroniques pour consolider la demande des particuliers et mettre en concurrence les offreurs. Priceline.com, par exemple, permet aux particuliers de proposer les tarifs qu'ils sont prêts à payer pour un billet d'avion. Les compagnies aériennes décident d'honorer ou non ces commandes. Sur un autre registre, plusieurs services sont destinés à agréger les demandes individuelles de plusieurs consommateurs afin de bénéficier de prix de gros auprès des industriels.

3. Les applications agents pour le commerce électronique:

De nos jours, les applications agents pour le commerce électronique dont les conceptions varient beaucoup dû à la variété des tâches dans ce domaine, ne possèdent pas un standard commun concernant leur développement. C'est la raison pour laquelle, nous allons les grouper en deux qui n'est qu'un choix arbitraire [27].

3.1. Le premier groupe

Le premier groupe maintient les applications agents qui réduisent le temps de recherche pour un produit en assistant le consommateur. Certaines sont des assistants « intelligentes » qui fournissent des produits alternatifs, vu que des autres fournissent des marchands alternatifs pour un produit. Or, une petite quantité d'elles permettent les utilisateurs (les consommateurs et les vendeurs) la possibilité de négocier pour un produit entre eux. L'inconvénient des applications qui font partie de ce groupe est qu'ils se concentrent sur le prix comme critère, mais ne tient pas compte qu'il existe d'autres critères (comme la durée de la garantie ou l'efficacité d'un service après vente par exemple) pour prendre une décision pertinente. De plus, la négociation est faite entre agents dans une façon naïve qui est insatisfaisante.

Classification des applications agents du premier groupe :

Bien qu'il y ait plusieurs classifications des applications agents pour le commerce électronique, le **Comportement d'Achat de Consommateur CAC** (Consumer Buying Behaviour, CBB) [35], qui est basé sur six étapes fondamentales afin de guider le comportement d'achat du consommateur, peut nous permettre de classer les applications agents pour le commerce électronique existantes le plus formel possible.

Les six étapes sont :

- 1) L'identification du besoin
- 2) Le courtage de produit (product brokering)
- 3) Le courtage de marchand (merchant brokering)
- 4) La négociation**
- 5) L'achat et la livraison
- 6) Le service et l'évaluation

Lors d'un classement des applications agents, le courtage de produit, le courtage de marchand et la négociation sont les trois étapes les plus fréquemment utilisées. **Le courtage de produit** est la seconde étape où les consommateurs déterminent ce qu'ils vont acheter comme produit. On utilise soit un filtrage, soit un système de recommandation des produits se correspondant aux besoins de l'acheteur. Cette étape consiste à comparer les alternatives des produits. **Le courtage de marchand** est la troisième étape, dans laquelle on ne compare que des alternatives des marchands. La négociation est la quatrième étape où on détermine le prix et/ou les autres critères de la transaction. **La négociation** est utilisée dans le marché boursier, dans la vente aux enchères, et dans les magasins électroniques où on vend/achète des produits de second main.

3.2. Le deuxième groupe

Les applications agents du deuxième groupe se concentrent pour éviter les défauts de ceux de premier groupe : elles se basent sur les techniques complexes de la négociation. Leur but est d'automatiser la négociation qui va réduire le temps de négociation et permettre de faire les transactions massivement. Cette automatisation va aussi enlever l'inconvénient majeur d'une négociation se passant entre deux personnes : la réticence à cause de l'embarras ou la personnalité d'un négociateur.

Le problème avec ces applications est qu'il faut un espace commun pour que les agents puissent communiquer. Le fait de définir : un langage commun pour les agents et un technique de la négociation sont deux autres problèmes. La solution est d'utiliser un marché électronique commun : chacun (soit l'acheteur, soit le vendeur) crée son agent propre (agent acheteur, agent vendeur), ils communiquent entre eux (il y a souvent un agent qui assiste à la communication de ces deux agents) dans l'espace fournit par ce marché, et s'il y a accord, ils font une transaction.

4. Outils de programmation

4.1. *Choix du langage de programmation JAVA*

Java présente un langage de programmation développé par *Sun Microsystem*, ou les premières versions sont lancées dès 1995, il a réussi à intéresser et intriguer beaucoup de développeurs à travers le monde.

Le choix du langage de programmation Java nous est en quelque sorte dicté par les contraintes d'exécution de l'application. En effet, Java permet l'écriture d'applications un peu particulières que l'on appelle appliquestes ou plus communément applets, qui s'exécutent dans un navigateur supportant Java après avoir été chargées à partir d'un serveur Web.

Les applets sont écrites dans un vrai langage de programmation et possèdent un potentiel bien plus important que toute combinaison de HTML (HyperText Markup Language), XML (eXtensible Markup Language) ou toute autre forme de scriptage. Cela donne la possibilité d'exécuter de véritables applications sur tout ordinateur disposant d'un navigateur supportant Java, et ce, sans installation préalable de logiciel sur cette même machine.

Java permet aussi de développer des applications qui sont comparables à tout autre programme écrit dans un autre langage, au détail près qu'elles nécessitent comme pour les applets, une « machine virtuelle » pour leur exécution. La machine virtuelle Java interprète des classes Java exécutables qui sont générées par la compilation du code source. L'intérêt d'une telle démarche est de permettre l'exécution d'un même programme (sans recompilation) sur n'importe quel système informatique qui possède une machine virtuelle Java, et donc, de façon indépendante du système d'exploitation.

4.2. Choix de la plate forme JADE

JADE est une plate-forme multi-agents développée en entier en JAVA, créé par le laboratoire TILAB. Jade a pour but de simplifier le développement des systèmes multi-agents tout en fournissant un ensemble complet de services et d'agents conformes aux spécifications FIPA.

La plate-forme JADE inclut tous les composants obligatoires qui contrôlent un SMA. Ces composants sont l'ACC, l'AMS et le DF (voire *figure 3*).

1. Le Système de gestion d'Agent (AMS- Agent Management System)

Agent qui exerce le contrôle de supervision sur l'accès et l'usage de la plateforme ; il est responsable d'authentifier les agents résidents et de contrôler la les enregistrements.

2. Le Canal De communication (ACC- Agent Communication Canal)

Agent qui fournit le chemin pour les interactions de base entre les agents dans et en d'hors de la plateforme ; c'est la méthode de communication implicite qui offre un service fiable et précis pour le routage des messages ; il (l'agent) doit aussi être compatible avec le protocole IIOP () pour assurer l'interopérabilité entre les différentes plateformes.

3. Le facilitateur d'Annuaire (DF- Directory Facilitator)

Agent qui fournit un service de pages jaunes à la plateforme.

FIPA spécifie aussi le Langage de Communication d'agents (ACL- Agent Communication Language). La communication entre agents ne se fait que par envoi de messages seulement.

Le but de JADE est de simplifier le développement des systèmes multi-agents en conformité avec la norme FIPA pour réaliser des systèmes multi-agents interopérables. Pour atteindre ce but, JADE offre les caractéristiques suivantes :

- La plate-forme multi-agents compatible FIPA, qui inclut le **AMS**, **DF**, **ACC** – voir ci dessus. Ces trois agents sont automatiquement créés et activés à l'activation de la plate-forme.
- La plate-forme d'agents distribuée. La plate-forme d'agents peut être distribuée sur plusieurs machines, à condition qu'il n'y ait pas de pare-feu entre ces machines. Une seule application Java (Machine Virtuelle Java) est exécutée sur chaque machine. Les agents sont implémentés comme des threads d'exécution Java et les événements Java sont utilisés pour la communication efficace et légère entre agents sur une même machine. Un agent peut exécuter des tâches parallèles et JADE planifie ces tâches d'une manière plus efficace (et même plus simple pour le programmeur) que la planification faite par la Machine Virtuelle Java pour les threads d'exécution.
- Un certain nombre de DF (Facilitateurs d'Annuaire) compatibles FIPA peuvent être activés quand on lance la plate-forme pour exécuter les applications multi-domaines, ou le domaine est logique comme décrit dans FIPA97 Part1.
- Une interface de programmation pour simplifier l'enregistrement de services d'agents avec un ou plusieurs domaines (exemple : **DF**).

- Un mécanisme de transport et une interface pour l'envoi et la réception des messages de et vers les autres agents.
- Le protocole IIOP compatible avec le document FIPA97 pour connecter les différentes plates-formes multi-agents.
- Le transport léger de messages **ACL** sur la même plate-forme d'agents. Dans le but de simplifier la transmission, les messages internes (sur la même plate-forme) sont transférés et codés comme des objets Java et non comme des chaînes de caractères. Quand l'expéditeur ou le récepteur n'appartient pas à la même plate-forme, le message est automatiquement converti en chaîne de caractères spécifiés par la FIPA. De cette façon, la conversion est cachée au programmeur d'agents, qui a seulement besoin de traiter la classe d'objets Java.
- Une bibliothèque de protocoles d'interaction compatibles FIPA.
- L'enregistrement automatique d'agents dans le Système de Gestion d'Agents (**AMS**).
- Un service d'attribution de noms compatible FIPA ; quand on lance la plate-forme, un agent obtient un identificateur unique (Globally Unique Identifier - **GUID**).
- Une interface graphique utilisateur (présenté par *figure5.1*) pour gérer plusieurs agents et plates-formes multi-agents en partant d'un agent unique. L'activité de chaque plate-forme peut être supervisée et enregistrée.

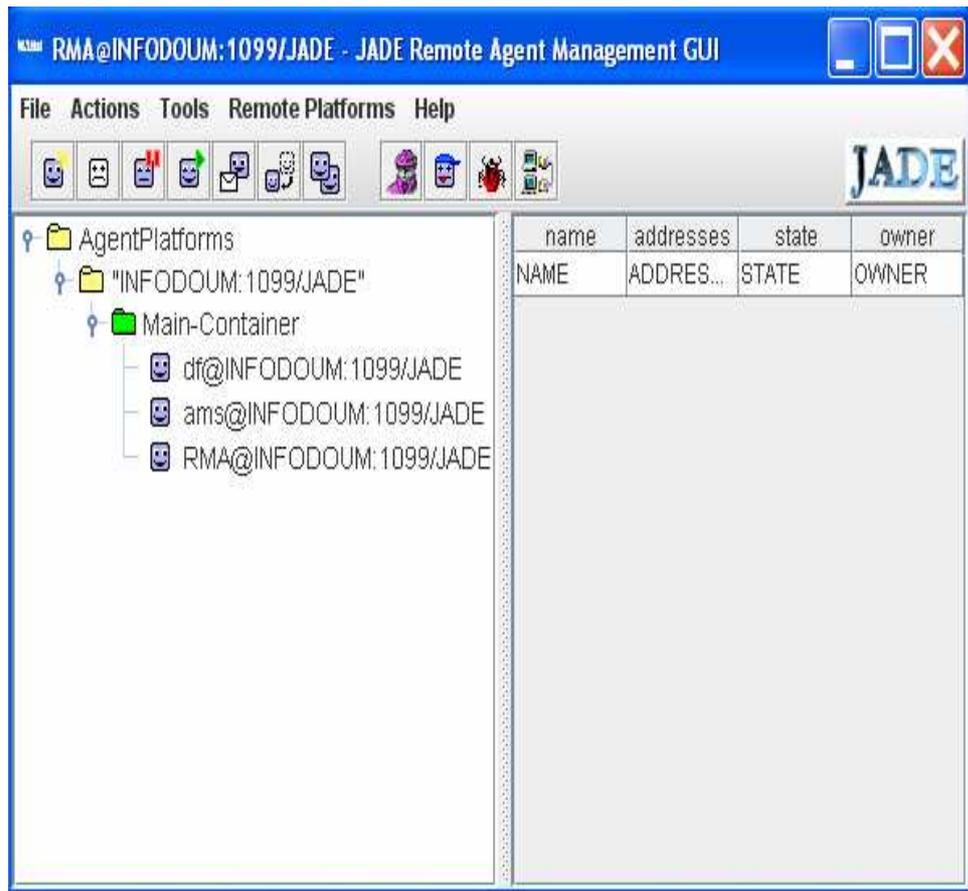


Figure 5.1 : L'interface graphique de la plate forme JADE

5. Description générale de l'application

Afin de montrer la validité, la fiabilité et l'extensibilité de notre architecture, nous allons faire une étude de cas d'où nous allons appliquer notre approche sur l'exemple de location des maisons, un exemple fréquemment utilisé et il est considéré comme un exemple typique pour les applications de négociation dans le commerce électronique.

Le scénario de notre exemple se déroule entre deux d'utilisateurs, un utilisateur propriétaire (initiateur) qui s'appelle *Abdelkader* et l'autre locataire (participant) *Mohamed* et notre système présente un espace de négociation virtuel pour ces deux utilisateurs, l'interface de notre application est présentée par la figure 5.2.



Figure 5.2 : Interface de notre système

Au début le propriétaire *Abdelkader Sahli* dépose une annonce de location d'une maison (VILLA) au niveau de la page **Annonce**, où il doit remplir un formulaire (voire figure 5.3) qui décrit son offre et ses souhaits et si ce propriétaire a envi de négocier ou non, si oui il doit choisir le bouton configuration de négociation sinon le bouton pas de négociation. Le choix de configuration de négociation oblige le propriétaire de remplir une autre fenêtre en précisant les marges de valeur des attributs sélectionnés dans le formulaire précédent et leurs importances c'est-à-dire leurs poids (voire figure 5.4).

Agent propriétaire

Soyez les biens venu Mr propriétaire , remplissez ce formulaire s'il vous plait pour vous enregistre

Nom Prénom Numéro telephone

E-mail

Mot de passe

Reentrez mot de passe

type de la maison à louer

Cochez et remplissez les attributs suivants:

<input checked="" type="radio"/> Prix de la location	<input type="text" value="100.000 DA"/>	<input type="radio"/> Paiement par mois ou semain	<input type="text"/>
<input checked="" type="radio"/> Période de la location	<input type="text" value="12 mois"/>	<input type="radio"/> bien meublé ou non	<input type="text"/>
<input checked="" type="radio"/> Avance de la location	<input type="text" value="6 mois"/>	<input type="radio"/> paiement par	<input type="text"/>

Figure 5.3 : Formulaire générant une offre d'un propriétaire

Configuration agent propriétaire

Attributs de l'offre:

Sélectionnez les attributs et leurs marges et importances pour configurer la négociation s'il vous plait.

Attributs	Marge minimale	Marge maximale	Importance
<input checked="" type="radio"/> Prix location	85.000 DA	100.000 DA	70
<input checked="" type="radio"/> Période location	6 mois	18 mois	20
<input checked="" type="radio"/> Avance location	6 mois	12 mois	10
<input type="radio"/> Paiement par moi ou semaine			
<input type="radio"/> Bien meublé ou non			
<input type="radio"/> Paiement par			

Lancer négociation Annuler

Figure 5.4 : Configuration de la négociation de l'agent propriétaire

De l'autre côté un locataire Mohamed Gasem cherche une maison à louer, alors il se dirige vers la page des annonces pour choisir l'offre convenable, s'il la trouvera il doit retourner à la page **Location** pour lancer une négociation avec le propriétaire qui a déposé cette annonce ou offre en remplissant un formulaire spécifiant le contre offre proposé par ce locataire (voire figure 5.4). Pour cet exemple nous allons proposer que l'offre convenable pour le locataire Mohamed Gasem soit lui-même proposé par Abdelkader.

De la même façon, le locataire Mohamed configure la négociation (voire figure 5.6) mais dans ce cas la réponse du propriétaire est le rejet car la contre offre est très loin ou n'appartient pas à l'intervalle d'utilité du propriétaire Abdelkader.

Le plus important dans notre application c'est que les deux cotés soit locataire ou propriétaire seront remplacé par deux agents intelligents pendant le processus de négociation.

Agent locataire

Soyez les biens venu Mr Locataire, remplissez ce formulaire s'il vous plait pour vous enregistrer

Nom: Gasem Prénom: Mohamed Numéro telephone: 033 73 35 14

E-mail: Gasem@yahoo.fr

Mot de passe: *****

Reentrez mot de passe: ***** Code de la maison: VILLA10

type de la maison à louer: VILLA

Cochez et remplissez les attributs suivants:

<input checked="" type="radio"/> Prix de la location	50.000 DA	<input type="radio"/> Paiement par mois ou semaine	
<input checked="" type="radio"/> Période de la location	6 mois	<input type="radio"/> bien meublé ou non	
<input checked="" type="radio"/> Avance de la location	0 mois	<input type="radio"/> paiement par	

Configuration négociation Pas de négociation

Figure 5.5 : Formulaire générant contre offre d'un locataire

Configuration agent locataire

Attributs de l'offre:

Sélectionnez les attributs et leurs marges et importances pour configurer la négociation s'il vous plaît.

Attributs	Marge minimale	Marge maximale	Importance
<input checked="" type="radio"/> Prix location	50.000 DA	90.000 DA	50
<input checked="" type="radio"/> Période location	6 mois	12 mois	10
<input checked="" type="radio"/> Avance location	0	0	40
<input type="radio"/> Paiement par moi ou semaine			
<input type="radio"/> Bien meublé ou non			
<input type="radio"/> Paiement par			

Lancer négociation Annuler

Figure 5.6 : Configuration de la négociation de l'agent locataire

6. Conclusion

Nous avons décrit et détaillé dans ce chapitre, l'implémentation de la négociation entre agents avec une étude de cas dans le domaine du commerce électronique. Il s'agit d'un modèle multi-attributs alternatif (échange d'offres et contre-offres) au lieu de se limiter à un seul attribut (le prix) comme il est le cas pour plusieurs modèles proposés. Les agents vont apprendre et modifier leurs croyances vis-à-vis des stratégies de leurs antagonistes au fur et à mesure que le système évolue dans le temps.

Conclusion générale et perspectives

Tout au long de ce mémoire, nous avons présenté les différentes technologies nécessaires pour proposer une approche formelle pour la négociation automatique entre agents. Donc nous nous sommes intéressés à la technologie d'agents intelligents et à son utilisation dans le domaine de négociation automatique.

Nous allons maintenant citer les objectifs atteints par notre projet qui sont :

- Définition d'une architecture de référence: notre architecture peut être utilisée pour concevoir différentes applications de négociations (selon le domaine de négociation suivie); et n'est pas propre à un seul projet. Dans ce cas, notre architecture sera essentiellement un cadre de développement.
- Définition d'un modèle de négociation multi-attributs intelligent et dynamique où les agents utilisent leurs croyances vis-à-vis de leurs antagonistes pour définir la tactique à appliquer au cours de processus de négociation.
- La proposition d'un nouveau protocole de négociation générique avec une nouvelle architecture d'agent négociateur.
- Une validation formelle de notre nouveau protocole de négociation par les RdPC.
- Une validation globale et exhaustive de notre architecture par la réalisation d'une instance de notre modèle en utilisant les outils les plus récents de programmation.

Pour nous la perspective la plus importante se présente par le délai de réponse qui n'est pas pris en compte dans notre travail. Il est à noter que notre protocole n'exige pas de l'agent (initiateur ou participant) de savoir quand il a reçu la réponse de l'autre agent antagoniste. Si un agent ne réussit pas à lancer un acte de proposition ou un acte de rejet, l'agent antagoniste peut potentiellement être mis en attente indéfiniment. Mais malheureusement à cause de contraintes de temps nous nous sommes pas arrivés à résoudre ce problème. Et pour éviter ce problème, nous donnons comme proposition (comme extension de ce travail) que l'appel d'offre contienne comme paramètre un délai d'attente d'une réponse avant lequel les réponses doivent être reçues par l'agent. Les réponses (accord, rejet, contre offre etc ...) reçues après le délai d'attente sont automatiquement rejetées avec comme justification que la

réponse a été faite trop tardivement. Pour bien résoudre ce problème nous allons proposer que si un délai est expiré l'agent qui attend la réponse doit arrêter la négociation par un rejet qui doit l'envoyer à l'agent antagoniste.

Bibliographie

- [1] **Austin, J. L.** 1962. « *How to Do Things with Words* ». Oxford University Press, Oxford, 1962.
- [2] **Babau, Jean-Philippe.** « *Le langage SDL pour les systèmes temps réel* ». CITI : Centre d'Innovation en Télécommunication et Intégration de services, INSA Lyon, F69621 Villeurbanne Cedex – France, 2003.
- [3] **Barbuceanu, Mihai et Fox, Mark S.** « *COOL: A Language for Describing Coordination in Multi Agent Systems* ». Enterprise Integration Laboratory, Department of Industrial Engineering, University of Toronto, 1995.
- [4] **Bourdon, François et al.** « *Systèmes multi-agents* »: Cours exposé dans le cadre du DEA-IAA-(TRONC COMMUN), Université de CAEN.
- [5] **Brousseau, Éric.** « *Commerce électronique : ce que disent les chiffres et ce qu'il faudrait savoir* », ParisX, 2000.
- [6] **Chaib-draa, B, Jarras, I et Moulin, B.** « *Systèmes multiagents: Principes généraux et applications* » : Agent et systèmes multiagents . Canada : s.n, 2001.
- [7] **Chaib- Draa, Brahim.** « *Systèmes multi agents* ». Version 1.0, 03 Janvier 2008.
- [8] **Courtia, Jean pierre et all.** « *Estelle: un langage ISO pour les algorithmes distribués et les protocoles* ». INRIA, Décembre 1986.
- [9] **Davies, Jim , Jim Woodcock.** «*Using Z* », Prentice Hall, 1996.
- [10] **Fadil, Hind et Koning, Jean-Luc. 2006.** « *Vers une spécification formelle des protocoles d'interaction des systèmes multi-agents en B* ». MOSIM'06 avril 2006 - Rabat – Maroc.
- [11] **Ferber, Jaques.** « *Les Systèmes Multi Agents: vers une intelligence collective* », Interédition 1995.
- [12] **Ferber, Jacques.** « *Technologie multi-agent* ». Université Pière et Marie Curie (ParisVI), 1999.
- [13] **Flavien Serge Mani Onana.** « *Vie privée en commerce électronique* », Thèse présentée pour l'obtention de grade PHD en informatique Montréal.
- [14] **Fourmanoit, S. E, Merlo. R, Roy.** « *Introduction au langage Z Cours LOG2000 - Eléments du génie logiciel* », Ecole Polytechnique de Montréal 2003

- [15] **Gamoura-Chehbi, Samia.** « *Modélisation et Simulation des Processus Collaboratifs dans les Chaînes Logistiques : Une Approche Multi Agents* ». Ecole Doctorale : Informatique et Information pour la Société Laboratoire LIESP (Laboratoire d'Informatique pour l'Entreprise et les Systèmes de Production), Thèse de docteur de l'université Lumière Lion II.
- [16] **Garavel, Hubert.** « *Compilation et vérification de programmes LOTOS* ». Université Josef Fourier- Grenoble I. Pour obtenir le grade de DOCTEUR. novembre 1989.
- [17] **Garavel, Hubert.** « *Introduction au langage LOTOS* ». Verilog Rhone-Alpes Centre d'Etudes Rhone-Alpes.
- [18] **Huget, Marc-philippe.** « *Une ingénierie des protocoles d'interaction pour les systèmes multi-agents* ». Thèse de docteur de l'université paris 9 Dauphine, juin 2001.
- [19] **Jarraya, Tarek.** « *Réutilisation des protocoles d'interaction et Démarche orientée modèles pour le développement multi-agents* ». Thèse de docteur de l'Université de Reims Champagne Ardenne en Informatique, décembre 2006.
- [20] **Jenings J.R.** « *Commitments and Conventions: The foundation of coordination in multi-agent systems* ». The knowledge Engineering review, 2(3): pages 223-250, 1993.
- [21] **Jennings, N. R et all.** « *A Service Oriented Negotiation model between Autonomous Agents* ». Dept Electronic Engineering, Queen Mary and Westfield College, university of London E1 4NS,UK. 1997.
- [22] **Mathieu, Philippe et Marie-Hélène, Verrons.** « *GeNCA : Un modèle général de négociation de contrats* ». LIFL (CNRS - UMR 8022) Université des Sciences et Technologies de Lille, 2005.
- [23] **Mazouzi, Hamza.** « *Ingénierie des protocoles d'interaction : des systèmes distribués aux systèmes Multi-agents* » : thèse de docteur de l'université paris 9 Dauphine, octobre 2001.
- [24] **Melville, P. R.J. Money and R. Nagarajan,** « *Content-Boosted Collaborative Filtering for improved Recommendations* », Proceeding of Eighteenth national conference on artificial intelligence, PP. 187-192, Edmonton, Canada, July 2002.
- [25] **Morge, Maxime.** « *Système dialectique multi-agents pour l'aide à la concertation* ». Thèse de docteur de l'école Nationale Supérieure des Mines de Saint Etienne et de l'Université Jean Monnet, 20 juin 2005.
- [26] **Pasquier, Philipe.** « *Aspects cognitifs des dialogues entre agents artificiels (L'approche par la cohérence cognitive)* ». Thèse de docteur de l'université Lava Québec Canada, aout 2005.

- [27] **Sansonnet, J-P.** « *Applications agents pour le commerce électronique Systèmes Multi-Agents* ».
- [28] **Sassi, Mourad et Chaib-draa, Brahim.** « *Stratégies de négociation entre agents dans le domaine du transport* ». Département d'Informatique, Faculté des Sciences et de Génie, Université Laval, Ste-Foy, PQ, Canada, G1K 7P4.
- [29] **Searle, J. R.** « *Speech Acts: An Essay in the Philosophy of Language* ». Cambridge University Press, Cambridge, London, 1969.
- [30] **Shaheen S. Fatima, Michael Woodbridge, Nicholas R. Jennings.** « *An agenda-based framework for multi-issue negotiation* ». Department of Computer Science, University of Liverpool, Liverpool L69 7ZF, UK : Department of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK, 2003.
- [31] **Spivey, John M.** « *The Z notation : a reference manual* », Prentice Hall, 1998.
- [32] **Turban, E . D. King, J Lee and D. Viehland,** « *Electronic commerce: A Managerial perspective* », 3rd, edition, Prentice Hall, 2004.
- [33] **Valerie, B.** « *Le pilotage et la conduite des systèmes de production décentralisés dans un environnement dynamique* », 2003.
- [34] **Verron, Marie-hélène.** « *GeNCA : un modèle général de négociation de contrats entre agents* ». Thèse de doctorat de l'Université des Sciences et Technologies de Lille (spécialité informatique), 2004.
- [35] **Wurman, P. M. Wellman, and W. Walsh,** « *The Michigan Internet AuctionBot: A Configurable Auction Server for Human and Software Agents* ». In proceedings of the Second International Conference on Autonomous Agents (Agents 98), May 1998.