

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Universitaire Mentouri de Constantine
Faculté de science et de l'ingénierie
Département Informatique

N° d'ordre : 354/ Mag /2009
Série : 011/ Inf /2009

Mémoire
Présenté en vue de l'obtention du diplôme de
Magister en informatique

Option : Systèmes distribués

Logique floue et SMA : Aide à la décision floue dans les systèmes multi-agents

Présenté par : Mr. DJAABOUB Salim

Directeur de thèse: Mr SAHNOUN Zaidi (Prof. Université Mentouri de Constantine)

Composition du jury

Président du jury: Mme. BELALA Faiza (M.C Université Mentouri de Constantine)

Rapporteur : Mr. MAAMRI Ramdane (M.C Université Mentouri de Constantine)

Examineurs : Mr. KAZAR Okba (M.C Université Khider de Biskra)

Mme. ZEGHIB Nadia (M.C Université Mentouri de Constantine)

Résumés

Résumé : Les Systèmes Multi-Agents (SMA) constituent actuellement une nouvelle approche pour l'analyse, la conception et la réalisation de systèmes complexes modernes, de ce fait, ils connaissent un intérêt croissant, dû à l'ampleur des problèmes abordés et aussi aux résultats encourageants obtenus. Cependant la mise en œuvre de systèmes modernes pose souvent des problèmes de représentation et de manipulation d'informations imparfaites, qui implique la nécessité de doter les SMA des techniques de représentation et de manipulation de connaissances leur permettent de tenir en compte cette imperfection. La voie que nous avons choisie pour remédier à ce problème est de doter les SMA des techniques floues de représentation et de manipulation d'informations. En effet, l'objet de nos travaux est de proposer un modèle de conception de SMA basé sur la logique floue et ses techniques, pour répondre aux problèmes auxquels sont confrontés de nombreux systèmes complexes, qui doivent traiter des informations qui sont de nature imparfaite. Nous présentons dans cette thèse une nouvelle approche pour la conception de systèmes multi-agents flous, le modèle FMAS (Fuzzy Multi-Agent System), et nous illustrons nos travaux sur une application d'aide à la décision.

Mots clés : Systèmes multi-agents, logique floue, Systèmes multi-agents flous, aide à la décision.

Abstract: Multi-Agents Systems (MAS) constitute currently a new approach for the analysis, design and implementation of modern complex systems, due to this fact; they receive an increasing interest, due to the wide range of the problems dealt with, and to the encouraging results obtained. However, the implementation of these systems often address problems of representation and manipulation of imperfect information, which imply the need for providing SMA with techniques of representation and knowledge manipulation, allowing them to take into consideration this imperfection. The approach we have chosen in order to remedy to this problem is to provide SMA with techniques of representation and manipulation of fuzzy information. In fact, the purpose of our study is to propose a sample for the conception of SMA based on fuzzy logic and its techniques, in order to resolve the problems that many of the complex systems are facing, that they have to process information which is of imperfect nature. We present in this thesis a new approach for conception of the fuzzy multi-agents systems, the FMAS model (fuzzy Multi-Agent System), and we are illustrating our works on an application of help to the decision.

Key Word: Multi-agent system, fuzzy logic, fuzzy multi-agent system, decision support.

ملخص: الأنظمة المتعددة الوكلاء تشكل حاليا نهجا جديدا لتحليل وتصميم وتنفيذ النظم الحديثة المعقدة، نظرا لهذه الحقيقة، فهي تلقي اهتماما متزايدا، وذلك بسبب مجموعة واسعة من المشاكل التي تمت معالجتها، وإلى النتائج المشجعة التي تم الحصول عليها. ومع ذلك، فإن انجاز هذه الأنظمة في كثير من الأحيان يطرح مشاكل معالجة و تمثيل المعلومات المنقوصة، مما يعبر عن الحاجة إلى تزويد هذه الأنظمة بتقنيات لتمثيل ومعالجة المعارف، لكي نسمح لهم الأخذ بعين الاعتبار هذا النقص. النهج الذي اخترناه من أجل علاج هذه المشكلة هو تزويد هذه الأنظمة بتقنيات تمثيل ومعالجة المعلومات الغامضة. في واقع الأمر ، فإن الغرض من دراستنا هو اقتراح نموذج لتصميم الأنظمة المتعددة الوكلاء على أساس المنطق الغامض وتقنياته ، من أجل حل مشاكل الكثير من الأنظمة المعقدة التي يجب عليها معالجة المعلومات التي هي من الطبيعة المنقوصة. نقدم في هذه الأطروحة نهجا جديدا لتصميم الأنظمة المتعددة الوكلاء الغامضة، النموذج (Fuzzy Multi-Agent System) FMAS، و سنستعمله في تطوير نظام للمساعدة على اتخاذ القرارات.

الكلمات المفتاح : الأنظمة المتعددة الوكلاء، المنطق الغامض، الأنظمة المتعددة الوكلاء الغامضة، المساعدة في اتخاذ القرارات.

Remerciement

Louange à dieu. Bienfaiteur miséricordieux.
Paix et bénédiction sur son prophète, Mohamed, ultime envoyé.

Je tiens avant toutes choses à remercier mon directeur de thèse, monsieur le professeur Zaidi SAHNOUN pour son aide et ses conseils. Je le remercie de la confiance qu'il m'a témoigné en m'intégrant au sein de son équipe. Mes remerciements vont aussi à Mr : MAAMRI Ramdane qui m'a encadré depuis le départ de Mr Sahnoun, pour avoir accepté de rapporter ce mémoire.

Mes remerciements vont également à ceux qui m'ont fait l'honneur d'être les membres de jury :

Mme Faiza BELALA, maître de conférences à l'Université Mentouri de Constantine, pour m'avoir fait l'honneur de bien vouloir présider le jury.

Mme Nadia ZEGHIB, maître de conférences à l'Université Mentouri de Constantine, ainsi que monsieur Okba KASAR, maître de conférences à l'Université Khider de Biskra, pour avoir accepté de rapporter ce manuscrit.

Je tiens à saluer les membres du Laboratoire LIRE, notamment ceux de l'équipe GL & IA. Je les remercie pour leur accueil et les innombrables échanges que nous avons eu durant ces années intenses.

Finalement, je remercie de façon spéciale ma famille : mes parents, mes sœurs et mes trois frères. Je remercie mes parents pour m'avoir toujours motivé à poursuivre mes études et pour avoir su me soutenir toutes ces années.

Table des matières

Introduction générale	01
Chapitre 1: Logique floue	04
1.1 Introduction	04
1.2 Logique classique et logique floue	05
1.3 Concepts et définitions	06
1.1.1 Sous ensembles flous	06
1.1.2 Variables linguistiques	09
1.1.3 Relations floues	10
1.1.4 Propositions floues	11
1.4 Inférence floue	13
1.4.1 Modus ponens généralisé	13
1.4.2 Méthodes d'inférence floue	14
1.5 Applications de la logique floue	15
1.5.1 Les systèmes de commande et les systèmes experts flous	15
1.5.2 L'aide à la décision	18
1.5.3 L'interrogation flexible de base de données	20
1.5.4 Le filtrage flou	21
1.6 Conclusion	21
Chapitre 2: Systèmes multi-agents	23
2.1 Introduction	23
2.2 Concepts et définitions	24
1.2.1 Concept d'agent	24
1.2.2 Systèmes multi-agents	25
1.2.3 Interaction entre agents	26
1.2.4 Communication entre agents	27

1.3	Principe de programmation.	28
1.3.1	Programmation orientée agent	28
1.3.2	Quelques plateformes de développement de SMA	29
1.4	Exemples de domaines d'application des SAM.	30
1.4.1	Systèmes complexes.	31
1.4.2	Systèmes d'information coopératifs.	31
1.4.3	Systèmes d'aide à la décision.	32
1.4.4	Commerce électronique et les agents du Web	32
1.5	Conclusion	33
Chapitre 3: Des approches d'aide à décision utilisées dans les SMA		35
3.1	Introduction	35
3.2	L'approche floue.	36
3.2.1	Pourquoi la logique floue avec les SMA	36
3.2.2	Les systèmes multi-agents flous.	37
3.3	Les autres approches	38
3.3.1	Les systèmes experts classiques	39
3.3.2	Les approches probabilistes	40
3.3.3	Modèles issus du «Soft computing».	41
3.3.3.1	Les réseaux de neurones	42
3.3.3.2	Les algorithmes évolutionnaires	44
3.3.3.3	Les approches hybrides.	45
3.4	Discussion	46
3.5	Conclusion	47
Chapitre 4: FMAS : Un modèle pour la réalisation de SMA Flou		48
4.1	Introduction	48
4.2	Vue globale du modèle	49
4.3	Fonctionnement générale	50
4.4	Agents flous du modèle.	51
4.4.1	Agents experts	51
4.4.2	Agents de choix	53
4.4.3	Agents de requête	55

4.5	Interaction et communication entre agents	57
4.5.1	Modélisation des interactions entre agents	57
4.5.2	Communication entre agents	58
4.6	Conclusion	59
Chapitre 5: Modélisation et réalisation d'une application d'aide à la		
décision dans un marché virtuel		60
5.1	Introduction	60
5.2	Présentation et objectif	61
5.3	Modélisation du système suivant le modèle FMAS.	62
5.3.1	Architecture du système	62
5.3.2	Agents flous utilisés.	63
5.4	Scenarios d'exécution	64
5.4.1	Scenario 1	64
5.4.2	Scenario 2	68
5.5	Implémentation	70
5.5.1	Choix techniques	70
5.5.2	Interfaces utilisateurs	71
5.5.3	Implémentation des agents par Jade	73
5.5.4	Communication entre agents	76
5.5.5	Simulation du système avec Jade	76
5.6	Conclusion	78
Conclusion générale		79
Bibliographie		81

Table des figures

Fig 1.1 : Classification des températures d'une pièce en deux sous ensembles.	06
Fig 1.2 : Représentation de l'âge par des sous-ensembles flous.	07
Fig 1.3 : Variable linguistique (V, T(V), X) pour décrire la température.	09
Fig 1.3 : Représentation de la relation x approximativement égal à 3.	10
Fig 1.5 : Valeur de vérité de la proposition floue p : la température est tiède.	11
Fig 1.6 : Structure d'un système flou.	16
Fig 1.7 : La défuzzification par méthode du centre de maximum.	18
Fig 1.8 : La défuzzification par la méthode du centre de gravité.	18
Fig 2.1 : Représentation imagée d'un agent en interaction avec son environnement et les autres agents [Ferber, 1995].	26
Fig 3.1 : Correspondance entre neurone biologique et neurone formel.	42
Fig 3.2 : Agent neural du projet AgNet.	43
Fig 4.1 : Vue globale du modèle FMAS.	49
Fig 4.2 : Fonctionnement générale du modèle.	50
Fig 4.3 : Structure du module de traitement des agents experts.	52
Fig 4.4 : Module d'évaluation des agents de choix.	54
Fig 4.5 : Agent de requête.	55
Fig 4.6 : Processus de traduction d'une requête floue en une requête simple.	56
Fig 4.7 : Diagramme de protocoles d'interactions entre agents.	58
Fig 5.1 : Architecture du marché basé agent	62
Fig 5.2 : Agent d'évaluation.	63
Fig 5.3 : Agent Expert Immobilier.	64
Fig 5.4 : Diagramme de protocole d'interaction pour le scenario 1.	65
Fig 5.5 : Variables linguistiques pour décrire la surface d'un l'apparement.	66
Fig 5.6 : Variables linguistiques pour décrire le loyer	66

Fig 5.7: Variables linguistiques pour décrire la distance à une ville.	67
Fig 5.8: Processus de dérivation de la requête du client en une requête simple.	67
Fig 5.9 : Résultat d'exécution de la requête du client	68
Fig 5.10 : Résultat d'évaluation des propositions par l'agent d'évaluation.	68
Fig 5.11: Les partitions floues utilisées pour calculer le prix	69
Fig 5.12 : Fenêtre de connexion.	70
Fig 5.13 : Exemple d'interface client.	71
Fig 5.14 : Exemple d'interface coté fournisseur.	72
Fig 5.16: Spécification partielle d'un agent exprimée avec JADE.	73
Fig 5.17: Spécification partielle du module de communication exprimée avec JADE	74
Fig 5.18 : Spécification partielle du module de traitement exprimée avec Jade.	75
Fig 5.19 : Spécification partielle du module d'évaluation avec Jade.	75
Fig 5.20 : Spécification partielle d'envoi d'un message dans Jade	76
Fig 5.21 : Le GUI de l'agent RMA.	77
Fig 5.21 : Simulation des interactions entre agents par l'agent Sniffer.	77

Introduction générale

Les systèmes modernes comme dans l'industrie, l'économie, le marketing et l'écologie reposent de plus en plus, souvent, sur des systèmes informatiques, qui sont chargés à aider les utilisateurs au traitement des problèmes et à la prise de décision. La complexité de systèmes informatiques devant être mis en place amène beaucoup de chercheurs à utiliser une modélisation reposant sur le paradigme multi-agents pour la réalisation de ces systèmes [Ferber, 1995] [Wooldridge and Jennings, 1994]. Cette approche permet en effet de tenir compte de la complexité de ces systèmes en extrayant les entités d'actions qui doivent être présentés dans la modélisation du système.

La mise en œuvre de systèmes modernes (avec les problèmes de communications homme machine qui les accompagnent) se caractérise souvent par la constante manipulation d'informations dont beaucoup sont imprécises, vagues, incertaines et incomplètes [Aliev, 2008] [Fazlollahi et al, 2004]. Or, l'utilisation du paradigme multi-agents implique la nécessité de y'intégrer des mécanismes lui permettent de tenir en compte de l'incertitude, de l'imprécision et de l'incomplétude d'information. Il est connu que la logique floue est le seul cadre dans lequel puissent être traitées des imprécisions et des incertitudes, qui autorisent également le traitement de certaines incomplétudes. C'est aussi le seul cadre dans lequel puissent être traitées des connaissances numérique et des connaissances exprimées symboliquement par des qualifications du langage naturel [Bouchon, 1995]. Cela rend nécessaire la conception de systèmes multi-agents (SMA) dotés des caractéristiques floues.

Les nombreuses applications de la logique floue qui ont été réalisées dans le monde entier ont été prouvé leur efficacité pour résoudre divers types de problèmes dans lesquels les connaissances disponibles sont imparfaites. La technologie multi-agent avec ses caractéristiques constitue le meilleur moyen pour la modélisation et la réalisation de

systèmes complexes distribués. La combinaison entre ces deux technologies doit ouvrir donc une nouvelle voie de recherche pour la conception et la réalisation de systèmes modernes qui sont souvent des systèmes complexes, composés avec plusieurs entités interagissant en mode distribué, et en outre ils sont caractérisés par l'imprécision et l'incertitude [Fazlollahi et al, 2004] [Aliev et al, 1999].

Dans le cadre de notre travail, notre objectif est la prise en compte de l'aspect flou dans les systèmes à base d'agents dans le but de pouvoir gérer des applications dans lesquelles certaines des connaissances conduisent à la décision sont imparfaites, ce qui est souvent le cas dans beaucoup des applications. En effet, l'objet de nos travaux est d'enrichir les SMA suffisamment pour que leur comportement soit acquérir des caractéristiques floues. Cela signifie que le comportement du système doit être en mesure, en particulier, de traiter des problèmes avec des connaissances imparfaites, de prendre des décisions (ou aider les décideurs dans leurs décisions) sous incertitude et avec des critères et des objectif flous, et de répondre à des requêtes des utilisateurs exprimées par des qualifications linguistiques vagues et imprécises.

Nous présentons dans ce mémoire, une nouvelle approche pour la prise en compte de l'aspect flou dans les systèmes multi-agents. Elle consiste essentiellement en la conception d'un modèle de SMA basé sur la logique floue: FMAS (abréviation de Fuzzy Multi-agent System). Nous allons illustrer nos travaux sur un problème de prise de décision dans une application de gestion d'un marché électronique, pour aider les utilisateurs à effectuer des transactions commerciales dans des conditions les meilleurs possibles. Ce mémoire sera organisé en cinq chapitres:

Dans le premier chapitre, consacré à la logique floue, nous commençons par énoncer les fondements de la logique floue. Nous voyons comment elle permet d'exprimer selon un formalisme unique des informations très diverses (données incertaines ou imprécises, connaissances exprimées sous forme linguistique, . .). Ensuite, nous présentons en détails les méthodes de raisonnement flou (inférence floue) qui constituent la base de la commande floue et de systèmes experts flous. Enfin, nous décrivons les applications les plus utilisées de la logique floue.

Dans le chapitre 2, nous présentons les systèmes multi-agents, en particulier les modèles sur lesquels nous fondons nos travaux. Ce chapitre présente les notions qui seront utilisées tout au long du document.

Le chapitre 3 sera consacré à l'étude des différentes approches de raisonnement et d'aide à la décision utilisées dans les SMA, et les travaux relatifs aux systèmes multi-agents flous. Ce chapitre a pour objectif de souligner l'originalité de nos travaux.

Le chapitre 4 présente le modèle FMAS : un modèle de conception pour la réalisation de systèmes multi-agents flous. Ce chapitre constitue la contribution de nos travaux présentés dans cette thèse.

Le chapitre 5 illustre l'application du modèle FMAS dans une étude de cas. Nous décrivons tout d'abord l'utilisation de notre modèle pour la modélisation et la réalisation d'une application de gestion d'un marché virtuel. Nous précisons ensuite la plateforme d'implémentation JADE et comment les différents agents flous issus du modèle FMAS peuvent être implémentés en utilisant cette plateforme.

Enfin, le mémoire se termine par une conclusion générale qui récapitule les travaux réalisés et propose des visions pour les travaux futurs.

Chapitre 1

Logique floue

1.1 Introduction

Dans le fonctionnement de l'esprit humain, les incertitudes sont particulièrement remarquables. La capacité d'établir des classes d'éléments de la nature ayant des propriétés analogues est très naturelle chez l'homme. Il sait déterminer l'âge approximatif d'un individu en l'observant. Il sait aussi rendre compte de données vagues «*Large ouverture*», imprécises «*de 3 à 5 m*», mal définies «*massif altéré*», dont la validité n'est pas absolue «*dans 85% des cas*», soumises à une incertitude «*très probable*». Il est tout aussi naturel à l'homme de traiter des données affectées d'incertitude, inhérentes à l'univers ou dues à sa méconnaissance de certains facteurs, que d'utiliser des critères subjectifs, donc imprécis [Bouchon, 1995].

Le souci d'automatiser ou d'assister de façon automatique les actions humaines, naturellement empiriques et empreintes d'imprécisions, dans le cadre d'une aide à la décision ou du contrôle par exemple, renforce l'intérêt des scientifiques pour l'approche floue et justifie son intense développement au cours de ces dernières années. C'est pour les possibilités qu'elle offre de gérer l'incertitude et l'imprécision, que nous nous sommes intéressés à la logique floue.

Les bases théoriques de la logique floue ont été formulées en 1965 par le professeur Lotfi A. Zadeh, de l'Université de Berkeley en Californie [Zadeh, 1965]. Il a introduit la notion de sous-ensemble flou pour répondre aux problèmes auxquels sont confrontés de nombreux systèmes complexes, qui doivent traiter des informations qui sont de nature imparfaite, son concept de base est de graduer l'appartenance à un ensemble, c'est un moyen efficace pour prendre en compte l'imprécision dans la connaissance et de formaliser le processus de raisonnement humain. De nombreuses applications basé sur la logique

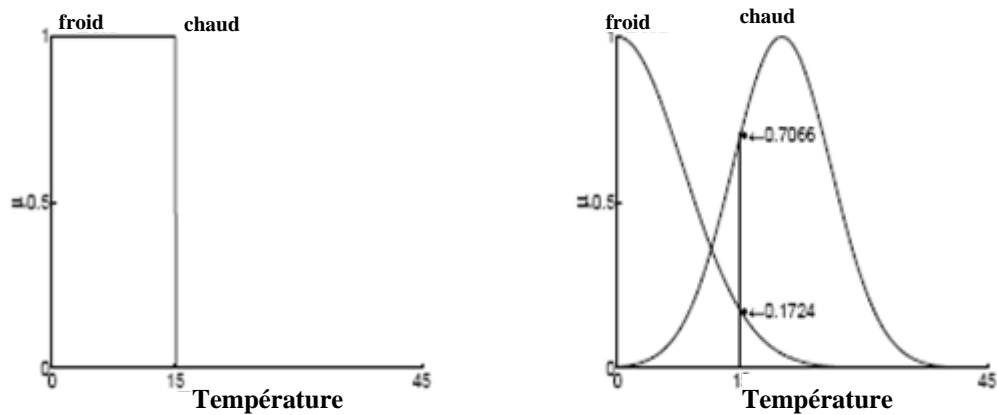
floue sont développées dans divers domaines, dans lesquelles aucun modèle déterministe n'existe ou n'est pratiquement implémentable. Ainsi que dans des situations pour lesquelles l'imprécision sur les données rend le contrôle par des méthodes classiques impossible.

1.2 Logique classique et logique floue

Dans le cadre de la logique classique, une proposition est soit vraie, soit fausse (1 ou 0). Par exemple, la logique classique peut facilement partitionner la température d'une pièce en deux sous-ensembles, «moins de 15 degrés» et «15 degrés ou plus». La figure 1.1a montre le résultat de cette partition. Toutes les températures de moins de 15 degrés sont alors considérées comme appartenant à l'ensemble «moins de 15 degrés». On leur affecte une valeur de 1. Toutes les températures atteignant 15 degrés ou plus ne sont pas considérées comme appartenant à l'ensemble «moins de 15 degrés». On leur attribue une valeur de 0.

Cependant, le raisonnement humain s'appuie fréquemment sur des connaissances ou des données inexacts, incertaines ou imprécises. Une personne placée dans une pièce dont la température est soit de 14.95 degrés soit de 15.05 degrés, ne fera certainement pas de distinction entre ces deux valeurs. Cette personne sera pourtant capable de dire si la pièce est «froide» ou «chaude», sans pour cela utiliser de température limite ni de mesure précise. La logique floue permet de définir des sous-ensembles, comme «froid» ou «chaud», en introduisant la possibilité pour une valeur d'appartenir plus ou moins à chacun de ces sous-ensembles.

La logique floue permet de faire intervenir les notions d'imprécision et d'incertitude dans un système. Cela permet par exemple de faire intervenir une température «d'environ 15 degrés» dans un contrôleur flou. L'incertitude et l'imprécision peuvent également être prises en compte dans le cadre de la logique floue quand on utilise une connaissance issue d'un expert humain. Comment pourrait-on utiliser avec des outils standards une connaissance humaine du genre : «il pleut souvent en hiver» ?



(a) Deux sous-ensembles selon la logique classique (b) Deux sous-ensembles selon la logique floue

Fig. 1.1 : Classification des températures d'une pièce en deux sous ensembles

1.3 Concepts et définitions

1.3.1 Sous ensembles flous

La notion de sous-ensembles flous a pour but de permettre des gradations dans l'appartenance d'un élément à une classe, c'est-à-dire d'autoriser un élément à appartenir plus ou moins à une classe prédéfinie, «plus la surface d'un appartement se rapproche de $200m^2$, plus son appartenance à la classe des grands appartements est forte». Cette notion permet l'utilisation de catégories aux limites mal définies «Large», de situations intermédiaires entre le tout et le rien «presque ouvert», le passage progressif d'une propriété à une autre «passage de peu profond à profond», l'utilisation de valeurs approximatives «environ 12 mètres».

Cette présentation évite l'utilisation arbitraire de limites rigides à des classes et répond au besoin de présenter des connaissances imprécises, soit parce qu'elles sont exprimées en langage naturel «on utilise aussi le terme variable linguistique», soit parce qu'elles sont obtenues avec des instruments d'observations qui produisent des erreurs de mesure.

Définition

Étant donné un ensemble de référence X , on peut indiquer les éléments de X qui appartiennent à une certaine classe de X et ceux qui n'y appartiennent pas. Cette classe est

alors un sous-ensemble de X. Si l'appartenance de certains éléments de X à une classe n'est pas absolue, on peut indiquer avec quel degré chaque élément appartient à cette classe. Celle-ci est alors un sous-ensemble flou de X [Bouchon, 1995].

Définition : Un sous-ensemble flou A dans un univers du discours X est caractérisé par sa fonction d'appartenance $\mu_A(x)$ qui associe à chaque élément x de X une valeur dans l'intervalle des nombres réels [0, 1].

$$\mu_A : X \rightarrow [0,1] \quad (1.1)$$

Ainsi un sous-ensemble flou A dans X peut être représenté par un ensemble de couples ordonnés :

$$A = \{(x, \mu_A(x)); x \in X, \mu_A(x) \in [0, 1]\} \quad (1.2)$$

Exemple : Pour évaluer la hauteur (H) d'une personne en utilisant des termes linguistiques, nous devons leur associer des fonctions de représentation. Nous présentons dans cet exemple, l'évaluation de la hauteur (H) d'une personne selon la logique floue. Pour cela, nous pouvons utiliser les mots, PETIT et GRAND. La représentation de PETIT et GRAND se fait au moyen de deux fonctions qui calculent l'appartenance de chaque personne à l'ensemble flou PETIT et/ou GRAND. Ainsi, une personne qui mesure 1m63 appartient en même temps au sous-ensemble flou PETIT et au sous-ensemble flou GRAND avec un degré de 0,7 (pour PETIT) et de 0,3(pour GRAND). Un sous-ensemble peut être représenté graphiquement par un trapèze ou une fonction plus complexe. La figure 1.2. Présente une forme de représentation.

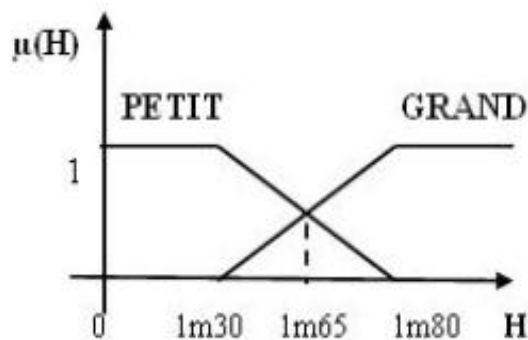


Fig 1.2 : Représentation de l'âge par des sous-ensembles flous

Opérations sur les sous ensembles flous

Supposons que A et B sont deux sous-ensembles flous définis dans un univers du discours X par les fonctions d'appartenance μ_A et μ_B . On peut définir des opérations ensemblistes telles que l'égalité, l'inclusion, l'intersection, l'union et le complément grâce à des opérations sur les fonctions d'appartenance.

Egalité : A et B sont dits égaux, propriété que l'on note $A = B$, si leurs fonctions d'appartenance prennent la même valeur en tout point de X :

$$\forall x \in X \quad \mu_A(x) = \mu_B(x) \quad (1.3)$$

Inclusion : A est dit inclus dans B, propriété que l'on note $A \subset B$, si tout élément x de X qui appartient :

$$\forall x \in X \quad \mu_A(x) \leq \mu_B(x) \quad (1.4)$$

Intersection : L'intersection de A et B, que l'on note $A \cap B$, est le sous-ensemble flou constitué des éléments de X affectés du plus petit des deux degrés d'appartenance μ_A et μ_B .

$$\forall x \in X \quad \mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)) = \mu_A(x) \wedge \mu_B(x) \quad (1.5)$$

Union : L'union de A et B, que l'on note $A \cup B$, est le sous-ensemble flou constitué des éléments de X affectés du plus grand des deux degrés d'appartenance μ_A et μ_B :

$$\forall x \in X \quad \mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)) = \mu_A(x) \vee \mu_B(x) \quad (1.6)$$

Complément : Le complément de A, que l'on note A^c , est le sous-ensemble flou de X constitué des éléments x lui appartenant d'autant plus qu'ils appartiennent peu à A :

$$\forall x \in X \quad \mu_{A^c}(x) = 1 - \mu_A(x) \quad (1.7)$$

1.3.2 Variables linguistiques

Toutes les variables que l'on considère, en physique ou en économie, par exemple, prennent une valeur unique de leur ensemble de définition X dans une situation donnée. Néanmoins, les conditions d'observations ne permettent pas toujours de connaître parfaitement cette valeur unique. Donc, une variable linguistique sert à modéliser les connaissances imprécises ou vagues sur une variable dont la valeur précise peut être inconnue.

Définition : Une variable linguistique est un triplet (V, X, T_v) , dans lequel : V est une variable définie sur un ensemble de référence X . L'ensemble $T_v = \{A_1, A_2, \dots\}$, fini ou infini, contient des sous-ensembles flous normalisés de X , utilisables pour caractériser V . Selon l'utilisation qui doit être faite, le nombre d'éléments de T_v est plus ou moins grand Bouchon [95].

Exemple : Afin de décrire la température d'une pièce par une variable linguistique, on peut utiliser l'ensemble des termes suivants : $T(V) = \{\text{froid, tiède, chaud}\}$. En considérant que l'univers du discours est l'intervalle $[0, 45]$, on peut définir les termes linguistiques comme suit : «froid» est «une température environ inférieure à 10 degrés», «tiède» est «une température d'environ 17 degrés» et «chaud» est «une température environ supérieure à 24 degrés». Ces termes peuvent être caractérisés par les fonctions d'appartenance représentées sur la figure 1.3. N'importe quelle fonction d'appartenance, par exemple la fonction «froid», définit un nombre flou.

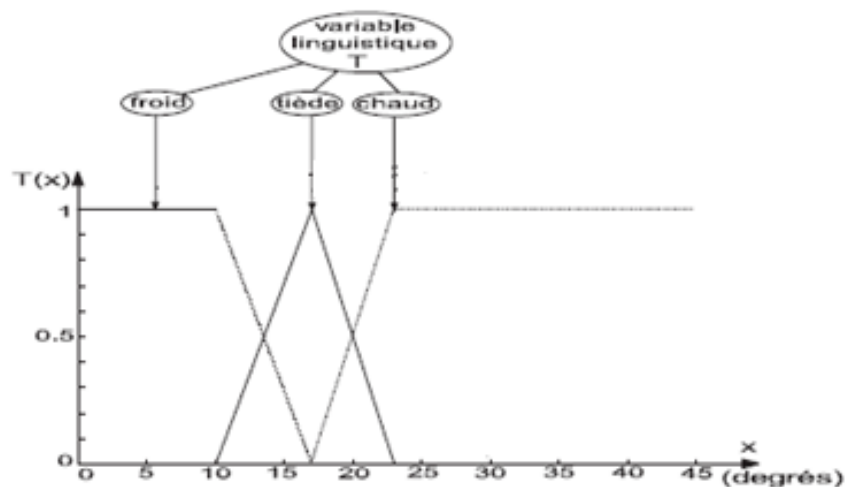


Fig 1.3 : Variables linguistiques $(V, T(V), X)$ pour décrire la température

1.3.3 Relations floues

Une relation floue est un concept qui permet de définir un sous-ensemble flou sur un univers du discours qui est un produit cartésien, tout en tenant compte des relations qui relient les univers du discours initiaux. La fonction d'appartenance définissant ce sous-ensemble flou ne se résume pas forcément à une simple combinaison de fonctions d'appartenance définissant des sous-ensembles flous dans les univers du discours initiaux.

Définition : Une relation floue R sur les univers de références X_1, X_2, \dots, X_n est définie comme un ensemble flou du produit cartésien $X_1 \times X_2 \times \dots \times X_n$ ayant la fonction d'appartenance μ_R .

$$R_X = \{((x_1, x_2, \dots, x_n), \mu_R(x_1, x_2, \dots, x_n)) \mid (x_1, x_2, \dots, x_n) \in X\}. \quad (1.8)$$

Où $\mu_R(x_1, x_2, \dots, x_n)$ est une fonction de mappage :

$$\mu_R(x_1, x_2, \dots, x_n) : X_1 \times X_2 \times \dots \times X_n \rightarrow [0, 1] \quad (1.9)$$

Exemple : La relation floue R : "approximativement égal à" peut être définie sur $\mathcal{R} \times \mathcal{R}$ par la fonction : $\mu_R(x, y) = \frac{1}{1 + (x-y)^2}$.

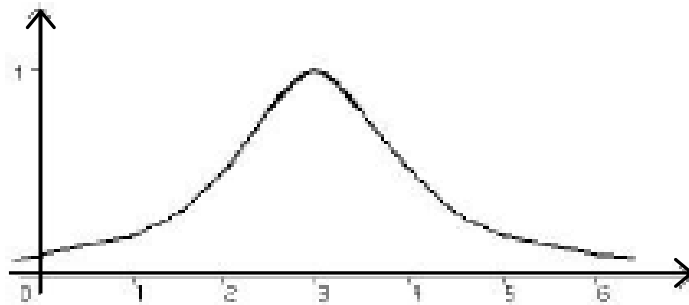


Fig. 1.4 : Représentation de la relation x approximativement égal à 3

La différence entre une relation floue et une relation classique (exacte) est que pour la première, toute valeur d'appartenance dans l'intervalle $[0,1]$ est permise alors que pour la seconde, seules les valeurs 0 et 1 sont permises.

1.3.4 Propositions floues

Une proposition floue est définie à partir d'un ensemble de variables linguistiques afin de représenter une connaissance. Par exemple, «la température de la pièce est froide».

Propositions floues élémentaires : Une forme élémentaire de proposition floue est définie à partir d'une seule variable linguistique $(V, T(V), X)$ et exprimée simplement par la phrase :

$$p : V \text{ est } A$$

Où V est une variable qui prend sa valeur dans l'univers du discours X , et A est l'un des termes linguistiques de $T(V)$. Une valeur particulière $V = v$ appartient à A avec le degré d'appartenance $\mu_A(v)$. Cela permet de définir la valeur de vérité $V(p)$ de la proposition lorsque V vaut v :

$$V(p) = \mu_A(v). \quad (1.10)$$

Exemple : Supposons que l'on veuille définir la température V d'une pièce en utilisant la variable linguistique définie dans la figure 1.3. La proposition floue correspondante est exprimée par la phrase :

$$p : V \text{ est tiède.}$$

A partir de la fonction d'appartenance définissant «tiède», nous trouvons que le degré d'appartenance d'une température de 15 degrés au sous-ensemble flou «tiède» est de 0.7, comme on le voit sur la figure 1.5. La valeur de vérité $V(p)$ de la proposition p lorsque la température vaut 15 degrés est donc de 0.7.

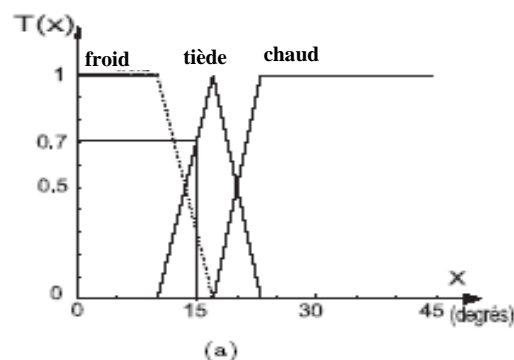


Fig 1.5 : Valeur de vérité de la proposition floue p : la température est tiède

Propositions floues générales : Une proposition floue générale est obtenue par la composition de propositions élémentaires «x est A», «y est B»,... pour des variables x, y,..... supposées non indépendantes. Habituellement les propositions floues générales sont classées en quatre types :

- **La conjonction de propositions floues élémentaires**

$$p : (x_1 \text{ est } A_1) \text{ et } \dots \text{ et } (x_n \text{ est } A_n)$$

Dans ce cas, la conjonction est associée au produit cartésien $A_1 \times A_2 \times \dots \times A_n$ caractérisant la variable conjointe (x_1, x_2, \dots, x_n) sur les univers de discours $X_1 \times X_2 \times \dots \times X_n$. Sa valeur de vérité est alors définie par :

$$V(p) = \min \{ \mu_{A_1}(x_1), \dots, \mu_{A_n}(x_n) \} \quad (1.11)$$

- **La disjonction de propositions floues élémentaires**

$$p : (x_1 \text{ est } A_1) \text{ ou } \dots \text{ ou } (x_n \text{ est } A_n)$$

La valeur de vérité de la disjonction sur les univers du discours $X_1 \times X_2 \times \dots \times X_n$ est définie par :

$$V(p) = \max \{ \mu_{A_1}(x_1), \dots, \mu_{A_n}(x_n) \} \quad (1.12)$$

- **Les implications entre propositions floues**

Une implication floue associe à toute règle floue de la forme si V est A alors W est B, construite à partir des variables linguistiques (V, X, TV) et (W, Y, TW) une relation floue R entre X et Y, de fonction d'appartenance : La fonction d'appartenance μ_R de cette relation dépend des fonctions d'appartenance μ_A et μ_B de A et B.

$$\forall x \in X, \forall y \in Y, \mu_R(x, y) = F(\mu_A(x), \mu_B(y)) \quad (1.13)$$

Où F est une fonction particulier appelée fonction d'implication floue, de $[0, 1] \times [0, 1] \rightarrow [0, 1]$. En logique floue, de nombreuses implications floues ont été développées, correspondant chacune à une interprétation différente des règles floues.

Les plus utilisées sont sans aucun doute les implications de MAMDANI (minimum) et LARSEN (produit).

- Mamdani

$$\mu_R(x, y) = \min(\mu_A(x), \mu_B(x)) \quad (1.14)$$

- Larsen

$$\mu_R(x, y) = \mu_A(x) \cdot \mu_B(x) \quad (1.15)$$

- **Les combinaisons** de conjonction, disjonction et implication de propositions floues élémentaires. Par exemple, «si (X1 est A11) et (X2 est A12) alors (Y est B1)», etc.

1.4 Inférence floue

Un des apports principaux de la logique standard a été la formalisation des méthodes de déduction, qui sont en quelque sorte un outil de raisonnement. Les méthodes de déduction utilisées en logique standard permettent de définir une nouvelle certitude à partir d'autres connaissances certaines. Dans le cadre de la logique floue, il est possible de généraliser les méthodes de raisonnement lorsqu'on dispose de connaissances incertaines ou imprécises.

Les méthodes d'inférence, utilisées habituellement en logique standard, peuvent être généralisées dans le cadre de la logique floue pour permettre de raisonner lorsque les règles ou les faits sont connus de façon imparfaite. La méthode d'inférence la plus connue est le modus ponens, qui permet de déduire une nouvelle connaissance en se basant sur la connaissance d'un seul fait et d'une seule règle.

1.4.1 Modus ponens généralisé

En logique booléenne traditionnelle, la règle d'inférence essentielle est appelée *modus ponens*, et elle est basée sur l'implication $p \rightarrow q$, où p et q sont deux propositions binaires, définie de la manière suivante : *Si (p → q) et (p est vrai) alors (q est vrai).*

En logique floue, les sous-ensembles flous en entrée d'une règle floue ne correspondent pas exactement aux sous-ensembles flous des prémisses de cette règle, initialement définies. Pour obtenir une conclusion à partir d'une règle dont la prémisse est différente de la prémisse réelle, il est nécessaire d'appliquer la règle du *modus ponens généralisé*, introduite par Zadeh [Zadeh, 1968]. A partir d'une règle floue et d'un fait observé A' pour V , on en déduit une valeur B' pour la conclusion :

$$\begin{array}{ll}
 \text{R :} & \text{si } V \text{ est } A \text{ alors } W \text{ est } B \\
 \text{Observation :} & V \text{ est } A' \\
 \hline
 \text{Conclusion :} & W \text{ est } B'
 \end{array}$$

La fonction d'appartenance de la conclusion B' est construite à partir de la fonction d'appartenance de l'observation A' et de la règle floue :

$$\forall y \in Y, \mu_{B'}(y) = \sup_{x \in X} T(\mu_{A'}(x), \mu_{A \rightarrow B}(x, y)) \quad (1.16)$$

La t-norme T utilisée, appelée opérateur de Modus Ponens Généralisé, est choisie pour conserver les propriétés d'un Modus Ponens classique (si $A' = A$ alors, on doit en déduire que $B' = B$). Elle doit dépendre de l'implication floue utilisée (et inversement).

$$\begin{array}{ll}
 \text{Lukasiewicz} & T_l(u, v) = \max(u + v - 1, 0) \\
 \text{Probabiliste} & T_p(u, v) = u \cdot v \\
 \text{Zadeh} & T_z(u, v) = \min(u, v)
 \end{array}$$

1.4.2 Méthodes d'inférence floue

Nous distinguons une variété importante de méthodes d'inférence floue, mais nous nous contentons d'en présenter les deux méthodes les plus utilisés. L'inférence de Mamdani (Min-Max) et l'inférence par la méthode additive. Les deux inférences diffèrent par la manière de déterminer les sorties.

Supposons que la base de connaissances contient n règles d'inférence contenant chacune m prémisses et une même conclusion y est B_i .

Règle 1 : Si (X_1 est A_{11}) et \dots et (X_m est A_{1m}) ; alors (Y est B_1)

Règle 2 : Si (X_1 est A_{21}) et \dots et (X_m est A_{2m}) ; alors (Y est B_2)

\dots

Règle n : Si (X_1 est A_{n1}) et \dots et (X_m est A_{nm}) ; alors (Y est B_n)

La méthode Min-Max : se décompose en deux étapes : pour chaque règle appliquée, le minimum de degré d'appartenance est retenu dans le résultat. Par contre, si plusieurs règles donnent un même résultat, le maximum de ces résultats est retenu [Cox, 1994]. Ces opérations sont expliquées par :

$$\mu_{R_i}(y) = \min (\mu_{A_{i1}}(X_1), \mu_{A_{i2}}(X_2), \dots, \mu_{A_{im}}(X_m)). \quad (1.17)$$

$$\mu_{Résultatfinal}(y) = \mu_{B_i}(y) = \min (\mu_{R_1}(x), \mu_{R_2}(x), \dots, \mu_{R_n}) \quad (1.18)$$

La méthode additive : Contrairement à la méthode Min-Max, dans la méthode additive le résultat final d'inférence est le minimum entre un «1», et l'addition de toutes les appartenances individuelles. Cette méthode est représentée par :

$$\mu_{R_i}(y) = \min (\mu_{A_{i1}}(X_1), \mu_{A_{i2}}(X_2), \dots, \mu_{A_{im}}(X_m)). \quad (1.19)$$

$$\mu_{Résultatfinal}(y) = \mu_{B_i}(y) = \min (1, \mu_{R_1}(x) + \mu_{R_2}(x) + \dots + \mu_{R_n}) \quad (1.20)$$

1.5 Applications de la logique floue

La logique floue trouve ses applications dans de nombreux domaines, comme modèle de représentation et de manipulation des connaissances. Elle est particulièrement utilisée pour la mise au point de contrôleurs et de systèmes expert flous, pour l'aide à la décision, pour l'interrogation flexible des bases de données et dans filtrage flou.

1.5.1 Les systèmes de commande et les systèmes experts flous

Il y a deux grands types de système dont le fonctionnement est basé sur la logique floue :

- **Les systèmes de commande floue** : De façon générale, un système de commande a pour objectif de piloter l'entrée d'un processus afin d'obtenir un fonctionnement correct de ce dernier. Lorsqu'un opérateur humain commande manuellement un système, les actions qu'il réalise sont dictées par une connaissance subjective du fonctionnement de ce système. Par exemple, s'il fait «froid» dans une pièce, on «augmente» le chauffage ; s'il fait «très froid», on «chauffe plus». Ce principe est à la base de la commande floue.
- **Les systèmes experts flous**: Ces systèmes peuvent être composés de plusieurs couches de règles avec des opérateurs ET et Ou. Les variables de sortie résultats de l'exécution d'un nombre de règles peuvent être appliquées aux entrées d'autres règles. Les applications de ces systèmes sont diverses : le contrôles linéaires et non linéaires, l'aide à la décision, la reconnaissance de forme, le système financières, la recherche opérationnelle, etc.

La mise en œuvre de ces deux systèmes s'effectue à travers un Système d'Inférence Flou composé généralement d'une base de connaissance avec les trois modules suivants (Fig 1.6):

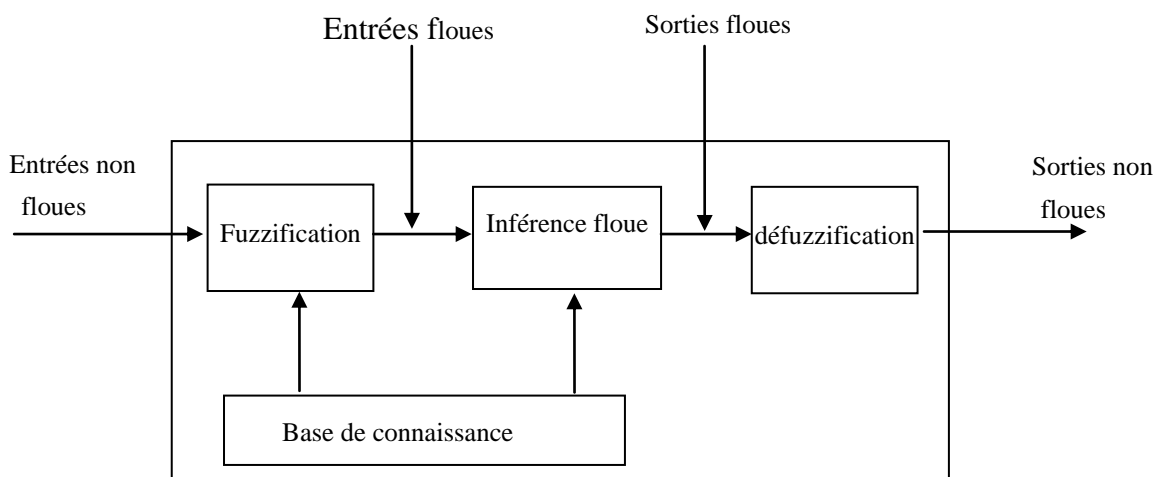


Fig 1.6 : Structure d'un système flou

Fuzzification

La fuzzification est réalisée dans l'interface d'entrée du système flou. Durant cette phase, les informations issues du système sont tout d'abord normalisées. Ensuite, les données normalisées sont transformées en qualifications linguistiques, en utilisant des règles sémantiques définies par un expert.

Les entrées dans un système flou sont en général mesurées à l'aide d'organes de mesures qui sont le plus souvent de type analogique. Etant donné que, l'implémentation du système flou se fait toujours en numérique, il faut d'abord convertir les entrées analogiques en digital, puis on procède à la fuzzification. L'adaptation des entrées permet de garantir l'appartenance de celles-ci à l'univers de discours choisi U . La dernière étape de fuzzification est la génération des degrés d'appartenance à chaque valeur linguistique définie par sa fonction d'appartenance. Par ailleurs, les fonctions d'appartenance les plus utilisées sont : triangulaire, trapézoïdale, gaussienne, sigmoïde, etc.

Inférence floue

L'inférence floue est le processus de formulation de la relation entre les entrées et les sorties par logique floue. Cette relation offre une base avec laquelle la décision est prise par le système flou. L'inférence floue fait appel alors aux concepts de fonctions d'appartenance, d'implication floue et les règles floues. Nous avons présenté dans la section 1.4 les principes d'inférence floue.

Défuzzification

Le résultat de l'inférence en utilisant une des méthodes d'implication floue, comme Max-min, est une valeur floue. Cette information ne peut être utilisée directement. Une transformation doit être prévue à la sortie du bloc d'inférence pour la convertir en grandeur fixe, cette transformation étant connue par le terme défuzzification (concrétisation). Plusieurs méthodes de défuzzification existent en logique floue, les plus utilisées sont les deux suivant [Cox, 1994].

➤ **La méthode du centre de maximum**

Dans cette méthode, la valeur de sortie est estimée par l'abscisse du point correspondant au centre de l'intervalle pour lequel la fonction d'appartenance est maximale (Fig 1.7).

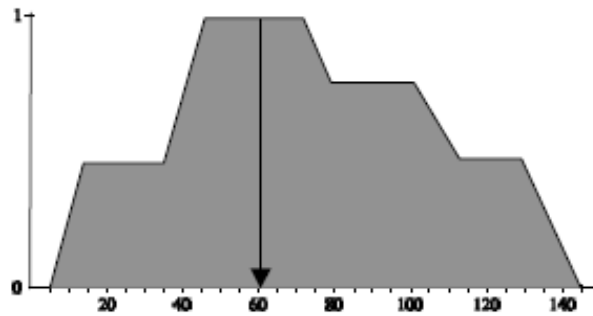


Fig 1.7 : La défuzzification par la méthode du centre de maximum

➤ **La méthode du centre de gravité (COG)**

La défuzzification par centre de gravité consiste à calculer l'abscisse du centre de gravité de la fonction d'appartenance (Fig 1.8):

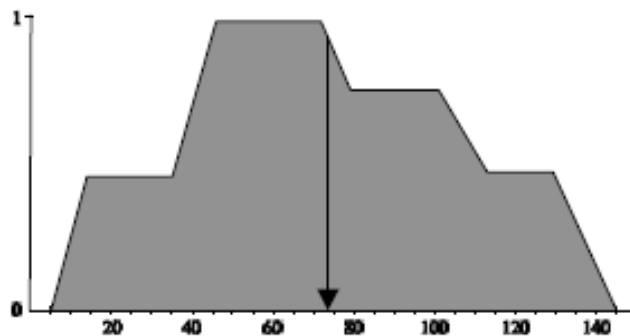


Fig 1.8 : La défuzzification par la méthode du centre de gravité

1.5.2 L'aide à la décision

Tous les concepts de l'aide à la décision et de la prise de décision ont été étudiés dans un contexte flou, pour traiter des situations dans les quelles certaines des connaissances

conduisent à la décision sont imparfaites. La prise de décision peut reposer sur des contraintes floues (la durée de réparation et approximativement de 120 heures), sur des critères dont les caractérisations sont floues (« le confort est satisfaisant »), l'objectif de la prise de décision lui-même pouvant être flou (« le résultat financier doit être largement supérieur à celui de l'année précédente »).

Plusieurs méthodes d'aide à la décision basées sur la théorie de sous-ensembles existent. La méthode la plus simple de décision en environnement flou est celle de Bellman et Zadeh [Bellman and Zadeh, 1970], qui consiste en la donnée d'un objectif G et d'une contrainte floue C comme deux sous-ensembles flous de D . Par exemple si D est l'ensemble des trains pour Constantine, l'objectif de la décision peut être de choisir un train qui arrive « relativement tard », étant donnée la contrainte indiquant que le train doit arriver « bien avant l'heure de la conférence ». Une décision floue est alors définie comme : $D = G \cap C$. Plus généralement, si l'on dispose de plusieurs objectifs $G_1 \cap G_2 \cap \dots \cap G_n$ et de plusieurs contraintes $C_1 \cap C_2 \cap \dots \cap C_m$ la décision est :

$$D = G_1 \cap G_2 \cap \dots \cap G_n \cap C_1 \cap C_2 \cap \dots \cap C_m. \quad (1.20)$$

D'autres méthodes sont aussi proposées, pour traiter des cas dans lesquelles les degrés d'importances des objectifs et des contraintes ne sont pas égaux, comme la méthode de Yager [Yager, 1977]. Yager a employé des poids exponentiels α_i pour accomplir cette différence dans les importances. Cette méthode est représentée par :

$$D(x) = G_1^{\alpha_1} \cap G_2^{\alpha_2} \cap \dots \cap G_n^{\alpha_n} \cap C_1^{\alpha_{n+1}} \cap C_2^{\alpha_{n+2}} \cap \dots \cap C_m^{\alpha_{n+m}}. \quad (1.21)$$

Dans le cadre de la décision de groupe ou de choix social, on considère un ensemble D d'options sur lesquelles une population P d'individus doit indiquer ses préférences par la mise au point d'un consensus, deux flexibilités pouvant essentiellement être introduites dans les méthodes traditionnelles : Les préférences peuvent être floues, Les majorités peuvent l'être également. Chaque individu i de P fournit ses préférences sous la forme 'une relation binaire R_i , dite de préférence, définie sur D , telle que :

$$\begin{aligned} R_i(x, y) &= 1 && \text{si } x \text{ est franchement préféré par } Y \\ R_i(x, y) &\in [0.5, 1] && \text{si } x \text{ est relativement préféré à } Y \end{aligned}$$

$R_i(x, y) = 0.5$ s'il y a indifférence entre X et Y

$R_i(x, y) \in [0, 0.5]$ si x est relativement non préféré a Y

Il existe de nombreuses travaux sur les relations de préférence et leur exploitation, on déduit une solution satisfaisante pour la population P de plusieurs façons, on peut par exemple utiliser le moyen pour une majorité m ($m < P$), c'est-à-dire l'ensemble des options y de D pour lesquelles il n'existe pas de X dans D tel que $R_i(x, y) > 0.5$ pour au moins m individus [Bouchon, 1995].

1.5.3 L'interrogation flexible des bases de données

Les Systèmes de Gestion de Bases de Données Relationnelles (SGBDR) sont devenus, sans conteste, le noyau de tout système informatique. Cependant, la diversification des applications des BD et de leurs utilisateurs a montré les limites des SGBDR, notamment sur le plan de l'interrogation flexible [Bosc et al. 1993] [Bosc et al, 1998]. Ainsi, plusieurs extensions du modèle relationnel et du langage SQL ont été proposées pour introduire une certaine forme de flexibilité dans l'interrogation des BD. En effet, l'interrogation classique d'une BDR est qualifiée d'interrogation booléenne dans la mesure où l'utilisateur formule une requête, avec SQL par exemple, qui retourne un résultat ou rien du tout. Cette interrogation pose des problèmes pour certaines applications [Larsen, 1999]. Le problème est que l'interrogation booléenne ne permet à l'utilisateur ni d'utiliser des termes linguistiques vagues et imprécis dans les critères de qualification des données recherchées ni d'exprimer des préférences entre ces critères, ce qui est souvent une demande légitime des utilisateurs. Pour illustrer ce problème, considérons un utilisateur qui consulte, via Internet, une BD d'offres de location de biens immobiliers. L'utilisateur souhaite trouver un appartement de préférence à Constantine ayant une surface « moyenne » et un loyer « modéré » avec une place de parking si possible.

Plusieurs travaux ont été proposés dans la littérature pour introduire la flexibilité dans l'interrogation des BD. La majorité de ces travaux ont utilisé le formalisme des ensembles flous pour modéliser les termes linguistiques tels que (« modéré », « moyen ») et pour évaluer des prédicats comportant de tels termes. L'idée essentielle dans ces travaux consiste à étendre SQL et à ajouter une couche supplémentaire au-dessus d'un SGBDR

pour évaluer les prédicats flous sur le résultat obtenu par l'évaluation d'une requête dite enveloppe. La requête enveloppe est générée à partir de la requête initiale de l'utilisateur.

Un prototype d'interrogation flexible d'une base de données relationnelle au travers d'une interface web a été mis en œuvre [Bartel, 2006]. Ce prototype est fondé sur le principe de la dérivation et permet de traiter des questions du type sélection-projection-jointure. L'interface permet de définir un profil utilisateur caractérisé par ses propres définitions de concepts graduels comme cher, loin, grand,... .A l'aide de ces concepts l'utilisateur construit les différents composants de sa requête. Celles-ci sont ensuite traitées pour en extraire une requête applicable à la base de données. Les résultats retournés sont ensuite affichés avec une évaluation de leur niveau d'adéquation.

1.5.4 Le filtrage flou

D'après Andres « au sens le plus général, le filtrage est l'opération qui consiste à rechercher dans une collection d'informations, une information dont le contenu correspond le mieux à une certaine description (le filtre) » [Andres, 89]. Il en résulte que si cette description est floue alors on parle plutôt de filtrage flou. Le filtrage flou est une technique de comparaison entre deux ensembles flous (ou distributions de possibilités) convexes¹⁰ et normalisés¹¹, ou bien entre la valeur d'une variable et un ensemble flou. La comparaison se réalise à partir des mesures de possibilités et de nécessités de la théorie des possibilités. Le filtrage flou est utilisé pour définir une technique de représentation de données linguistiques sous forme de classes imprécises de termes, qui peuvent être mises en relations afin de représenter les liens sémantiques et l'ambiguïté propres au langage naturel.

1.6 Conclusion

Dans ce chapitre après la présentation de différents concepts de la logique floue, et les principes d'inférence floue, nous avons décrit les principales applications de la logique floue et nous avons vu l'intérêt de cette approche dans les domaines du contrôle, d'aide à la décision, d'interrogation flexible des base de données et de filtrage floue. Cette approche permet de tenir compte à la fois des connaissances d'un expert humain, de l'incertitude et de l'imprécision des données traitées. Les variables linguistiques permettent aussi de traiter

ces deux informations initialement très différentes à l'aide d'un formalisme unique. Cela va permettre de créer des systèmes intelligents de manière facile et qui possèdent des capacités de raisonnement et de prise de décision proche à celles de l'être humain.

Chapitre 2

Systemes Multi-agents

2.1 Introduction

Les systèmes multi agents sont un domaine issu de la rencontre de l'Intelligence Artificielle et des Systèmes Distribués. Cette discipline a vu le jour pour remédier aux insuffisances et enrichir l'approche classique de l'IA pour résoudre certains problèmes qui sont distribués de manière inhérente, et que seule une résolution distribuée peut les résoudre, par exemple la gestion de trafic aérien, la gestion d'un réseau électrique, ...etc. L'idée est que plutôt de tenter de résoudre un problème par une seule entité intelligente, il serait mieux de faire coopérer autres plus simple, afin de résoudre le problème par une équipe. Chaque entité a une perception et des compétences limitées qui ne leur permettent pas individuellement de résoudre le problème, ils utilisent la coopération et la coordination afin de résoudre ensemble le problème. Cette approche a rendu les systèmes plus performants et a lui permettent de donné des résultats encourageantes.

Dans ce chapitre nous allons décrire les concepts qui découlent du domaine de recherche que constituent les systèmes multi-agents. Nous donnerons un état d'avancement des travaux sur ce domaine avant de conclure sur les possibilités et les voies de recherche qu'offrent ce domaine.

2.2 Concepts et définitions

2.2.1 Concept d'agent

Dans la littérature, on trouve une multitude de définitions d'agents. Elles se ressemblent toutes, mais diffèrent selon le type d'application pour laquelle est conçu l'agent. À titre d'exemple, voici l'une des premières définitions de l'agent due à Ferber [Ferber, 1995]: « Un agent est une entité autonome, réelle ou abstraite, qui est capable d'agir sur elle-même et sur son environnement, qui, dans un univers multi-agent, peut communiquer avec d'autres agents, et dont le comportement est la conséquence de ses observations, de ses connaissances et des interactions avec les autres agents ». Il ressort de cette définition des propriétés clés comme l'autonomie, l'action, la perception et la communication. D'autres propriétés peuvent être attribuées aux agents. Nous citons en particulier la réactivité, la rationalité, l'engagement et l'intention.

Récemment, une autre définition a été proposée par Jennings, Sycara et Wooldridge dans [Jennings et al, 1998]: « Un agent est un système informatique, situé dans un environnement, et qui agit d'une façon autonome et flexible pour atteindre les objectifs pour lesquels il a été conçu ».

Les notions “situé”, “autonomie” et “*flexible*” sont définies comme suit :

- *situé* : l'agent est capable d'agir sur son environnement à partir des entrées sensorielles qu'il reçoit de ce même environnement. Exemples : systèmes de contrôle de processus, systèmes embarqués, etc.
- *autonome* : l'agent est capable d'agir sans l'intervention d'un tiers (humain ou agent) et contrôle ses propres actions ainsi que son état interne.
- *flexible* : l'agent dans ce cas est :
 - *capable de répondre à temps* : l'agent doit être capable de percevoir son environnement et élaborer une réponse dans les temps requis,
 - *proactif* : l'agent doit exhiber un comportement proactif et opportuniste, tout en étant capable de prendre l'initiative au “bon” moment;

- *social* : l'agent doit être capable d'interagir avec les autres agents (logiciels et humains) quand la situation l'exige afin de compléter ses tâches ou aider ces agents à accomplir les leurs.

Bien entendu, dépendamment des applications certaines propriétés sont plus importantes que d'autres, il peut même s'avérer que pour certains types d'applications, des propriétés additionnelles soient requises. Il convient cependant de souligner que la présence des propriétés comme l'autonomie, la flexibilité, la sociabilité, etc., donne naissance au paradigme agent tout en le distinguant des systèmes conventionnels comme les systèmes distribués, les systèmes orientés objets et les systèmes experts.

2.2.2 Systèmes multi-agents

Initialement, le domaine de l'Intelligence Artificielle (IA) cherche surtout à décrire et à résoudre des problèmes complexes identifiés par des experts. Dans ce domaine, il est possible de construire des programmes informatiques, capables d'exécuter un nombre important de tâches en centralisant « l'intelligence » au sein d'un système unique. Il est cependant difficile d'entrer dans une même « base », les connaissances, les compétences d'individus totalement différents qui communiquent entre eux. Contrairement aux systèmes d'IA, qui simulent dans une certaine mesure les capacités du raisonnement humain, les IAD et en particulier les SMA sont conçus et implantés idéalement comme un ensemble d'agents interagissant, le plus souvent, selon des modes de coopération, de concurrence ou de coexistence [Chaib-draa, 1999].

Comme dans le cas avec le concept d'agent, on trouve une multitude de définitions des systèmes multi-agents, parmi ceux nous retiendrons celle de Ferber [Ferber, 1995], qui le définit comme un système composé des éléments suivants (Fig 2.1):

- un environnement E identifié et muni d'un système de repérage dans l'espace (souvent Euclidien) ;
- un ensemble d'objets O passifs pouvant être perçus, créés, modifiés ou détruits par des agents ;

- un ensemble d'agents A actifs ($A \subseteq O$) ;
- un ensemble de relations R qui unissent des objets entre eux ;
- un ensemble d'opérations Op offrant la possibilité aux agents de A de percevoir, produire, consommer, transformer et manipuler des objets de O ;
- un ensemble de lois universelles qui sont des opérateurs chargés de représenter l'application des actions des agents sur le monde et la réaction du monde à ces actions

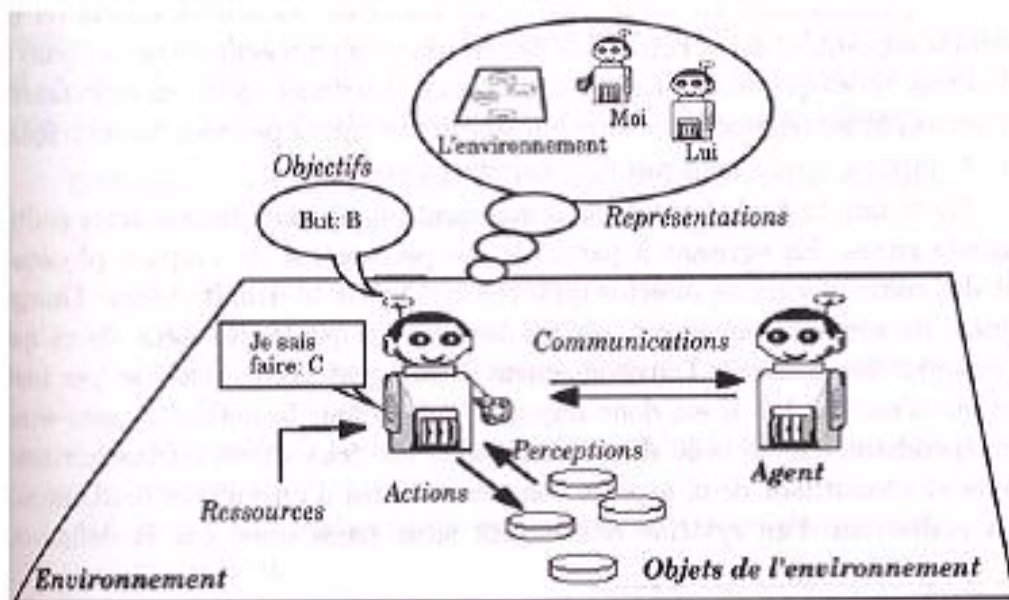


Fig 2.1 : Représentation imagée d'un agent en interaction avec son environnement et les autres agents [Ferber, 1995].

2.2.3 Interaction entre agents

Un agent peut agir sur le monde qui l'environne, c'est à dire sur les autres agents présents dans son univers et sur l'environnement lui-même (réel ou simulé). Cette intervention peut prendre la forme d'une modification de l'état des autres agents qu'il côtoie, que ce soit au niveau de leurs connaissances (ex : envoi d'information) ou au niveau de leur activité (ex : demande d'engagement). Un agent "situé" (i.e. relié à un système physique) peut même

influer sur un environnement autre qu'informatique. C'est l'ensemble de ces actions réalisables par un agent que l'on appelle interaction.

Pourquoi donner une telle importance à cette caractéristique ? Parce que comme l'affirme Ferber [Ferber, 1995] : "un agent sans interaction avec d'autres agents n'est plus qu'un corps isolé, qu'un système de traitement d'information, dépourvu de caractéristiques adaptatives". Dans une telle éventualité, le SMA (pour peu qu'on puisse appeler ainsi un tel regroupement d'agents) présenterait les mêmes lacunes que les systèmes experts sans toutefois en avoir tous les avantages.

Ainsi, le concepteur d'architecture multi-agents doit définir des modes de comportement individuel de sorte que les répercussions au niveau du système tout entier, n'aboutissent pas à l'anarchie mais à un comportement harmonieux et cohérent. Ceci se traduit par la définition de nombreux protocoles régissant la vie collective des agents (à l'image des lois humaines).

2.2.4 Communication entre agents

Les agents ont besoin de communiquer pour pouvoir interagir et échanger de l'information. Ils peuvent interagir soit en accomplissant des actions linguistiques (en communiquant entre eux), soit en accomplissant des actions non-linguistiques, qui modifient leur environnement. En communiquant, les agents peuvent échanger des informations et coordonner leurs activités. Dans les SMA, deux stratégies principales ont été utilisées pour supporter la communication entre agents : les agents peuvent échanger des messages directement, ou ils peuvent accéder à une base de données partagée (appelée tableau noir ou "blackboard") dans laquelle les informations sont postées. Les communications sont à la base des interactions et de l'organisation sociale d'un SMA [Chaib-draa, 1999].

Il existe plusieurs langages de communication, qui se basent sur des actes avec des locutions comme « demander » ou « commander ». Le plus connu parmi ces langages est le KQML, « Knowledge Query Manipulation Language », et FIPA-ACL (« Foundation for Intelligent Physical Agent-Agent Communication Language ») [FIPA, 1997]. Ces deux langages de communication entre agents, ont émergé des efforts de standardisation de la communauté des systèmes multi-agents.

FIPA ACL est syntaxiquement similaire à KQML, mis à part certains noms de primitives réservées. La particularité de FIPA – ACL consiste en un ensemble de messages avec une sémantique associée, c'est-à-dire les conditions que doivent respecter l'expéditeur et les effets attendus sur le destinataire.

2.3 Principe de programmation

Dans la première partie de ce chapitre, nous avons présenté les différentes notions concernant les agents et les systèmes multi-agents. Ces notions sont celles sur lesquelles nous fondons nos travaux. Elles peuvent apparaître assez abstraites et doivent donc être ramenées dans un cadre beaucoup plus concret qu'est celui de la conception d'applications informatiques. Dans cette section, nous décrivons ce que peut être la programmation de systèmes multi-agents au travers de la notion de programmation orientée agent et du concept de plate-forme de développement de SMA.

2.3.1 Programmation orientée agent

Les applications reposant sur le paradigme multi-agent doivent permettre d'exprimer des caractères radicaux, tels que la réification de la notion d'émergence, la propriété de reproduction auto-contrôlée de groupes d'agents et le comportement non linéaire [Cardon, 1998]. La conception d'applications informatiques relève aujourd'hui d'une démarche basée sur les principes du Génie Logiciel dans son approche orientée objets. Il existe en effet de nombreux langages à objets et des méthodes telles que UML [Lai, 1997] [Muller, 1997] utilisent le concept objet comme entité de base de la modélisation.

La démarche de modélisation par objets se fonde sur un a priori qui restreint le domaine des systèmes que l'on y modélise. En effet, le système est, dans l'approche objet, considéré comme fonctionnellement bien décomposable [Cardon, 1996]. Or, dans les systèmes basés sur le paradigme agent, on aborde la conception de systèmes complexes qui ne sont pas fonctionnellement bien décomposables. Le problème qui se pose alors est celui de l'agentification. Ce terme désigne le passage de la modélisation d'un système reposant sur le paradigme agent vers la réalisation de ce système. Quelques méthodes tentent d'apporter

des réponses à la modélisation d'un système basé sur le paradigme agent [Gutknecht and Ferber, 1997].

Pour en terminer avec la programmation orientée agent, nous citerons les travaux de Yoav Shoham [Shoham, 1993] qui propose un langage de programmation orienté agent de la même manière qu'il existe des langages de programmation orientés objets. Il s'agit de considérer que l'on programme des agents à l'aide d'éléments de haut niveau tels que des buts, des choix, des compétences, des croyances, etc. et que les types de messages que les agents s'échangent se réfèrent à des mécanismes de communication de haut niveau eux aussi, en définissant des messages d'informations, de requête, d'offre, de promesse, de refus, d'acceptation, etc. Malgré tous ces travaux, il n'existe pas encore de langage de programmation orienté agent qui serait l'équivalent de langages tels que C++ ou Java dans le monde des langages orientés objets.

2.3.2 Quelques plates-formes de développement de SMA

Afin de réaliser une opérationnalisation plus accessible des systèmes multi-agents, des travaux ont tenté de réutiliser des architectures et des langages existants pour construire des environnements de développement de ces systèmes. Les environnements de développement ou les plate-formes multi-agents sont nécessaires pour renforcer le succès de la technologie multi-agents. Les plates-formes multi-agents permettent aux développeurs de concevoir et de réaliser leurs applications sans perdre de temps à réaliser des fonctions de base pour la création et l'interaction entre agents et éliminent, dans la plupart des cas, la nécessité d'être familier avec les différents concepts théoriques des systèmes multi-agents. Parmi les plate-formes les plus connues on peut citer JADE [Bellifemine et al, 2003], Swarm [Minar, 1996], MadKit [MadKit, 2003] et Zeus [Azvine et al. 2000], etc.

JADE (Java Agent Development Framework) est une plate-forme multi-agents développée en Java par CSELT (Groupe de recherche de Gruppo Telecom, Italie) qui a comme but la construction des systèmes multi-agents et la réalisation d'applications conformes à la norme FIPA (FIPA, 1997). JADE comprend deux composantes de base : une plate-forme agents compatible FIPA et un paquet logiciel pour le développement des agents Java. 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 inter-opérables.

SWARM est une plate-forme multi-agents avec agents réactifs. L'inspiration du modèle d'agent utilisé vient de la vie artificielle. SWARM [Minar, 1996] est l'outil privilégié de la communauté américaine et des chercheurs en vie artificielle. L'environnement offre un ensemble de bibliothèques qui permettent l'implémentation des systèmes multi-agents avec un grand nombre d'agents simples qui interagissent dans le même environnement. De nombreuses applications ont été développées à partir de SWARM qui existe aujourd'hui implémenté en plusieurs langages (Java, Objective-C).

La plate-forme MADKIT est une plate-forme développée par le Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM) de l'Université Montpellier II. MADKIT est libre pour l'utilisation dans l'éducation. Elle est écrite en Java et est fondé sur le modèle organisationnel Alaadin. Il utilise un moteur d'exécution où chaque agent est construit en partant d'un micro-noyau. Chaque agent a un rôle et peut appartenir à un groupe. Il y a un environnement de développement graphique qui permet facilement la construction des applications.

Zeus est une plate-forme de développement de SMA générique, personnalisable et pourrait être augmentée par l'adjonction de nouveaux composants. Elle est présente sous forme d'un ensemble de classes implémentées dans le langage Java. En cela, elle est complètement portable et peut être utilisée sur tout système disposant d'une machine virtuelle Java. Un agent est composé de plusieurs objets et threads. Ces classes sont catégorisées dans trois groupes fonctionnels: une librairie de composants, un outil de développement et un outil de visualisation. De par sa généricité, elle peut être utilisée dans la conception d'un grand nombre d'applications mais en retour nécessitera des augmentations assez nombreuses et spécifiques à chaque type d'application.

2.4 Exemples de domaines d'application des SMA

Les systèmes multi-agents étant issus du domaine de l'intelligence artificielle distribuée, ils permettent de modéliser des applications où une modélisation classique est inadéquate, et où le système est souvent naturellement distribué. Si un problème n'est pas naturellement

distribué, on peut choisir de le distribuer pour des raisons de simplification. C'est alors à la charge du concepteur de trouver la meilleure distribution possible afin qu'elle corresponde au problème qu'il souhaite résoudre.

2.4.1 Systèmes complexes

Les systèmes complexes sont des systèmes où les techniques de modélisation classique sont difficilement utilisables. En effet, dans ces systèmes, les paramètres sont beaucoup trop nombreux ou contradictoires pour pouvoir être pris en compte. Parfois même, il n'est pas possible de connaître l'ensemble des paramètres qui interviennent pour la modélisation. L'approche multi-agent permet alors d'avoir recours à une modélisation locale. Cette modélisation permet grâce aux principes d'émergence présents dans les SMA d'obtenir un système ayant les propriétés attendues. Un bon exemple concerne la simulation d'une fourmilière [Drogoul, 1993]. En effet, lors de ses travaux de thèses, Alexis Drogoul a modélisé une fourmilière en utilisant des agents pour modéliser les fourmis. Il a alors réussi à démontrer que l'on pouvait obtenir un but global qui était la survie de la fourmilière sans jamais avoir programmé cet élément dans le système mais uniquement à partir de l'interaction des agents par émergence organisationnelle.

2.4.2 Systèmes d'informations coopératifs (SIC)

Les SIC sont généralement caractérisés par la grande variété et le grand nombre de sources d'informations. Ces sources d'informations sont hétérogènes et distribuées soit sur un réseau local (Intranet) soit sur l'Internet. De tels systèmes doivent être capables d'exécuter principalement les tâches suivantes:

- la découverte des sources: trouver la bonne source de données pour l'interroger;
- la recherche d'informations: identifier les informations non structurées et semi-structurées;
- le filtrage des informations: analyser les données et éliminer celles qui sont inutiles;
- la fusion des informations: regrouper les informations d'une manière significative.

Les agents et les systèmes multi-agents sont utilisés dans plusieurs domaines d'applications, comme par exemple: la gestion des réseaux, la recherche d'informations, le

commerce électronique et la planification des tâches. Pour une grande partie de ces applications, les agents sont utilisés dans les systèmes d'informations coopératifs ou comme assistants personnels dans certains travaux. À titre d'exemple, sur l'utilisation des SMA dans ce domaine, le système NETSA. Ce système est un système multi-agent coopératif, développé à l'université Laval [Côté, 1999], et destiné aux environnements riches en informations.

2.4.3 Systèmes d'aide à la décision

Les systèmes d'aide à la décision (SAD) sont présents dans de nombreux domaines et ont pour objectif d'aider le décideur dans sa tâche en lui fournissant tous les éléments pertinents pour la prise de décision. Cela consiste très souvent à extraire de l'information depuis de multiples sources et à la traiter. Des traitements de l'information pour une prise de décision multi-critères sont alors nécessaires. Les systèmes multi-agents apparaissent comme étant bien adaptés pour traiter de l'information qui peut revêtir diverses formes et provenir de diverses sources. En effet, il faut pouvoir faire la corrélation entre l'ensemble des éléments obtenus pour les présenter à l'utilisateur. L'approche à base d'agents permet d'effectuer cette corrélation en utilisant la négociation et la coopération entre agents. Il s'agit donc d'un domaine d'application pour les SMA.

2.4.4 Commerce électronique et les agents du Web

Le domaine du commerce électronique est un domaine en plein essor qui permet de favoriser les transactions commerciales. Ce domaine utilise l'outil informatique et plus particulièrement les ressources mises à disposition par Internet pour rapprocher les acteurs commerciaux dans certains domaines. Ce domaine se rapproche de celui de l'aide à la décision et peut même être confondu avec celui-ci dans certaines circonstances car il est caractérisé par la mise en place de moyens permettant d'extraire de l'information sur les produits, les marchés, les acteurs du marché [Ghidini and Serafini, 1998].

L'utilisation de systèmes multi-agents est une bonne solution ici aussi car elle permet comme pour le domaine de l'aide à la décision de faire de la fusion d'informations. La dénomination « agents du WEB » désigne les agents qui circulent sur le réseau Internet

pour extraire de l'information. Ces agents qui sont en général mobiles se déplacent au travers des sites WEB. Certains systèmes ont pour objectif de fournir au consommateur une valeur ajoutée en lui fournissant des informations supplémentaires afin qu'il fasse un choix approprié selon les critères qu'il s'est fixé [Guttman and Maes, 1998]. Il peut s'agir d'informations comparatives entre produits ou encore d'études qualitatives.

Prenons un exemple concret: vous souhaitez acheter une nouvelle voiture mais vous n'avez pas encore décidé du modèle exact. Jusqu'à maintenant, vous deviez visiter un certain nombre de concessionnaires, étudier des documentations techniques (parfois ardues), acheter des magazines spécialisés qui effectuent eux-mêmes des tests et des études comparatives. Tous ce travail est fastidieux et coûteux en temps. Le but des applications dans le domaine du commerce électronique va être d'extraire toute cette information qui est disponible à divers endroits du WEB et de n'en retenir que celle qui vous intéresse. Il va s'agir d'un travail d'extraction, de filtrage et d'analyse de l'information disponible. La multiplicité des sources d'informations disponibles sur le WEB et la complexité des traitements à effectuer rend particulièrement appropriée l'utilisation du paradigme agent pour le développement des applications de commerce électronique.

2.5 Conclusion

Dans ce chapitre, nous avons présenté les systèmes multi-agents et les concepts sur lesquels nous avons fondé nos travaux. De nombreux travaux ont été effectués concernant les architectures d'agents et de systèmes multi-agents. Pour permettre la conception de systèmes multi-agents, des plate-formes de développement apparaissent un peu partout dans le monde. Certaines sont complètement génériques (Jade, MadKit, Zeus) et fournissent uniquement les mécanismes de base nécessaires au développement de SMA; d'autres plate-formes sont plus spécialisées. Il s'agit par exemple de la plate-forme SWARM qui se spécialise dans les applications de simulation.

Après la domination de la technologie objet, la technologie agent semble prendre de plus en plus d'importance. On parle désormais de conception orientée agent comme il avait été question en d'autres temps de conception orientée objet. En effet, elle permet de répondre aux besoins de nombreux domaines d'application jusque là difficilement abordables par des

moyens traditionnels. Il s'agit notamment des systèmes complexes, des systèmes d'aide à la décision, du commerce électronique, etc.

Nous allons compléter cet état de l'art dans le chapitre suivant en présentant les travaux qui nous intéressent plus particulièrement: les systèmes multi-agents flous.

Chapitre 3

Des approches d'aide à la décision utilisées dans les SMA

3.1 Introduction

Dans le domaine de l'intelligence artificielle, on tente souvent de doter les agents artificiels des techniques de représentation et d'utilisation de connaissances leur permettent de résoudre des problèmes et d'atteindre leurs objectifs. Ce problème de représentation et d'utilisation des connaissances est au centre d'une discipline scientifique relativement nouvelle et en tout cas controversée, qu'on appelle l'intelligence artificielle. Cette discipline a eu un impact limité, jusqu'à une date récente et de façon exclusive, sur le traitement symbolique de la connaissance, par opposition à la modélisation numérique utilisée traditionnellement dans les sciences de l'ingénieur. Plus récemment, on a assisté à un retour du numérique dans ces problèmes d'intelligence artificielle, avec les réseaux neuro-mimétiques, les algorithmes génétiques et la logique floue. Alors que les réseaux neuro-mimétiques proposent une approche implicite de type «boîte noire» et les algorithmes génétiques sont des algorithmes itératifs d'optimisation, la logique floue est plus conforme à l'intelligence artificielle symbolique, qui met en avant la notion de raisonnement, et où les connaissances sont codées explicitement.

L'objectif principal de ce chapitre est de souligner l'originalité de nos travaux et de motiver notre choix d'utiliser la logique floue avec les SMA. La première section de ce chapitre sera consacrée à l'utilisation de la logique floue avec les SMA et les différents travaux réalisés dans le domaine de SMA flous. Dans la deuxième section, nous présentons les autres

approches d'aide à la décision les plus utilisées avec les SMA. Ensuite nous allons motiver pourquoi nous avons choisi la logique floue parmi tous ces approches.

3.2 L'approche floue

3.2.1 Pourquoi la logique floue avec les SMA

La mise en œuvre de systèmes modernes pose souvent des problèmes de représentation et de manipulation d'informations imparfaites, cette imperfection est due principalement de la nature du vrai environnement du monde qui ne peut pas toujours être représentée dans un format quantitatif précis, des instruments d'observations qui produisent des erreurs, et des problèmes de communication homme machine qui caractérisent ces systèmes [Aliev, 2008] [Fazlollahi et al, 2004]. Or, l'utilisation des SMA implique la nécessité de doter les agents intelligents des capacités de raisonnement et de prise de décision proche à celles de l'être humain, tel que sa capacité de résoudre des problèmes et de prendre des décisions sous incertitude et avec des connaissances imprécises et incomplètes. Cela amène les chercheurs à associer la logique floue aux agents intelligents pour établir de nouvelles sortes de systèmes autonomes.

La logique floue constituée le moyen le plus efficace pour prendre en compte l'imprécision et l'incertitude d'informations, et pour formaliser les processus de raisonnement et de prise de décision humain. Elle suscite actuellement un intérêt général de la part de tous ceux qui éprouvent le besoin de formaliser des méthodes empiriques, de généraliser des modes de raisonnement naturels, d'automatiser la prise de décision dans leur domaine, et de créer des agents intelligents effectuant les tâches habituellement prises en charge par les humains.

C'est pour les possibilités qu'elle offre de gérer l'incertitude et l'imprécision, et pour sa capacité de modéliser des mécanismes de raisonnement et de prise de décision humain, que nous nous sommes intéressés à la logique floue.

3.2.2 Les systèmes multi-agents flous

Les systèmes complexes modernes comme dans l'industrie, l'économie, la finance, le marketing et l'écologie sont composée avec plusieurs agents interagissent en mode distribuée et, presque toujours, ils sont caractérisés par l'imprécision, l'incertitude et l'incomplétude d'informations. Il est évident que les systèmes multi-agents basé sur la logique floue est l'approche la plus adéquate pour l'analyse, la conception et la réalisation de systèmes avec tels propriétés [Fazlollahi et al, 2004]. Dans la suite de cette section nous allons présenter un aperçu rapide sur l'utilisation de la logique floue avec les SMA dans la littérature avec des exemples sur les travaux réalisées dans ce domaine.

Parmi les premiers travaux sur les SMA flous on trouve ceux d'Aliev et al [Aliev et al, 1997] [Aliev et al, 1999], qui ont introduit le concept de système multi-agent intelligent distribué flou (Fuzzy Distributed multi-agent Intelligent System (FDIS)). Ils ont considéré en particulier le problème de coordination entre agents autonomes, à savoir, la planification floue, le scheduling flou, le dispatching, et les systèmes dynamiques en ligne. Leur modèle est utilisé pour la mise en œuvre d'un système de contrôle dans un système industriel. Une autre approche pour la conception de systèmes multi-agents flous est celle présentée dans [Fazlollahi et al, 2000]. Ce modèle est un système d'aide à la décision flou et multicritères, son idée principale est décomposé la tâche entre un nombre d'agents parallèles et compétitif; chaque agent intelligent composé avec un système à base de connaissance floues ; chaque agent propose une solution au problème totale (pas seulement pour un problème partiel) ; la solution totale du problème est déterminée comme proposition d'un des agents après une procédure de compétition (pas avec coordination et intégration des solutions partielle des agents comme dans les système distribué classique). Dans [Yager, 2001], Yager traite le problème de la prise de décision en groupe dans les SMA, une décision dans laquelle une collection de n agents doit collaborer sur le choix d'une alternative entre un ensemble, $X = \{x_1, x_2, \dots, x_n\}$, d'alternatives possibles. Basé sur la théorie des ensembles flous, il a décrit certaines méthodes pour l'aide dans ce type de décision. Dans [Herrera et al, 2004] et [Herrera et al, 2006], des modèles linguistiques flous pour la conception de système multi-agent sont proposé. Ces modèles sont proposés pour le filtrage coopératif et linguistique d'informations sur le Web.

On trouve aussi de nombreux travaux qui utilisent la logique floue pour créer de nouvelles sortes d'agents intelligents rassemble de plus en plus les êtres humains. Dans [Gomez and Martinez, 1997], Gomez et Martinez ont proposé une architecture basée sur des agents intelligents flous pour le contrôle de robot mobile. El-Nasr, Yen et Ioerger [El-Nasr et al, 2000] ont utilisé une représentation basée sur la logique floue pour modéliser les émotions et les comportements dans leur agent émotionnel. Un agent intelligent flou, qui sert à simuler le comportement humain par l'adaptation de son processus de prise de décision, est présenté dans [Araujo et al, 2003]. Ce modèle utilise un contrôleur adaptatif flou pour permettre à l'agent d'adapter ses objectifs selon ses nouvelles perceptions, et pour lui permettre d'accomplir ses buts dans des environnements où présente l'incertitude, avec la présence du vague et de l'information imprécise. He, leung et Jennings [He et al, 2001] ont conçu et implémenté des stratégies d'offres basées la logique floue, ces stratégies sont utilisées par les agents vendeur et acheteur dans les enchères continue. En effet, les agents utilisent des règles de contrôle floues pour sélectionner leurs actions d'offres, dépendamment de l'historique et l'état actuel de marché.

Les systèmes multi-agents flous peut être utilisé dans la plupart des grands domaines où la mise au point de systèmes intelligents pose des problèmes, de la complexité et de l'imperfection des informations. Il existe des applications du système multi-agent flou à des problèmes réels dans des domaines aussi variés que filtrage d'information dans web, le commerce électronique, l'aide à la décision, la robotique, le contrôle de systèmes complexes. Dans [Herrera et al, 2006], on trouve un aperçu sur les différents domaines d'utilisations des systèmes multi-agents basé sur la logique floue.

3.3 Les autres approches

Nous abordons dans cette section, le principe de fonctionnement de différentes approches de décisions les plus utilisés avec les SMA, ainsi que leurs avantages et leurs limites notamment dans le contexte de représentation et de manipulation d'informations imparfaites.

3.3.1 Les systèmes experts classiques

L'approche la plus utilisée jusqu'à maintenant dans les SMA est les systèmes experts basés sur la logique classique (booléenne). Dans le cadre de ce logique, une proposition est soit vraie, soit fausse (1 ou 0). La règle d'inférence essentielle est appelée *modus ponens*, et est basée sur l'implication $p \rightarrow q$, où p et q sont deux propositions binaires, définie de la manière suivante : *Si ($p \rightarrow q$) et (p est vrai) alors (q est vrai)*. Les connaissances et les règles sont stockées dans une *base de connaissances*, qui est interprétée par un *moteur d'inférence* utilisant une *base de faits* pour stocker les données propres au problème à traiter et les déductions faites par le système [ROYER, 1990]. Les connaissances sont souvent exprimées sous la forme de clauses de Horn [Genesereth and Nilsson, 1987]. Ces clauses sont d'habitude appelées des *règles de production* qui contiennent une partie prémisses et une partie actions. Si les prémisses sont vraies le moteur d'inférence exécute les actions.

La puissance d'un système expert est dépend de la logique utilisée. La logique utilisée le plus souvent jusqu'à maintenant est la logique des prédicats. Un des problèmes principaux avec la logique des prédicats est l'explosion combinatoire qui se produit très vite quand il existe beaucoup d'entrées avec beaucoup de valeurs possibles. Il devient notamment difficile de s'assurer de la maintenance, de la cohérence et la complétude d'une telle base de connaissances [AyeI and Rousset, 1990]. D'autres logiques plus évoluées ont été développées pour combler les insuffisances de la logique de prédicats. Ce sont des logiques comme la logique des défauts, les logiques modales non-monotones, les logiques des conditionnels etc.

Evaluation des systèmes experts

L'avantage des systèmes experts basé sur la logique classiques (booléenne) est qu'il est très facile d'exprimer des connaissances certaines et précises. Il est, par exemple, facile d'exprimer sous la forme d'une règle que si la couleur d'une voiture est jaune il est sûr qu'elle ne plaît pas. Ils sont spécialement bien adaptés aux problèmes où la réponse est certaine et précise. Dans le domaine de l'aide à la décision il semble très utile de pouvoir exprimer de telles connaissances notamment pour les valeurs de critères directement

éliminatoires pour la décision finale. Par contre, quand les connaissances sont approximatives, imprécises, vagues et incertaines, l'utilisation de systèmes experts est très difficile et peut être impossible. Il est impossible de représenté et de traiter des informations imprécise (la longueur est approximativement de 100 mètre) par des règle et des fait logique.

3.3.2 Les approches probabilistes

Deux approches coexistent pour traiter l'incertitude et l'imprécision: celle de la logique floue et celle de la théorie des probabilités. Chacune de ces deux approches permet de modéliser la notion d'imprécision au sens d'un choix parmi un ensemble disjonctif d'éléments. Les approche probabiliste abordent l'imprécision de manière quantitative et donc plus riche mais, en revanche, moins flexible (nuancée) que la logique floue.

Dans les approches probabilistes, il s'agit le plus souvent d'estimer la fréquence (les chances) d'occurrence d'une valeur relativement à celles de toutes les valeurs de comparaison jugées possibles. Il existe plusieurs approche de décision basé sur la théorie des probabilités, les plus souvent utilisées jusqu'à maintenant sont les réseaux bayesiens. Ces réseaux reposent sur un formalisme basé sur les théories des probabilités et des graphes. Un réseau bayésien est un modèle probabiliste graphique permettant d'acquérir, de capitaliser et d'exploiter des connaissances. Il s'agit d'un modèle graphique dans lequel les connaissances sont représentées sous forme de variable. Chaque variable est un nœud du graphe et prend ses valeurs dans un ensemble discret ou continue. Le graphe est toujours dirige et acyclique. Les arcs dirigés représentent un lien de dépendance directe (la plupart du temps il s'agit de causalité). Ainsi un arc allant de A à B exprimera le fait que B dépend directement de A. L'absence d'arc ne renseigne alors que sur la non-existence d'une dépendance directe. Les paramètres expriment le poids donne à ces relations et sont les probabilités conditionnelles des variables sachant leurs parents (exemple: $p(B|A)$) ou les probabilités a priori si la variable n'a pas de parents.

Evaluation des approches probabilistes

L'utilisation de la théorie des probabilités dans plusieurs domaines a donné des résultats importants [Bonnal, 2004]. Cependant, parce que c'est une théorie particulièrement

axiomatique, plusieurs limitations existent par rapport à des situations de processus décisionnel, particulièrement pour la qualité des environnements et des systèmes réels. Ces limitations résultent notamment du fait du nombre de données qui ne permet pas une utilisation raisonnable de la loi de distribution.

Les techniques fondées sur la théorie des probabilités ne permettent de quantifier que l'incertitude (la probabilité de garantir un volume annuel de commandes de N unités, pour le produit A , est égale à P). En réalité les informations sont également imprécises (le volume annuel de commandes du produit A est compris N_1 et N_2). De nombreuses évaluations sont vagues (le volume annuel de commandes du produit A est environ égal à N) et ne peuvent être immédiatement quantifiées. Le formalisme actuellement le mieux adapté, pour intégrer l'ensemble de ces estimations, est celui de la théorie des sous-ensembles flous.

3.3.3 Modèles issus du "soft computing"

Afin d'améliorer les médiocres résultats des méthodes statistiques traditionnelles obtenus dans des contextes incertains et perturbés, les nouveaux outils du "soft computing" ont été développés. Ceux-ci peuvent être caractérisés par la définition suivante: " le soft computing est un ensemble de techniques dont l'objectif est d'exploiter la tolérance de l'imprécision et de l'incertitude afin de construire des systèmes interprétables, robustes et peu coûteux" [Zadeh, 1994]. Constitué essentiellement de la logique floue, des techniques neuronales, des algorithmes évolutionnistes et du raisonnement probabiliste. Le soft computing joue un rôle de plus en plus important dans de nombreuses applications. Le principe de ces outils est fondé sur le fonctionnement du cerveau humain (techniques neuronales ou logique floue) ou sur des lois établies dans la nature (algorithmes évolutionnistes).

Le soft computing peut être utilisé dans la plupart des grands domaines où la mise au point de systèmes intelligents pose des problèmes, de l'apprentissage à la commande de processus en passant par les bases de données ou le traitement d'images. Il existe des applications du soft computing à des problèmes réels dans des domaines aussi variés que la recherche d'information, la fouille de données (data mining), l'aide à la décision, la

robotique, le contrôle de systèmes complexes. Dans cette section, nous présentons les différentes utilisations des techniques issues du soft computing pour la conception de modèles de prévision. Nous établissons ensuite les avantages et inconvénients de ces méthodes.

3.3.4 Les réseaux de neurones

Les réseaux de neurones artificiels sont des méthodes qui cherchent à s'inspirer du traitement de l'information effectué par le cerveau humain, cette méthode a vu le jour surtout après l'insuffisance des approches existantes comme l'IA symbolique et les méthodes algorithmiques classiques pour résoudre les problèmes existants, Citons par exemple : le domaine de la reconnaissance de formes (images ou signaux), le diagnostic, la traduction automatique, la compréhension du langage naturel [Touzet, 1992].

On peut définir les réseaux de neurones comme étant des réseaux fortement connectés de processeurs élémentaires fonctionnant en parallèle. Chaque processeur élémentaire calcule une sortie unique sur la base des informations qu'il reçoit. Les processeurs élémentaires sont les neurones formels. Le fonctionnement des réseaux de neurones artificiels peut être vu comme une interprétation mathématique des neurones biologiques, parce que les neurones de cerveau peuvent être considérés comme un automate comprenant (Fig 3.1) : la cellule centrale (ou soma) contenant le noyau, les dendrites recevant les signaux d'autres neurones par l'intermédiaire des synapses, et l'axone qui achemine l'information, en sortie de la cellule vers les autres neurones.

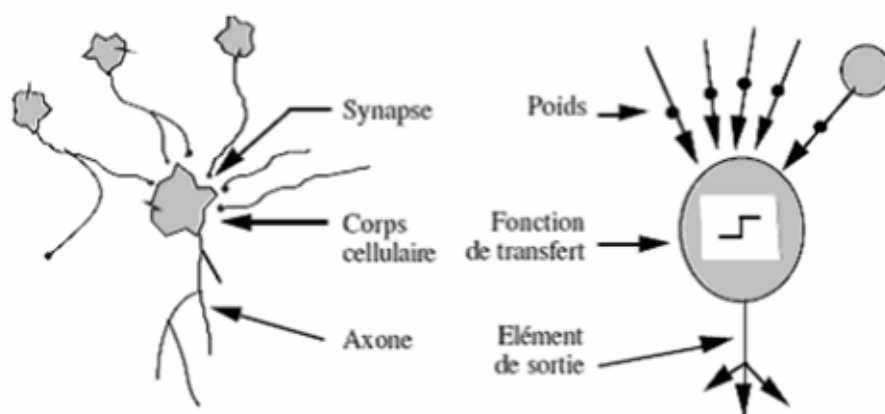


Fig 3.1 : Correspondance entre neurone biologique et neurone formel

Les problèmes typiques pour les réseaux de neurones sont le traitement du signal et le traitement d'images sous toutes ses formes. Ils sont utilisés aussi pour générer les fonctions d'appartenance dans les systèmes flous [Fuller, 2000]. A titre d'exemple sur l'utilisation de réseaux de neurones de avec la technologie agent le projet de recherche AgNet [Wermter, 2000]. Ce projet développe des agents neuraux (Fig 3.2) pour la classification sémantique de texte sur internet. Les messages textuels arrivent à un certain point de la livraison, par exemple une adresse de courrier électronique ou adresse du World Wide Web. Ces messages seront scannés et ensuite distribué à un ou plusieurs agents experts.

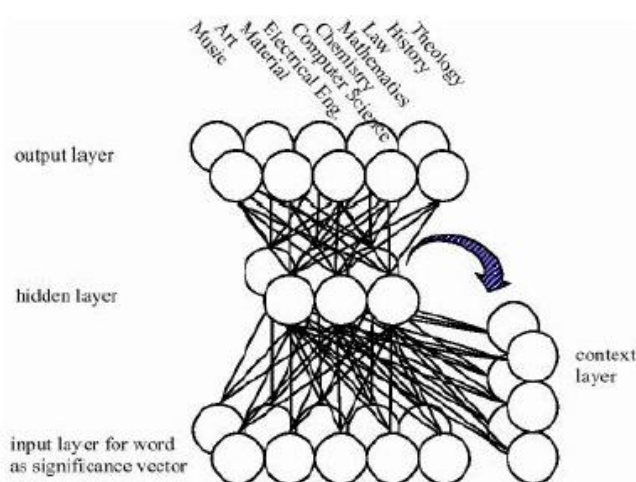


Fig 3.2 : Agent neural du projet AgNet [Wermter, 2000]

Evaluation des réseaux de neurones

Cette technique peut être utilisée dans les cas où nous avons des informations en entrée et nous voulons obtenir certains résultats en sortie, mais il n'existe pas de description du processus, pour passer de l'entrée à la sortie. Cependant l'approche neuronique a trois limitations qui l'empêchent d'être acceptée comme une pratique usuelle en prise de décision dans les SMA:

- L'approche neuronique est considérée comme étant une approche *boîte noire*. Par conséquent, il est difficile d'expliquer et d'interpréter son processus.

- Les réseaux de neurones ont été utilisés spécifiquement pour les problèmes de classification et de catégorisation et non pas de pour le raisonnement et la prise de décision.
- Il n'existe aucune démarche standard pour le choix des différents paramètres de la topologie d'un réseau de neurones (nombre de couches, nombre d'unités de traitement par couche, valeurs initiales des poids de connexions, etc.).

3.3.5 Les algorithmes évolutionnaires

Les algorithmes évolutionnaires sont des métaheuristiques basées sur des métaphores biologiques inspirées des mécanismes d'évolution darwinienne. En vue d'imiter les processus d'évolution observés dans la nature, la première adaptation des premiers des AE, à travers les algorithmes génétiques, aux problèmes d'optimisation combinatoire, a été réalisée par Holland, dans les années [Holland, 1970]. Les algorithmes génétiques constituent l'exemple le plus connu (ou populaire) des algorithmes évolutionnaires. Ces algorithmes, initiés dans les années 1970 par John Holland, sont des algorithmes d'optimisation s'appuyant sur des techniques dérivées de la génétique et des mécanismes d'évolution de la nature : croisement, mutation, sélection.

Les AE sont des algorithmes itératifs de recherche globale, fondés sur une analogie avec le monde biologique. En effet, pour un problème donné, une solution est un individu et un ensemble de solutions correspond à une population d'individus. Chaque individu peut être appelée chromosome, et chaque chromosome est constitué d'un ensemble de caractéristiques, appelés les gènes. L'algorithme élabore par itérations une population de chromosomes par l'intermédiaire d'opérateurs tels que la reproduction, le croisement (crossover) et la mutation. A chaque boucle, la performance de l'ensemble des chromosomes est quantifiée par une fonction d'évaluation (fonction fitness). La probabilité de sélectionner un chromosome pour générer la population suivante est en fonction de la qualité de celui-ci en terme de fitness.

Evaluation des algorithmes évolutionnaires

L'avantage des algorithmes génétiques est qu'ils peuvent donner des solutions aux problèmes complexes qui ne pas des solutions en temps raisonnables. Elles sont utilisées en science et en ingénierie comme des algorithmes adaptatifs pour la résolution des problèmes pratiques et comme un modèle des systèmes de l'évolution naturelle [Fogel, 1995].

Cependant comme dans le cas des réseaux de neurones, la configuration d'un algorithme génétique (ou un programme génétique) nécessite le choix de plusieurs paramètres tels que la fonction de fitness, et la taille de chaque population. Ces paramètres sont souvent déterminés par expérimentations. Ainsi, nous pouvons adresser aux algorithmes génétiques la même critique que celle mentionnée dans le cas des réseaux de neurones au sujet de l'interprétation et de la compréhension du processus d'un algorithme génétique. Les algorithmes génétiques ne fournissent pas des mécanismes de raisonnement pour l'agent, qui est peut-être une caractéristique essentielle pour l'agent. Les algorithmes génétiques fournissent des solutions approximatives pour les problèmes, mais il est impossible de traiter les données imprécises et incertaines par de tels algorithmes.

3.3.6 Les Approches hybrides

Le principal intérêt du soft computing réside dans l'utilisation conjointe de plusieurs de ces composantes dans des systèmes hybrides tirant parti d'une synergie entre les différentes méthodes afin d'exploiter les avantages de chacune tout en compensant ses inconvénients par l'utilisation d'une autre dont les propriétés sont complémentaires. Par exemple, les compétences en apprentissage des réseaux neuronaux correspondent à une insuffisance en termes d'expressibilité des résultats, que l'on peut compenser par l'utilisation de la logique floue qui manipule aisément des descriptions linguistiques. Les systèmes neuro-flous trouvent ainsi leur justification. La logique floue aide à gérer des systèmes complexes de façon simple et facilement explicitable, la difficulté de sa mise en œuvre réside dans la mise au point des paramètres et des fonctions d'appartenance et ce problème peut être résolu par l'utilisation d'algorithmes évolutionnaires.

Il existe bien d'autres exemples de synergie entre les différentes méthodes et les moyens de les combiner sont multiples. Il suffit de considérer les nombreux moyens d'utiliser conjointement logique floue et réseaux neuronaux pour s'en convaincre. L'association de la logique floue et des algorithmes génétiques est également fructueuse et il existe aussi des utilisations conjointes de la logique floue et du chaos. Le soft computing ne se limite pas aux combinaisons mettant en jeu la logique floue et les combinaisons de réseaux de neurones et d'algorithmes génétiques ou, plus couramment de réseaux de neurones et de méthodes probabilistes, trouvent de nombreuses utilisations. Cependant, le soft computing n'est pas la solution générale à toutes les applications, mais peut apporter un complément intéressant aux méthodes traditionnelles.

3.4 Discussion

Le problème de prise de décision est un problème complexe, et comme il ya une grande variante de problèmes de décision il y a une grande variante de techniques, méthodes et outils pour résoudre le problème de prise de décision. En générale, le choix d'un modèle fait intervenir différents critères, comme la facilité, le cout, le délai de mise en œuvre d'une solution, mais en particulier les caractéristiques de problème et de l'environnement. Quelques modèles sont plus convenables qu'autres pour un problème de décision particulier. Cependant, à l'exception de la logique floue, tous ces techniques ne proposent pas une solution au problème de représentation et de manipulation d'information imparfaite. Nous insistons ici sur le fait que la logique floue est le seul cadre dans lequel puissent être traitées des incertitudes, imprécisions, et qui autorise également le traitement de certaines incomplétudes. C'est aussi le seul cadre dans lequel puissent être traitées des connaissances numériques et des connaissances exprimées symboliquement par des qualifications du langage naturel. [Bouchon, 95].

La logique floue et la théorie des ensembles floue permette aux agents d'utilisé des variables comme grand, très grand et avec des degrés d'appartenance graduelles qui est le caractéristiques de raisonnement humain, aucune modèles entre les modèles précédent ou les modèles mathématique peut permette ce traitement. D'autre par un des caractéristiques principale de décision humain est l'utilisation des objectifs flous, des contraintes floues, et

des critères dont les caractérisations sont floues, la théorie de décision floue est le seul moyen possible pour la prise de décision dans ce cas.

Le souci d'automatiser ou d'assister de façon automatique les actions humaines, naturellement empiriques et empreintes d'imprécisions, dans le cadre d'une aide à la décision ou du diagnostic par exemple, renforce l'intérêt des scientifiques pour l'approche floue et justifie son intense développement au cours de ces dernières années. C'est pour les possibilités qu'elle offre de gérer l'incertitude et l'imprécision, et pour sa capacité de modéliser des mécanismes de raisonnement et de prise de décision humain, que nous sommes intéressés à la logique floue.

3.5 Conclusion

Nous avons présenté dans ce chapitre, le principe de fonctionnement de différentes approches de décisions les plus utilisés avec les SMA, ainsi que leurs avantages et leurs limites notamment dans le contexte de représentation et de manipulation d'informations imparfaites. Nous avons vu que parmi ces approches la logique floue est le moyen le plus efficace pour traiter des informations imparfaites, et pour créer des systèmes autonomes capables d'effectuer les tâches réalisées par les êtres humains.

Les caractéristiques de la logique floue, les limites des autres approches, et les besoins de créer des entités autonomes rassemblées de plus en plus l'être humain dans sa manière de raisonnement et de prise de décision et capable de le remplacer dans la réalisation des tâches, tout cela a contribué à la naissance d'une nouvelle technologie, dans laquelle la logique floue est associée aux agents intelligents pour établir de nouvelles sortes de systèmes intelligents. Cette nouvelle technologie est constituée d'une moyenne efficace pour la mise en œuvre de systèmes complexes modernes. Cela implique la nécessité de proposer un modèle générique pour la conception de systèmes multi-agents flous. Dans le chapitre suivant nous allons présenter une nouvelle approche pour la conception de SMA flous, le modèle FMAS.

Chapitre 4

FMAS: un modèle pour la réalisation de SMA Flou

4.1 Introduction

La conception de système multi-agents (SMA), basé sur l'utilisation des techniques floues, est une nouvelle approche pour la prise en compte de l'incertitude, l'imprécision et l'incomplétude d'informations dans un SMA. Concevoir un SMA flou signifie, en particulier, que le système doit contenir des agents ayant la capacité : de traiter des problèmes avec des connaissances imparfaite par inférence floue, de prendre des décisions (ou aider les décideurs dans leurs décisions) sous incertitude et avec des critères et des objectifs flous, et de répondre à des requêtes exprimées par des qualifications linguistiques vagues et imprécises.

Notre objectif est d'enrichir les SMA suffisamment, indépendamment de leurs Architectures, pour qu'ils puissent traiter des problèmes flous. Nous présentons dans ce chapitre une nouvelle approche pour la prise en compte de l'aspect flou dans un SMA: Le modèle FMAS, un modèle de système multi-agent basé sur la logique floue. Nous allons détailler aussi l'architecture de différents agents *flous* que nous avons proposé pour l'utiliser dans les SMA, lorsque les conditions demandent, à fin que leur comportement soit acquis des caractéristiques *floues*.

4.2 Vue globale du modèle

L'idée principale de notre contribution est d'enrichir les SMA suffisamment par des agents flous de différents types, indépendants des entités de système, leur rôle est d'aider les agents de système à le traitement des problèmes et à la prise de décision. Elle s'agit d'introduire des agents flous au sein des SMA, afin que leur comportement soit régi par des techniques floues. En effet, Le modèle FMAS est reposé sur des agents systèmes (modélisent les entités de système réel et dépend de l'application), et des agents flous intégrés dans le système pour réaliser des tâches flous (Fig 4.1). Notre modèle est fondé donc sur les systèmes multi-agents classiques [Ferber, 95] auxquels ils sont intégrés des agents *flous*. Chaque agent système peut solliciter un nombre d'agents flous de trois types suivants:

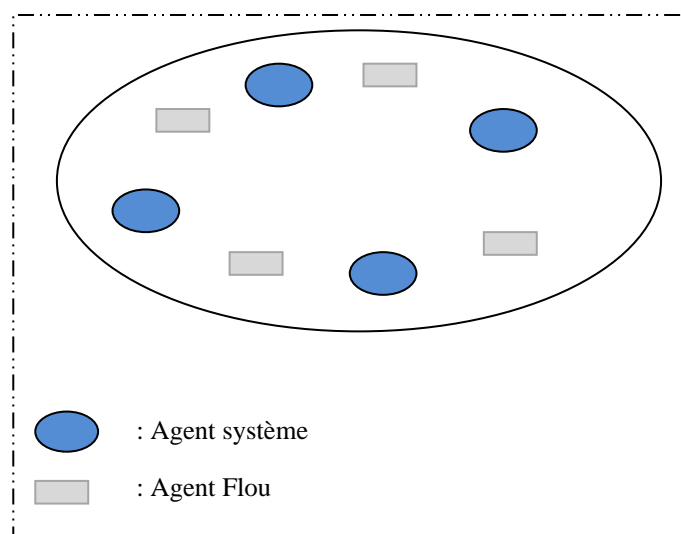


Fig 4.1 : Vue globale du modèle FMAS

- **Agents experts** : Ces agents, qui sont des systèmes à base de connaissances floues, possèdent la capacité de résoudre des problèmes avec des connaissances imparfaites par inférence floue.
- **Agents de choix** : Ces agents sont des systèmes d'aide à la décision flous et multicritères. Un agent de choix reçoit en entrée un ensemble d'alternatives de décision,

son rôle est de les évaluer et de les ordonner selon un nombre de critères flous pour aider le décideur dans sa décision.

- **Agents de requête** : Ces agents permettent le traitement des requêtes avec des contraintes exprimées par des variables linguistiques vagues et imprécises (requêtes floues). Ils utilisent la technique de dérivation pour traiter ce type de requêtes.

4.3 Fonctionnement générale

Dans notre modèle nous avons introduit deux modes fonctionnement (Fig 4.2). Le mode normale (courant), dépend de l'application, dans lequel les entités de systèmes (*agent système*) interagissent et coopèrent pour la résolution de problèmes, et le mode flou qui fait intervenir des agents flous pour réaliser des tâches floues.

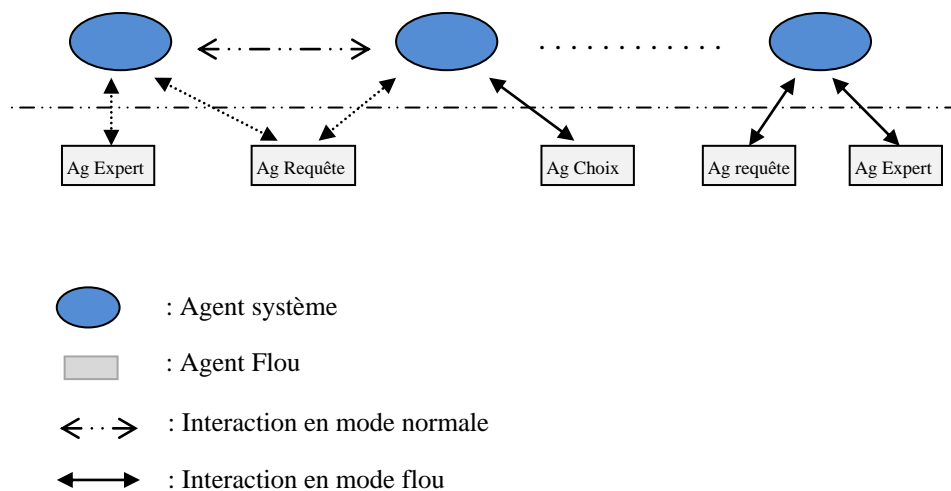


Fig 4.2 : Fonctionnement générale du modèle

Lorsqu'un agent travaille, il est amène pour résoudre un problème à faire appel à d'autres agents. Lorsque l'on considère des agents flous, un agent qui fait appeler un agent flou, est pour : lui traiter ou calculer la solution d'un problème par inférence floue (*agent expert*),

l'aider dans son choix (*agent de choix*), ou pour lui traiter une requête floue d'un utilisateur ou d'un autre agent (*agent de requête*).

Lors d'un processus de résolution d'un problème ou de calcul d'une solution, l'*agent système* sollicite l'*agent expert* chargé de traiter ce problème. Un agent expert lorsque reçoit une requête d'un agent systèmes, acquise toute la donnée nécessaire (à partir d'une source de données ou reçu de l'agent système), calcule la solution par inférence floue, et envoie le résultat à l'agent appelant.

Pour les problèmes de choix sous incertitude, on demande l'aide des agents de choix. Un agent de choix reçoit un ensemble d'alternatives et un ensemble de critères. Fait l'évaluation des alternatives selon les critères reçus, et retourne un choix (la meilleure alternative) ou un ensemble ordonné d'alternatives.

Pour traiter les requêtes floues reçues des utilisateurs ou des autres agents, l'agent système transmettent ces requêtes à l'agent de requête pour les traiter. L'agent de requête traduit ces requête en requête simple, extrait les données, et retourne les données trouvés vers l'agent appelant.

4.4 Agents flous du modèle

Dans cette section nous allons présenter en détail l'architecture de différents agents flous que nous avons proposé pour l'utiliser dans les SMA, lorsque les conditions demandent, à fin que leur soit acquérir des caractéristiques floues. Pour la modélisation de ces agents nous avons choisi de nous baser sur des architectures modulaires, et à partir de ces architectures nous attachons à l'augmenter de modules nécessaires à la réalisation d'une architecture d'agents flous.

4.4.1 Agents experts

Ces agents, qui sont des systèmes experts flous, possèdent la capacité de traiter des problèmes par inférence floue. Un agent qui fait appeler un agent expert est pour lui calculer la solution d'un problème, à partir d'un ensemble de données floues ou crispes (numériques), par inférence floue. Chaque agent expert est composé d'un module de communication, gère la communication avec les autres agents, et d'un module de traitement qui calcule les solutions

des problèmes par inférence floue. Dans ce qui suit, nous décrivons brièvement chacun de ces modules.

Le module de communication

Ce module gère la réception et l'interprétation des messages provenant des autres agents. Il reçoit des autres agents les requêtes et les données sous la forme d'un message, et suite à cela, il déclenche le module de traitement. Il transmet aussi les résultats calculés par le module de traitement vers les agents appelants.

Module de traitement

Ce module est un système à base de connaissances floues avec un nombre de règles floues. Il reçoit en entrée un ensemble d'informations, crispes ou floues, $X = \{x_1, x_2, \dots, x_n\}$. Il génère en sortie, par inférence floue, la solution y_1, y_2, \dots, y_m . L'architecture interne de ce module, quant à elle, est composée de quatre éléments suivant (Fig 3) :

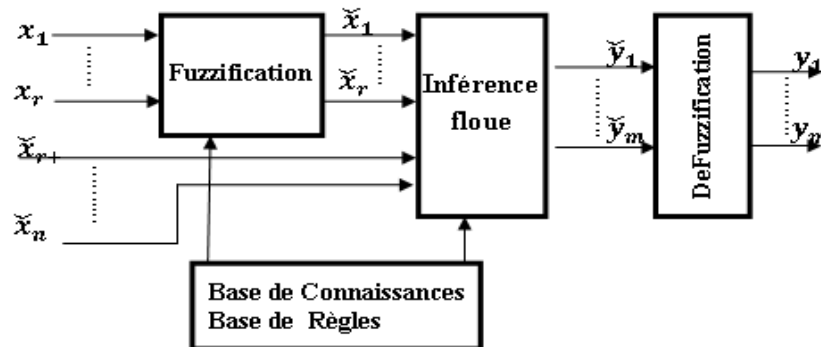


Fig 4.3 : Structure du module de traitement des agents experts

- **Base de connaissances et base de règle :** elle est composée d'une base de connaissances, fournissent les informations nécessaires pour les fonctions de normalisation (d'appartenances), et de la base de règle, qui est constituée de n règles de production à prédicat imprécis de la forme :

SI x_1 est A_1 **ET** ... **ET** x_n est A_n **ALORS** y_1 est B_1 ... **Et** y_k est B_m

- **Fuzzification** : ce module reçoit un ensemble de données numériques, son rôle est de les transformer en données floues.
- **Le module d'inférence** : Le module d'inférence est le noyau du module de traitement. Il utilise les techniques des systèmes à base de règles pour la déduction de la solution de problème.
- **Défuzzification**: Après l'inférence et la composition d'un résultat flou, si nous sommes obligés de déduire la valeur numérique des résultats obtenus à des fins d'interprétation ou de comparaison. Cette valeur peut être obtenue à l'aide de module de défuzzification. En effet, ce module reçoit les valeurs floues des variables de sortie, puis la conversion de l'entrée des données réelles (nettes) en ensembles flous.

4.4.2 Agents de choix

Ces agents sont des systèmes d'aide à la décision flou et multicritères. Leur rôle est d'aider les décideurs dans leur choix, lorsque il y a plusieurs alternatives de décision. On utilise la méthode de choix multi-critère proposé dans [Bellman and Zadeh, 1970], pour la sélection de la meilleure alternative ou pour ordonner les alternatives. Un agent de choix est composé d'un module de communication et d'un module d'évaluation qui évalue les alternatives reçues des autres agents.

Le module de communication

Il reçoit des autres agents les requêtes, les alternatives de décision et les critères selon lesquels ces alternatives seront évalués, et suite à cela, il appelle le module d'évaluation pour faire la tâche d'évaluation.

Le module d'évaluation

Ce module reçoit en entrée un ensemble d'alternatives $D = \{a_1, a_2, \dots, a_n\}$. Chaque alternative est caractérisée par M critères, $C = \{c_1, c_2, \dots, c_m\}$, selon lesquelles la désirabilité des alternatives sera déterminée. Le problème traité par ce module est la détermination de l'alternative optimale, qui est mieux que les autres en termes de critères d'évaluation. Nous

utilisons la méthode proposée par Bellman et Zadeh [Bellman and Zadeh, 1970] présentée dans le chapitre 1, pour trouver l'alternative optimale.

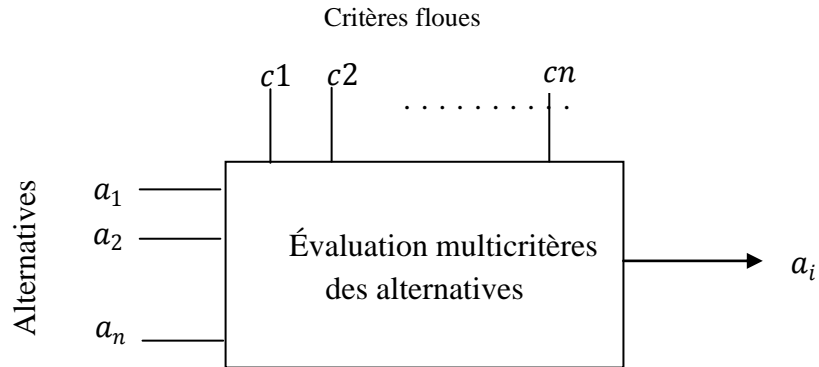


Fig 4.4 : Module d'évaluation des agent de choix

Nous définissons l'estimation de l'alternative a_i selon le critère c_j par $\mu_{c_j}(a_i)$. Après la détermination du degré de satisfaction de chaque alternative aux critères C_j nous obtenons les ensembles flous suivant.

$$\begin{aligned}
 C(a_1) &= \{ \mu_{c_1}(a_1), \mu_{c_2}(a_1), \dots, \dots, \mu_{c_m}(a_1) \} \\
 C(a_2) &= \{ \mu_{c_1}(a_2), \mu_{c_2}(a_2), \dots, \dots, \mu_{c_m}(a_2) \} \\
 &\vdots \\
 &\vdots \\
 C(a_n) &= \{ \mu_{c_1}(a_n), \mu_{c_2}(a_n), \dots, \dots, \mu_{c_m}(a_n) \}
 \end{aligned}$$

Il est reconnu que l'alternative optimale a_d , selon la méthode de Bellman et Zadeh, est:

$$a_d = \max_{i=1 \dots n} (\min_{j=1 \dots m} \mu_{c_j}(a_i)) \quad (4.1)$$

4.4.3 Agents de requête

Un agent de requête est un agent qui possède la capacité de traiter des requêtes floues. Une requête floue est une requête qui utilise des termes linguistiques vagues et imprécis pour exprimer les critères de qualification des données recherchées, ce qui est souvent le cas dans beaucoup d'applications. On utilise le formalisme des ensembles flous et de la logique floue pour modéliser l'imperfection et l'incertitude du monde réel et l'évaluation des requêtes dites vagues, floues ou graduelles. Chaque terme linguistique flou est modélisé par un ensemble flou dont la fonction d'appartenance est de type trapézoïdal.

Dans FMAS l'agent de requête a pour but d'extraire des données dans une source de données de manière flexible. Il reçoit d'abord une requête floue d'un agent appelant, traduit cette requête en une requête simple, interroge la base de données et construit une réponse qui sera acheminée vers l'agent appelant.

L'architecture interne de l'agent de requête est, quant à elle, composée d'un module de traitement et de deux interfaces comme indiquées en Figure 4.5. Ces trois éléments donnent aux agents une plus grande extensibilité étant donné leur découpage en tâches. Ils sont comme suit :

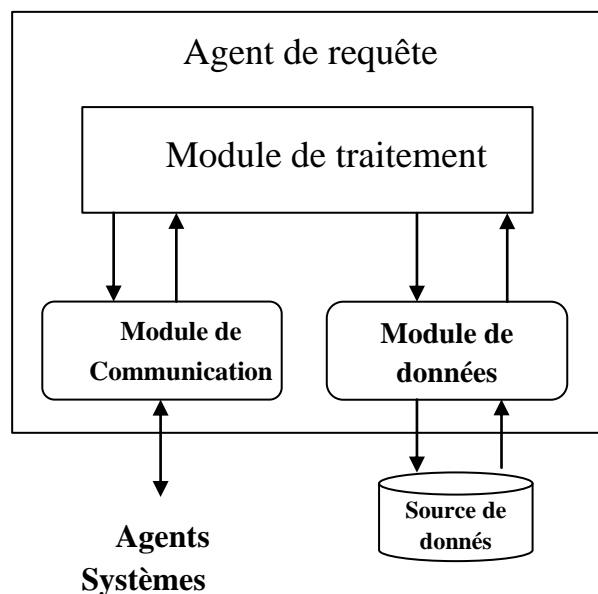


Fig 4.5 : Agent de requête

Le module de communication

Il reçoit des autres agents les requêtes sous la forme d'un message et suite à cela, il appelle le module de traitement. Il reçoit également des demandes de transmission de messages du module de traitement. Ces demandes de transmissions constituent des réponses aux requêtes reçues.

Le module de traitement

Ce module traite les requêtes reçues par le module de communication. Pour cela, il les transforme en requête simple et les achemine vers le module de données. Il construit ensuite une réponse sous forme de message qu'il donne au module de communication pour être expédiée.

Pour traiter les requêtes floues, on utilise la technique de dérivation présentée dans [Bosc et al, 1992]. Le principe de base de cette technique est d'utiliser les fonctions d'appartenances des variables linguistiques pour traduire la requete floue en requete simple, et cette dernière sera exécutée pour extraire les données (Fig 4.6).

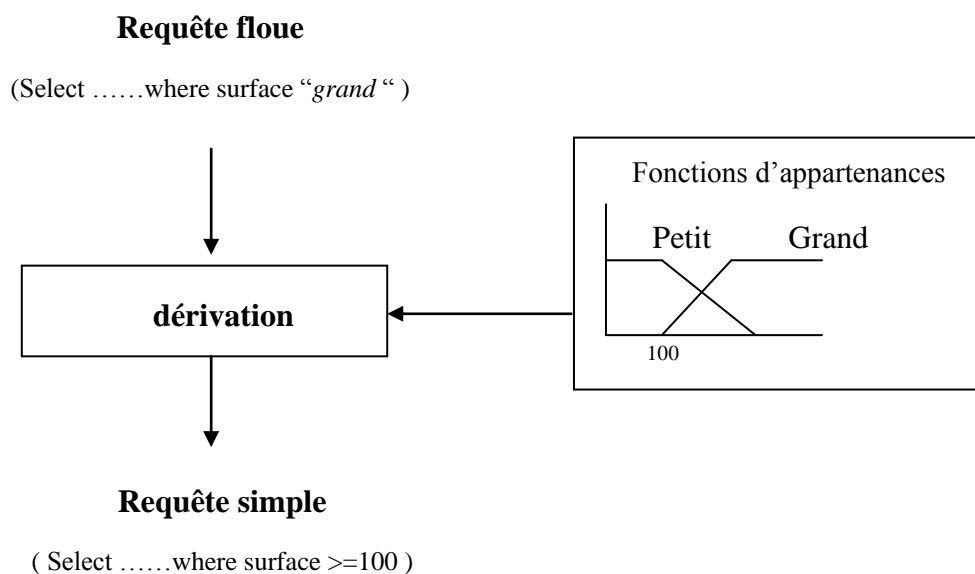


Fig 4.6 : Processus de traduction d'une requête floue en une requête simple

Le module de données

Elle reçoit du module de traitement une demande de recherche de variable dans une source de données. Ce module exécute la requête SQL simple demandée par le module de traitement et retourne à ce module les valeurs trouvées.

4.5 Interaction et communication entre agents

4.5.1 Modélisation des interactions entre agents

Les interactions entre les agents sont souvent modélisées avec des approches logiques, des outils comme des graphes de transition d'automates à états finis, des réseaux de Pétri, etc. En effet, l'OMG (Object Management Group) et FIPA (Foundation for Intelligent Physical Agents) ont proposés une extension d'UML, nommé AUML (pour Agent UML) comme un langage de modélisation qui doit permettre la spécification de système d'agents.

Dans notre travail, nous utilisons le langage AUML pour modéliser les interactions entre les Agents Système et les Agent flous de notre modèle. Le diagramme de protocoles d'interaction présenté dans la figure 4.7, modélise le mécanisme de fonctionnement du modèle FMAS décrit dans la section 3.

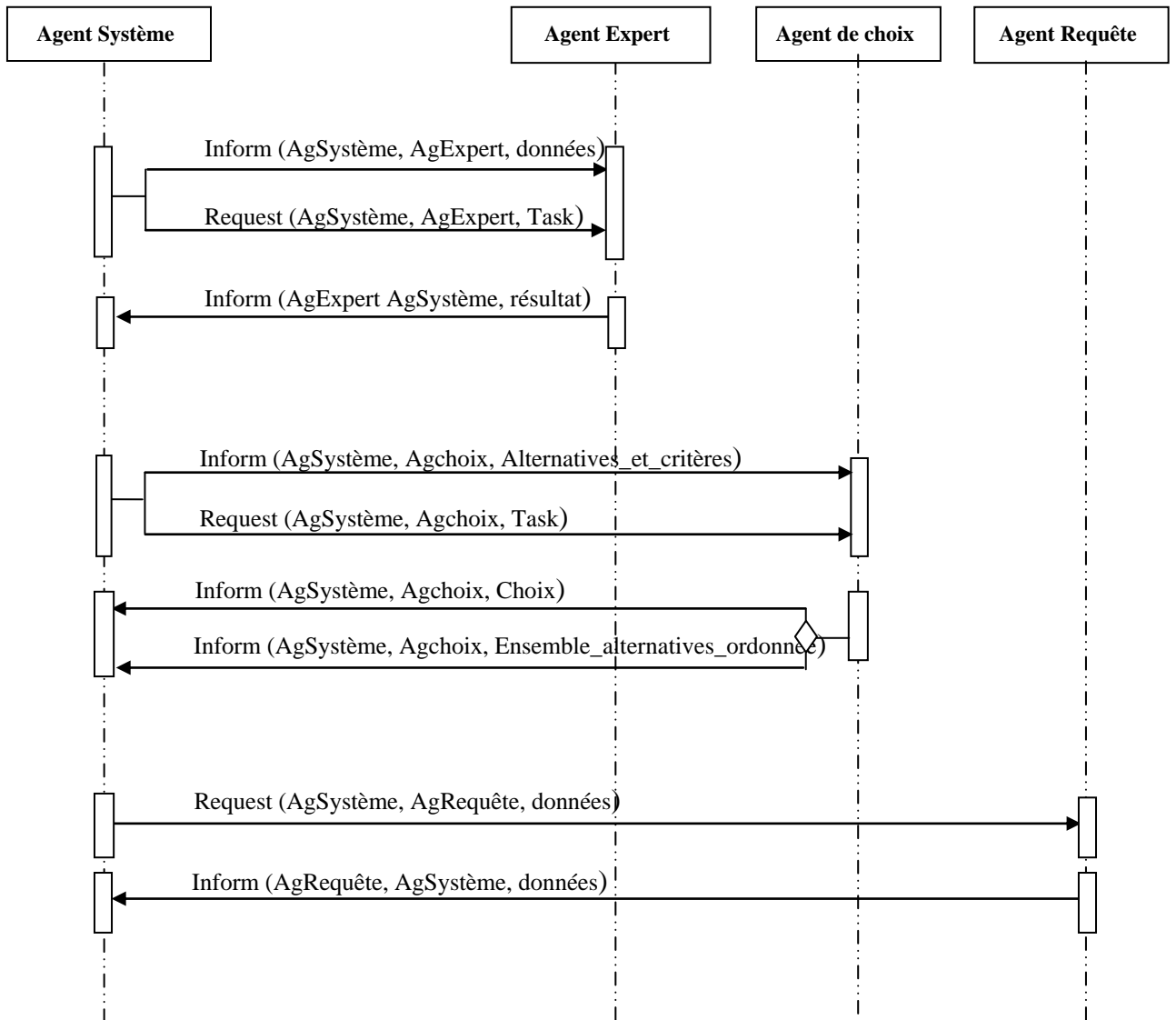


Fig 4.7 : Diagramme de protocoles d'interactions entre agents

4.5.2 Communication entre agents

Dans notre modèle, les agents communiquent à travers les échanges de messages. Ces messages sont représentés dans le langage de communication FIPA-ACL. La plate-forme JADE, que nous utilisons a retenu comme langage d'expression des messages le langage de communication ACL de la FIPA [FIPA, 1999]. Les primitives de ces langages sont appelées actes de communication (et performatifs pour KQML). Elles définissent les actions que les

agents peuvent faire pour communiquer les uns avec les autres. Dans un cas comme dans l'autre, un message est essentiellement constitué d'une primitive qui porte la valeur illocutoire et d'un contenu propositionnel. Le format de représentation du contenu n'est pas imposé. La définition des messages ACL est très importante, puisqu'elle détermine les actions adéquates du comportement d'un agent. Les détails sur les messages échangés peuvent aussi être inspectés.

La plate-forme Jade nous permet aussi d'ajouter des nouvelles ontologies. Une ontologie est une spécification d'objets, de concepts et de relations dans un domaine d'intérêt. L'intérêt d'une ontologie quand elle est partagée par des agents pour représenter leur connaissance est qu'ils ont les moyens de comprendre les « mots » utilisés dans une communication.

4.6 Conclusion

Dans ce chapitre, nous avons présenté une nouvelle approche pour la prise en compte de l'aspect flou dans un SMA: le modèle FMAS. La principale différence qu'il y a entre nos travaux et ceux déjà effectués sur les SMA flous est que, dans notre modèle, l'aspect flou est intégré dans un niveau multi-agent par l'utilisation des agents flous indépendants des autres entités de systèmes. Nous avons détaillé aussi la structure interne de différents agents flous, issus de modèle FMAS, que nous avons proposés.

Le modèle de conception de SMA flou que nous avons proposée dans ce travail est développé suivant une approche générique, i.e., indépendamment de toute application ou domaine d'application. Pour motiver nos travaux de recherche d'un point de vue pratique, le chapitre suivant décrit l'application de notre modèle avec une étude de cas, ainsi que sa projection sur un environnement opérationnel (c.-à-d, comment implémenté les différents agents flous du modèle FMAS avec la plateforme multi-agents JADE).

Chapitre 5

Modélisation et réalisation d'une application d'aide à la décision dans un marché virtuel

5.1 Introduction

L'utilisation de systèmes d'aide à la décision (SAD) en commerce électronique est augmentée dans les dernières années. L'objectif de ces systèmes est de permettre aux utilisateurs des marchés électroniques de faire des transactions dans les meilleures conditions et de les aider dans leurs décisions. La conception et la mise en œuvre de SAD (avec les problèmes de communications homme machine qui les accompagnent) se caractérise souvent par la constante manipulation d'informations dont beaucoup sont imparfaites. L'utilisation du paradigme multi-agents pour la réalisation de ces systèmes implique la nécessité d'y intégrer des mécanismes qui tiennent en compte le traitement des informations floues. Cela rend nécessaire la conception de systèmes multi-agents (SMA) dotés de caractéristiques floues. Pour la modélisation et la réalisation de ce système nous avons choisi d'utiliser le modèle FMAS.

Ce chapitre concerne la présentation de l'utilisation du modèle FMAS pour le développement d'une application de gestion et d'aide à la décision dans un marché virtuel. Après la présentation du système à réaliser dans la première section, nous décrivons dans la deuxième section l'application du modèle FMAS pour la modélisation et la réalisation de ce système. Nous précisons ensuite la plateforme d'implémentation JADE et comment les agents flous de notre modèle (Expert, de choix et de requête) peuvent être implémentés en utilisant cette plateforme.

5.2 Présentation et objectif

Le sujet de cette étude de cas est la réalisation d'un système d'aide à la décision en ligne dans un marché virtuel. Ce système doit permettre aux utilisateurs du marché électronique d'accomplir leurs transactions en ligne d'une manière flexible et doit les aider dans leur décision.

Notre système doit permettre aux clients du marché d'utiliser des termes linguistiques vagues et imprécis dans les critères de qualification des données recherchées et d'exprimer des préférences entre ces critères, ce qui est souvent une demande légitime des utilisateurs. Par exemple, le système doit permettre à un utilisateur qui cherche, via Internet, des annonces d'offres de location de biens immobiliers, d'utiliser des requêtes floues comme : surface *moyenne*, loyer *modéré* et lieu *proche* à Constantine. Ainsi, les données retournées par le système doivent être ordonnées et présentées à l'utilisateur selon ces préférences: Loyer, Surface, Nbr pièces, lieu....

D'autre part, notre système doit aider les vendeurs à estimer leurs biens immobiliers au juste prix. En générale, les vendeurs qui font recours à ce genre de transactions ne sont pas des experts, et pour réaliser une vente rapide, leur bien immobilier mis en vente doit être estimé au juste prix, parce que, comme il est indiqué dans le site www.Seloger.com, dans 90% des cas, quand la vente ne se fait pas, c'est que le prix n'est pas le bon. Il existe de nombreux sites Web qui proposent ce service aux utilisateurs comme le site www.Seloger.com et le site www.MeilleursAgents.com. Il faut doter donc notre marché d'un agent expert, remplacer l'expert immobilier humain, pour aider les vendeurs à estimer leurs biens au juste prix.

L'objectif donc est de concevoir un système d'aide à la décision permettant à des clients d'obtenir des informations sur les appartements, leur prix, surface.... et de passer des commandes en se basant sur les informations ainsi obtenues. Et permette aux vendeurs d'estimer leurs biens aux justes prix.

Afin de réaliser ce système, nous avons choisi d'utiliser notre modèle de conception de SMA flous. La section suivante présente l'application du modèle FMAS pour la modélisation de système d'aide à la décision désirée.

5.3 Modélisation de système suivant le modèle FMAS

5.3.1 Architecture du système

L'architecture du système est présentée dans la figure 5.1. Notre marché est composé avec trois types d'agents systèmes, modélisent les acteurs de notre marché. Les *Agents Clients* simulent les clients du marché, les *Agents Vendeurs* simulent les vendeurs (ou les propriétaires des biens immobiliers), et l'*Agent broker* qui s'occupe de la médiation entre les clients et les vendeurs dans la place de marché. Avec les agents systèmes nous avons introduit les trois types d'agents flous suivants: les *Agents Ressources* pour la recherche flexible d'informations, l'*Agent Expert Immobilier* pour l'estimation des prix des biens immobiliers, et l'*Agent d'évaluation* pour aider les clients dans leurs choix. Nous allons détailler par la suite les rôles de ces agents flous.

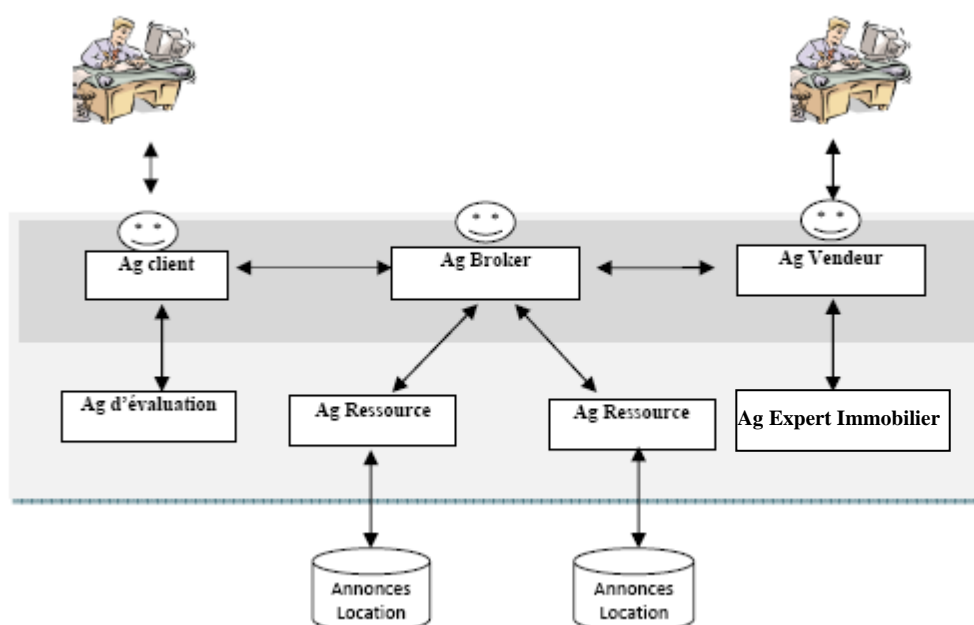


Fig 5.1 : Architecture du marché basé agents

5.3.2 Agents flous utilisés

Dans cette section, nous allons présenter de manière détaillée les rôles des agents flous introduits dans notre marché.

Agents Ressources

Ces agents sont des agents flous de requête (section 4.4.3). Dans notre système, l'agent ressource a pour but est d'extraire des données, dans une base de données ou dans une page Web structurée, de manière flexible. Cet agent reçoit des requêtes envoyées par le broker et les transforme en requêtes SQL simple, afin d'extraire l'information requise des bases de données ou de rechercher l'information demandée dans une page HTML. L'information trouvée est ensuite traduite en message pour répondre à l'agent broker.

Agent d'évaluation

Cet agent est un agent flou de choix (voir section 4.4.2), son rôle est d'aider les clients dans leurs choix. Il reçoit un ensemble de propositions (appartements ou maisons), et retourne un ensemble de propositions ordonné selon les préférences du client (Fig 5.2).

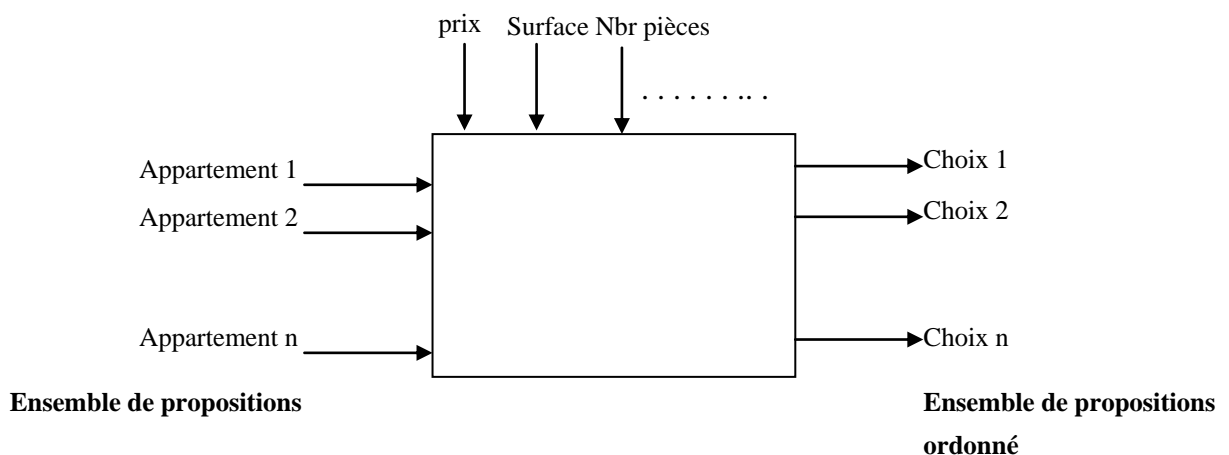


Fig 5.2: Agent d'évaluation

Agent expert immobilier

Cet agent est un agent expert flou (voir section 4.4.1), son expertise permet de calculer les prix approximatifs des biens immobiliers, selon leurs caractéristiques. Cette agent reçoit en entrée les caractéristiques des biens immobiliers et génère en sortie, par inférence floue, leurs prix approximatifs (Fig 5.3).

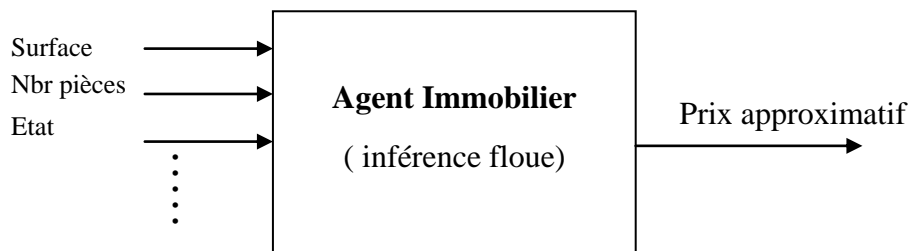


Fig 5.3: Agent expert immobilier

5.4 Scénarios d'exécution

Dans notre système deux scénarios d'exécutions sont envisageables. Le premier lié aux clients et le deuxième lié aux vendeurs ou propriétaires des biens immobiliers.

5.4.1 Scénario 1

Lorsque un client s'introduit dans notre marché, le système lui crée un agent client, et après que le client définit ses préférences pour un appartement (prix, surface, Nbr Pièces, ...), cet agent contacte l'agent broker pour lui fournir la liste des appartements qui vérifiés la requête du client. Pour traiter cette requête, qui peut être floue, l'agent broker appelle les agents Ressources, qui la traduit en une requête simple, extrait les données et les transmet vers l'agent broker.

Lorsque le client reçoit l'ensemble des propositions possibles, il demande l'aide de l'agent d'évaluation, cette agent évalue et ordonne les appartements selon les préférences de client.

En fin, et après que le client décide d'acheter ou d'aloier un appartement, l'agent client informe le broker de cette choix pour contacter le vendeur. Ce processus d'achat est montré dans le diagramme d'interaction suivant (Fig 5.4).

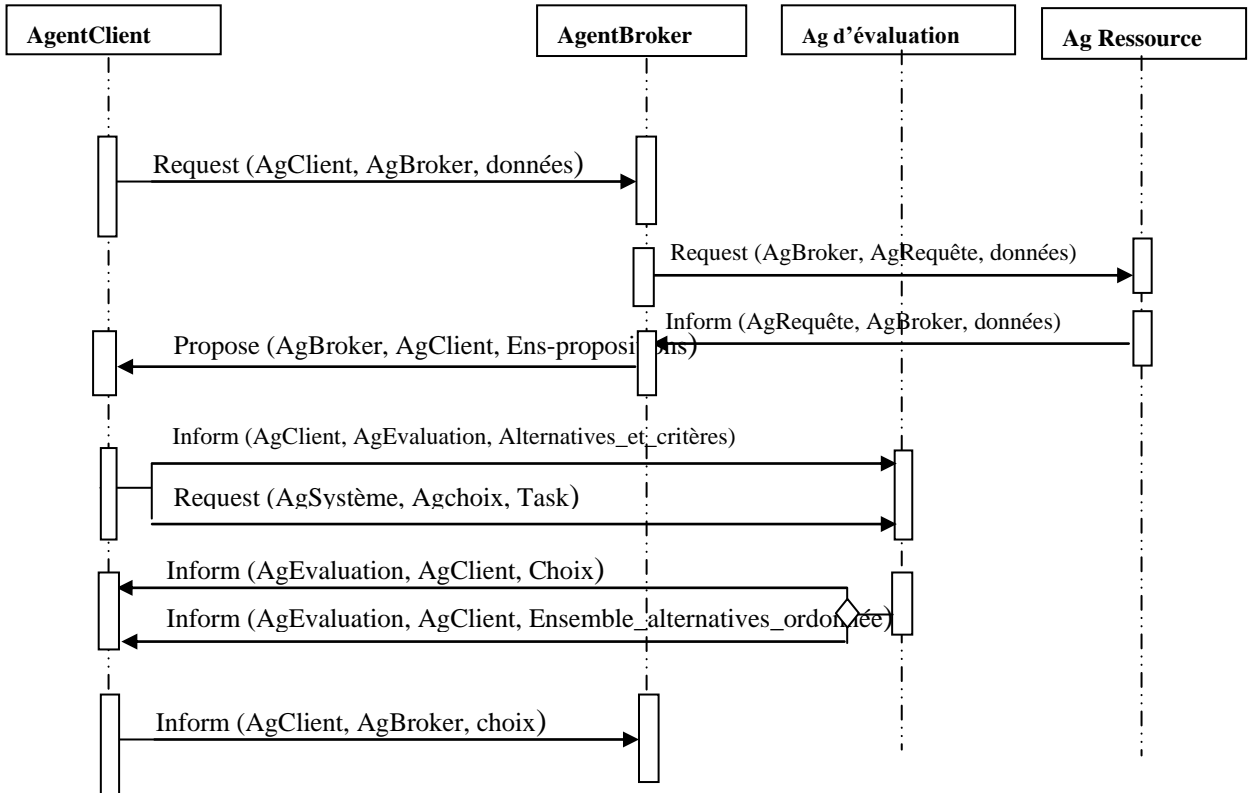


Fig 5.4 : Diagramme de protocole d'interaction pour le scenario 1

Exemple : Considérons un utilisateur qui consulte, via Internet, une BD d'offres de location de biens immobiliers. L'utilisateur souhaite trouver un appartement de préférence à Constantine ayant une surface « moyenne » et un loyer « modéré » avec une Nbr-pièces entre 3 et 5. D'autre part, l'utilisateur souhaite que les données retournées soient ordonnées et présentées à lui selon ces préférences (surface non petit, loyer non grand, et lieu proche de la nouvelle ville).

Représentation de la requête du client

Pour représenter la requête du client nous utilisons les variables linguistiques suivantes :

- **Surface de l'appartement** : pour décrire la surface d'un appartement on utilise l'ensemble des termes suivants : {petite, moyenne, grande}. Ces termes peuvent être caractérisés par les fonctions d'appartenance représentées sur la figure suivante.

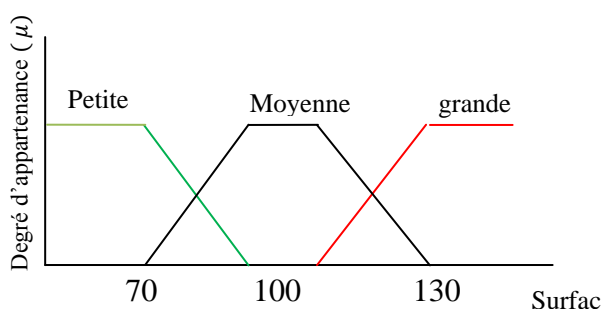


Fig 5.5: Variables linguistiques pour décrire la surface d'un appartement

- **Loyer** : pour décrire le loyer d'allocation d'un appartement on utilise l'ensemble des termes suivants: {petit, moyen, grand}. Ces termes peuvent être caractérisés par les fonctions d'appartenance représentées sur la figure 5.2.

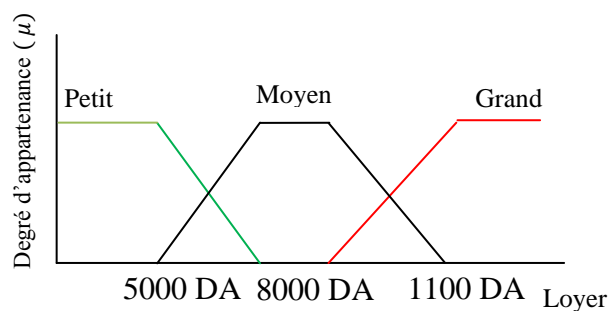


Fig 5.6: Variables linguistiques pour décrire le loyer

- **Distance**: Afin de décrire la distance à une ville on utilise l'ensemble des termes suivants : {proche, loin, très-loin}. Ces termes peuvent être caractérisés par les fonctions d'appartenance représentées sur la figure suivante.

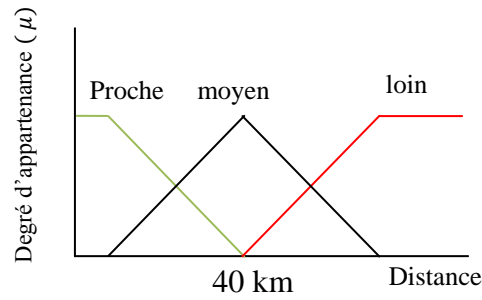


Fig 5.7: Variables linguistiques pour décrire la distance à une ville

Traitement de la requête par l'Agent Ressource (Fig 5.3)

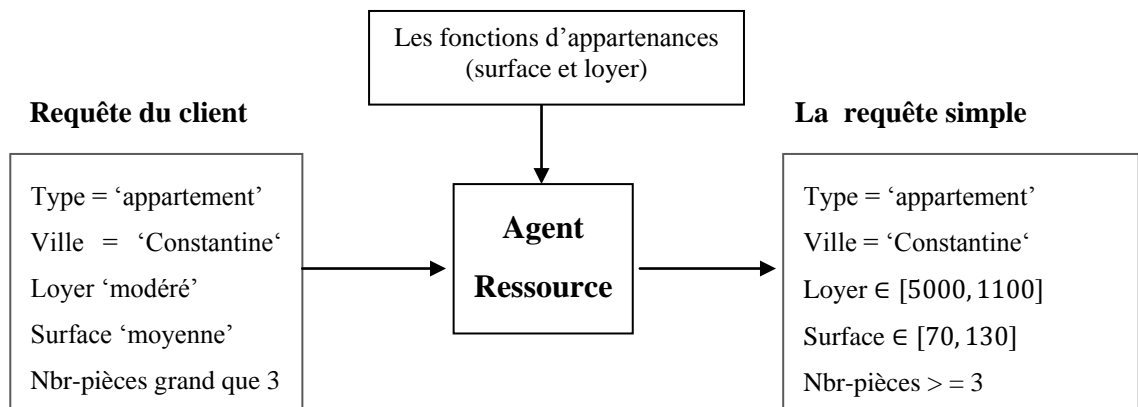


Fig 5.8: Processus de dérivation de la requête du client en une requête simple

Donc la requête SQL exécutée est la suivante :

```
Select * From Announce Where Ville = 'Constantine'
and Loyer between (7000 , 13000) and Surface between (70, 130)
and Nbr_pièces >= 3
```

Le résultat de l'exécution de cette requête est exprimé dans le tableau suivant :

	Lieu	Loyer (moyen)	Surface (moyen)	Nbr pièces
A_1	Constantine	80000	100	4
A_2	Constantine	9000	80	4
A_3	Constantine	10000	110	5

Fig 5.9: Résultat d'exécution de la requête du client

L'évaluation des propositions par l'agent d'évaluation

Le résultat de cette évaluation avec les niveaux d'adéquation des propositions est montré dans le tableau suivant.

	$\mu_{c1}(ai)$ (loyer non grand)	$\mu_{c1}(ai)$ (proche au centre ville)	$\mu_{c1}(ai)$ C3 (Surface non petit)	niveaux d'adéquation Min ($\mu_{c1}(ai), \mu_{c1}(ai), \mu_{c1}(ai)$)
A_1	1	0.6	1	0.6
A_2	0.8	0.8	0.8	0.8
A_3	0.6	0.9	1	0.6

Fig 5.10 : Résultat d'évaluation des propositions par l'Agent d'évaluation

La meilleure proposition donc est la deuxième avec un degré de satisfaction de : 0.8.

5.4.2 Scénario 2

Lorsque un vendeur ou un propriétaire d'un bien immobilier entré dans notre marché, le système à lui créé un agent vendeur, qui lui permet de déposer son bien immobilier dans notre marché. Dans le cas où le vendeur demande l'estimation de son bien, l'agent vendeur envoie les caractéristiques de son bien vers l'agent expert immobilier. L'agent expert immobilier calcule le prix approximatif par inférence floue et retourne le résultat vers l'agent vendeur.

Exemple : Considérons un vendeur qui a un appartement avec une surface de 100 m^2 , et 5 pièces. Ce vendeur demande l'aide de notre système pour lui aider à estimer son bien immobilier.

Pour des raisons de simplicité le prix approximatif sera calculé seulement selon la surface et le Nbr de pièces. Les variables linguistiques et les règles utilisées sont présentés ci-dessous.

Variables linguistiques utilisées

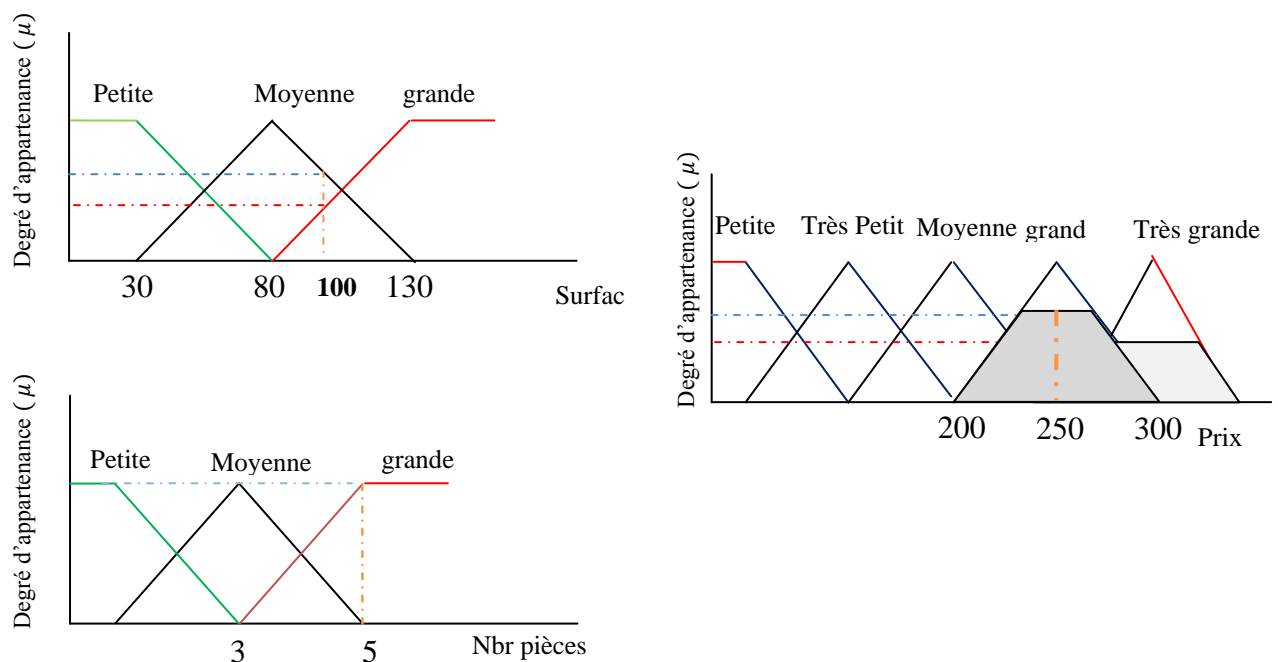


Fig 5.11: Les partitions floues utilisées pour calculer le prix

Quelques règles utilisées :

- R1:** SI surface est *grande* ET nbrpièce est *moyen* Alors prix est *grand*
- R2:** SI surface est *moyenne* ET nbrpièce est *grand* Alors prix est *grand*
- R3:** SI surface est *moyenne* ET nbrpièce est *moyen* Alors prix est *moyen*
- R4:** SI surface est *grande* ET nbrpièce est *grand* Alors prix est *tresgrand*

Le prix approximatif : Après l'exécution des règles floues en utilisant la méthode de Mamdani, on utilise la méthode de défuzzification par centre du maxima, le prix approximatif trouvé est : 250 million centimes.

5.5 Implémentation

5.5.1 Choix techniques

Le meilleur moyen pour construire un système multi-agent (SMA) est d'utiliser une plate-forme multi-agent. Une plate-forme multi-agent est un ensemble d'outils nécessaire à la construction et à la mise en service d'agents au sein d'un environnement spécifique. Ces outils peuvent servir également à l'analyse et au test du SMA ainsi créé. Ces outils peuvent être sous la forme d'environnement de programmation (API) et d'applications permettant d'aider le développeur. Il existe actuellement plusieurs plates-formes pour le développement des systèmes multi-agents, telles que Madkit [Madkit, 2003], Zeus [Azvine et al. 2000], Swarme [Minar, 1996] et JADE [Bellifemine et al, 2003], etc. Cependant, ces plates-formes ne proposent pas une solution faciliter l'utilisation des protocoles d'interaction, à l'exception de la plate-forme JADE.

La réalisation de notre application, sur l'étude de cas qui concerne la réalisation d'une application de gestion et d'aide à la décision dans un marché nous impose de choisir des outils d'implémentation. L'étape de développement nécessite des outils technologiques adaptés au domaine de recherche, et à l'architecture de l'application concernée. Notre choix s'est porté sur le langage de programmation Java et la plateforme de développement des SMA : JADE pour les raisons suivantes :

- Il est simple de faire une mapping des agents flous issus de notre modèle dans un code Jade.
- Il est simple de créer des agents avec jade.
- Jade gère la communication entre les agents et offre des interfaces de gestion des agents.
- La plateforme Jade répond à plusieurs fonctionnalités et offre une large gamme de bibliothèques.
- les agents développés en JADE sont écrits totalement en JAVA qui est un langage facile et basé sur la notion d'objet.

5.5.2 Interfaces utilisateurs

L'interface homme machine du marché permet aux utilisateurs d'être identifiés, soit en tant que client ou fournisseur (Vendeur). La distinction entre les utilisateurs et l'interface qui leur est liée se fera au moment de leur connexion (voir fenêtre en figure 5.6). Deux types d'interface seront disponibles :

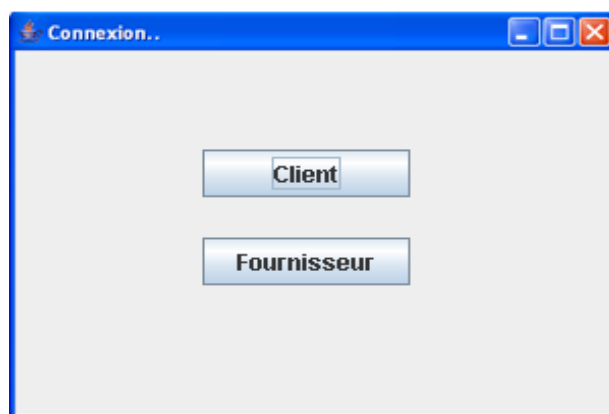


Fig 5.12 : Fenêtre de connexion

- L'interface pour les utilisateurs de type client.
- L'interface pour les utilisateurs de type vendeur.

Interface coté client

Lorsqu'un client se connecte, la fenêtre représentée dans la figure 5.13 s'affiche. Elle lui permettra de chercher un bien immobilier en utilisant des requêtes floues.

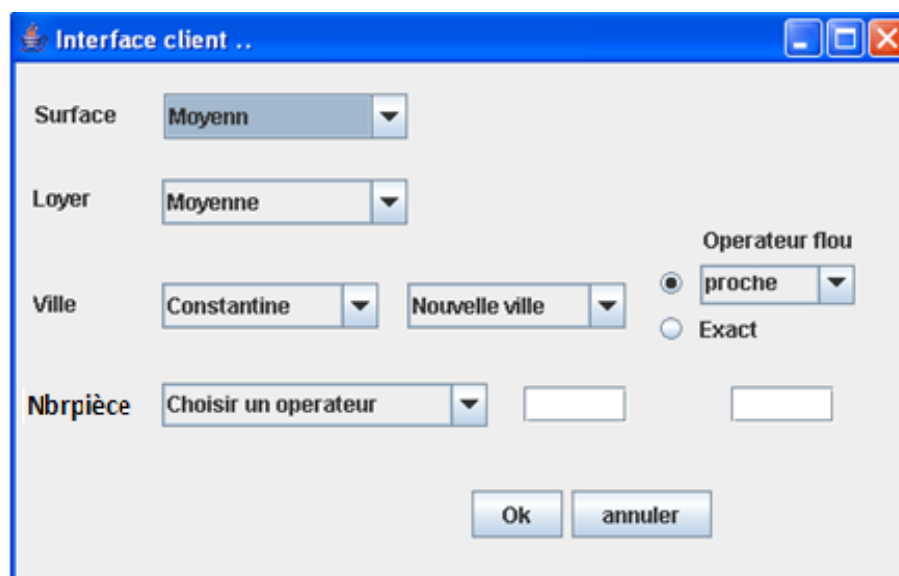


Fig 5.13 : Exemple d'interface client

Interface coté vendeur

Lorsqu'un fournisseur se connecte et demande l'estimation de son bien immobilier la fenêtre ci-dessous s'affichera. Elle lui permettra d'entrer les caractéristiques (Types, Surface, Nombre de pièces, Etat....) de son bien immobilier, qui sera utilisés pour calculer le prix.

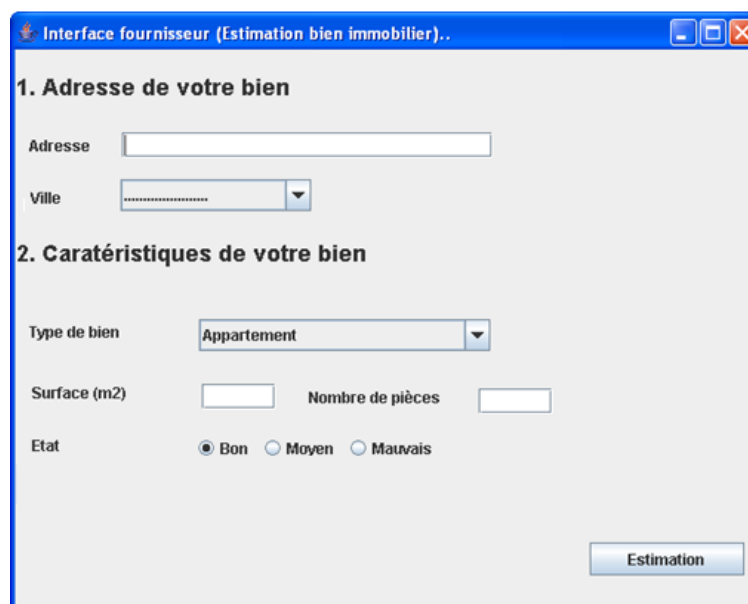


Fig 5.14 : Exemple d'interface coté vendeur

5.5.3 Implémentation des agents avec Jade

Nous présentons dans cette section les règles qui nous permettent d'implémenter facilement les agents flous issus du modèle FMAS avec la plateforme JADE.

La création des agents

La création d'un agent JADE est simple que la définition d'une classe qui hérite la classe `Jade.core.agent` et implémente la méthode `setup()`. Cette dernière est destinée d'inclure les initialisations d'un agent. Le comportement réel qu'un agent doit faire est représenté comme des comportements (behaviours) de JADE. Dans notre cas, le module communication est lancé par la méthode `setup()` comme indiqué dans le code JADE ci-dessous (Fig 5.16).

```
import jade. core.*;
public class Agent_Requete extends Agent {
    // Possible variables
    protected void setup {
        // to do: add necessary start-up code
        addBehaviour(new module_communication(this));
    }
}
```

Fig 5.16: Spécification partielle d'un agent exprimée avec JADE

Les modules de communication

Les modules de communication de différents agents flous de notre modèle sont dérivés de la classe `CyclicBehaviour`. Il s'exécutera donc continuellement. Une fois qu'un message est arrivé, interprète les informations obtenues, déclenche le module de traitement et retourne à l'état initial (attendre les messages). Le code de comportement `Module_communication` est montré ci-dessous (Fig 5.17).

```
import java.util.*;
import jade.core.*;
import jade.core.behaviour.*;
import jade.lang.acl.ACLMessage.*;
class Module_communication extends CyclicBehaviour {
    public Module_communication (Agent a) {super(a);}
    public void action() {
        // attente pour message
        ACLMessage received = myAgent.receive();
        if(ACLMessage == null) {block();}
        else {
            // interpretation du message
            .....
            // Start Module_traitement (agent expert)
            myAgent.addBehaviour(new ModuleTraitement(x1,x2,..xn));
        }
    }
}
```

Fig 5.17 : Spécification partielle du module de communication exprimée avec JADE

Le module de traitement des agents experts

Le module de traitement est dérivé de la classe `SequentialBehaviour`. Ce module prend comme paramètres d'entrées un ensemble d'informations (les résultats d'interprétation des messages par le module de communication) et calcule la solution par inférence floue. La logique suivie dans les `SequentialBehaviour` est simple et intuitive. Le `Behaviour` commence par exécuter le premier sous-`Behaviour` et lorsque celui-là termine son exécution, il passe au prochain `Behaviour`, et ainsi de suite. Les sous-`Behaviours` (Fuzzification, module Inférence, Défuzzification) sont ajoutés par la méthode `addSubBehaviour()`. Voici le code du Module de traitement montré dans la figure 518.

```
public class ModuleTraitement extends SequentielBehaviour {
    ensemble informaton;// x1, x2, ...xn
    public Moduletraitement (ensemble information) {
        this.information = information;
    }
    public void action() {
        this.addSubBehaviour(new Fuzzificztion());
        this.addSubBehaviour(new Infrence());
        this.addSubBehaviour(new Defuzzification());
    }
    public boolean done() {
        return true;
    }
}
```

Fig 5.18 : Spécification partielle du module de traitement exprimée avec Jade

Les autres modules

Les autres modules des agents flous issus du modèle FMAS (module de traitement et module de données pour les agents de requêtes, module d'évaluation pour les agents de choix) sont implémenté comme des comportements de type SimpleBehaviour. La figure suivante présente une partie de code du module d'évaluation qui ordonne les propositions.

```
public class ModuleEvaluation extends SimpleBehaviour {
    Ensemble propositions pros;
    public PlanRetrieval( Ensemblepropositions p) {
    }
    public void action(){
        le code qui permet d'ordonne les proposition
        .....
    }
}
```

Fig 5.19 : Spécification partielle du module d'évaluation avec Jade

5.5.4 Communication entre agents

En adoptant l'environnement JADE, la communication entre agents est facilitée par des méthodes prédéfinies. Par exemple, un agent obtient un identificateur unique (GUID : Globally Unique Identifier) à l'aide de la méthode `setName (AID)` de la classe `DFAgentDescription`, enregistrer à la plate-forme JADE (la méthode `register (this, DFAgentDescription)` de la classe `DFService`). La communication des agents est basée sur l'envoi de messages (les méthodes `send (message)` et `receive (message)` de la classe `ACLMessage`). L'environnement JADE offre la possibilité de suivre la communication entre des agents au moyen de l'agent "sniffer" qui est un GUI (voir Fig 5.22). La figure suivante présente le code Jade d'un message envoyé par le Broker vers le client.

```
Enproposition pros;
// l'ensemble des proposition ou d'alternatives à évaluer
.....
try {
    ACLMessage msg = new ACLMessage (ACLMessage.Request);
    msg.addReceiver (new AID);
    msg.setContentObject (pros);
    msg.setLanguage ("JavaSerialization");
    send(msg);
}
catch (IOException e ) {
    e.printStackTrace ();
}
```

Fig 5.20 : Spécification partielle d'envoi d'un message dans Jade

5.5.5 Simulation des interactions entre les agents

Nous allons maintenant simuler l'interaction entre les différents agents de notre système. L'interaction est basée sur l'envoi de message ACL entre agents.

La figure 5.4 correspond à l'interface graphique (GUI) de l'agent RMA (Remote Monitoring Agent) dont le rôle est de contrôler et superviser la plate-forme; il est responsable de l'authentification et l'enregistrement des agents du système.

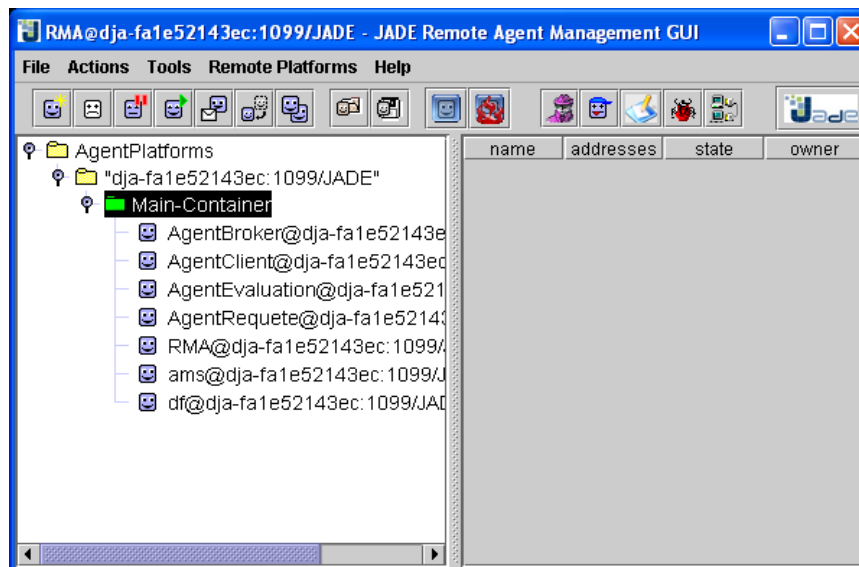


Fig 5.21 : Le GUI de l'agent RMA

La figure ci-dessous représente la façon et l'ordre d'échange de messages entre les agents du marché virtuel, selon le protocole Contract-net de FIPA.

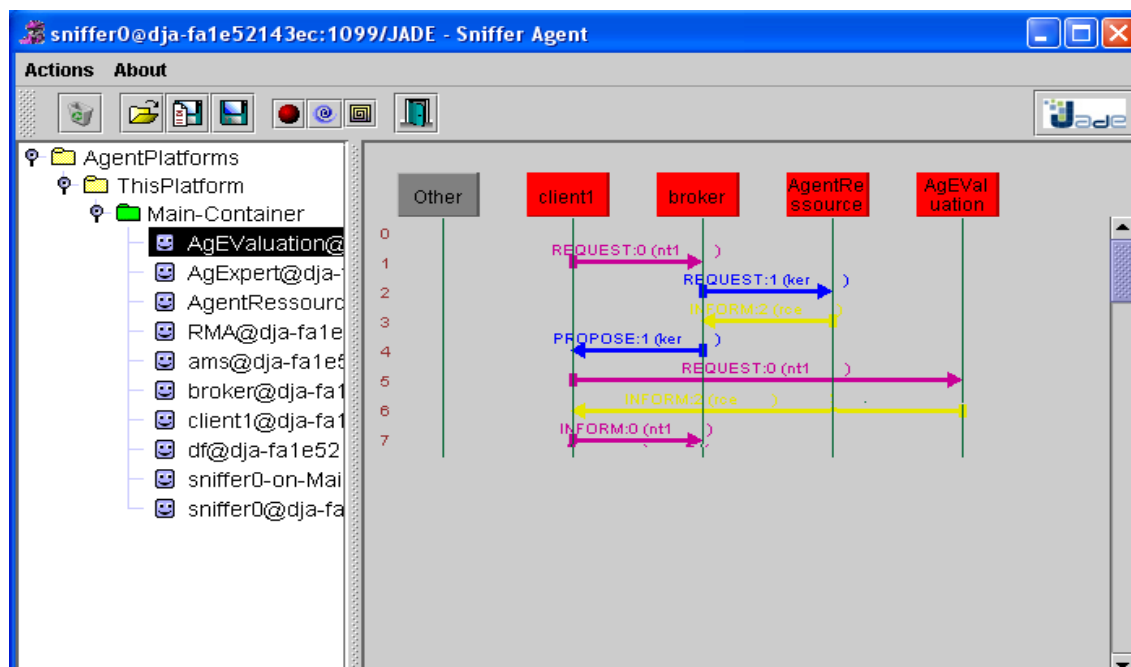


Fig 5.22 : Simulation des interactions entre agents par l'agent Sniffer

5.6 Conclusion

Dans ce chapitre, nous avons présenté l'application de notre modèle, le modèle FMAS, dans une étude de cas, qui consiste en la modélisation et la réalisation d'un système d'aide à la décision dans un marché virtuel de biens immobiliers. Après la modélisation de système en utilisant notre modèle avec ses différents agents flous, nous avons présenté les scénarios d'exécution possibles avec des exemples sur l'utilisation des agents flous de requête, expert, et de choix. Et pour l'implémentation de système, nous avons utilisé la plateforme JADE. Dans ce cas, nous avons proposé des règles qui permettent d'implémenter facilement les agents flous issus du modèle FMAS, ainsi que leurs différents modules. Des portions de codes sont présentées pour clarifier certaines opérations.

Conclusion générale

Dans notre travail, nous nous intéressons au problème de conception de systèmes multi-agents dotés des caractéristiques flous pour répondre aux besoins des systèmes complexes modernes qui doivent traiter avec des informations imparfaites. Pour résoudre ce problème nous avons proposé dans ce mémoire une nouvelle approche pour la conception de systèmes multi-agents flous, le modèle FMAS, avec ses trois types d'agents flous (Agent expert, Agent de choix, et agent de requête), qu'on peut l'utiliser dans n'importe quelle application lorsque les conditions demandent. Pour mettre en œuvre le modèle FMAS, nous avons choisi une application dans le domaine de l'aide à la décision. Cette application est pour l'instant restreinte à des interrogations simples dans le domaine de biens immobiliers. Nous espérons pouvoir prochainement l'étendre à un environnement plus large et plus complexe.

Dans notre contribution, le modèle proposé est d'abord générique puisque nous avons constaté, dans la littérature, l'absence d'un modèle générique pour la conception de SMA flou. Et contrairement aux travaux actuellement réalisés sur les systèmes multi-agents flous, qui proposent des architectures pour des applications spécifiques ou utilisent la logique pour traiter des problèmes dans les SMA, notre modèle de conception s'appuie sur l'idée d'intégrer la logique floue dans un niveau multi-agent, par l'utilisation des agents flous indépendants des autres agents de systèmes, il est donc indépendant de l'application dans laquelle nous allons l'utiliser. En effet, Notre modèle va permettre de construire les applications complexes pouvant bénéficier des avantages de l'approche multi-agent et des capacités de la logique floue, tel que sa capacité de représenter et de manipuler des connaissances imparfaites.

Dans l'état actuel de cette étude, nous envisageons plusieurs améliorations ou extensions possibles au modèle FMAS. Ces travaux se situent à plusieurs niveaux.

1. la validation de notre modèle à travers diverses applications dans lesquelles les conditions de la complexité de système et de l'imprécision, de l'incertitude et de l'incomplétude d'informations demandent l'utilisation de systèmes multi-agents basé sur la logique floue.
2. L'enrichissement du modèle FMAS par d'autre type d'agents flous plus spécialisés, pour traiter des problèmes spécifiques dans les SMA comme la coordination, la négociation, la planification...etc.
3. L'exploitation de la technologie du soft computing pour faire face à d'autres problèmes technique des SMA.

Bibliographie

- [Aliev, 2008] R.A. Aliev. Modelling and stability analysis in fuzzy economics. *Appl. Comput. Math.* 7 (2008), no.1, pp. 31-53.
- [Aliev et al, 1997] R.A. Aliev, R.R. Aliev „Fuzzy Distributed System for Continuous Production, In *Application of Fuzzy Logic: Towards High Machine Quotient Systems*". (M.Jamshidi, A.Titli, L.Zadeh, S.Boverie, eds.), Prentice Hall PTR, New Jersey, 1997, pp.301-321.
- [Aliev et al, 1999] R.A. Aliev, R.R. Aliev, Fuzzy Distributed Intelligent manufacturing System, In *first European congress on Fuzzy and Intelligent Technologie. EUFIT*, Aachen, Germany, 1999, Part II, pp.229-235.
- [Andres, 1989] V. Andres. *Filtrage sémantique dans une base de données imprécises et incertaines: un système souple autorisant la formulation de requêtes composites pondérées*. Thèse, Université Paul Sabatier, Toulouse, France, 1989.
- [Araujo et al, 2003] J.E. Araujo Filho, K.H. Kienitz, “Adaptive reference driven decision-making process”, *Proc Intern. Conf. on Fuzzy Systems (FUZZ-IEEE)*, v.1, pp.452-457, St. Louis, 2003.
- [Azvine et al. 2000] B. Azvine, N. Azarmi, , and D. Nauck, (2000). *Intelligent Systems and Soft Computing: Prospects, Tools and Applications*, volume 1804 of *LNCS*. Springer-Verlag.
- [Bartel, 2006] S. Bartel, *Interrogation floue de base de données*. Rapport de projet de 3ème année ENSSAT, avril 2006.
- [Bellifemine et al, 2003] F Bellifemine, G Caire, T Trucco, et G Rimassa. Jade Programmer's Guide. JADE 3.0b1. Available at <http://sharon.cselt.it/projects/jade/>, 2003.
- [Bellman and Zadeh, 1970] R.E. Bellman, L.A. Zadeh, “Decision Making in fuzzy environment”, *Management Science*, 1970, v. 17, pp. B141-B264.
- [Bonnal, 2004] P. Bonnal, « Where do we stand with fuzzy project scheduling », *journal of construction engineering and management*, Vol. 130, no 1, 2004, p. 114-128.

- [Bosc et al, 1992] P. Bosc, O. Pivert, Some Approaches for Relational Databases Flexible Querying, *International Journal of Intelligent Information Systems*, 1 :323-354 (1992).
- [Bosc et al, 1998] P. Bosc, L. Liétard, O. Pivert, Bases de Données et Flexibilité : Les Requêtes Graduelles, *Techniques et Sciences Informatiques*, 17(3) :355-378 (1998).
- [Bouchon, 1995] B. Bouchon-Meunier. La logique floue et ses applications. Addison-Wesley France, 1995.
- [Burkhart, 1994] R. Burkhart,. The SWARM multi-agent simulation system. In *Object Oriented Programming Systems, Languages and Applications (OOPSLA) '94*. 1994.
- [Cardon, 1998] A. Cardon, la complexité des systèmes d'expression du sens. Technical report IBP, 1998.
- [Cardon, 1998] A. Cardon,. *Modélisation des Systèmes Adaptatifs par Agents: vers une Analyse-conception Orientée Agent*. LIP6. 1998.
- [Chaib-draa, 1999] B. Chaib-draa, (1999). *Agents et systèmes multi-agent*. Université de Laval, Québec. Notes de cours.
- [Côté, 1999] M. Côté and N. Troudi. Une architecture multi-agent pour la recherche. *L'expertise Informatique*, 3(3), 1999.
- [Cox, 1994] E.. COX, «The fuzzy systems handbook». Academic press, 1994
- [Daniels, 2000] M. Daniels, An open framework for agent-based modeling. In *Applications of Multi-Agent Systems in Defense Analysis*, Los Alamos National Labs. 2000.
- [Drogoul, 1993] A. Drogoul. De la simulation multi-agents à la réalisation collective de problèmes : un étude de l'émergence de structure d'organisation dans les systèmes multi-agents. PhD thesis, Université de paris VI. 1993.
- [El-Nasr et al, 2000] M.S. El-Nasr, J.Yen, R.R. Ioerger, “FLAME – Fuzzy logic adaptive model of emotions”, *Automatic Agent and Multiagent*, v.3, n.3, pp.219-257, 2000.
- [Fazlollahi et al, 2000] B. Fazlollahi, R.M. Vahidov, R.A. Aliev, “Multi-agent Distributed Intelligent System based on Fuzzy-Decision Making”, *Intern. Journal of Intelligent System*, 2000, v.15, n.9, pp. 383-398.
- [Fazlollahi et al, 2004] B. Fazlollahi, R.A. Aliev, R.R. Aliev, *Soft Computing and its Applications in Business and Economics*. Springer-Verlag, Berlin, Heidelberg, 2004.
- [Ferber, 1995] J. Ferber *Les Systèmes Multi-Agents : Vers une Intelligence Collective*. InterEditions, Paris, 1995.

- [FIPA, 1997] Foundation for Intelligent Physical Agents (FIPA), "FIPA 07 Specification part 2 – Agent Communication Language", 1997, disponible sur : <http://www.fipa.org>.
- [Fogel, 1995] D. Fogel, Evolutionary computation, IEEE Press, 272 p , 1995.
- [Fuller, 2000] R. Fuller, Introduction to Neuro-Fuzzy Systems, Springer-Verlag, 289 p, 2000.
- [Genesereth and Nilsson, 1987] R.M GENESERETH , N.J. NILSSON Logical Foundations of Artificial Intelligence. Los Altos California: Morgan Kaufmann Publishers Inc., 1987. 401 p.
- [Gomez and Martinez, 1997] A.F. Gomez-Skarmeta, H. Martinez Barbera, "Fuzzy logic based intelligent agents for reactive navigation in autonomous systems", *In Proc. Fifth Intern. Conf. on Fuzzy Theory and Technology*, Raleigh, 1997, pp. 125-131.
- [Ghidini and Serafini, 1998] C. Ghidini and L. Serafini, Information integration for electronic commerce. In Pablo Noriega, C.S, editor, International Work-shop on Agent Mediated Electronic Trading (AMET 98) LNCS Minneapolis. Springer-Verlag, 1998.
- [Gutknecht and, 1997] O. Gutknecht and J. Ferber Madkit : Organizing heterogeneity with groups in a platform for multiple multi-agent systems. Technical report, LIRMM, UMR 9928, université de Montpellier II.
- [Guttman and Maes, 1998] R. Guttman and P . Maes, Agent-Mediated integrative negotiation for retail electronic commerce. In Pablo Noriega, C.S, editor, International Work-shop on Agent Mediated Electronic Trading (AMET 98) LNCS Minneapolis. Springer-Verlag, 1998.
- [He et al, 2001] M. He, HF leung and N Jennings, A fuzzy logic based bidding strategy in continuous double auctions, IEEE transaction on knowledge and Data Engineering (2001).
- [Herrera et al, 2004] E. Herrera-Viedma, F. Herrera, L. Mart´inez, J.C. Herrera, A.G. L´opez, Incorporating Filtering Techniques in a Fuzzy Multi-Agent Model for Gathering of Information on the Web, *Fuzzy Sets and Systems* 148 (1) (2004), 61–83.
- [Herrera et al, 2006] E. Herrera-Viedma, F. Herrera, L. Mart´inez, J.C. Herrera, A.G. L´opez: *A Fuzzy Linguistic Multi-agent Model Based on Semantic Web Technologies and User Profiles*, *StudFuzz* **197**, 105–120 (2006)
- [Holland, 1975] J.H. Holland. Adaptation in natural and artificial systems. Ann Arbor : The University of Michigan Press, 1975.

- [Jennings et al, 1998] N. R Jennings, M. Wooldridge et K. Sycara. A roadmap of agent research and development. *Int Journal of Autonomous Agents and Multi-Agent Systems*, 1(1): pp.7- 38, 1998.
- [Lai, 1997] H. lai UML, la notation unifiée de la modélisation objet. InterEditions, 1997
- [Larsen, 1999] Larsen H. L., An Approach to Flexible Information Access Systems Soft Computing, Proceedings of the 32nd Hawaii International Conference on System Sciences, 1999.
- [Madkit, 2003] Plate-forme MADKIT, 2003 <http://www.madkit.org/>
- [Muller, 1997] P. Muller, Modélisation objet avec UML. Editions Eyrolles.
- [Minar, 1996] N. Minar The swarm simulation system: a toolkit for building multi-agent simulations, 1996, <http://www.santafe.edu/project/swarm>
- [ROYER, 1990] J.C ROYER. : Un système d'apprentissage pour l'aide au réglage d'un instrument. Thèse Doct.: Institut National Polytechnique de Grenoble, 1990.
- [Shoham 93] Y. Shoham. Agent-oriented programming. *Artificial Intelligence*, 60:51-92, 1993.
- [Touzet, 1992] Claude Touzet. Les Réseaux de Neurones Artificiel. Introduction au Connexionnisme. Cours, Exercices et Travaux Pratiques. Juillet 1992.
- [Wermter, 200] S. Wermter, 2000, Neural Network Agents for Learning Semantic Text Classification, *Information Retrieval*. Vol.3 No.2, 29 p.
- [Wooldridge and Jennings, 1994] M. Wooldridge and N. Jennings, Agent theories, architectures an practice. A survey. In M. Wooldridge and N. Jennings, editors, *Intelligent Agents*, ECAI 1994, volume LNAI 890, page 1-32. Springer Verlag.
- [Yager, 1977] R.R. Yager, 1977. "Multiple Objective Decision-Making Using Fuzzy Sets," *Intl. J. Man-Machine Studies*, 9, 375-382.
- [Yager, 2001] R.R. Yager, "Penalizing Strategic Preference Manipulation in Multi-agent Decision Making", *IEEE Trans. Fuzzy Systems*, 2001, v.9, n.3, pp. 393-403.
- [Zadeh, 1965] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8 :338–353, 1965.
- [Zadeh, 1968] L.A. Zadeh. Fuzzy algorithm. *Information and Control*, 12 :94–102, 1968.
- [Zadeh, 1994] L.A. Zadeh. Soft computing and fuzzy logic. *IEEE Software*, 11(6) :48–56, 1994.