

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Larbi Ben M'hidi Oum El Bouaghi



Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie

Département de Mathématiques et Informatique

Thème

Une approche de renforcement flou des normes pour les Systèmes Multi-Agents

Présenté Par : SAHBI Abderrahim

Devant Le Jury:

Encadrant	Dr. Mohamed Sedik CHEBOUT	MCA	Université d'Oum Bouaghi
Président	Dr. Tahar GUERRAM	MCA	Université d'Oum Bouaghi
Examineur	Mr. Hamza MEROUANI	MAA	Université d'Oum Bouaghi

2022/2023

DÉDICACE

Je dédie ce modeste travail :

*A mes très chers parents de m'avoir toujours donné
le courage et d'être l'une des raisons qui me
pousse à me battre face à chaque obstacle.*

A tout ma famille

*A tous mes amis, surtout « SOLTANI NOOR
EDDEN » « NAIT SI MOHAND AYMEN »
« DJERMANE ABDELLAH »*

-A tous mes amis de la promotion 2023.

ABD ERRAHIM

REMERCIEMENTS

Nous tenons à remercier tout d'abord « Allah », le tout puissant, qui nous a donné le courage, la force et la patience pour bien mener ce modeste travail.

*Je tiens tout d'abord à remercier **Dr. Mohamed Sedik CHEBOUT** pour son soutien, son sérieux sa gentillesse et surtout pour son aide précieuse tout au long de l'élaboration de ce travail.*

Nous tenons à exprimer notre grande gratitude à toute personne qui, de près ou de loin, a contribué à la réalisation de ce travail.

Finalement, nos remerciements affectueux et chaleureux vont aussi à nos parents pour leur patience, et à toutes nos familles.

ABDERRAHIM



Sommaire

SOMMAIRE

Sommaire

DÉDICACE	III
REMERCIEMENTS.....	IV
Liste des Figures	IX
Liste des tableaux.....	XI
Liste des abréviations.....	XII
Résumé.....	XIV
Introduction générale	1

Chapitre I : Systèmes Multi-Agent Normatifs

1. Introduction.....	5
2. Les Systèmes Multi-Agents	5
2.1 Les agents :	6
2.1.1 Définition d'un agent :	6
2.1.2 Caractéristique d'un agent :	7
2.2 Environnement :	8
2-3 Interaction :	8
3- Les Systèmes Normatifs	10
4- Les Systèmes Multi-Agents Normatifs.....	11
4.1. La représentation des normes :	12
4.2. Mécanisme de renforcement :	12
4.2.1. Enforcer :	13
4.3. La Gestion et la mise à jour des normes :	13
5. Conclusion	15

Chapitre II : La logique floue

1-Introduction	17
2- Historique de la logique floue	17
3- Définition.....	18
4- Domaines d'application.....	18
5- Concept et définition	19
5.1 Définition d'un ensemble flou:	19
5.2 Variable et valeur linguistique:	19
5.3 Univers De Discours [29]:	20
5.4 Les opérateurs de la logique floue :	20
5.4.1 Complément (négation ou opérateur Non)	20
5.4.3 Intersection (opérateur ET)	20
5.5 Fonction d'appartenance:	21

SOMMAIRE

5.5.1 Type des fonctions d'appartenances:	21
5.5.2 Caractéristiques d'une fonction d'appartenance:	22
6- Principe d'un contrôleur flou:.....	22
6.1 La Fuzzification:	23
6.2 Base de règles floues :.....	23
6.3 L'inférence floue:.....	24
6.4 La Défuzzification :	26
6.4.1 Méthode de centre de gravité (COG):.....	26
6.4.2 Méthode de centre de maximum (COM):.....	27
6.4.3 Méthode de moyenne de maximum (MOM):	27
7- Pourquoi utiliser la logique floue?.....	27
8- Conclusion	28
Chapitre III : Approche proposée, Implémentation, Résultats et Discussion	
1-Introduction	30
2-Approche proposée	30
3-Processus de raisonnement Flou.....	30
4-Processus de renforcement flou :.....	31
4-1-Comportement de l'agent Monitor :.....	32
4-2-Comportement de l'agent flou Enforcer :	32
4-3-Communication entre agents :.....	33
5-Environnements de développement :.....	34
5-1-Madkit :	35
5-2-jFuzzyLogic :.....	35
6-Etudes de cas :	36
6-2-Résultats et analyse :	42
7-Conclusion:.....	46
Conclusion générale.....	48
Références.....	49

Liste des figures

LISTE DES FIGURES

Liste des Figures

Figure 1: Le modele AGR	10
Figure 2:Représentation d'un système flou	23
Figure 3:La Méthode D'inférence Takagi-Sugeno	25
Figure 4:La Méthode D'inférence Mamdani	26
Figure 5:Architecture de FE4NorMAS.....	31
Figure 6:Diagramme d'état transition modélisant le comportement de l'agent Monitor.	32
Figure 7:Diagramme d'état transition modélisant le comportement de l'agent flou Enforcer.	33
Figure 8: Diagramme de séquence modélisant la communication entres les agents dans le cas d'une punition.	34
Figure 9 : 1ère étude de cas.....	37
Figure 10: Base de règles floues pour le 1ere étude de cas.....	39
Figure 11 : Résultat de la décision floue prise par l'enforcer.	39
Figure 12: Résultat de la défuzzification généré par le plugin jFuzzyLogic.	39
Figure 13 : 2ème étude de cas.....	40
Figure 14: Résultat de la défuzzification généré par le plugin jFuzzyLogic pour les voitures : jaune et rouge.	43
Figure 15 : Résultat de la défuzzification généré par le plugin jFuzzyLogic pour la voiture bleue.	43
Figure 16: Résultat de la décision floue prise par l'enforcer.	43
Figure 17: Résultat de la décision floue prise par l'enforcer.	45
Figure 18 : Résultat de la défuzzification généré par le plugin jFuzzyLogic pour la voiture bleue.	45
Figure 19: Résultat de la défuzzification généré par le plugin jFuzzyLogic pour la voiture rouge.	46
Figure 20: Résultat de la défuzzification généré par le plugin jFuzzyLogic pour la voiture jaune.	46



Liste des tableaux

LISTE DES TABLEAUX

Liste des tableaux

Tableau 1: Les Fonctions d'appartenance.....	21
Tableau 2: L'outillage utilisé pour implémenter notre proposition et l'étude de cas associée.....	34
Tableau 3:La configuration du premier scénario	42
Tableau 4:La configuration du deuxième scénario	44



Liste des abréviations

Liste des abréviations

Liste des abréviations

IA : Intelligence Artificielle.

IAD : Intelligence Artificielle Distribuée.

SMA : System Multi Agent.

SMAN : System Multi Agent Normatif.

COG : Centre De Gravité.

COM : Center De Maximum.

MOM : Moyenne De Maximum.

FE4NorMAS : **F**uzzy **E**nforcement For **NOR**mative **M**ulti **A**gent **S**ystem

UML : Unified Modeling Language

GUI : Graphical User Interface.

MADKIT : Multi Agent Development Kit.

FCL : Fuzzy Control Language.

Résumé-Abstract-ملخص

Résumé

Dans le contexte des Systèmes Multi-Agent Normatifs (SMAN), les normes ont été conçues de manière à contrôler le comportement des agents. Ces derniers peuvent ne pas se comporter conformément à la norme s'ils ont maintenu leur autonomie. A cet effet, le renforcement des normes consiste en un mécanisme permettant à motiver les agents pour respecter les normes par le biais des sanctions (récompenses ou punitions). En revanche, le raisonnement flou montre son efficacité dans la gestion de l'incertitude et/ou l'incomplétude des données en rapprochant le raisonnement au cerveau humain. Ainsi, le renforcement flou des normes exprime le fait de sanctionner l'agent suite aux résultats fournis par un processus de raisonnement flou. Dans le cadre de notre projet de fin d'études, nous proposons une nouvelle approche de renforcement de normes basée sur le raisonnement flou. L'approche proposée a été déployée sous le nom : FE4NorMAS pour Fuzzy Enfoncement FOR Normative Multi-Agent Systems et est supporté par une étude de cas comportant différents scénarios sous la plateforme MaDKit.

Mots-clés : Systèmes Multi-Agent Normatifs - Renforcement des normes, logique floue.

Abstract

In the context of Normative Multi-Agent Systems (NMAS), norms have been designed to control the behavior of agents. However, agents may not conform to the norms if they maintain their autonomy. Therefore, norm enforcement is a mechanism that aims to motivate agents to adhere to norms through sanctions (rewards or punishments). On the other hand, fuzzy reasoning has shown its effectiveness in managing uncertainty and/or data incompleteness by approximating human-like reasoning. Thus, fuzzy norm enforcement involves sanctioning agents based on the results provided by a fuzzy reasoning process. As part of our final project, we propose a new approach to norm enforcement based on fuzzy reasoning. This approach, called FE4NorMAS (Fuzzy Enfoncement FOR Normative Multi-Agent Systems), is supported by a case study comprising different scenarios implemented on the MaDKit platform.

Keywords: Normative Multi-Agent System, Norm enforcement, Fuzzy Logic.

الملخص:

في سياق الأنظمة متعددة الوكلاء (SMAN)، تم تصميم القوانين بهدف مراقبة سلوكهم ففد لا يلتزم الوكلاء بالقوانين إذا حافظوا على استقلاليتهم. وبالتالي، يشكل تعزيز القوانين آلية تحفيز الوكلاء للامتثال للقوانين من خلال فرض عقوبات (مكافآت أو عقوبات). من ناحية أخرى، يظهر التفكير المبهم (الضبابي) فعاليته في إدارة عدم اليقين و/أو نقص البيانات عن طريق تقريب التفكير إلى العقل البشري. وبالتالي، يعبر تعزيز القواعد المبهمة عن عقاب الوكيل بناءً على النتائج المقدمة من عملية التفكير المبهم.

في إطار مشروعنا للتخرج، نقتح نهجًا جديدًا لتعزيز القواعد بناءً على التفكير المبهم يطلق عليه اسم (Fuzzy Enforcement FOR Normative Multi-Agent System) وهو اختصار

FE4NorMAS ويتم دعمه بدراسة حالة تشمل سيناريوهات متنوعة على منصة **Madkit**

الكلمات المفتاحية: الأنظمة متعددة الوكلاء المعيارية، تعزيز القوانين، المنطق الضبابي



Introduction Générale

Introduction Générale

Contexte générale de la problématique :

Les systèmes multi-agents (SMA) sont un domaine clé de l'intelligence artificielle (IA) qui étudie la manière dont des agents autonomes interagissent pour atteindre des objectifs communs. L'intégration des SMA dans l'IA permet de développer des systèmes plus complexes et plus adaptatifs, capables de prendre des décisions intelligentes et de résoudre des problèmes de manière collaborative.

L'intelligence des agents qui composent les systèmes multi-agents et rendre leurs comportements comparables aux comportements des êtres humains, les chercheurs pensent à l'intégration des normes dans ce type des systèmes. En fait, les normes jouent un rôle très important dans la société humaine et impose une influence importante sur leurs comportements. Ce type des systèmes s'appelle les Système Multi-Agents Normatifs (NMAS, Normatif Multi-Agents System).

Les systèmes multi-agents normatifs (NMAS) sont des systèmes composés d'agents autonomes interagissant selon des règles et des normes préétablies. Ces systèmes jouent un rôle crucial dans de nombreux domaines, tels que la gestion du trafic, la logistique, les systèmes de recommandation et les réseaux sociaux. Cependant, la coordination et l'application efficace des normes dans ces systèmes posent souvent des défis complexes en raison de l'incertitude, de l'ambiguïté et de la subjectivité qui peuvent entourer les normes.

La logique floue joue un rôle important dans la modélisation et la gestion des normes en offrant une approche adaptée pour représenter et raisonner sur les aspects incertains et vagues liés aux normes. Son utilisation permet de prendre en compte les nuances, les variations et les conflits normatifs, tout en intégrant des facteurs subjectifs dans l'évaluation des normes. Elle contribue ainsi à une meilleure prise de décision et à une gestion plus efficace des normes dans divers domaines d'application.

Contribution

Dans ce mémoire nous proposons une approche, que nous appelons "Fuzzy Enfoncement for Normative Multi-Agent Systems" (FE4NorMAS), qui vise à renforcer les normes en fournissant un cadre flexible et adaptatif.

Introduction générale

Le processus de renforcement des normes adopté dans notre proposition consiste à profiter des avantages de la logique floue pour procéder à la sanction des agents. Pour ce faire, nous avons implémentés deux types d'agents formant un SMA qui se comportent comme une tierce partie avec le SMA soumis aux normes. Un agent enforceur doté d'un mécanisme de raisonnement flou. Ainsi, l'agent enforceur est un agent qualifié de flou dont son architecture est similaire à celle d'un système flou (module de fuzzification, moteur d'inférence flou et module de défuzzification). L'agent moniteur, en revanche, est un agent réactif qui observe en permanence le comportement des agents et le compare avec celui décrit dans les normes.

Une fois qu'un comportement non normatif a été détecté par l'agent moniteur, ce dernier informe l'enforceur par la configuration observée (état de l'agent fautif, la spécification de la norme dans cette situation et la sanction correspondante). Comme l'agent enforceur a une autorité exécutive, il procède à sanctionner l'agent fautif selon le résultat produit par le raisonnement flou. Ainsi, la sanction a été conçue de manière flexible de sorte à aller progressivement de la punition ou de la récompense du plus faible au plus haut niveau. Les sanctions classiques, en outre, consistent à appliquer ou non la sanction comme si nous procédions avec la logique classique dans laquelle deux états possibles peuvent se figurer (sanctionner ou non). Les normes dans notre proposition ont été spécifiées sous forme de règles floues.

Pour implémenter notre approche, nous avons choisi la plateforme MaDKit pour le développement des SMA. Le choix de MaDKit a été justifié par le fait qu'elle est conçue en se basant sur le modèle AGR (Agent-Group-Role). Ce dernier offre un moyen efficace pour modéliser l'ouverture dans un SMA. En ce qui concerne la logique floue, nous avons utilisé le fameux plugin jFuzzyLogic qui représente une bibliothèque open source écrite en JAVA permettant le développement des systèmes flous.

Organisation du mémoire :

Ce mémoire est organisé en trois chapitres. Tout d'abord, nous présentons les systèmes multi-agents normatifs dans le premier chapitre. Bien entendu, la présentation des systèmes multi-agents normatifs passe par la présentation des concepts généraux de systèmes multi-agents et les systèmes normatifs. Ensuite, nous avons consacré le deuxième chapitre à la présentation de la logique floue et les concepts clés à savoir : les ensembles flous, les fonctions d'appartenance, les opérateurs flous, l'architecture d'un système flou, etc. Finalement nous allons expliquer notre proposition de renforcement des normes suivie d'une implémentation d'une étude de cas avec plusieurs scénarios possibles dont l'objectif est de montrer l'applicabilité de notre approche.

Introduction générale

Enfin nous concluons notre mémoire par une conclusion générale, suivie par des perspectives pouvant être apportés pour compléter la réalisation.



Chapitre I : Systèmes Multi-Agent Normatifs

1. Introduction

Les systèmes multi-agents constituent une branche de l'intelligence artificielle distribuée dans laquelle un comportement global intelligent émerge à partir des comportements d'entités plus simples appelées agents. Dans ces systèmes, les comportements individuels des agents ainsi que leurs interactions sont spécifiés. Parfois, ces comportements sont régis par des lois sociales ou des règles imposées par l'environnement.

On qualifie de systèmes multi-agents normatifs ceux dans lesquels les comportements des agents sont régis par des lois.

Dans ce chapitre, nous nous concentrons sur la présentation des systèmes multi-agents normatifs, en accordant une attention particulière à la problématique de leur mise en œuvre.

2. LES SYSTEMES MULTI-AGENTS

A la fin des années soixante-dix, les rétablissements des systèmes avancés comme l'raison Artificielle (IA) ont conduit à l'émergence de l'appréciation artificielle assignée (IAD). Le rudiment de base de l'IAD consiste à distribuer un système lumineux de conseil d'un aléa sur différents essences.

Ainsi, trois branches ont été apparues à cause de ce champ : la résolution

Distribuée des problèmes, les systèmes multi-agents et l'intelligence artificielle parallèle.

Cependant, les systèmes multi-agents représentent la branche la plus prometteuse [1].

La définition d'un système multi-agents peut être énoncée comme un ensemble d'agents.

Toutefois, cette définition simpliste ne capture pas toutes les caractéristiques essentielles d'un tel système.

En réalité, un système multi-agents ne se résume pas à un simple regroupement d'agents travaillant de manière indépendante.

Des notions telles que l'interaction et l'organisation jouent un rôle crucial dans les systèmes multi-agents [2].

Ferber [3] a défini un système multi-agents « un système composé des éléments suivants :

Un système multi-agent est « un ensemble composé de :

- Un environnement **E** : un espace disposant généralement d'une métrique.

- Un ensemble d'objets **O** situés : pour tout objet, il est possible, à un moment donné, d'associer une position dans **E**. Ces objets sont passifs, Ils peuvent être perçus, créés, détruits et modifiés par les agents.
- Un ensemble d'agents **A**, qui sont des objets particuliers $A \subseteq O$, lesquels représentent les entités actives du système.
- Un ensemble de relations **R** qui unissent des objets (et donc des agents) entre eux.
- Un ensemble d'opérations Op permettant aux agents de **A** de percevoir, produire, consommer, transformer et manipuler des objets de **O**.
- Des opérateurs chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification. ».

L'approche VOYELLE, proposée par Demazeau dans son article [4], Est une méthode de décomposition des systèmes multi-agents en quatre dimensions :

L'environnement (**E**), l'agent (**A**), l'organisation (**O**) et l'interaction (**I**).

Cette décomposition permet de présenter les différents composants d'un système multiagents de manière simple et dans un cadre unifié, Comme décrit dans l'article de référence [1].

2.1 Les agents :

2.1.1 Définition d'un agent :

Plusieurs définitions existent, nous citons entre autres les suivantes :

Pour Wooldrige [5] : « Un agent est un système informatique qui est situé dans un certain Environnement, et qui est capable d'agir de manière autonome dans cet environnement afin de répondre à ses objectifs de conception.

Pour Ferber [1] : un agent est une entité physique ou virtuelle :

- a. qui est capable d'agir dans un environnement.
- b. qui peut communiquer directement avec d'autres agents.
- c. qui est mue par un ensemble de tendances (sous la forme d'objectifs individuels ou d'une fonction de satisfaction, voire de survie, qu'elle cherche à optimiser).
- d. qui possède des ressources propres.

- e.* qui est capable de percevoir (mais de manière limitée) son environnement.
- f.* qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune).
- g.* qui possède des compétences et offre des services.
- h.* qui peut éventuellement se reproduire.
- i.* dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences.

Dont elle dispose, et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit.

2.1.2 Caractéristique d'un agent :

L'agent se caractérise par plusieurs propriétés [6], nous citons entre autres :

- **Autonomie** : Un agent est capable de prendre des décisions et d'agir de manière autonome, sans intervention directe d'un utilisateur ou d'un superviseur externe.
- **Comportement adaptatif** : Un agent est capable d'adapter son comportement en fonction des informations perçues, des objectifs qu'il cherche à atteindre et des changements dans son environnement. Il peut également apprendre et s'améliorer au fil du temps.
- **La réactivité** : est une propriété importante pour un agent. Cette caractéristique rend l'agent capable d'interagir avec l'environnement et avec les autres agents. C'est-à-dire perçoit les événements de son environnement et réagit par conséquence.
- **La proactivité** : Prendre l'initiative pour satisfaire les buts.
- **La sociabilité** : L'agent doit obligatoirement avoir des capacités de communication (un comportement social) pour vivre dans son environnement avec les autres agents.

2.1.3 Classification Des Agents :

Les agents peuvent être classés en trois types :

- **Les agents cognitifs** : Ce type d'agents est capable de déclencher des actions en utilisant ses capacités de raisonnement pour la délibération.

La délibération permet à l'agent d'atteindre son objectif en se basant sur ses croyances et ses plans [7].

- **Les agents réactifs** : Comme son nom l'indique, un agent réactif réagit en fonction des perceptions de son environnement.

Les agents réactifs n'ont pas de capacité de communication et sont généralement utilisés dans le domaine de la simulation [8].

- **Les agents hybrides** : L'utilité de chaque type d'agent est limitée dans la pratique.

En fait, les agents réactifs peuvent être utilisés dans certains types de problèmes et moins pour les autres. De la même façon, un agent doté seulement de capacité de délibération n'a pas une grande utilité pratique. En effet, un agent hybride est conçu pour rassembler à la fois les caractéristiques de l'agent cognitif et les caractéristiques de l'agent réactif [8].

2.2 Environnement :

L'environnement est considéré comme le lieu où les agents travaillent pour atteindre leurs objectifs.

Un environnement d'un système multi-agents se caractérise par [9] :

- *Structuration* : l'environnement est un espace dans lequel les agents opèrent. Cet espace peut avoir une structure spatiale ou organisationnelle.

- Assure la gestion des ressources et des services.

- Le maintien des processus écologiques : l'environnement peut allouer des ressources à des activités spécifiques.

- Activation des communications : l'environnement doit fournir des méthodes de communication entre les agents, et l'envoi de messages entre les agents est la méthode la plus couramment utilisée.

- Observabilité : l'environnement doit être observable.

2-3 Interaction :

Le concept d'interaction est essentiel dans un système multi-agent car il permet aux agents de générer des comportements collectifs complexes. La fonction d'interaction est basée sur un

ensemble de mécanismes qui lui permettent d'établir des liens avec son environnement et d'autres agents [10].

D'un autre point de vue, l'interaction est une relation dynamique établie entre deux ou plusieurs agents par le biais d'un ensemble d'actions réciproques. Lorsqu'un agent interagit avec un autre, cela peut engendrer à la fois des opportunités et des problèmes. L'interaction permet à un agent de partager des informations et des services afin d'atteindre ses objectifs et d'éviter les conflits. Lorsqu'une interaction est initiée, elle doit se dérouler et se terminer correctement. À cet effet, les interactions sont structurées selon des schémas typiques appelés protocoles. Les protocoles d'interaction permettent aux agents d'échanger des messages structurés, de contrôler cet échange de messages et de faciliter leur coordination. Un protocole d'interaction définit des règles qui doivent être respectées par les agents pendant la conversation en cours [11].

En général, on peut distinguer différentes situations d'interaction entre les agents : la communication, la coopération, la négociation et la coordination. La classification de ces situations est basée sur plusieurs critères [12]:

- La présence d'objectifs communs ou compatibles.
- L'accès à des ressources communes.
- La répartition des compétences entre les agents.

L'organisation :

L'organisation est le regroupement des composants en fonction de leur fonctionnalité dans un système multi-agents. Elle est représentée par un ensemble des rôles adoptés par les agents, ce qui spécifie les interactions et les relations entre ces agents. L'organisation représente le niveau macro de l'agent. On distingue deux types d'organisations : les organisations structurelles, qui sont prédéfinies avec des rôles et des relations statiques, et les organisations émergentes, qui sont construites automatiquement par les agents [13].

Il existe plusieurs modèles d'organisation, parmi lesquels le modèle AGR (Agent, Groupe, Rôle) qui représente l'organisation en utilisant les concepts d'agents, des rôles et des groupes. La figure 1 illustre ce Modèle [13].

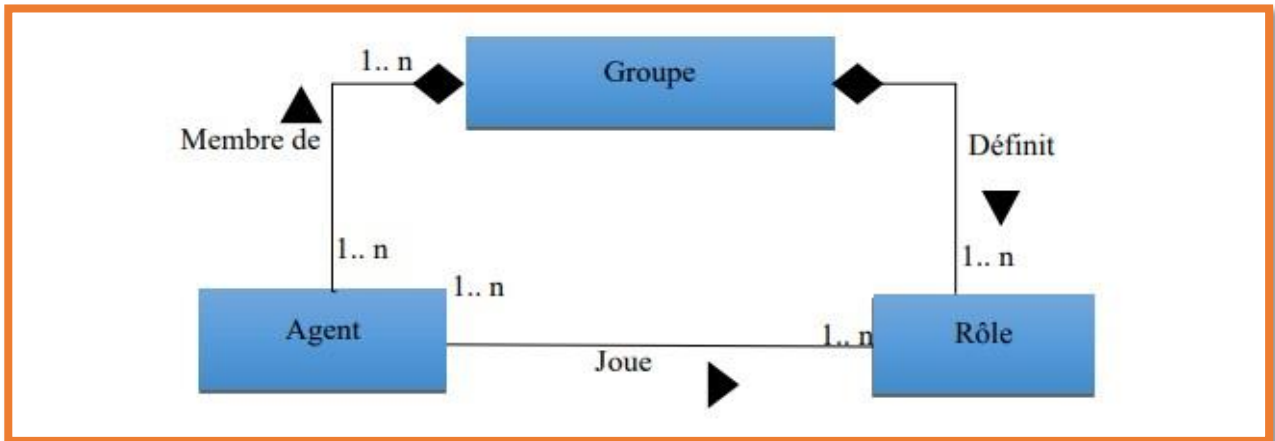


Figure 1: Le modelé AGR [14]

3- Les Systèmes Normatifs

Le concept de normes constitue la base des systèmes normatifs. Selon le dictionnaire en ligne Larousse, une norme peut être définie comme [15] :

« Une règle, principe ou critère se réfère tout jugement. ». On peut l'utiliser dans le domaine industriel pour décrire les règles qui fixent les conditions d'exécution d'un processus ou les caractéristiques d'un produit.

Les systèmes normatifs sont des systèmes où les normes jouent un rôle dans la spécification ou la description de leurs comportements [16].

Les normes sont un ensemble de règles et de standards qui sont compris par les membres d'une société et qui guident leurs comportements. En réalité, les normes contribuent à réduire les conflits et facilitent la coordination. Elles peuvent prendre la forme de traditions, de codes éthiques ou de lois [17].

Il existe deux grandes catégories de normes : les normes conventionnelles et les normes essentielles. Les normes conventionnelles sont des normes qui émergent naturellement sans qu'il y ait une application spécifique. En revanche, les normes essentielles sont celles qui facilitent la collaboration en résolvant les conflits entre les intérêts individuels et collectifs.

En distingue trois types de normes essentielles [17] :

- **Les normes constitutives** : Les normes constitutives sont établies dans le but de générer de nouvelles normes ou de nouveaux faits normatifs.

-**Les normes procédurales** : on distingue dans cette catégorie deux catégories de normes : les normes objectives et les normes subjectives.

Les normes de procédure objectives représentent les règles qui expriment la manière dont les décisions sont réellement prises dans un système normatif,

Dans les normes subjectives on spécifie les procédures permettant l'exécution d'une décision donnée.

-**Les normes régulatrices** : Les normes réglementaires sont conçues pour réguler les activités en imposant des obligations, des interdictions ou des autorisations :

Obligation : Dans ces normes, il est obligatoire d'adopter un comportement spécifique. Le non-respect de l'adoption de ce comportement est considéré comme une violation de la norme. Par conséquent, cette omission entraîne l'application d'une sanction.

L'interdiction : dans ce cas, l'adoption du comportement est considérée comme une violation de la norme. Bien entendu, cette violation entraîne l'application d'une sanction.

La permission : dans cette situation, le comportement peut être effectué ou ignoré sans aucune répercussion.

4- Les Systèmes Multi-Agents Normatifs

Un système multi-agents normatif est défini comme [18] : « a MAS governed

By restrictions on patterns of behaviour of the agents in the system that are actively or passively transmitted and have a social function and impact ». En d'autres termes, un système multi-agents normatif est un SMA soumis à des contraintes sur les modèles de comportement des agents du système, qui sont transmis de manière active ou passive et qui ont une fonction et un impact social.

Jessica.S, ET Jean.O [19] définissent les systèmes multi-agents normatifs comme suit: « *a multi-agent system that explicitly represents norms in order to govern the behaviour of its software agents and maintain social order. Such systems have norm enforcement mechanisms and are able to represent, interpret, police, manage and update norms* ».

Cette définition met en évidence les défis des systèmes multi-agents normatifs, tels que la représentation des normes, le mécanisme d'application, la gestion et la mise à jour des normes. Dans cette section, nous nous concentrons sur la présentation de ces défis.

4.1. La représentation des normes :

La représentation des normes constitue l'un des défis majeurs des systèmes multi-agents normatifs [19].

On peut classer les approches de présentation des normes en quatre approches [17] :

- La logique déontique : la logique déontique est une extension de la logique classique qui s'occupe des notions d'obligations, d'interdictions et de permissions.

- La théorie des jeux : Dans les systèmes multi-agents normatifs basés sur la théorie des jeux, les agents disposent d'un ensemble d'actions qu'ils peuvent appliquer. Chaque agent choisit l'action à exécuter en cherchant à maximiser sa fonction d'utilité.

- Les systèmes à base de règles : Ces systèmes sont constitués d'un ensemble de règles gérées par un moteur d'inférence. Une règle est une combinaison de conditions et d'actions. En conséquence, les normes sont implicitement représentées dans le système de prise de décision de l'agent.

- Les chaînes binaires : Une chaîne binaire est une séquence de chiffres binaires. Le chiffre binaire « 0 » représente l'absence de la norme, tandis que le chiffre « 1 » représente l'occurrence d'une norme. Cette structure est généralement utilisée pour la transmission et l'émergence des normes.

4.2. Mécanisme de renforcement :

Pour garantir le respect des normes spécifiées, il est essentiel que les agents soient motivés à s'y conformer. Les mécanismes les plus efficaces pour assurer l'application des normes sont généralement l'application de sanctions et de récompenses.

Ces mécanismes sont utilisés pendant et après le processus de diffusion des normes. De plus, ils peuvent être dirigés aussi bien vers l'extérieur que vers l'intérieur du système [17].

Dans le cas où un agent viole les normes, il peut s'infliger lui-même des sanctions. Les émotions sont considérées comme un facteur essentiel de motivation pour l'exécution interne, mais

dans la pratique, la sanction pour non-conformité est généralement intégrée en tant que coût de l'exécution d'une action en présence d'une norme [21].

L'exécution externe ou l'application par un tiers se produit lorsqu'un observateur (moniteur) constate qu'un agent viole une norme. En conséquent, il informera l'agent d'exécution (Enforcer) pour appliquer une sanction correspondant à cette violation. Toutefois, le renforcement par un tiers est mise en place dans le but de réduire les comportements indésirables dans la société [22].

4.2.1. Enforcer :

L'Enforcer est un agent spécifique qui peut être soit un agent du système possédant d'autres comportements, soit un agent dédié exclusivement à l'application des normes. Un agent Enforcer doit contenir deux méthodes spécifiques. La première méthode " **Punish(j)** ", est utilisée pour sanctionner les agents. La deuxième méthode " **Reward(j)** ", est utilisée pour récompenser les agents [20].

Cette stratégie soulève un problème important qui concerne la possibilité de violation de la norme par l'enforcer. En d'autres termes, il est possible que l'enforcer évite de sanctionner un agent qui viole une norme, ou inversement. Pour résoudre ce problème, on peut ajouter des normes qui régularisent le comportement de l'enforcer en appliquant l'une des deux stratégies : régimentation ou l'enforcement. Ainsi, l'enforcer est contraint de respecter les normes [20].

4.3. La gestion et la mise à jour des normes :

Il est essentiel de comprendre que les normes ne sont pas des règles immuables. Lors de l'exécution d'un système, il est nécessaire de mettre à jour les normes et les règles qui le régissent. Cette mise à jour peut être réalisée par la création, l'internalisation, la modification et la suppression des normes. En ce qui concerne la création des normes, celles-ci peuvent être établies dans le monde réel de différentes manières : par l'émergence naturelle de l'interaction sociale, par un décret émis par une autorité compétente, ou par la négociation entre les membres d'un groupe [21].

La suppression des normes indésirables est une phase cruciale dans la gestion des normes. Elle consiste à éliminer une norme obsolète et à la remplacer par une nouvelle norme. Le processus de suppression revêt une importance théorique lorsque le système est dynamique. De plus, il devient plus complexe lorsque les ressources du système sont limitées [17].

Chapitre I : Systèmes Multi-Agent Normatifs

L'acceptation et la modification des normes sont des étapes préalables à leur intégration. Avant qu'une norme puisse être intégrée, elle doit tout d'abord être acceptée par un agent. L'acceptation d'une norme est un processus de résolution de conflits dans lequel des contraintes sociales externes se confrontent aux désirs et motivations internes de l'agent. Si la nouvelle norme entre en conflit avec les normes existantes et peut conduire à des comportements incohérents, il peut être nécessaire d'apporter des modifications pour assurer une cohérence globale [21].

5. Conclusion

Nous avons présenté les systèmes multi-agents et les systèmes normatifs, avant présenter les systèmes multi-agents normatifs. Les systèmes multi-agents normatifs sont un nouveau type de systèmes multi-agents qui intègrent des normes dans leur fonctionnement. Ces normes sont utilisées pour orienter et contrôler les comportements des agents.



Chapitre II : La logique floue

1-Introduction

La logique floue suscite actuellement un intérêt général de la part des chercheurs, des ingénieurs et des industriels, et plus généralement de la part de tous ceux qui éprouvent le besoin de formaliser des méthodes empiriques, de généraliser des modes de raisonnement naturel, d'automatiser la prise de décision dans leur domaine et de construire des systèmes artificiels effectuant les tâches habituellement prises en charge par les humains. Dans ce chapitre, nous allons présenter les notions de base de la logique floue, telles que les ensembles flous, les opérations sur les ensembles flous, les relations entre ensembles flous, la structure d'un système flou et certains domaines d'application.

2- Historique de la logique floue

Historiquement, la logique floue a été marquée par plusieurs dates [23]:

- En 1965, Lotfi A. Zadeh, de l'Université de Californie à Berkeley, publie «Fuzzy Sets», qui expose les mathématiques de la théorie des ensembles flous et, par extension, de la logique floue. Zadeh avait observé que la logique informatique conventionnelle ne pouvait pas manipuler des données qui représentaient des idées subjectives ou vagues, donc il a créé une logique floue pour permettre aux ordinateurs de déterminer les distinctions entre les données avec des nuances de gris, semblables au processus du raisonnement humain.

- Seiji Yasunobu et Soji Miyamoto d'Hitachi ont suscité l'intérêt pour les systèmes flous.

En 1985, ils ont fourni des simulations qui démontraient la supériorité des systèmes de contrôle flou pour le chemin de fer de Sendai. Leurs idées ont été adoptées, et les systèmes flous ont été utilisés pour contrôler l'accélération et le freinage lorsque la ligne a ouvert en 1987.

- Toujours en 1987, lors d'une réunion internationale de chercheurs flous à Tokyo, Takeshi Yamakawa a démontré l'utilisation du contrôle flou, à travers un ensemble de puces logiques floues dédiées, dans une expérience de "pendule inversé". C'est un problème de contrôle classique, dans lequel un véhicule essaie de maintenir un poteau monté sur son sommet par une charnière en se déplaçant d'avant en arrière.

- Les observateurs ont été impressionnés par cette démonstration, ainsi que par des expériences ultérieures de Yamakawa au cours desquelles il amonté un verre de vin contenant de l'eau ou même une souris vivante au sommet du pendule.

Chapitre II : La logique floue

- Le système a maintenu la stabilité dans les deux cas. Yamakawa a finalement organisé son propre laboratoire de recherche sur les systèmes flous pour l'aider à exploiter ses brevets sur le terrain.

3- Définition

Plusieurs définitions ont été proposées pour le terme "logique floue". Dans la suite, nous allons présenter quelques-unes de ces définitions:

Définition de zadeh : la logique floue s'intéresse aux principes formels du raisonnement approximatif, avec un raisonnement précis considéré comme limitant. Il vise à modéliser le mode de raisonnement imprécis qui joue un rôle essentiel dans la capacité humaine remarquable à prendre des décisions dans un environnement d'incertitudes et d'imprécisions [24].

Autre Définition : la logique floue est une méthodologie de résolution de problèmes qui fournit un moyen simple de tirer des conclusions définitives à partir d'informations vagues et imprécises [25].

4- Domaines d'application

La logique floue a un potentiel d'application important dans tous les domaines. Parmi les principaux domaines d'application de la logique floue, on peut citer [26].

- Contrôle simplifié des robots (Hirota, Fuji Electric, Toshiba, Omron)
- Caméra visant la télédiffusion d'événements sportifs (Omron)
- Contrôle efficace et stable des moteurs de voitures (Nissan)
- Régulateur de vitesse pour automobiles (Nissan, Subaru)
- Substitution d'un expert pour l'évaluation des activités boursières (Yamaichi, Hitachi)
- Planification optimisée des horaires d'autobus (Toshiba, NipponSystem, Keihan-Express)
- Système d'archivage pour documents (Mitsubishi Elec.)
- Système de prévision pour la reconnaissance précoce des tremblements de terre (Bureau de sismologie de métrologie, Japon)
- Technologie de la médecine : diagnostic du cancer (École de médecine de Kawasaki)
- Systèmes de décision, diagnostic, reconnaissance.

- Systèmes de contrôle/commande dans la plupart des domaines industriels de production, transformation, traitement de produit et déchet.

5- Concept et définition

5.1 Définition d'un ensemble flou:

L'ensemble flou est un outil puissant qui nous permet de représenter des objets ou des membres d'une manière vague ou ambiguë. L'ensemble flou fournit également une manière qui est similaire aux concepts et au processus de pensée d'un être humain. Cependant, seul l'ensemble flou lui-même ne peut mener à aucun produit utile et pratique tant que le processus d'inférence floue n'est pas appliqué [27].

Dans la théorie classique des ensembles, un sous-ensemble **A** de **B** est défini par une fonction d'appartenance $\mu_A(X)$ qui caractérise tout élément **X** appartenant à **B**. Cette fonction prend la valeur **1** si **X** appartient à **A** et la valeur **0** dans le cas contraire [28]:

$$\mu_A(X) = (1 \text{ si } x \in A) \text{ et } (0 \text{ si } x \notin A)$$

En logique floue, un sous-ensemble flou **A** de **B** est défini par une fonction d'appartenance $\mu_A(X)$

Pouvant prendre différentes valeurs comprises entre **0** et **1**, selon le degré d'appartenance de l'élément au sous-ensemble **A** [28].

$$\mu_A(X) \in [0 \ 1]$$

5.2 Variable et valeur linguistiques:

La variable linguistique désigne une grandeur ou un phénomène qui est décrit en utilisant des termes linguistiques tels que "Quelque", "Beaucoup", "Souvent", "Négative", "Environ Zéro" ou encore "Positive". Cependant, il est important de noter que ces termes ne représentent pas les valeurs linguistiques en elles-mêmes, mais plutôt des catégories linguistiques permettant de quantifier et de décrire la variable linguistique.

Les valeurs linguistiques, quant à elles, correspondent aux différentes valeurs que peut prendre la variable linguistique dans un contexte donné. Par exemple, pour la variable linguistique "taille" d'un objet physique, les valeurs linguistiques pourraient être "Petit", "Moyen" ou "Grand".

5.3 Univers de discours :

L'univers de discours représente l'ensemble de référence ou le domaine de variation de la variable linguistique, ou domaine de fonctionnement du processus dans le cas de réglage [29].

5.4 Les opérateurs de la logique floue :

Il s'agit de la généralisation des opérateurs négation, intersection et union de la théorie classique des ensembles.

Soient deux ensembles flous A et B définis sur l'ensemble de référence U ayant comme fonctions d'appartenance associées μ_A et μ_B .

5.4.1 Complément (négation ou opérateur Non)

Le complément \bar{B} de l'ensemble flou B est défini par la fonction d'appartenance :

$$\forall x \in X : \mu_{\bar{B}}(x) = 1 - \mu_B(x)$$

5.4.2 Union (opérateur OU)

L'intersection de deux ensembles flous A et B est un ensemble flou $(A \cup B)$ de fonction d'appartenance :

$$\forall x \in X : \mu_{A \cup B}(x) = \mu_A(x) + \mu_B(x)$$

Avec:

$$\mu_A(x) + \mu_B(x) = \text{MAX}(\mu_A(x), \mu_B(x))$$

5.4.3 Intersection (opérateur ET)

L'intersection de deux ensembles flous A et B est un ensemble flou $(A \cap B)$ de fonction d'appartenance :

$$\forall x \in X : \mu_{A \cap B}(x) = \mu_A(x) * \mu_B(x)$$

Le symbole « * » représente la norme triangulaire, qui caractérise le plus petit degré d'appartenance à la fois à A et à B . Les normes les plus utilisées dans le domaine de la commande sont les suivantes :

$$\mu_A(x) * \mu_B(x) = \text{min}(\mu_A(x), \mu_B(x))$$

$$\mu_A(x) * \mu_B(x) = (\mu_A(x) \times \mu_B(x))$$

5.5 Fonction d'appartenance:

La fonction d'appartenance est l'ensemble de degrés d'appartenance à chaque valeur linguistique est matérialisé par un nombre compris entre 0 et 1 et notée $\mu_A(x)$.

5.5.1 Type des fonctions d'appartenances:

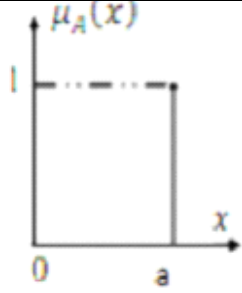
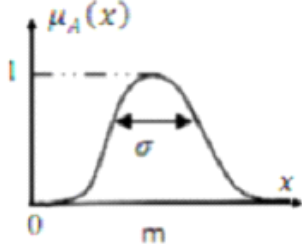
Il existe plusieurs formes de la fonction d'appartenance, notamment [30] :

- o Fonction gaussienne.
- o Fonction carrée ou trapézoïdale.
- o Fonction triangulaire.
- o Fonction singleton.

Les fonctions d'appartenance les plus utilisées sont représentées dans le tableau suivant :

Tableau 1: Les Fonctions D'appartenance

Fonction	Forme Algébrique	Forme Graphique
<i>Fonction trapézoïdale</i>	$\mu_{\bar{A}}(x) = 0 \text{ si } x < a$ $\frac{x-a}{b-a} \text{ si } a \leq x < b$ $= 1 \text{ si } b \leq x < c$ $= \frac{d-x}{d-c} \text{ si } c \leq x < d$ $= 0 \text{ si } x > d$	
<i>Fonction triangulaire</i>	$\mu_{\bar{A}}(x) = 0 \text{ si } x < a$ $= \frac{x-a}{b-a} \text{ si } a \leq x < b$ $= \frac{c-x}{c-b} \text{ si } b \leq x \leq c$ $= 0 \text{ si } x > c$	
	$\mu_{\bar{A}}(x) = 1 \text{ si } x = a$ $= 0 \text{ si } x \neq a$	

<p><i>Fonction singleton</i></p>		
<p><i>Fonction gaussienne</i></p>	$\mu_{\bar{A}}(x) = \exp\left(\frac{-1}{2} \left(\frac{x - m}{\sigma}\right)^2\right)$	

5.5.2 Caractéristiques d'une fonction d'appartenance:

Les caractéristiques suivantes sont propres aux fonctions d'appartenance :

- **Forme** : la forme du nombre qui peut être triangulaire, trapézoïdale, gaussienne ou sigmoïdale
- **Domaine** : une fonction d'appartenance est définie sur un domaine qui est généralement un ensemble d'éléments.
- **Hauteur** : La hauteur du sous-ensemble flou **A** de **X**, notée **h(A)**, est le plus fort degré avec lequel un élément de **X** appartient à **A**.
- **Cardinalité** : Cardinalité d'un sous-ensemble flou **A** de **X**, noté **A**, est le nombre d'éléments appartenant à **A** pondéré par leur degré d'appartenance.

Ces caractéristiques sont importantes pour comprendre le fonctionnement des fonctions d'appartenance et leur utilisation en théorie des ensembles flous.

6- Principe d'un contrôleur flou:

Il existe trois modules pour un régulateur à logique flou [31]:

1. La fuzzification des entrées et sorties.
2. L'inférence floue selon une base de règle.

3. La défuzzification des sorties.

L'architecture d'un contrôleur flou est donnée par le schéma suivant :

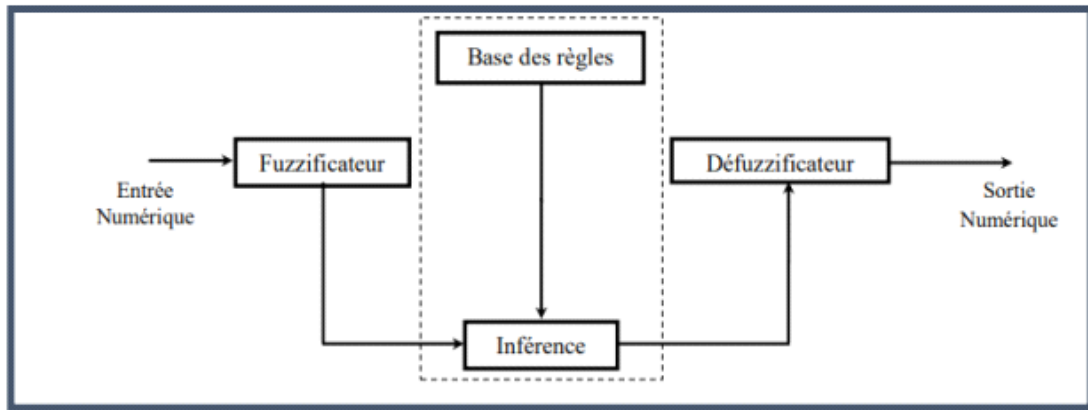


Figure 2: Représentation d'un système flou [31]

6.1 La Fuzzification:

La fuzzification est la première étape dans le FLC qui transforme les entrées numériques X_i en un ensemble de valeurs d'appartenance dans l'intervalle $[0,1]$ à des ensembles flous correspondants $\mu(x_i)$.

Les fonctions suivantes sont exécutées par le bloc de fuzzification [32] :

- Définition des fonctions d'appartenance de toutes les variables d'entrées.
- Transformation des grandeurs physiques (réelles ou numériques) à des grandeurs linguistiques ou floues.
- Représentation d'échelle transférant la plage des variables d'entrées aux univers de discours correspondants.

6.2 Base de règles floues :

La base de règles floues, également appelée base de connaissances, contient des règles décrivant le comportement du système. Elle est le cœur du système car tous les autres composants sont utilisés pour interpréter et combiner ces règles afin de former le système final. Les règles peuvent être fournies par un expert ou extraites de données numériques et prennent la forme « Si prémisse Alors conclusion ». [33]

Chapitre II : La logique floue

Pour assurer le bon fonctionnement d'un système flou, la base de règles doit respecter les conditions de complétude et de consistance. La complétude exige que pour chaque vecteur d'entrée, au moins une règle floue soit activée. Pour garantir cette propriété, les fonctions d'appartenance doivent couvrir tout l'espace des variables d'entrée.

D'autre part, l'inconsistance d'une base de règles se produit lorsqu'il existe deux règles floues ayant la même prémisse mais des conclusions différentes. La propriété de consistance permet d'éviter les contradictions dans une base de règles et de maintenir la fiabilité du système.

6.3 L'inférence floue:

Le mécanisme d'inférence, également connu sous le nom de moteur d'inférence, est l'élément clé de l'algorithme de logique floue. Il permet d'émuler la prise de décision de l'expert en interprétant et en appliquant les connaissances sur la meilleure façon de contrôler le système.

Le mécanisme d'inférence est chargé de combiner les règles floues de la base de connaissances pour déterminer les actions à entreprendre pour chaque entrée du système. Il utilise des méthodes telles que la méthode de Mamdani ou la méthode de Takagi-Sugeno pour effectuer cette tâche. [34]

❖ La méthode de Takagi-Sugeno:

Le mécanisme d'inférence de Takagi-Sugeno combine les règles floues en utilisant une fonction linéaire qui est définie pour chaque règle. Cette fonction linéaire est déterminée en utilisant les valeurs d'entrée du système pour chaque règle et permet de calculer la contribution de chaque règle à la sortie crisp. [28]

La méthode de Takagi-Sugeno est souvent utilisée dans les systèmes de contrôle pour son efficacité et sa capacité à fournir des réponses précises et rapides. Elle est également utile pour les systèmes qui ont une base de connaissances moins complexe que celle nécessitant l'utilisation de la méthode de Mamdani. [28]

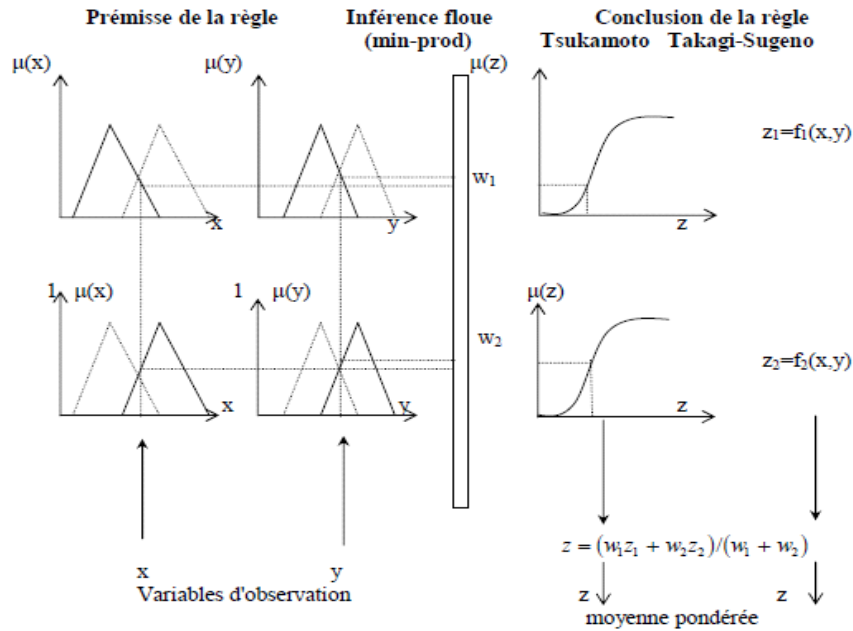


Figure 3: La Méthode D'inférence Takagi-Sugeno [28].

- La méthode de Mamdani:

Le mécanisme d'inférence de Mamdani fonctionne en évaluant chaque règle floue de la base de connaissances pour chaque entrée du système. La règle floue est activée si la valeur de la prémisse de la règle est vraie pour cette entrée. Une fonction d'appartenance est ensuite appliquée à la conclusion de la règle floue pour obtenir une valeur floue qui représente la contribution de cette règle à la réponse globale du système. Les valeurs floues des règles sont ensuite agrégées à l'aide d'une méthode de composition, comme la somme pondérée ou le maximum. [28]

Une fois que les valeurs floues des règles ont été agrégées, une méthode de déflation est souvent utilisée pour obtenir une réponse crisp à partir de la réponse floue. Cette méthode consiste à déterminer le centre de gravité de la réponse floue agrégée et à l'utiliser comme la réponse crisp du système. [28]

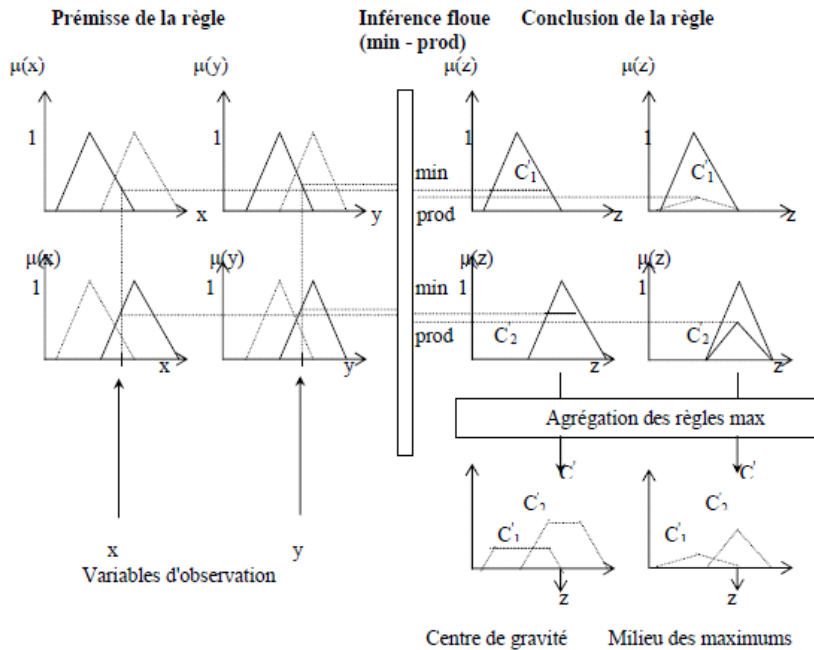


Figure 4: La Méthode d'inférence Mamdani [28].

6.4 La Défuzzification :

La défuzzification est la dernière étape d'un contrôleur flou. Elle consiste à convertir les valeurs floues en une sortie numérique, ce qui correspond à l'opération inverse de la fuzzification. Cette étape est essentielle pour obtenir une réponse concrète et précise à partir des valeurs floues [35].

Il existe plusieurs méthodes de défuzzification les plus utilisées sont :

- Méthode de centre de gravité (COG).
- Méthode de centre de maximum (COM).
- Méthode de moyenne de maximum (MOM).

6.4.1 Méthode de centre de gravité (COG):

La méthode du centre de gravité est l'une des méthodes les plus couramment utilisées pour la défuzzification, car elle offre une haute précision dans les résultats. Cette méthode calcule la totalité de l'espace actif sous les fonctions d'appartenance pour déterminer le centre de gravité de la courbe de la fonction d'appartenance agrégée [35].

Cependant, le calcul de plusieurs surfaces peut présenter des inconvénients en termes de complexité et de temps de calcul. Il est donc important de prendre en compte ces facteurs lors du

choix de la méthode de défuzzification et de considérer les performances du système en termes de vitesse de traitement et de ressources informatiques disponibles [35].

6.4.2 Méthode de centre de maximum (COM):

La méthode du maximum est connue pour sa simplicité, car elle ne prend en compte que le sommet de l'espace actif sous la fonction d'appartenance. Cette méthode est particulièrement adaptée aux fonctions d'appartenance de type échelon, où elle offre une précision similaire à celle de la méthode du centre de gravité [35].

6.4.3 Méthode de moyenne de maximum (MOM):

La méthode de la moyenne des maximums consiste à prendre la moyenne des maximums de chaque fonction d'appartenance pour déterminer la valeur de sortie. Cette méthode est particulièrement adaptée aux cas où la grandeur de sortie a des valeurs fixes et discontinues. Elle est souvent utilisée dans les systèmes de contrôle de processus industriels, où les valeurs de sortie sont limitées à un ensemble de valeurs prédéfinies [35].

7- Pourquoi utiliser la logique floue?

- Les systèmes flous étant non linéaires, ils sont plus adaptés pour la commande de processus non linéaires.
- Les systèmes flous ne nécessitent pas l'existence d'un modèle analytique précis du processus à contrôler et peuvent fonctionner avec peu d'informations pour mettre en place la boucle de commande.
- La logique floue permet d'inclure facilement des informations qualitatives dans la conception d'un contrôleur. [36]

8- Conclusion

Dans ce chapitre, nous avons présenté les concepts fondamentaux de la logique floue, son mode de raisonnement, ainsi que ses avantages et ses domaines d'application.



Chapitre III : Approche proposée, Implémentation, Résultats et Discussion

1-Introduction

Le présent chapitre consiste en une description technique détaillée de notre approche de renforcement flou des normes dans les SMA ouverts. L'approche proposée porte l'acronyme : **FE4NorMAS** pour **Fuzzy Enforcement for Normative Multi-Agent Systems**. On va tester également, dans ce chapitre, l'applicabilité de FE4NorMAS avec une étude de cas sur plusieurs scénarios possibles. L'étude de cas proposé concerne la modélisation du trafic routier où les voitures sont développés sous formes d'agent soumis au code de la routes (normes). La base de regles flous créées pour notre étude de cas comporte en somme 90 règles. Une analyse des résultats obtenus a été fournis à la fin de chaque processus de renforcement flou.

2-Approche proposée

L'idée centrale de notre proposition de renforcement flou de normes (figure 5) consiste à profiter les avantages du raisonnement flou en appliquant les sanctions sur les agents. Les sanctions sont réparties en deux catégories à savoir : les punitions et les récompenses. Un agent sera puni suite à un comportement non conforme à celui décrit dans la norme. Une punition traditionnelle va être mise en place par un agent particulier qui s'appelle *Enforcer* sur l'agent produisant un comportement non normatif. Autrement dit, l'Enforcer lance un processus de renforcement qui correspond à la situation observée et communiquée par l'agent *Monitor*. Dans le cadre de notre approche, l'Enforcer déclenche un processus de renforcement flou qui consiste à appliquer les sorties (outputs) produits par le système de raisonnement flou utilisé. Ainsi, le raisonnement flou, dans notre cas, montre la sanction (punition ou récompense) convenable à la situation observée. A l'opposée du renforcement classique où on obtient un seul type de sanction pour chaque situation de non-conformité à la norme, notre approche rendre le renforcement des normes plus flexible de sorte à dispatcher la sanction sur plusieurs niveaux (sanction plus faible, sanction faible, sanction moyenne, sanction sévère et sanction plus sévère). De même pour les récompenses, un agent qui adapte son comportement avec la norme sera récompensé avec un degré convenable (récompense faible, moyenne ou forte).

3-Processus de raisonnement Flou

Dans le contexte de FE4NorMAS, on a procédé premièrement par l'identification des variables linguistiques et les fonctions d'appartenance de chaque variable. Il est intéressant de noter que le choix des variables linguistiques dépend fortement du domaine étudié (*domain-dependent*). Ainsi, les normes sont représentées par des règles floues de la forme (IF ... THEN ...). La méthode de raisonnement flou sélectionnée est celle de **Mamdani** (le And et OR sont

Chapitre III : Approche proposée, Implémentation, Résultats et Discussion

réalisés par les deux fonctions MIN et MAX respectivement, voir chapitre 2). Ainsi, on a adopté la méthode de Center Of Gravity (COG) dans la phase de défuzzification grâce à la haute précision qu'elle offre.

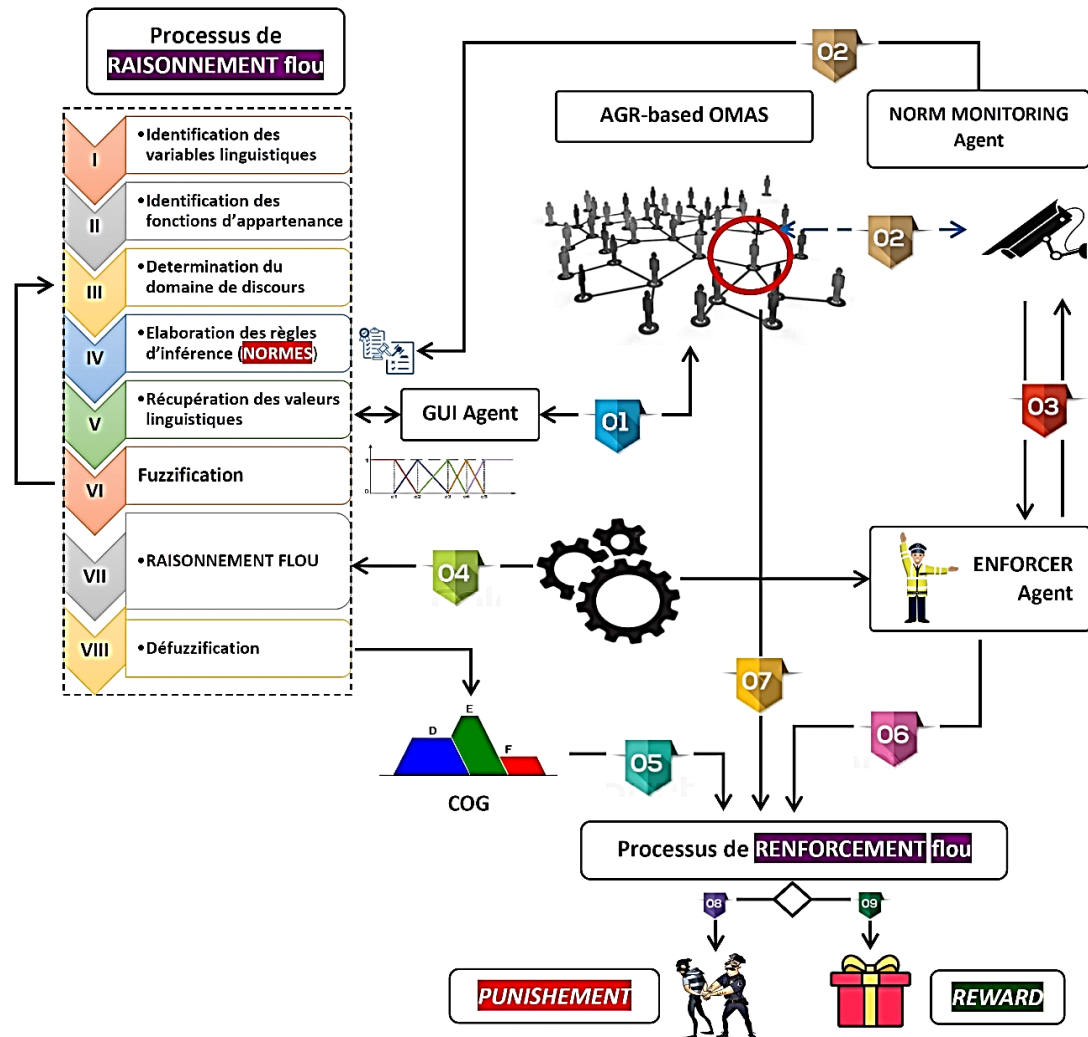


Figure 5: Architecture de FE4NorMAS.

4-Processus de renforcement flou :

Le processus de renforcement flou est le fruit de la combinaison de trois paramètres (étapes : 5, 6 et 7 dans la figure 5) : le résultat de la défuzzification, la décision floue prise et l'effet de la sanction sur l'agent en question. Dans le contexte de notre travail, un agent sanctionné sera blacklisté. Pour ce faire, on a maintenu une liste pour les agents punis et une autre liste pour les agents récompensés [39].

Dans ce qui suit on va présenter le comportement des agents Monitor et Enforcer sous forme de diagrammes d'état transition UML.

4-1-Comportement de l'agent Monitor :

L'agent moniteur se comporte comme étant un surveillant des comportements des agents systèmes par rapport à la base de norme. Dans le cas où un comportement non normatif figure, il procède pour capturer l'état actuelle du système (configuration courante) et informe en conséquence l'agent enforcer en lui passant la configuration capturée. La configuration actuelle du système représente une capture en temps réel des valeurs des paramètres qui contribuent à la bonne spécification de la sanction. Dans notre cas, la configuration actuelle a été implémentée sous forme d'une structure de donnée qui contient :

- AgentID,
- normID,
- normType : Obligation, Interdiction
- Variables linguistiques : speed, rainy, lighth,
- Valeurs des variables linguistiques et le
- Temps.

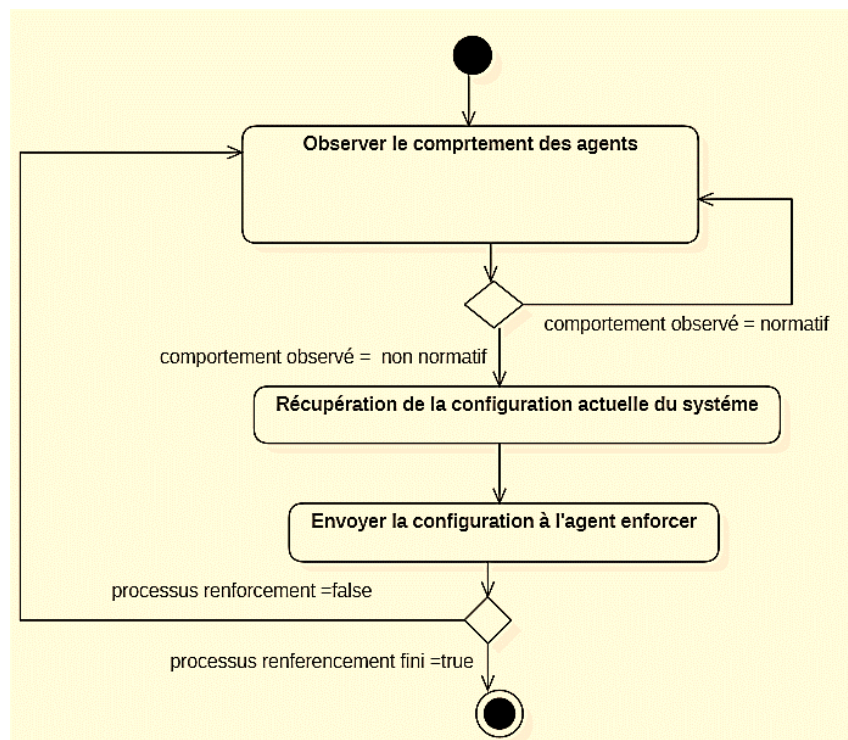


Figure 6:Diagramme d'état transition modélisant le comportement de l'agent Monitor.

4-2-Comportement de l'agent fluu Enforcer :

L'agent enforcer, après avoir reçu la configuration courante du système, déclenche le processus de raisonnement fluu en passant les valeurs réels des paramètres reçus à l'étape de fuzzification pour obtenir des valeurs floues. Ensuite, le moteur de raisonnement fluu va activer

Chapitre III : Approche proposée, Implémentation, Résultats et Discussion

les règles qui correspondront à la situation traitée en utilisant la méthode de Mamdani. A la fin du processus de raisonnement, on obtient, par le biais de la méthode de défuzzification des quantités réels représentant le degré de sanction applicable à l'agent. A cet effet, l'agent enforcer va prendre la décision appropriée suivant les valeurs obtenues et exécute le plan de sanction correspondant.

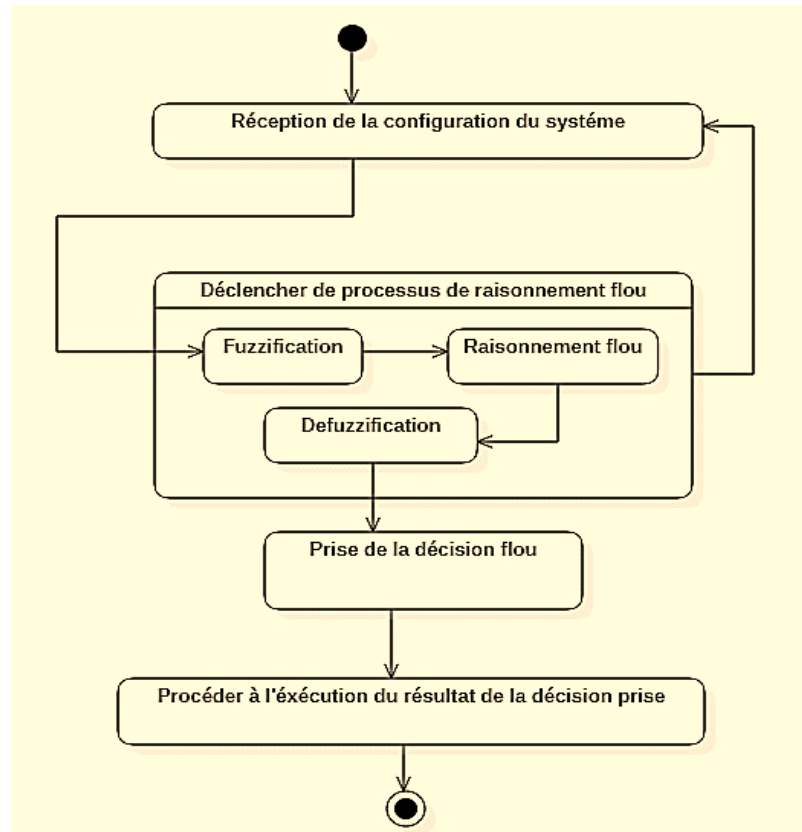


Figure 7:diagramme d'état transition modélisant le comportement de l'agent flou Enforcer.

4-3-Communication entre agents :

En ce qui concerne les étapes : 1, 2 et 3 mentionnés dans la figure de l'approche(figure5), on a procédé pour une modélisation de la communication effectuée entre les agents : GUI Agent, Monitor, Enforcer et éventuellement l'utilisateur du système soumis aux normes à l'aide du diagramme de séquence UML de la figure suivante :

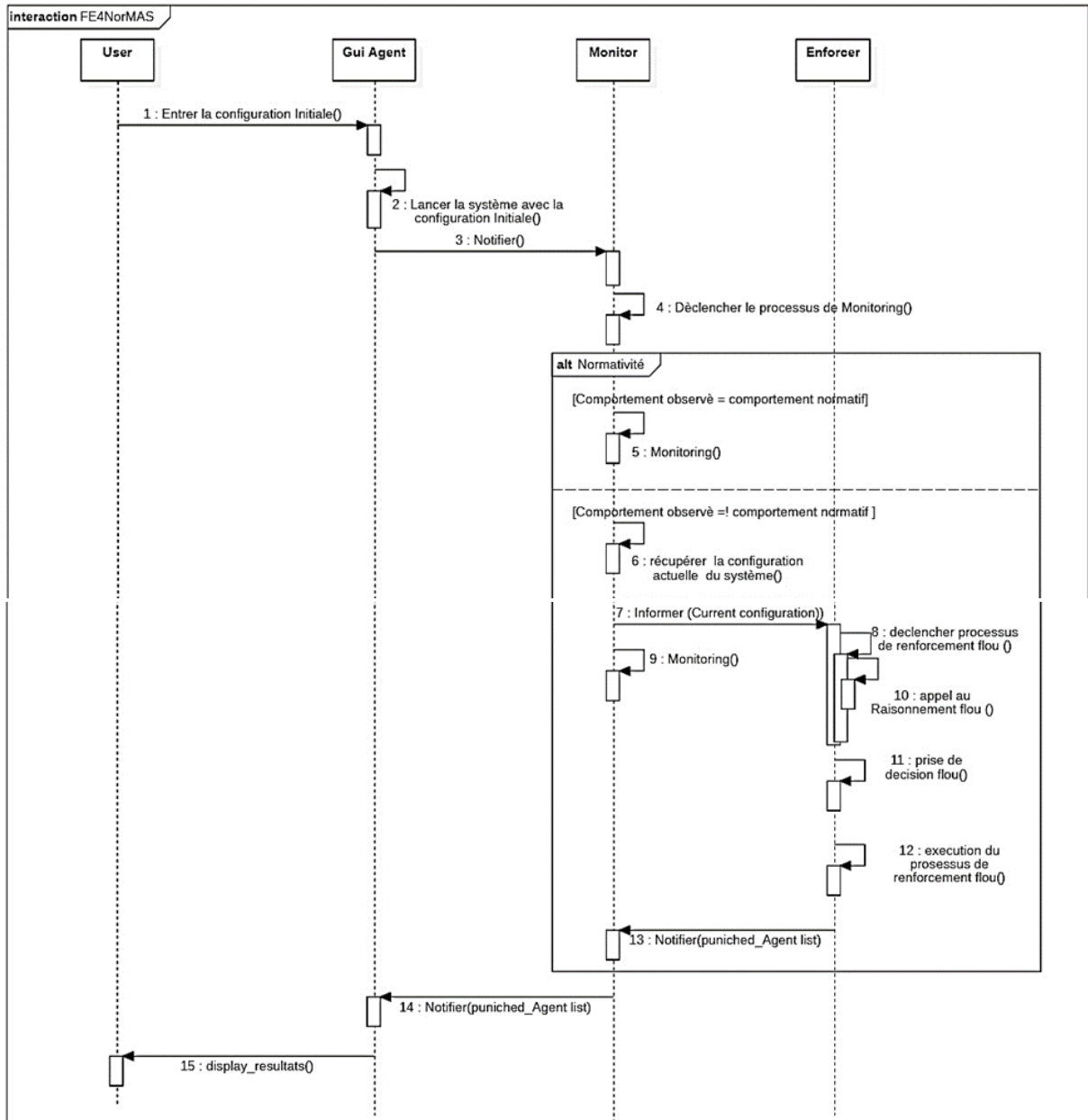


Figure 8: Diagramme de séquence modélisant la communication entre les agents dans le cas d'une punition.

5-Environnements de développement :

La table suivante montre l'outillage utilisé pour implémenter notre proposition et l'étude de cas associé :

Tableau 2: l'outillage utilisé pour implémenter notre proposition et l'étude de cas associé

Outil	nom	Version
Plateforme de développement des systèmes multi agent	MaDKit	5.3.2
Logique floue	jFuzzyLogic	LGPLv3

5-1-Madkit :

Madkit est une plateforme de développement d'applications multi-agents open-source. Elle fournit un environnement et une infrastructure pour la conception, la simulation et l'exécution de systèmes multi-agents. Cette plateforme est conçue pour faciliter le développement d'applications autonomes et distribuées basées sur le paradigme multi-agents. [37]

Voici quelques caractéristiques clés de Madkit :

- ❑ Architecture multi-agents : Madkit est basé sur une architecture multi-agents, où les entités autonomes appelées agents interagissent entre elles et avec leur environnement. Les agents sont capables de prendre des décisions indépendantes et de communiquer avec d'autres agents pour atteindre des objectifs communs.
- ❑ Plateforme open-source : Madkit est une plateforme open-source, ce qui signifie qu'elle est librement accessible et modifiable par la communauté des développeurs. Cela permet une collaboration ouverte, des contributions et des améliorations continues.
- ❑ API et bibliothèques : Madkit propose une API riche et des bibliothèques qui simplifient le développement d'applications multi-agents. Il fournit des fonctionnalités pour créer, déployer et gérer les agents, ainsi que des mécanismes de communication inter-agents, de coordination et de planification.
- ❑ Simulation et exécution : Madkit offre des capacités de simulation qui permettent de tester et d'analyser le comportement des agents dans des scénarios virtuels. Il permet également le déploiement des systèmes multi-agents en environnement réel, ce qui permet de mettre en œuvre des applications pratiques.
- ❑ Communauté active : Madkit bénéficie d'une communauté active de développeurs et de chercheurs qui contribuent à son développement et à son amélioration. Il y a des forums de discussion, des ressources en ligne et une assistance communautaire pour aider les utilisateurs de Madkit.

5-2-jFuzzyLogic :

JFuzzyLogic (ou simplement JFuzzy) est une bibliothèque open-source écrite en JAVA qui permet de concevoir, développer et exécuter des systèmes d'inférence floue. Elle fournit une interface conviviale pour créer des systèmes de logique floue en utilisant le langage FCL (Fuzzy Control Language). Le langage FCL est une représentation textuelle des systèmes de

Chapitre III : Approche proposée, Implémentation, Résultats et Discussion

logique floue. Il permet de définir les variables linguistiques, les ensembles flous, les règles et les méthodes d'inférence utilisées pour résoudre des problèmes complexes basés sur la logique floue. Le FCL est conçu pour être facilement compréhensible et modifiable par les utilisateurs, même sans connaissances approfondies en logique floue.

JFuzzyLogic facilite la création et la manipulation des éléments clés d'un système de logique floue en fournissant des classes Java spécifiques. Elle permet de définir les variables linguistiques, les ensembles flous et les règles du système, ainsi que d'effectuer des opérations d'inférence floue pour obtenir des résultats en utilisant différentes méthodes d'inférence telles que la somme de produits (Mamdani) ou le maximum. La bibliothèque JFuzzyLogic offre également des fonctionnalités pour visualiser et analyser les résultats des systèmes de logique floue, ce qui facilite le processus de développement et de débogage. [38]

6-Etudes de cas :

Afin de tester l'applicabilité de notre approche, on a procédé pour le développement de deux études de cas qui concernent le code de la route : *les panneaux de signalisation* et *les priorités de circulation*. Ce dernier a été implémenté de sorte à faire circuler les voitures selon deux scénarios différents.

1^{ere} étude de cas :

La figure suivante montre une voiture qui circule sur une route avec un panneau de signalisation qui indique une obligation de ne pas dépasser la vitesse 40 km/h.

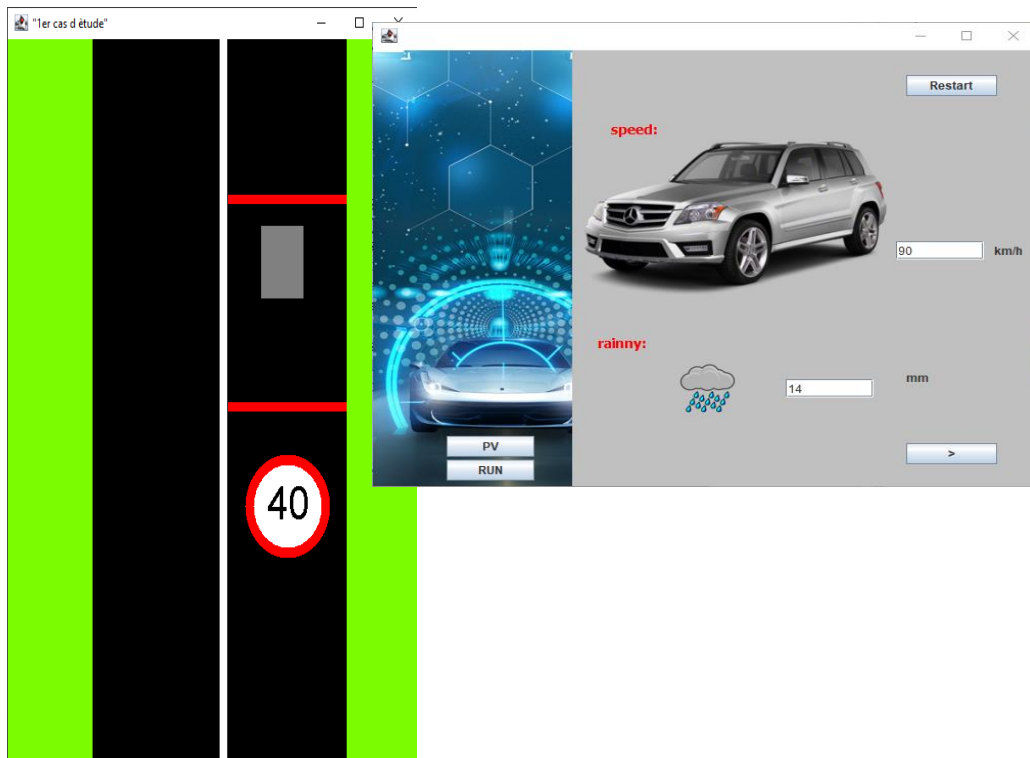
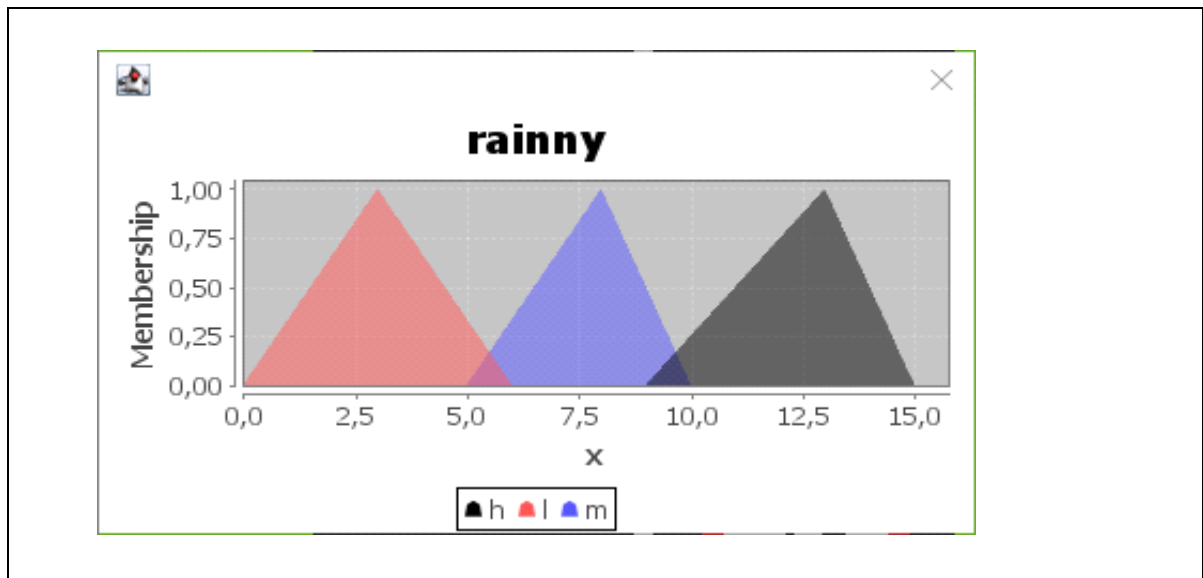


Figure 9 : 1ère étude de cas.

On a proposé deux variables linguistiques : *SPEED* et *RAINNY*. Ainsi, L'étude de cas a été lancée avec la configuration suivante : une vitesse de 90 km/h et une quantité de pluie élevée 14 mm.

Variable linguistique	Valeur linguistique	Fonction d'appartenance	Domaine de discours
Speed	High = 90	Triangulaire	0 – 180 km/h
Variable linguistique	Valeur linguistique	Fonction d'appartenance	Domaine de discours
Rainny	Low = 14	triangulaire	0 – 15 mm



Le fragment de code suivant (figure 10) montre la base de règles correspondante (fichier fcl) :

```

RULE 1 : IF (speed IS vh) AND (rainny IS l) THEN punishment IS vh;
RULE 2 : IF (speed IS vh) AND (rainny IS m) THEN punishment IS vh;
RULE 3 : IF (speed IS vh) AND (rainny IS h) THEN punishment IS vh;
RULE 4 : IF (speed IS h) AND (rainny IS l) THEN punishment IS h;
RULE 5 : IF (speed IS h) AND (rainny IS m) THEN punishment IS vh;
RULE 6 : IF (speed IS h) AND (rainny IS h) THEN punishment IS vh;
RULE 7 : IF (speed IS m) AND (rainny IS l) THEN punishment IS m;
RULE 8 : IF (speed IS m) AND (rainny IS m) THEN punishment IS h;
RULE 9 : IF (speed IS m) AND (rainny IS h) THEN punishment IS h;
RULE 10 : IF (speed IS l) AND (rainny IS l) THEN punishment IS vl;
RULE 11 : IF (speed IS l) AND (rainny IS m) THEN punishment IS vl;
RULE 12 : IF (speed IS l) AND (rainny IS h) THEN punishment IS vl;
RULE 13 : IF (speed IS vl) AND (rainny IS l) THEN punishment IS vl;
RULE 14 : IF (speed IS vl) AND (rainny IS m) THEN punishment IS vl;
RULE 15 : IF (speed IS vl) AND (rainny IS h) THEN punishment IS vl;
RULE 16 : IF (speed IS vh) AND (rainny IS l) THEN reward IS vl;
RULE 17 : IF (speed IS vh) AND (rainny IS m) THEN reward IS vl;
RULE 18 : IF (speed IS vh) AND (rainny IS h) THEN reward IS vl;
RULE 19 : IF (speed IS h) AND (rainny IS l) THEN reward IS l;
RULE 20 : IF (speed IS h) AND (rainny IS m) THEN reward IS l;
RULE 21 : IF (speed IS h) AND (rainny IS h) THEN reward IS vl;
RULE 22 : IF (speed IS m) AND (rainny IS l) THEN reward IS l;
RULE 23 : IF (speed IS m) AND (rainny IS m) THEN reward IS l;
RULE 24 : IF (speed IS m) AND (rainny IS h) THEN reward IS vl;
RULE 25 : IF (speed IS l) AND (rainny IS l) THEN reward IS vh;
RULE 26 : IF (speed IS l) AND (rainny IS m) THEN reward IS h;
RULE 27 : IF (speed IS l) AND (rainny IS h) THEN reward IS m;
RULE 28 : IF (speed IS vl) AND (rainny IS l) THEN reward IS vh;
RULE 29 : IF (speed IS vl) AND (rainny IS m) THEN reward IS vh;
    
```

RULE 30 : IF (speed IS v1) AND (rainny IS h) THEN reward IS vh;

Figure 10: base de règles floues pour le 1ere étude de cas.

6-1-Résultats et discussion :

La figure suivante montre le résultat du processus de renforcement flou montré par l'agent GUI (figure11, figure12) après avoir reçu la décision prise par l'enforcer. La figure de l'étude de cas montre un panneau de signalisation indiquant une vitesse max de 40 km/h or le moniteur à capturé une vitesse de **90 KM/h** et un taux de pluie élevé (**14 mm**). Les règles activées dans ce cas sont : Rule 6 et Rule 21. Par conséquent, le conducteur (agent) de cette voiture sera puni avec 8250.00 DZD et récompensé avec 702.22 DZD. En somme, la valeur de la punition sera : $8250.00 \text{ DZD} - 702.22 \text{ DZD} = 7,547.78 \text{ DZD}$.



Figure 11 : résultat de la décision floue prise par l'enforcer.

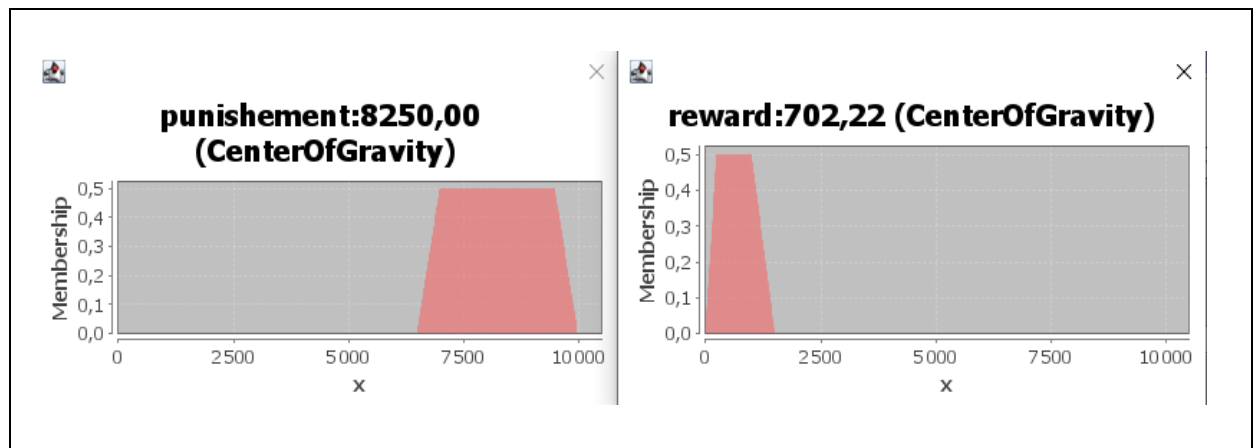


Figure 12: résultat de la défuzzification généré par le plugin jFuzzyLogic.

2^{eme} étude de cas :

La deuxième étude de cas représente une priorité de circulation dans une intersection avec des feux tricolores (figure13). Cette étude de cas a été tester selon deux scénarios.

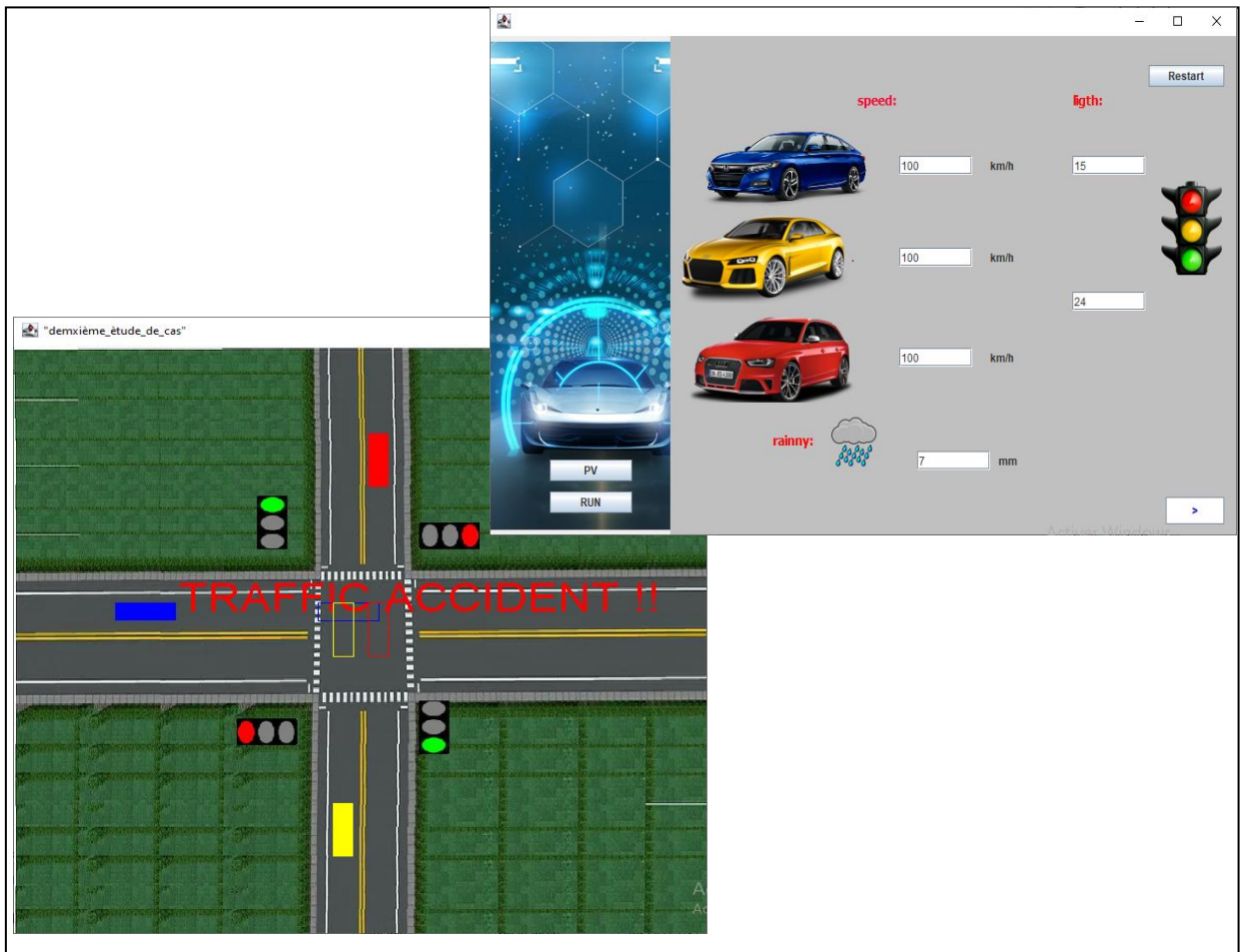


Figure 13 : 2ème étude de cas.

Le fragment du code suivant montre la base de règles floues élaborée pour cette étude de cas.

```

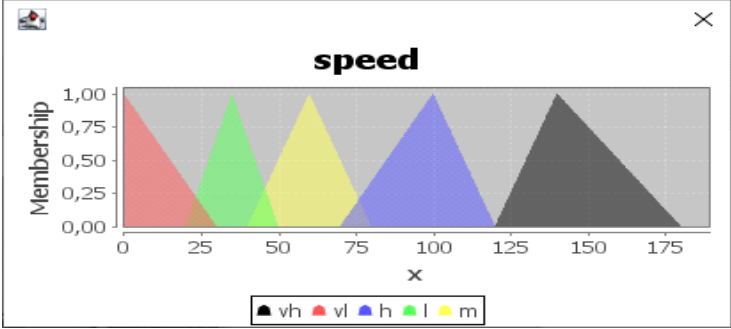
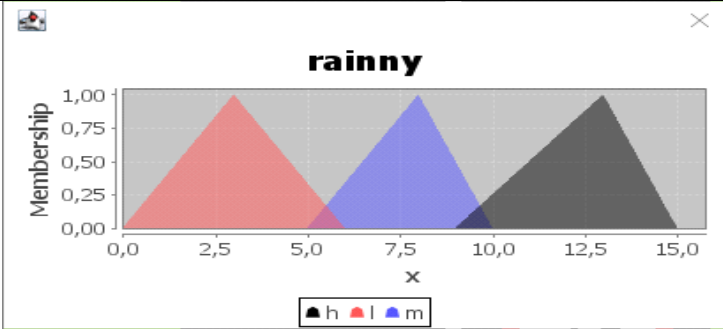
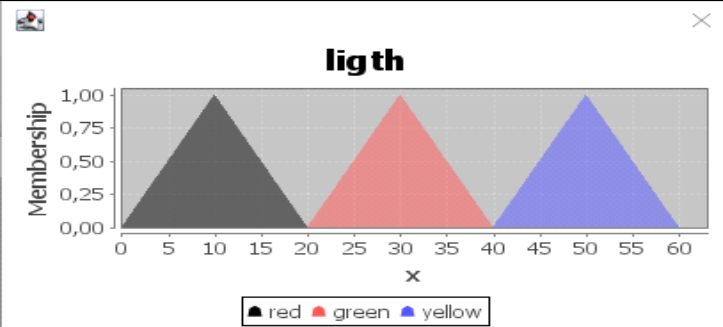
RULE 1 : IF (speed IS vh) AND (ligh IS red)AND (rainny IS l) THEN punishment IS vh ;
RULE 2 : IF (speed IS vh) AND (ligh IS red)AND (rainny IS m) THEN punishment IS vh ;
RULE 3 : IF (speed IS vh) AND (ligh IS red)AND (rainny IS h) THEN punishment IS vh ;
RULE 4 : IF (speed IS h) AND (ligh IS red)AND (rainny IS l) THEN punishment IS vh ;
RULE 5 : IF (speed IS h) AND (ligh IS red)AND (rainny IS m) THEN punishment IS vh ;
RULE 6 : IF (speed IS h) AND (ligh IS red)AND (rainny IS h) THEN punishment IS vh ;
RULE 7 : IF (speed IS m) AND (ligh IS red)AND (rainny IS l) THEN punishment IS h ;
RULE 8 : IF (speed IS m) AND (ligh IS red)AND (rainny IS m) THEN punishment IS h ;
RULE 9 : IF (speed IS m) AND (ligh IS red)AND (rainny IS h) THEN punishment IS vh ;
RULE 10 : IF (speed IS l) AND (ligh IS red)AND (rainny IS l) THEN punishment IS m ;
RULE 11 : IF (speed IS l) AND (ligh IS red)AND (rainny IS m) THEN punishment IS m ;
RULE 12 : IF (speed IS l) AND (ligh IS red)AND (rainny IS h) THEN punishment IS h ;
RULE 13 : IF (speed IS vl) AND (ligh IS red)AND (rainny IS l) THEN punishment IS m ;
RULE 14 : IF (speed IS vl) AND (ligh IS red)AND (rainny IS m) THEN punishment IS m ;
RULE 15 : IF (speed IS vl) AND (ligh IS red)AND (rainny IS h) THEN punishment IS h ;
RULE 16 : IF (speed IS vh) AND (ligh IS green)AND (rainny IS l) THEN punishment IS m ;
RULE 17 : IF (speed IS vh) AND (ligh IS green)AND (rainny IS m) THEN punishment IS m ;
RULE 18 : IF (speed IS vh) AND (ligh IS green)AND (rainny IS h) THEN punishment IS h ;
RULE 19 : IF (speed IS h) AND (ligh IS green)AND (rainny IS l) THEN punishment IS l ;
RULE 20 : IF (speed IS h) AND (ligh IS green)AND (rainny IS m) THEN punishment IS l ;
    
```

RULE 21: IF (speed IS h) AND (ligth IS green)AND (rainny IS h) THEN punishment IS m ;
RULE 22: IF (speed IS m) AND (ligth IS green)AND (rainny IS l) THEN punishment IS vl ;
RULE 23 : IF (speed IS m) AND (ligth IS green)AND (rainny IS m) THEN punishment IS vl ;
RULE 24 : IF (speed IS m) AND (ligth IS green)AND (rainny IS h) THEN punishment IS vl ;
RULE 25 : IF (speed IS l) AND (ligth IS green)AND (rainny IS l) THEN punishment IS vl ;
RULE 26 : IF (speed IS l) AND (ligth IS green)AND (rainny IS m) THEN punishment IS vl ;
RULE 27 : IF (speed IS l) AND (ligth IS green)AND (rainny IS h) THEN punishment IS vl ;
RULE 28 : IF (speed IS vl) AND (ligth IS green)AND (rainny IS l) THEN punishment IS vl ;
RULE 29 : IF (speed IS vl) AND (ligth IS green)AND (rainny IS m) THEN punishment IS vl ;
RULE 30 : IF (speed IS vl) AND (ligth IS green)AND (rainny IS h) THEN punishment IS vl;
RULE 31 : IF (speed IS vh) AND (ligth IS red)AND (rainny IS l) THEN reward IS vl ;
RULE 32 : IF (speed IS vh) AND (ligth IS red)AND (rainny IS m) THEN reward IS vl;
RULE 33 : IF (speed IS vh) AND (ligth IS red)AND (rainny IS h) THEN reward IS vl ;
RULE 34 : IF (speed IS h) AND (ligth IS red)AND (rainny IS l) THEN reward IS vl;
RULE 35 : IF (speed IS h) AND (ligth IS red)AND (rainny IS m) THEN reward IS vl ;
RULE 36 : IF (speed IS h) AND (ligth IS red)AND (rainny IS h) THEN reward IS vl;
RULE 37 : IF (speed IS m) AND (ligth IS red)AND (rainny IS l) THEN reward IS l;
RULE 38 : IF (speed IS m) AND (ligth IS red)AND (rainny IS m) THEN reward IS l;
RULE 39 : IF (speed IS m) AND (ligth IS red)AND (rainny IS h) THEN reward IS l;
RULE 40 : IF (speed IS l) AND (ligth IS red)AND (rainny IS l) THEN reward IS l;
RULE 41 : IF (speed IS l) AND (ligth IS red)AND (rainny IS m) THEN reward IS l;
RULE 42 : IF (speed IS l) AND (ligth IS red)AND (rainny IS h) THEN reward IS l ;
RULE 43 : IF (speed IS vl) AND (ligth IS red)AND (rainny IS l) THEN reward IS l;
RULE 44 : IF (speed IS vl) AND (ligth IS red)AND (rainny IS m) THEN reward IS l ;
RULE 45 : IF (speed IS vl) AND (ligth IS red)AND (rainny IS h) THEN reward IS l ;
RULE 46 : IF (speed IS vh) AND (ligth IS green)AND (rainny IS l) THEN reward IS l;
RULE 47 : IF (speed IS vh) AND (ligth IS green)AND (rainny IS m) THEN reward IS l ;
RULE 48: IF (speed IS vh) AND (ligth IS green)AND (rainny IS h) THEN reward IS vl ;
RULE 49: IF (speed IS h) AND (ligth IS green)AND (rainny IS l) THEN reward IS m;
RULE 50: IF (speed IS h) AND (ligth IS green)AND (rainny IS m) THEN reward IS l;
RULE 51: IF (speed IS h) AND (ligth IS green)AND (rainny IS h) THEN reward IS vl;
RULE 52: IF (speed IS m) AND (ligth IS green)AND (rainny IS l) THEN reward IS vh ;
RULE 53 : IF (speed IS m) AND (ligth IS green)AND (rainny IS m) THEN reward IS h;
RULE 54 : IF (speed IS m) AND (ligth IS green)AND (rainny IS h) THEN reward IS m ;
RULE 55 : IF (speed IS l) AND (ligth IS green)AND (rainny IS l) THEN reward IS vh;
RULE 56 : IF (speed IS l) AND (ligth IS green)AND (rainny IS m) THEN reward IS vh ;
RULE 57 : IF (speed IS l) AND (ligth IS green)AND (rainny IS h) THEN reward IS h;
RULE 58 : IF (speed IS vl) AND (ligth IS green)AND (rainny IS l) THEN reward IS vh;
RULE 59 : IF (speed IS vl) AND (ligth IS green)AND (rainny IS m) THEN reward IS vh;
RULE 60 : IF (speed IS vl) AND (ligth IS green)AND (rainny IS h) THEN reward IS h;

1er scénario :

La table suivante montre la configuration du premier scénario :

Tableau 3:la configuration du premier scénario

Variable linguistique	Valeur linguistique	Fonction d'appartenance	Domaine de discours
<i>Speed yellow car</i>	High = 100	<i>triangulaire</i>	0 – 200 km/h
<i>Speed red car</i>			
<i>Speed blue car</i>			
			
Variable linguistique	Valeur linguistique	Fonction d'appartenance	Domaine de discours
<i>Rainny</i>	Medium = 7	<i>triangulaire</i>	0 – 15 mm
			
Variable linguistique	Valeur linguistique	Fonction d'appartenance	Domaine de discours
Light blue car	RED = 15 s	<i>triangulaire</i>	0 – 60 secondes
Light yellow car & red car	GREEN = 24 s		
			

6-2-Résultats et analyse :

La configuration précédente montre que les trois voitures roulent avec une vitesse 100 km/h or le feu tricolore indique une priorité de passage pour les voitures jaune et rouge (feu vert) et un stop obligatoire pour la voiture bleue. Comme les vitesses des trois voitures était

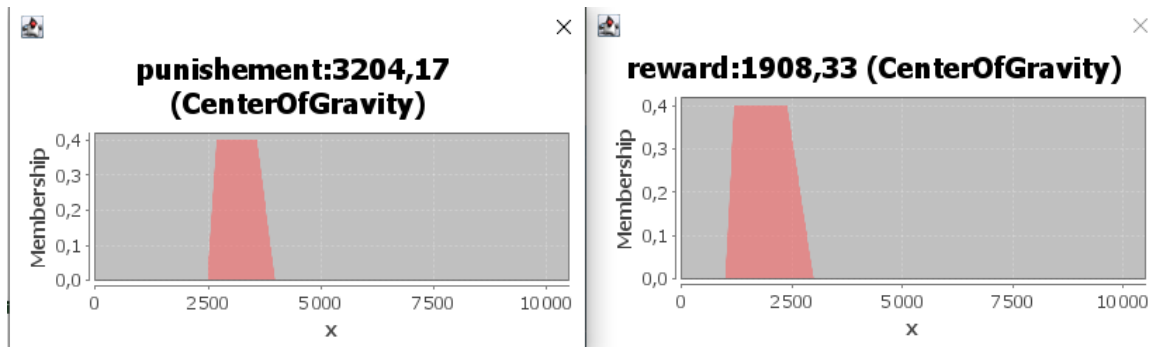


Figure 14: résultat de la défuzzification généré par le plugin jFuzzyLogic pour les voitures : jaune et rouge.

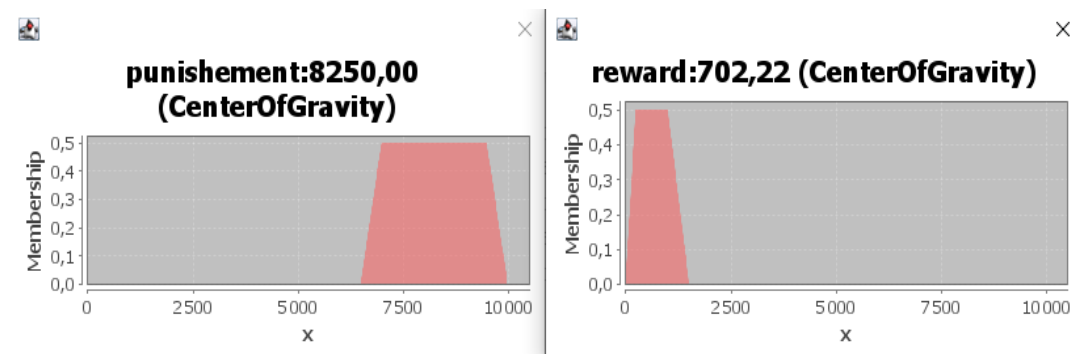


Figure 15 : résultat de la défuzzification généré par le plugin jFuzzyLogic pour la voiture bleue.

trop élevé et ont tous le même point de départ un accident a été enregistré au milieu de l'intersection ce qui montre une punition signalée pour les trois voitures avec un degré faible pour les voitures jaune et rouge puisque le passage a été libre (feu vert) et une punition élevée pour la voiture bleue (feu rouge).



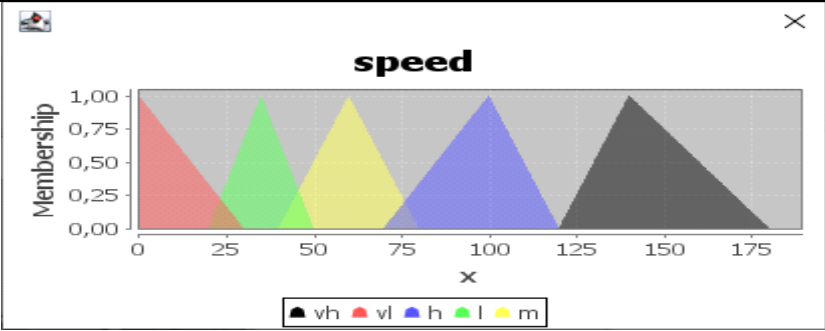
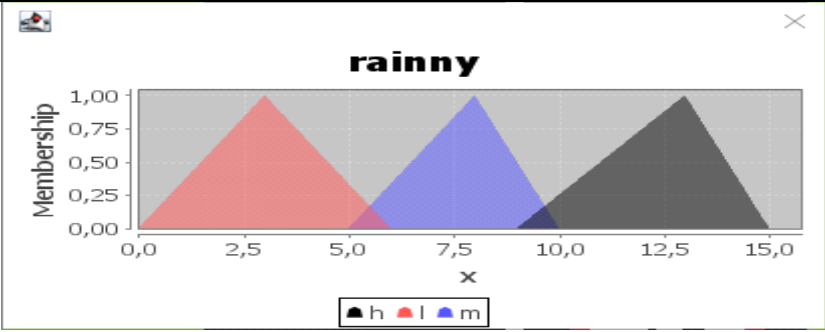
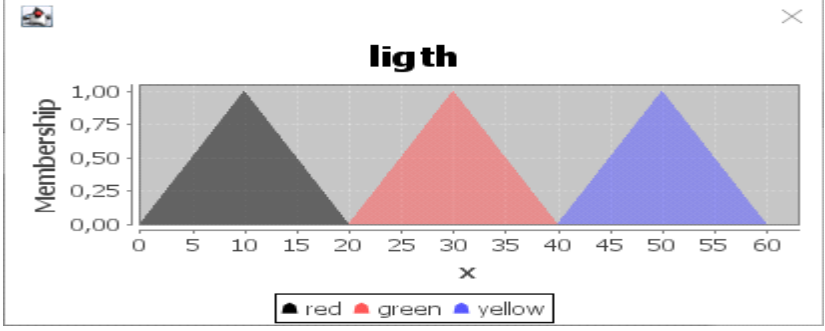
Figure 16: résultat de la décision floue prise par l'enforcer.

Chapitre III : Approche proposée, Implémentation, Résultats et Discussion

2eme scénario :

La table suivante montre la configuration du deuxième scénario :

Tableau 4:la configuration du deuxième scénario

Variable linguistique	Valeur linguistique	Fonction d'appartenance	Domaine de discours
Speed yellow	Very High = 170	<i>triangulaire</i>	0 – 180 km/h
Speed red	High = 100		
Speed blue	Low = 40		
			
Variable linguistique	Valeur linguistique	Fonction d'appartenance	Domaine de discours
Rainny	Low = 3	<i>triangulaire</i>	0 – 15 mm
			
Variable linguistique	Valeur linguistique	Fonction d'appartenance	Domaine de discours
Light car blue	Green = 24 s	<i>triangulaire</i>	0-60 secondes
Light car yellow & car red	Red = 15 s		
			

6-3-Résultats et analyse :

Après l'exécution du 2em scénario, la voiture bleue a été permise de passer (feu vert) et elle a respecté la norme donc elle a été récompensée et les voitures jaune et rouge doivent se stopper (feu rouge) mais ils n'ont pas respecté la norme donc ils ont été punis.



Figure 17: résultat de la décision floue prise par l'enforcer.

La figure suivante montre le montant de récompense calculé par l'enforcer pour la voiture bleue, soit 8250.00 DZD et un montant de punition de 702.22 DZD. En somme : la voiture bleue sera récompensé avec $8250.00 - 702.22 = 7547.78$ DZD.

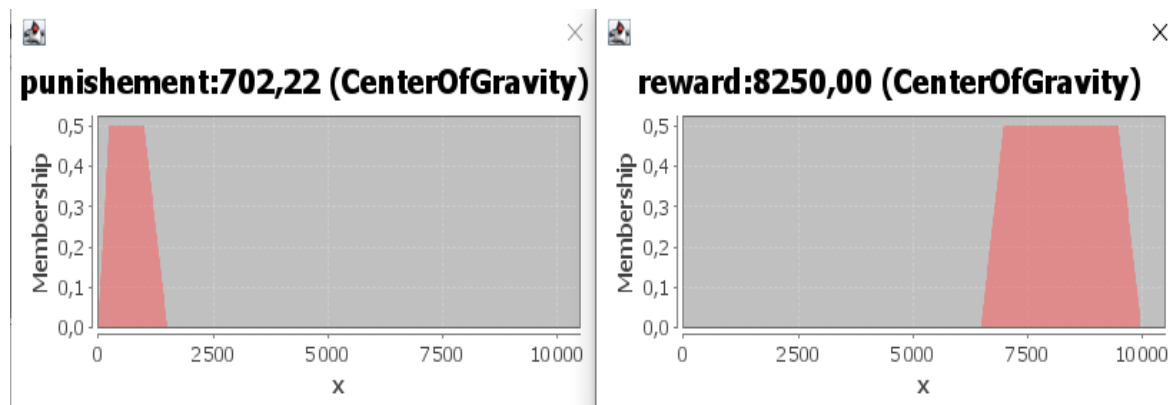


Figure 18 : résultat de la défuzzification généré par le plugin jFuzzyLogic pour la voiture bleue.

En revanche, la voiture rouge sera punie parce qu'elle n'a pas respecté la norme. L'enforcer à montrer un montant de punition égale à 5136.36 DZD et un montant de récompense de 1833.33 DZD. En somme, le conducteur de la voiture rouge doit payer une amende de : $5136.36 - 1833.33 = 3303.03$ DZD

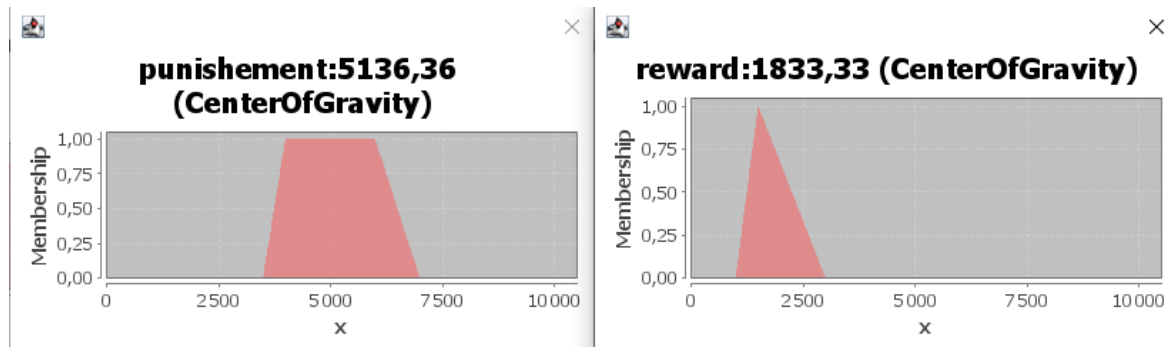


Figure 19: résultat de la défuzzification générée par le plugin jFuzzyLogic pour la voiture rouge.

De même, le conducteur de la voiture jaune doit payer une amende de : $8250.00 - 729.07 = 7,520.93$ DZD.

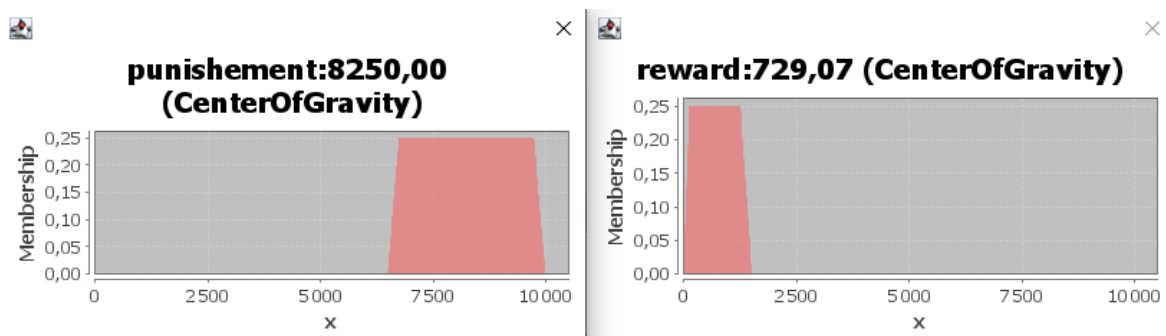


Figure 20: résultat de la défuzzification générée par le plugin jFuzzyLogic pour la voiture jaune.

7-Conclusion:

Dans ce chapitre, nous avons présenté notre idée sur le renforcement flou des normes en montrant l'architecture de notre approche proposée nommée FE4NorMAS pour **Fuzzy Enforcement for Normative Multi-Agent Systems**. Nous avons également justifié l'applicabilité de notre approche sur une étude de cas dans le domaine du trafic routier sur plusieurs scénarios possible.



Conclusion générale

Conclusion générale

Bilan

Dans le présent manuscrit, nous avons proposé une approche de renforcement flou des normes dont l'objectif était de profiter les points forts du raisonnement flou pour maîtriser le processus de sanction des agents de sorte à le rendre plus flexible et adaptatif. A cet effet, nous avons délégué deux types d'agent qui s'occupent du processus de renforcement flou : un agent moniteur de type réactif responsable de l'observation et la surveillance du comportement des agents systèmes vis-à-vis les normes et un agent enforcer qui est un agent flou qui produit des décisions de renforcement selon un mécanisme de raisonnement flou. La base de règles floues dans notre proposition a été établie au préalable (i.e. conception offline) dont on a représenté les normes sous forme de règles floues

Ainsi, nous avons procédé pour une implémentation d'une étude de cas traitant le trafic routier sous la plateforme MaDKit. Le processus de raisonnement flou a été implémenté en utilisant le plugin jFuzzyLogic : l'outil populaire offrant un moyen d'implémentation des architectures floues pour les développeurs JAVA. Enfin, on a testé l'applicabilité de notre approche sur une étude de cas dans le domaine du trafic routier comportant plusieurs scénarios possibles.

Perspectives

Nous envisageons, comme perspective à ce travail, de multiplier le nombre de variables linguistiques afin de couvrir la majorité des aspects liés au domaine du trafic routier et d'étendre notre approche de renforcement flou pour qu'elle soit opérationnelle sur plusieurs plateformes agents (JADE, AgentFactory, etc.), qui ne dépendent pas du modèle comme le cas de MaDKit.

Références

Références

Références :

- [1]. T. Marir, "Les Systèmes Multi-Agents," Cour, Master 2, Architecture Distribuée, Université d'Oum El Bouaghi, Oum El Bouaghi, Algeria, 2018
- [2]. B. Benaissa, "Mesure de la qualité des systèmes multi-agents Basés DIMA," Mémoire de Master, Université Larbi Ben M'hidi d'Oum El Bouaghi, 2013.
- [3]. J. Ferber, "Les systèmes multi-agents : un aperçu général Multi-Level Emergence View project Exception handling in agent-based systems View project," 2014. Available: <https://www.researchgate.net/publication/242623967>
- [4]. P.-M. Ricordel and Y. Demazeau, "Environnement de développement de systèmes multi-agents-la plate-forme volcano. modularité et réutilisation pour les systèmes multi-agents," *TSI-Technique et Science Informatiques-RAIRO*, vol. 21, no. 4, pp. 447–472, 2002.
- [5]. M. Wooldridge, *An introduction to multiagent systems*. John Wiley & Sons, 2009.
- [6]. M. Wooldridge, "An Introduction to Multi Agent Systems," John Wiley & Sons Ltd, 2002.
- [7]. M. Savall, J.-P. Pécuchet, N. Chaignaud, and M. Itmi, "YAMAM—un modèle d'organisation pour les systèmes multi-agents. Implémentation dans la plate-forme Phoenix," in *Proceedings of 3ème Conférence Francophone MOSIM (Modélisation et simulation)*, 2001, pp. 25–27.
- [8]. F. Klein, "Contrôle d'un SMA Réactif par Modélisation et Apprentissage de sa Dynamique Globale.," 2009.
- [9]. http://liawww.epfl.ch/~schumacher/publications/environmentForMAS_ker_2005.
- [10]. Boussebough, "Les systèmes multi-agents dynamiquement adaptables," *Doctoral dissertation, Thèse de doctorat*, 2011.
- [11]. N. Bakhta, "Modèle Multi-Agent pour la conception de systèmes d'aide à la décision collective," Thèse de doctorat en informatique, sous la direction d'A. Aalaa, Université d'Oran, Oran, 2014, 155 p.
- [12]. K. Benhamza, "Conception d'un système multi-agents adaptatif pour la résolution de problème," *UNIVERSITE BADJI MOKHTAR, ANNABA*, 2016.
- [13]. Mastria, *Systèmes complexes et SMA, Informatique Bio-Inspirée*.
- [14]. O. Boissier, *Système multi-agents Organisation, support de cours, SMA/ENS Mines Saint-Etienne*, 2011.

Références

- [15]. www.larousse.fr. (2020). Consulté le 6, 6, 2020, sur Dictionnaire online : <https://www.larousse.fr/dictionnaires/francais>.
- [16]. G. Boella, L. Van Der Torre, and H. Verhagen, "Introduction to normative multiagent systems," *Computational & Mathematical Organization Theory*, vol. 12, pp. 71-79, 2006.
- [17]. M. A. Mahmoud, M. S. Ahmad, M. Z. Mohd Yusoff, and A. Mustapha, "A review of norms and normative multiagent systems," *The Scientific World Journal*, vol. 2014, 2014.
- [18]. T. Balke *et al.*, "Norms in MAS: definitions and related concepts," in *Dagstuhl Follow-Ups*, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2013.
- [19]. J. S. Santos, J. O. Zahn, E. A. Silvestre, V. T. Silva, and W. W. Vasconcelos, "Detection and resolution of normative conflicts in multi-agent systems: a literature survey," *Auton Agent Multi Agent Syst*, vol. 31, pp. 1236–1282, 2017.
- [20]. D. G. L. v. d. T. Davide Grossi, The Norm Implementation Problem.
- [21]. C. D. Hollander and A. S. Wu, "The current state of normative agent-based systems," *Journal of Artificial Societies and Social Simulation*, vol. 14, no. 2, p. 6, 2011.
- [22]. Chebout Mohamed sedik : "Contrôlabilité des systèmes multi agents ouverts," Thèse de doctorat, Université Larbi ben M'hidi, Oum-El-Bouaghi, 2019.
- [23]. [Http: //www.computerworld.com/news/2004/story/0,11280,95282,00.html](http://www.computerworld.com/news/2004/story/0,11280,95282,00.html)
- [24]. L. A. Zadeh, "Dispositional logic and commonsense reasoning," in *NASA. Johnson Space Center, the 1988 First Joint Technology Workshop on Neural Networks and Fuzzy Logic*, 1988.
- [25]. S. Omar, M. Waweru, and R. Rimiru, "A literature survey: Fuzzy logic and qualitative performance evaluation of supply chain management," *Int. J. Engineeirng Sci*, vol. 4, no. 5, pp. 56–63, 2015.
- [26]. R. Leonid, "Fuzzy controllers," *Newnes*, vol. 1, pp. 1–9, 1997.
- [27]. Y. Bai and D. Wang, "Fundamentals of fuzzy logic control—fuzzy sets, fuzzy rules and defuzzifications," *Advanced fuzzy logic technologies in industrial applications*, pp. 17–36, 2006.
- [28]. F. Lahouazi, "Mise en oeuvre d'une stratégie de commande neuro floue, application à un Pendule inversé," Mémoire de Magister en Automatique, Université Mouloud Mammeri, Tizi-Ouzou, 2011.

Références

- [29]. M Madaoui, “ commande floue et neuro-floue d’un convertisseur direct AC/AC monophasé à structure minimale, Application à l’alimentation d’une machine asynchrone monophasé ”, Thèse de Magister, ENP Alger, (2007).
- [30]. M. Billel and N. BOUHABZA, “Commande par logique floue de type 2 et commande neuro-floue appliquées à un pendule inversé,” *Mémoire de Master, Université de Blida*, vol. 1, 2015.
- [31]. Antoine Cornuéjols : “Introduction à la logique floue, présentation”, janvier 2008.
- [32]. N. S. M. Aymen, K. M. Ines, and C. M. Sedik, “Un Système multi-agents basé flou pour le contrôle d’un smart street lighting,” 2021.
- [33]. S. Labiod, “ Contribution à la commande adaptative floue des systèmes non linéaire ”, Thèse Doctorat, (2004).
- [34]. C. Dualibe, M. Verleysen, and P. G. A. Jespers, “Mixed-Signal Programmable Fuzzy Logic Controllers: Design, fabrication and test of demonstrators,” *Design of Analog Fuzzy Logic Controllers in CMOS Technologies: Implementation, Test and Application*, pp. 107–154, 2003.
- [35]. Salim Boukaka, “ réalisation d'une bibliothèque de lois de commande adaptative pour msap “, UNIVERSITÉ DU QUÉBEC, (2015).
- [36]. L.-X. Wang, *Adaptive fuzzy systems and control: design and stability analysis*. Prentice-Hall, Inc., 1994.
- [37]. Gutknecht, O., & Ferber, J. (2000). MadKit: a generic multi-agent platform. In: Wagner T., Rana O.F. (eds) Infrastructure for Agents, Multi-Agent Systems, and Scalable Multi-Agent Systems. AGENTS 2000. Lecture Notes in Computer Science, 1887. doi:https://doi.org/10.1007/3-540-47772-1_5
- [38]. P. Cingolani and J. Alcalá-Fdez, "jFuzzyLogic: a robust and flexible Fuzzy-Logic inference system language implementation," in 2012 IEEE International Conference on Fuzzy Systems, June 2012, pp. 1-8.
- [39]. Chebout, M., Mokhati, F. and Badri, M. NC4OMAS: A Norms-based Approach for Open Multi-Agent Systems Controllability. DOI: 10.5220/0010793600003116 In Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022) - Volume 1, pages 164-171 ISBN: 978-989-758-547-0; ISSN: 2184-433X