

*République Algérienne Démocratique et Populaire*  
*Ministère de L'enseignement Supérieur et de la Recherche Scientifique*



*Université d'Oum El Bouaghi*



*Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie*  
*Research Laboratory on Computer Science' Complex Systems (ReLa(CS)<sup>2</sup>)*

*Département des Mathématiques et d'Informatique*

## **Extension du framework NorJADE Pour supporter la programmation organisationnelle**

**Mémoire présenté pour l'obtention du diplôme de Master en *Informatique***  
***Spécialité : Architectures Distribuées***

*Travail réalisé par :*

- Ala Eddine Ounadi .
- Zoghmar Badis .

*Travail encadré par :*

- Dr MARIR Toufik .

*Jury :*

- Président : Dr. Belhouchet Kenza
- Encadrant : Dr. Marir Toufik
- Examineur : Dr. Silem Abdelhaq .

# Remerciements

The realization of this memorial was possible thanks to the help of several people to whom I would like to express my gratitude.

First of all, I would like to express my gratitude to the Director of this brief, Mr Toufik MARIR, for his patience, his availability and above all his wise advice, which have contributed to my reflection.

I would especially like to thank Toufik MARIR, who was the first to introduce me to the subject that guided my brief.

I would also like to thank the professors at the University of Oum-EL-Bouaghi, who provided me with the tools I needed to succeed in university.

Finally, I would like to express my gratitude to all the researchers and specialists, too many to mention, who took the time to discuss my subject. Each of these exchanges helped me advance my analysis.

# Dédicace

I dedicate this dissertation thesis to the unwavering love and support of my incredible parents. From the very beginning, you believed in me and instilled in me the values of hard work, perseverance, and resilience. Your constant encouragement and sacrifices have been the guiding force behind my academic journey, and I am forever grateful for your unwavering belief in my abilities. This achievement is as much yours as it is mine.

To my beloved family, your unconditional love and unwavering faith in my dreams have been a constant source of inspiration throughout this academic pursuit. Your unyielding support, understanding, and patience have given me the strength to overcome challenges and reach for the stars. Thank you for being my pillars of strength and for always being there to celebrate my successes.

To all the dear friends who have played a significant role in my journey, thank you for your invaluable contributions. Your guidance, discussions, and shared knowledge have been instrumental in shaping my ideas and expanding my horizons. Your presence and encouragement have made this experience not only academically enriching but also immensely enjoyable.

Finally, I would like to express my deep gratitude to every individual who has supported me in ways both big and small. Your words of encouragement, acts of kindness, and belief in my abilities have provided me with the strength to overcome obstacles and strive for excellence. This achievement would not have been possible without your unwavering support.

May this work stand as a testament to the collective efforts of my family, dear friends, and all those who have touched my life. Your presence in my journey has made it richer, and I am forever grateful for the love and support that has shaped me into the person I am today.

## Résumé

La plateforme JADE est l'une des plateformes les plus connues pour le développement des systèmes multi agent. Cette plateforme focalise sur les concepts d'agents et leurs interactions. Cependant, les concepts organisationnels sont quasi ignorés dans cette plateforme. Dans ce contexte, le framework NorJADE propose une extension de JADE pour prendre en considération les concepts normatifs. Ce framework propose la possibilité de manipuler les comportements des agents par des normes. Considérant les normes comme une façade des organisations multi-agents, nous pensons qu'un enrichissement de NorJADE par des concepts organisationnels va offrir une meilleure expressivité. Ce travail propose une extension de NorJADE pour supporter la programmation des concepts organisationnels.

Mots clés : Systèmes Multi-agents, Organisations, JADE, NorJADE.

## Abstract

The JADE platform is one of the best-known platforms for the development of multi-agent systems. This platform focuses on agent concepts and their interactions. However, organizational concepts are almost ignored in this platform. In this context, the NorJADE framework proposes an extension of JADE to take into account normative concepts. This framework offers the possibility to manipulate the behaviors of agents by standards. Considering standards as a front for multi-agent organizations, we think that an enrichment of NorJADE by organizational concepts will offer a better expressiveness. This work proposes an extension of NorJADE to support the implementation of organizational concepts.

Keywords: Multi-agent Systems, Organizations, JADE, NorJADE.

## ملخص

تعد منصة JADE واحدة من أشهر المنصات لتطوير الأنظمة متعددة الوكلاء. تركز هذه المنصة على مفاهيم الوكيل وتفاعلاتها. ومع ذلك، يتم تجاهل المفاهيم التنظيمية تقريباً في هذه المنصة. وفي هذا السياق، يقترح إطار NorJADE تمديد تمثيل JADE لمراعاة المفاهيم المعيارية. يوفر هذا الإطار إمكانية التحكم بسلوكيات الوكلاء بالمعايير. بالنظر إلى المعايير كواجهة للمنظمات متعددة الوكلاء، نعتقد أن إثراء NorJADE من خلال المفاهيم التنظيمية سيوفر تعبيراً أفضل. نقدم في هذا العمل، امتداد لمنصة NorJADE لبرمجة المفاهيم التنظيمية.

الكلمات الرئيسية: JADE - NorJADE - المنظمات - الأنظمة متعددة الوكلاء

## Sommaire

<b>Introduction générale .....</b>	<b>12</b>
<b>Chapitre 01 :.....</b>	<b>13</b>
<b>Les Systèmes Multi-Agents Normatifs.....</b>	<b>13</b>
<b>1- Introduction.....</b>	<b>14</b>
<b>2- Les systèmes multi-agents.....</b>	<b>14</b>
2.1- Agent :.....	15
2.2- Environnement :.....	17
2.3- Interaction :.....	17
2.4- l'organisation .....	19
<b>3- Les systèmes normatifs.....</b>	<b>19</b>
<b>4- Les systèmes multi-agents normatifs (NorMAS).....</b>	<b>20</b>
4.1- La notion de «NorMAS» .....	20
<b>L'idée de réguler le système multi-agents.....</b>	<b>20</b>
4.2- La présentation des normes dans NorMAS : .....	21
4.2.1- La notion : .....	21
4.2.2- Classification des normes :.....	21
4.2.3- Cycle de vie de la norme :.....	24
4.3- Implémentation des normes dans NorMAS.....	25
<b>5- Conclusion.....</b>	<b>26</b>
<b>Chapitre 02 :.....</b>	<b>27</b>
<b>Les Organisations Multi-Agents. ....</b>	<b>27</b>
<b>Introduction : .....</b>	<b>28</b>
<b>2- L'organisation multi-agent .....</b>	<b>28</b>
2.1 C'est quoi une organisation : .....	28
2.2 Définition d'organisation : .....	28
<b>3- L'organisation structurelles et concrète .....</b>	<b>28</b>
3.1- Les structures organisationnelles.....	29
3.2- Les structures concrètes.....	30
<b>4- Dimensions des organisations .....</b>	<b>30</b>
4.1- Structure organisationnelle.....	31
4.2- Fonctions dans des organisations.....	31

---

<b>4.3- Paramètres de concrétisation .....</b>	<b>32</b>
<b>5- Modèles organisationnels .....</b>	<b>33</b>
<b>5.1- AGR .....</b>	<b>33</b>
5.1.1- Agent.....	34
5.1.2- Rôle.....	34
5.1.3- Groupe.....	34
<b>5.2- Moise : .....</b>	<b>35</b>
5.2.1- les principes de Moise.....	35
5.2.2- Les spécifications de MOISE .....	35
<b>5.3- Jason .....</b>	<b>36</b>
5.3.1- AgentSpeak .....	36
5.3.2- les avantage de Les avantage de Jason.....	36
<b>6- Programmation des organisations multi-agents.....</b>	<b>37</b>
<b>6.1- MADKit .....</b>	<b>37</b>
6.1.1- Définition .....	37
6.1.2- Architecture de MADKit :.....	38
6.1.3- Fonctionnalités clés de MADKit.....	39
6.1.4- Conclusion .....	39
<b>6.2- JeCaMo .....</b>	<b>40</b>
6.2.1- Définition .....	40
6.2.2- Les avantages de JeCaMo.....	40
6.2.3- Conclusion .....	40
<b>7- Conclusion.....</b>	<b>41</b>
<b>Chapitre 03 : L'approche Proposée.....</b>	<b>42</b>
<b>1- Introduction.....</b>	<b>43</b>
<b>2- Les plateformes et les technologies utilisées .....</b>	<b>43</b>
2.1- Les ontologies.....	43
2.2- la programmation orienté aspect (AOP) .....	45
2.3- NorJade .....	46
2.3.1- NorJADE ontologie .....	46
2.3.2- NorJADE Aspects .....	47
<b>3- Approche proposée.....</b>	<b>50</b>
3.1- Description de l'ontologie .....	50
3.2 Description des aspects et java:.....	51
<b>4- Étude de cas .....</b>	<b>52</b>
<b>5- Conclusion.....</b>	<b>56</b>
<b>Conclusion générale :.....</b>	<b>57</b>

---

**Bibliographie..... 58**



## Table de figure

Figure 1: Les trois dimensions d'un agent <sup>6</sup> .....	16
Figure 2: Origine des systèmes multi-agents normatifs (NorMAS) <sup>13</sup> . .....	20
Figure 3 : Classification des normes <sup>16</sup> .....	21
Figure 4 : l'organisation structurelle et l'organisation concrète .....	29
Figure 5 : paramètre de concrétisation et analyse structurelle d'organisation 2 .....	32
Figure 6 : le modèle Agent/Groupe/Rôle (AGR) .....	34
Figure 7 : Architecture générale de MADKit .....	38
Figure 8 : Les domaines d'utilisations de l'ontologie .....	44
Figure 9 : Concepts utilisés pour spécifier les normes dans NorJADE 4.....	47
Figure 10: Les relations entre les classes (Object propriétés) .....	50
Figure 11: Les notions d'approche proposée dans NorJADE . .....	51
Figure 12: Montrez ce que nous avons ajouté dans la classe NorJADEOntologyBase .....	52
Figure 13: Exemple expliquer communication entre les rôles .....	53
Figure 14: Exemple montrant punir ou récompenser un agent en fonction du comportement d'un rôle.....	55
Figure 15: Une image d'ontologie montrant les propriétés de la loi et conséquence . .....	56

---

## Liste des tableaux

<b>Tableau 1 : Classification des situations d'interactions <sup>7</sup> .....</b>	<b>18</b>
<b>Tableau 2: Représentation la dimension des organisations .....</b>	<b>31</b>
<b>Tableau 3: Représentation schématique des déférentes fonctions d'une organisation .....</b>	<b>32</b>
<b>Tableau 4: Représentation de la spécification de Moise .....</b>	<b>35</b>
<b>Tableau 5: Détails sur les packages, les aspects, les classes et les méthodes 4.....</b>	<b>49</b>



## Introduction générale

Les systèmes multi-agents ont suscité un intérêt croissant dans de nombreux domaines de recherche et d'application. Ces systèmes, composés d'entités autonomes appelées agents, interagissent entre eux pour atteindre des objectifs communs ou individuels. L'importance des systèmes multi-agents réside dans leur capacité à résoudre des problèmes complexes, tels que la planification, la coordination et la prise de décision dans des environnements dynamiques et incertains. Grâce à leur nature décentralisée et distribuée, les systèmes multi-agents offrent une flexibilité et une adaptabilité qui les rendent particulièrement pertinents pour résoudre des problèmes complexes.

Dans la modélisation des systèmes multi-agents, les concepts organisationnels jouent un rôle crucial. Les entités autonomes dans un système multi-agent peuvent être dotées de rôles spécifiques, définis par des règles, des normes et des relations sociales. Ces concepts organisationnels permettent de structurer et de réguler les interactions entre les agents, facilitant ainsi la coordination et la coopération au sein du système. La prise en compte des concepts organisationnels dans la modélisation des systèmes multi-agents permet de mieux représenter les interactions humaines et d'améliorer la compréhension et la prédictibilité des comportements collectifs.

Cependant, malgré l'importance des concepts organisationnels, de nombreuses plateformes multi-agents actuelles ignorent ces aspects. Les modèles et les langages de programmation traditionnels pour les systèmes multi-agents se concentrent souvent sur la représentation des connaissances individuelles et des interactions directes entre les agents, négligeant ainsi les aspects organisationnels cruciaux. Cette lacune limite la capacité des développeurs à modéliser et à simuler des environnements socialement complexes, et à étudier les interactions entre les agents et les normes qui les régissent.

Dans ce mémoire de Master, notre objectif est de proposer une extension du framework NorJADE pour supporter la programmation organisationnelle dans les systèmes multi-agents. Nous cherchons à combler le fossé entre les aspects individuels et organisationnels en fournissant des outils et des mécanismes pour modéliser les rôles, les normes et les relations sociales dans les systèmes multi-agents. En intégrant ces concepts organisationnels dans NorJADE, nous visons à améliorer la capacité du framework à représenter et à simuler des environnements socialement réalistes,

## Résumé

permettant ainsi une meilleure compréhension des systèmes multi-agents normatifs et de leurs comportements collectifs.

Ce mémoire est organisé en trois chapitres. Dans le premier nous expliquons les concepts relatifs aux systèmes multi-agents en concentrant sur les concepts des systèmes multi-agents normatifs. Dans le deuxième chapitre, nous expliquons les concepts organisationnels. Ensuite, nous proposons notre extension dans le troisième chapitre.

# **Chapitre 01 :**

# **Les Systèmes**

# **Multi-Agents**

# **Normatifs**



## 1- Introduction

Le système multi agents (SMA) est considéré comme une branche de l'intelligence artificielle distribuée. Représente un paradigme de programmation qui modélise le programme en tant que groupe d'agents d'interaction indépendants. Ce paradigme est considéré comme l'un des paradigmes les plus appropriés pour développer des systèmes complexes (tels que la simulation de phénomènes sociaux et biologiques). Récemment, le paradigme est combiné avec le système normatif pour contrôler le comportement de l'agent. Par conséquent, un système multi-agents normatifs est apparu.

Dans ce chapitre, nous présenterons les concepts de base des systèmes multi-agents. Nous présenterons ensuite des termes liés aux normes et nous finissons par une présentations des systèmes multi agents normatifs.

## 2- Les systèmes multi-agents

En termes simples, un système multi-agents (SMA) est défini comme un ensemble d'entités en interaction, appelées agents.

Ou plus précisément : « *Un Système Multi-Agents comporte plusieurs agents qui interagissent entre eux dans un environnement commun. Certains de ces agents peuvent être des personnes ou leurs représentants (avatars), ou même des machines mécaniques. S'il y a moins de trois agents, on parle plutôt d'interaction homme/machine, ou machine/machine que de systèmes multi-agents.* » Anne Nicole<sup>1</sup>.

Et Ferber propose une définition plus complète de ces systèmes : « *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. »<sup>2</sup>.*

Malgré la diversité de définitions du système multi-agents, on peut remarquer l'existence de certains concepts communs entre les définitions présentées. En effet, on peut conclure que les concepts de base partagés par toutes les définitions sont: l'agent, l'interaction, l'environnement et la communication.

## 2.1- Agent :

Un agent est une entité de comportement, car le terme "agent" provient de "agere" latin, ce qui signifie faire <sup>3</sup>. Dans le domaine des systèmes multi-agences, il n'y a pas de consensus sur la définition du terme agent. En effet, plusieurs définitions ont été proposées, Par exemple, nous avons constaté que la définition de l'agent est "une entité qui perçoit son environnement et agisse sur celui-ci " <sup>4</sup>. Dans cette définition, nous avons remarqué que L'agent a un environnement. Il peut réaliser les deux actions il peut percevoir ou bien agir .La définition proposée par Jacques Ferber <sup>2</sup> est plus détaillée : "*l'agent est une entité autonome physique ou abstraite qui est capable d'agir sur elle-même et sur son environnement, qui dans un univers Multi Agents peut communiquer avec d'autres agents et dont le comportement est la conséquence de ses observations, de ses connaissances, et des interactions avec les autres agents* " .

En fait, cette définition met l'accent sur les communications entre les agents (La coopération, La coordination, La négociation) En effet, un agent est caractérisé par <sup>5</sup> :

- **L'autonomie** : c'est la caractéristique la plus importante parmi les caractéristiques de l'agent. Un agent autonome est un agent qui peut agir sans l'intervention des autres agents.
- **La situation dans l'environnement** : dans la plupart des cas l'environnement est considéré comme un espace métrique. En fait, un agent situé est un agent capable d'agir sur son environnement.
- **La réactivité** : un agent réactif est un agent qui perçoit son environnement et répond dans un temps requis.
- **La proactivité** : un agent proactif est un agent capable de prendre l'initiative au bon moment.
- **La sociabilité** : un agent doit être capable d'interagir avec les autres agents en raison d'accomplir les objectifs et les buts attendus.

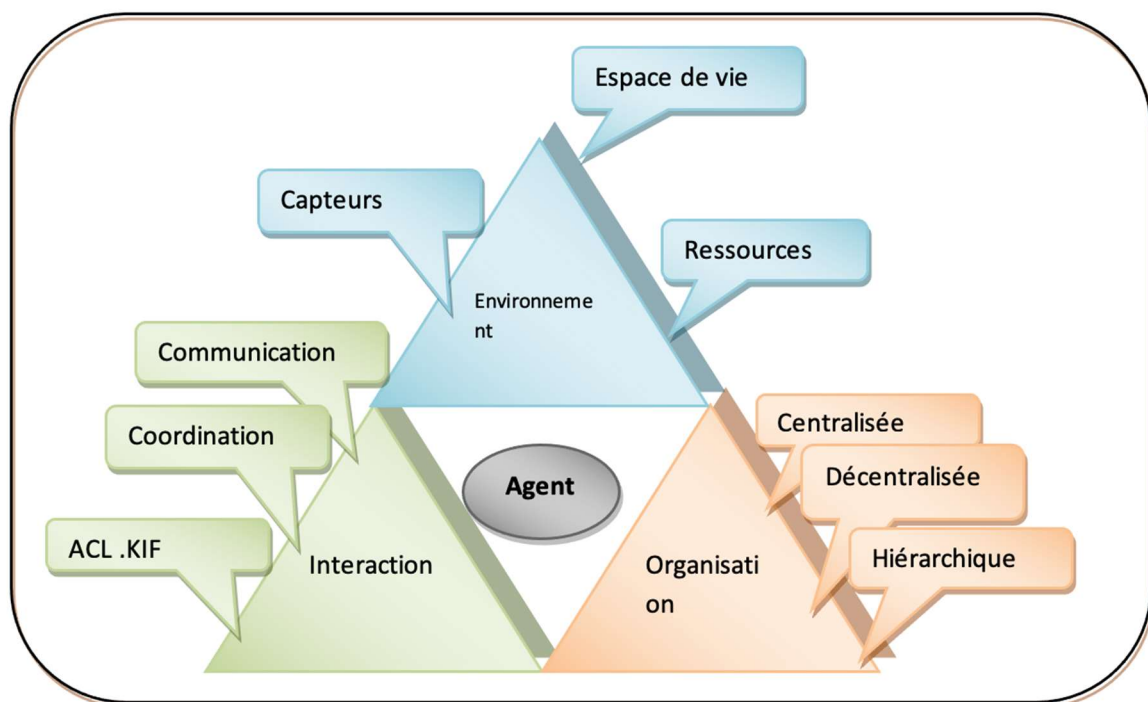


Figure 1: Les trois dimensions d'un agent <sup>6</sup>.

## 2.2- Environnement :

En générale, l'environnement est ainsi :

- ❖ Un médium d'interaction : signaux, traces, .... Avec des lois physiques ou non.
- ❖ Un lieu ou des actions individuelles ou collectives sont réalisées, ou des réactions sont perçues.
- ❖ Un espace de déplacement : grilles, positions des agents.
- ❖ Un moyen de structuration des agents : relations de proximité, définition de topologies spatiales et temporelles.
- ❖ Une source de données pour le système.
- ❖ Un lieu des ressources disponibles.

## 2.3- Interaction :

Pour qu'un agent puisse atteindre ses objectifs et les réaliser, il doit interagir avec d'autres agents. L'interaction est donc un concept important dans les systèmes multi-agents.

Toutefois, ce concept soulève des questions pratiques quant à sa réalisation. Il s'agit de savoir comment faire interagir, communiquer et coopérer les agents entre eux. En fait, ce concept est devenu un axe de recherche indépendant dans les systèmes multi-agents.

*« Une interaction est une mise en relation dynamique de deux ou plusieurs agents par le biais d'un ensemble d'actions réciproques. Les interactions s'expriment ainsi à partir d'une série d'actions dont les conséquences exercent en retour une influence sur le comportement futur des agents ».*<sup>2</sup> Les agents s'interagissent entre eux pour se coopérer, se coordonner, et se collaborer. La classification de Ferrière définit les différentes formes d'interaction entre les agents. Cette classification est basée sur trois critères : la compatibilité des buts, la suffisance des ressources et la suffisance des compétences.

Buts	Ressources	Compétences	Types de situation	Remarque
Compatibles	Suffisantes	Suffisantes	Indépendance	Simple situation de coopérations
		Insuffisantes	Collaboration	
	Insuffisantes	Suffisantes	Encombrement	
		Insuffisantes	Coordonnée	
Incompatibles	Suffisantes	Suffisantes	Compétition individuelle pure	Situations d'antagonismes
		Insuffisantes	Collective pure	
	Insuffisantes	Suffisantes	Conflits individuels pour des ressources	
		Insuffisantes	Conflits collectifs pour des ressources	

Tableau 1 : Classification des situations d'interactions <sup>7</sup>.

## 2.4- l'organisation

Une organisation est définie, par Morin <sup>8</sup> , comme: « *Un agencement de relations entre composants ou individus qui produit une unité, ou système, dotée de qualités inconnues au niveau des composants ou individus. L'organisation lie de façon interrelationnelle des éléments ou évènements ou individus divers qui d'es lors deviennent les composants d'un tout. Elle assure solidarité et solidité relative, donc assure au système une certaine possibilité de durée en dépit de perturbations aléatoires.* ».

Selon Fox <sup>9</sup> , une organisation peut être définie comme une structure décrivant la manière dont les membres d'une organisation sont en relation les uns avec les autres et interagissent pour atteindre un objectif commun - les résultats de l'organisation. La structure d'une organisation est la fonction de l'environnement dans lequel elle opère, des ressources disponibles pour produire ses résultats et de la nature de ces derniers. Cette structure peut être statique, c'est-à-dire conçue a priori par le programmeur du système, ou dynamique, comme c'est le cas dans les SMA ouverts. En d'autres termes, le concepteur du système doit choisir a priori cette structure de manière appropriée afin de contrôler la complexité du système. La complexité peut être considérée comme la quantité d'informations que le système peut traiter<sup>10</sup> .

## 3- Les systèmes normatifs

Les systèmes normatifs sont définis comme des systèmes où les normes jouent un rôle dans le comportement, ces systèmes nécessitent des concepts normatifs afin d'être décrits ou spécifiés <sup>11</sup> .

## 4- Les systèmes multi-agents normatifs (NorMAS)

### 4.1- La notion de «NorMAS»

L'idée de réguler le système multi-agents normatifs s'inspire des principes de base des systèmes normatifs dans le domaine des systèmes multi-agents. Elle a comme normes dans une certaine mesure. En termes de société et d'organisation. Standardiser les normes du système multi-agent comme un outil de contrôle et de régulation <sup>12</sup>.

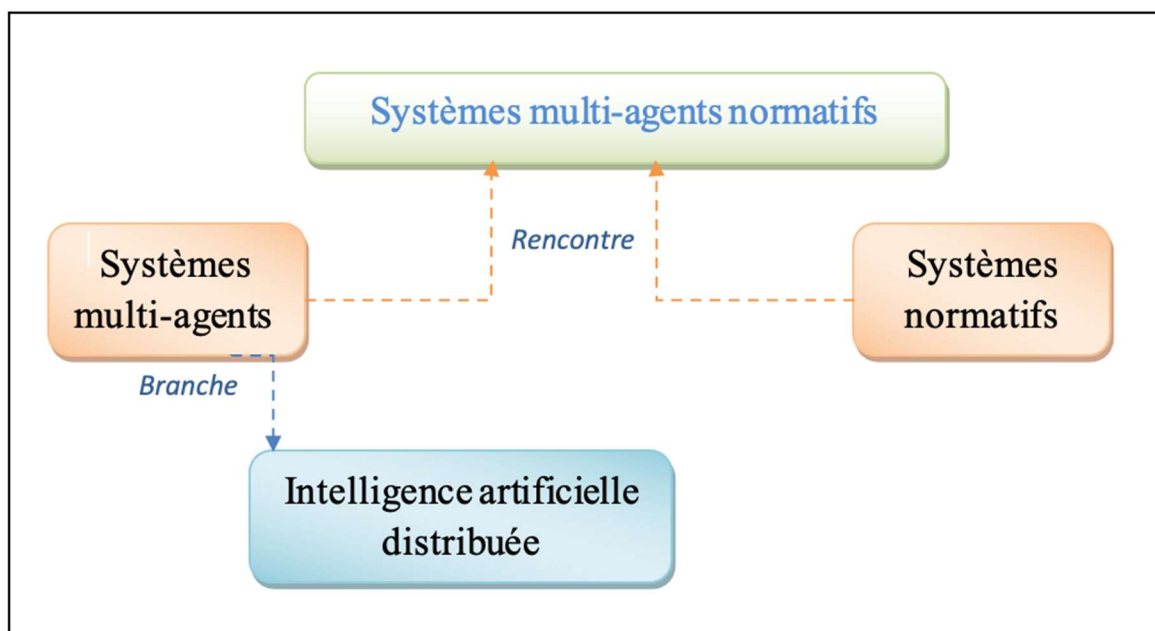


Figure 2: Origine des systèmes multi-agents normatifs (NorMAS) <sup>13</sup>.

## 4.2- La présentation des normes dans NorMAS :

### 4.2.1- La notion :

Les normes représentent les concepts de différentes disciplines, telles que: sociologie, psychologie, philosophie, droit, économie, biologie et informatique et en représentent le concept de base des systèmes normatifs. Ainsi, le dictionnaire en ligne Larousse propose une définition du terme norme comme : « une règle, principe ou critère se réfère tout jugement. »<sup>14</sup>. Elle présente une règle fixant les conditions de réalisation d'une opération ou de l'exécution d'un objet.

### 4.2.2- Classification des normes :

Dans la littérature des systèmes multi-agents normatifs, plusieurs termes ont été utilisés pour présenter le concept de normes, qui sont : les conventions, la norme sociale et la loi sociale<sup>15</sup>. Défini deux grandes catégories de normes : les conventions et les normes essentielles.

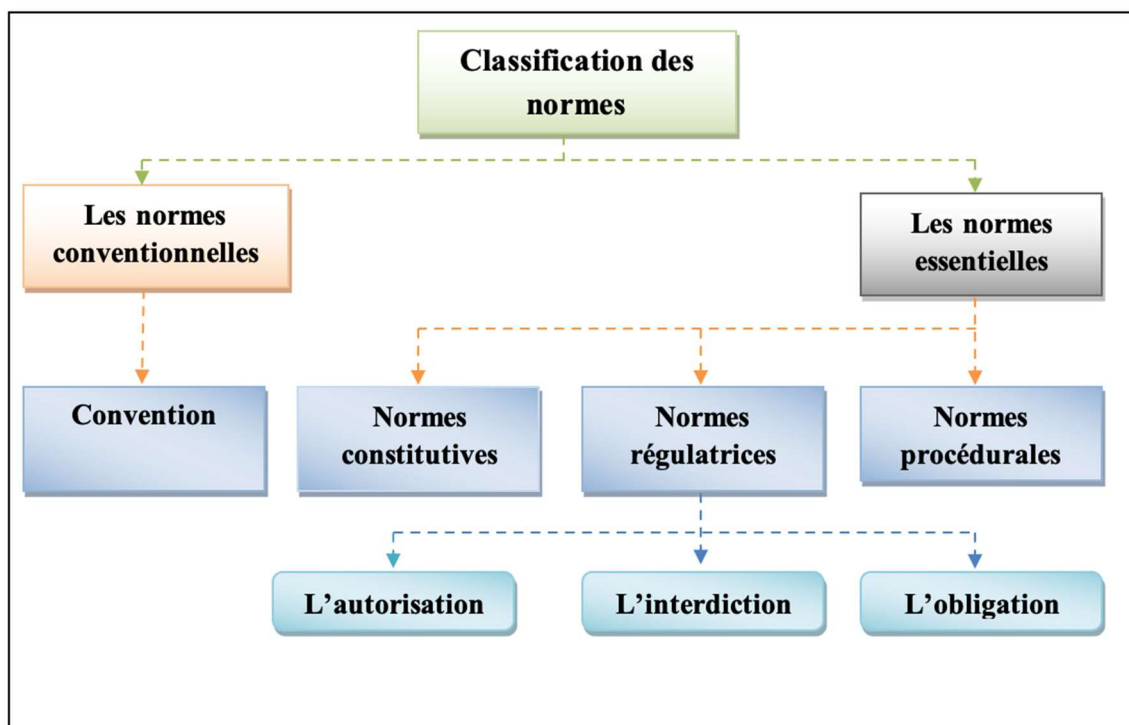


Figure 3 : Classification des normes<sup>16</sup>.

## I. Conventions :

Les conventions sont des normes naturelles qui surgissent sans aucune application <sup>15</sup>. Les conventions traitent des problèmes de coordination lorsqu'il n'y a pas de conflit entre les intérêts individuels et collectifs, par exemple, chacun se conforme au comportement souhaité <sup>15</sup>. Young <sup>17</sup> définit la convention comme « un modèle de comportement coutumier, attendu et auto-exécutoire. Tout le monde se conforme, tout le monde s'attend à ce que les autres se conforment, et tout le monde veut se conformer étant donné que tout le monde se conforme. Les conventions fixent une norme parmi un ensemble de normes qui est toujours efficace tant que chacun dans la communauté emploie la même norme, c'est-à-dire les salutations, le côté conducteur de la route »<sup>15</sup>.

## II. Normes essentielles :

Les normes fondamentales peuvent résoudre ou atténuer les problèmes d'action collective lorsque les intérêts individuels et collectifs entrent en conflit<sup>18</sup>. Par exemple, « la norme de ne pas polluer les rues urbaines est essentielle en ce qu'elle impose aux individus de transporter leurs déchets, plutôt que de les jeter sur place, un acte qui profite à tous »<sup>19</sup>.

### a) Normes procédurales :

La norme de procédure est divisée en norme objective et norme subjective. Les normes procédurales objectives représentent des règles qui expriment comment les décisions sont réellement prises dans un système canonique, tandis que les normes procédurales subjectives représentent les outils que les individus utilisent pour travailler dans le système, tels que les programmes d'arrière-plan <sup>20</sup>.

### b) Normes constitutives :

L'affirmation de normes constitutives conduit à de nouvelles normes ou états de choses cibles, telles que les règles d'un jeu comme les échecs <sup>21</sup>.

**Boella et van der Torre** observent un certain nombre de caractéristiques qui constituent des normes sexuelles ; l'une d'entre elles est un concept intermédiaire illustré par une affirmation telle que « ceci est un président de cérémonie de mariage » ou « ce bout de papier compte comme un billet de cinq euros »<sup>22</sup>. Deux autres caractéristiques sont les normes organisationnelles et structurelles, faisant référence à la manière dont les rôles

définissent le pouvoir et la responsabilité, et à la manière dont les hiérarchies structurent les groupes et les individus<sup>23</sup>. Les normes sont introduites à la fois par des agents, qui jouent un rôle législatif, et par des agents ordinaires, qui créent de nouvelles obligations, interdictions et licences pour des agents spécifiques<sup>24</sup>. **Boella** et **van der Torre** <sup>22</sup> ont différencié les normes régulatrices et les normes constitutives avec un exemple ; par exemple, si la norme réglementaire stipule que les véhicules sont interdits dans le parc, alors la norme constitutive est que « les vélos sont également comptés comme des véhicules dans le parc ».

c) **Normes Réglementaires :**

Les normes réglementaires visent à réglementer les activités en imposant une obligation ou une interdiction d'accomplir une action<sup>21</sup>. Comme l'a commenté Peczenik<sup>21</sup>, une norme régulatrice qualifie une action ou un état de fait comme prescrit, permis ou interdit. Parce qu'une norme réglementaire qualifie une action, elle peut être traitée comme une norme de conduite, par exemple, la responsabilité de déposer un rapport de police dans un délai raisonnable en cas de découverte d'objets perdus ou volés. Une norme de conduite peut prescrire une punition ou une sanction à une personne qui viole une norme. On peut ainsi faire la distinction entre une norme sanctionnée et une norme sanctionnante. Peczenik <sup>21</sup> ont également décrit des normes morales qui servent des normes d'objectif, par exemple, la garantie que " tout le monde devrait avoir un niveau de vie décent ". Dans d'autres récits, les normes régulatrices produisent une régulation d'une situation problématique antérieure en fixant des règles de comportement des acteurs, qui représentent des obligations et des interdits<sup>25</sup>, par exemple la règle selon laquelle une personne « doit conduire sur la voie de droite »<sup>21</sup>.

➤ **La norme d'obligation :**

Sont des critères qui peuvent entraîner une récompense ou une punition ; c'est-à-dire que si l'agent applique la norme, il évite la sanction, mais s'il ne le fait pas, il est puni <sup>26</sup>. De la définition, nous pouvons déduire que lorsque l'agent agit sur l'événement, il évite la punition, et lorsque l'agent n'agit pas sur l'événement, il est puni.

➤ **Les normes d'interdiction :**

Sont des normes qui peuvent aussi conduire à la punition, mais dans le sens d'une norme de devoir rejetée ; c'est-à-dire que si l'agent n'exerce pas la norme, il évite la punition, mais s'il le fait, il est puni<sup>26</sup>. Intuitivement, à partir de la définition, nous pouvons également déduire que lorsque l'agent agit sur l'événement, il est puni, et lorsque l'agent n'agit pas sur l'événement, il évite la punition.

➤ **La norme de permission :**

Fait partie de la norme d'obligation, mais exempte l'agent de certaines actions obligatoires dans certaines circonstances <sup>27</sup>.

#### 4.2.3- Cycle de vie de la norme :

La littérature sur les normes sociales montre qu'il n'existe pas de vision unifiée des mécanismes par lesquels les normes sont produites et diffusées au sein d'une société ou d'un groupe social.

La norme a traversé plusieurs événements depuis son apparition jusqu'à sa disparition. Ces événements constituent le cycle de vie des normes .Il se compose principalement des étapes suivantes <sup>28</sup> :

**1. Création de normes :**

Dans les systèmes de normes, la création de normes est le mécanisme par lequel les agents d'une société apprennent ce que sont les normes sociales.

**2. Diffusion de normes :**

La diffusion d'une norme dans un groupe, afin que tous les membres du groupe connaissent ce processus, ce processus est appelé diffusion standard.

**3. Application de la norme :**

Cette étape peut sanctionner la personne qui enfreint la norme ou en recommander d'autres qui respectent la norme.

**4. Émergence des normes :**

Cette étape est définie comme l'atteinte d'un seuil important dans le degré de diffusion de normes, si plus de X% de la population suit

une norme alors on dit que cette société a une norme, la valeur X varie d'une société à une autre, et aussi d'un type à l'autre

### 4.3- Implémentation des normes dans NorMAS

Dans les systèmes normatifs, les normes utilisées dans les systèmes à base d'agents doivent être présentées de manière à pouvoir être traitées par des agents logiciels <sup>29</sup>. Selon Savarimuthu <sup>30</sup>, les chercheurs ont représenté les spécifications dans des structures de données explicites et implicites. Hollander et Wu <sup>29</sup> ont mentionné quatre principaux schémas de représentation utilisés dans les recherches récentes, à savoir la logique déontique, les systèmes basés sur des règles, les chaînes binaires et la théorie des jeux.

#### 1. La logique déontique :

Se développe à partir de la logique modale, une version étendue de la logique formelle classique qui traite du "nécessaire" et du "possible". D'autre part, la logique déontique traite des obligations, des interdictions et des permissions <sup>31</sup>.

#### 2. Un système à base de règles :

Est un ensemble de paires condition/action codées avec un moteur d'inférence. Il est souvent utilisé par les systèmes qui exploitent la conception hors ligne, où les critères sont implicitement exprimés dans le système de décision de l'agent <sup>32</sup>.

#### 3. Une chaîne binaire :

Est une séquence numérique de 1 et de 0, où le chiffre 1 indique l'occurrence et le chiffre 0 indique l'absence de norme. Ce format est couramment utilisé dans les études de population pour tester la propagation et l'émergence de normes <sup>33</sup>.

#### 4. Dans la théorie des jeux :

Chaque agent est capable de faire un choix simple, étant donné le gain correspondant, et à chaque tour tente de maximiser son gain en agissant selon ses attentes vis-à-vis de l'adversaire <sup>34</sup>.

## 5- Conclusion

Dans ce chapitre, nous avons présenté d'une manière générale les systèmes multi agents normatifs qui représentent l'intersection entre les systèmes multi agents et les systèmes normatifs. Ainsi, nous avons commencé par la présentation des systèmes multi-agents avant de présenter les systèmes normatifs. Le chapitre suivant est consacré à la présentation des organisations multi-agents.

# **Chapitre 02 :**

# **Les Organisations**

# **Multi-Agents.**

## Introduction

Les organisations multi-agent sont un domaine de recherche émergent qui explore les interactions complexes et dynamiques entre plusieurs agents autonomes dans un environnement commun. Ces agents peuvent être des individus, des robots, des logiciels ou même des systèmes physiques, et ils interagissent entre eux pour atteindre des objectifs spécifiques. Les OMA offrent une approche novatrice pour modéliser, analyser et concevoir des systèmes où l'interaction et la coopération entre les agents jouent un rôle central.

## 2- L'organisation multi-agent

### 2.1 C'est quoi une organisation

Le concept l'organisation multi-agents est utilisé dans de nombreux domaines de la recherche scientifique, en particulier dans le domaine de l'intelligence artificielle et d'un système multi-agents, car c'est l'un des principaux éléments d'un système multi-agents (SMA) afin de gérer les gestions des systèmes complexes composés d'agents et d'écrire leur approche de modélisation.

### 2.2 Définition d'organisation :

*« Une organisation fournit un cadre pour l'activité et l'interaction à travers la définition des rôles, des attentes comportementales et des relations d'autorité. »<sup>1</sup>*

## 3- L'organisation structurelles et concrète

Les structures organisationnelles et les organisations concrètes sont des éléments essentiels dans les systèmes multi-agents. Elles permettent de définir la manière dont les agents interagissent. Dans cette partie nous expliquerons les organisations structurelles et les organisations concrète.

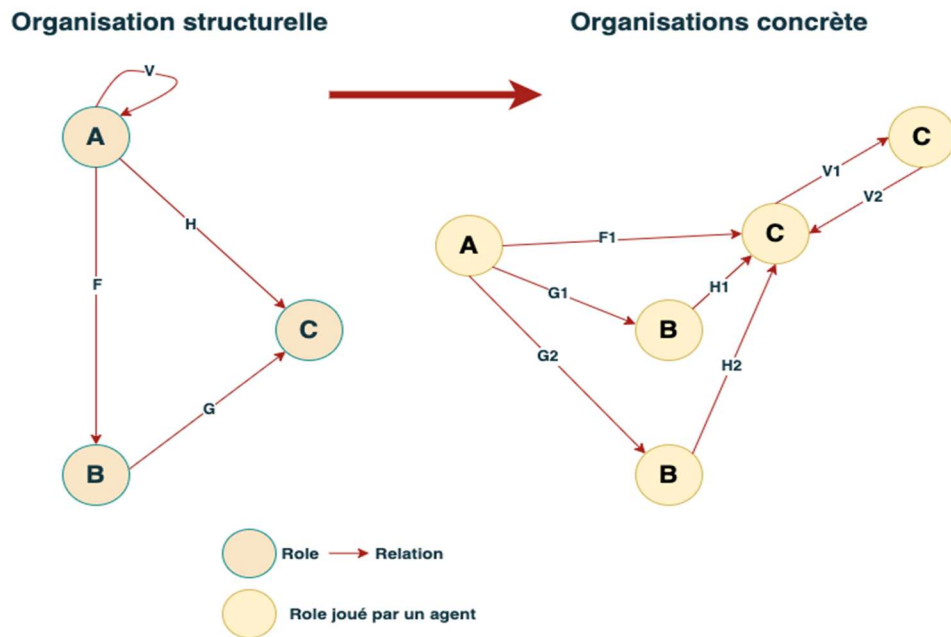


Figure 4 : l'organisation structurelle et l'organisation concrète

- Note : (G1, G2, H1, H2, ...) : ensembles des liens entre les entités concrètes correspondent aux liens abstraits qui existent entre les entités abstraites.

### 3.1- Les structures organisationnelles

La structure organisationnelle représente les caractéristiques abstraites d'une organisation, Elle est composée d'un ensemble de classes d'agents dotés de leurs rôles avec les relations abstraites existantes entre ces rôles, Certains des avantages des organisations structurelles sont <sup>2</sup>:

- Formalisée et structurée.
- Ensemble de relations abstraites existant entre ces rôles.
- Un ensemble de classes d'agents caractérisées par les rôles affectés aux agents
- Il se concentre sur les relations différentes d'agents d'une organisation.
- Il définit les relations entre les individus au sein d'une organisation

### 3.2- Les structures concrètes

Représente les instances de l'organisation structurelle, Certains des avantages des organisation concrètes <sup>2</sup>:

- L'organisation concrète fait référence à la manière dont les activités réelles sont menées au sein d'une organisation.
- Concentre sur les aspects pratiques et opérationnels de la réalisation des tâches et des objectifs.
- Inclure des éléments tels que les processus de travail, le flux d'informations, la communication, et la coordination.
- Il s'agit de la mise en œuvre effective des plans et structures organisationnels.

### 4- Dimensions des organisations

Les organisations sont des entités complexes qui impliquent une combinaison de structures, de fonctions et de paramètres pour atteindre des objectifs spécifiques, La compréhension de ces dimensions est essentielle pour concevoir et gérer efficacement les organisations le tableau sous dessous représente la différence pour chaque dimension :

<b>Dimension</b>	<b>Description</b>
<b>La dimension organisationnelle physique</b>	C'est la partie responsable de la mise la mise en œuvre de l'infrastructure et de la mise en œuvre des mécanismes de contrôle avec leur plantation et leur gestion dans le cadre de ses activités organisées <sup>2</sup> .
<b>La dimension organisationnelle sociale</b>	Cela dépend de la définition des tâches et des rôles qui permettent aux organisations de bas niveau d'effectuer des tâches utiles et de haut niveau, car elles sont basées sur l'organisation sociale pour planifier les tâches sociales <sup>2</sup> .
<b>La dimension organisationnelle relationnelle</b>	Elle travaille sur la planification des relations entre les différentes interactions, la coordination des procédures et la communication les uns avec les autres, car il dépend de la caractéristique de la coopération sur un traille spécifique et poursuit ce processus <sup>2</sup> .
<b>La dimension organisationnelle environnementale</b>	Cette partie travaille à contrôler le contrôle de la mise en œuvre des actions dans certains environnements, Certains considèrent les problèmes d'interaction <sup>2</sup> .

<b>La dimension organisationnelle personnelle</b>	Elle travaille sur les problèmes de méta-organisation de l'agents, c'est-à-dire le contrôle des caractéristiques internes de l'agent et la gestion de sa structure interne <sup>2</sup> .
---	---

Tableau 2: Représentation la dimension des organisations <sup>2</sup>

#### 4.1- Structure organisationnelle

La structure organisationnelle représente la manière dont les tâches (les relations, les composant) qui sont organisées au sein d'une organisation, la structure organisationnelle contient deux tâches essentielles qui sont <sup>2</sup>:

- Les composant (les individus) : cette tâche décrits par le type (individuel, composite, récursivité), et le rôle (coordinateur, exécutant, expert).
- Les relations : cette tâche représente la relation entre les composants (autorité, communication).

#### 4.2- Fonctions dans des organisations

Les fonctions organisationnelles sont les différentes activités ou domaines au sein d'une organisation chaque fonction à des objectifs spécifiques et contribuent à la réalisation des objectifs, le tableau sous dessous représente la différence pour chaque fonction :

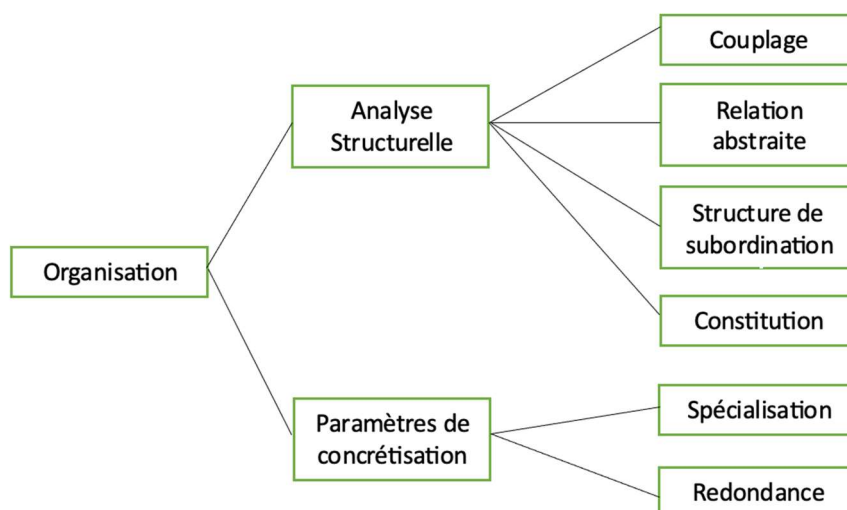
Fonction	Description
<b>La fonction représentationnelle</b>	Les fonctions de représentation comprennent toutes les fonctions qui modélisent l'environnement et les autres organisations, ainsi que la mémoire des événements pouvant affecter l'organisation. Cette fonction est particulièrement développée dans l'organisation des individus cognitifs (c'est-à-dire les agents cognitifs), qui ont des représentations internes <sup>2</sup> .
<b>La fonction organisationnelle</b>	Les fonctions organisationnelles impliquent tout ce qui concerne la gestion des activités organisationnelles, notamment la planification, l'attribution et le suivi des tâches, la coordination des actions et la gestion des engagements <sup>2</sup> .
<b>La fonction conative</b>	Cette fonction elle-même est divisée en trois sous-fonctions : décrire les motivations de la formation des buts ou des dispositions, qu'il s'agisse de besoins internes ou externes, élaborer des contraintes, décrire les normes de limites et inhiber les actions possibles de l'organisation à la suite de la décision. Faire des fonctions associées à la portée du système organisationnel et la portée de la décision finale <sup>2</sup> .

<b>La fonction interactionnelle</b>	Cette fonctionnalité relie une organisation à son environnement. Elle est responsable de la gestion des interactions avec son environnement et de toutes les communications avec les autres organisations. Par conséquent il fournit toutes les fonctions d'entrée, de sortie et d'interface. Il se compose de deux sous-fonctions la fonction de perception qui est chargée d'obtenir des informations du monde extérieur, et la fonction exécutive qui est chargée d'effectuer des actions sélectionnées par le système conatif et exécutées par le système organisationnel <sup>2</sup> .
<b>La fonction productive</b>	Cette fonction concerne d'ensemble des activités primitives qui doivent être mises en œuvre pour résoudre un problème ou effectuer un travail <sup>2</sup> .
<b>La fonction végétative</b>	Cette fonction traite de la conservation de la structure et des agents, de l'acquisition et de l'entretien des ressources et des tâches employées pour maintenir et reproduire d'organisation, mais aussi de l'intégration de nouveaux agents <sup>2</sup> .

**Tableau 3: Représentation schématique des différentes fonctions d'une organisation <sup>2</sup>**

### 4.3- Paramètres de concrétisation

C'est le passage d'une structure organisationnelle à une organisation concrète, C'est-à-dire c'est la réalisation effective d'une organisation multi-agent <sup>2</sup>.



**Figure 5 : paramètre de concrétisation et analyse structurelle d'organisation <sup>2</sup>**

Après avoir choisi la fonction et la structure de l'organisation, les concepteurs de la MAS sont confrontés à une question connexe : comment attribuer toutes les compétences aux différents agents pour que le travail soit réellement fait ?

La réponse à cette question concerne le paramètre de réification. Autrement dit, des paramètres qu'il est permis de passer de l'organisation structurelle à l'organisation concrète. Notez qu'un attribut important sur lequel repose la réponse à cette question est le niveau de spécialisation de l'agent. Par conséquent, il est possible d'attribuer chaque rôle à un agent spécifique ou d'attribuer plusieurs rôles au même agent. Bien entendu, en fonction des caractéristiques de l'application et des besoins de l'utilisateur, le choix est discutable <sup>2</sup>.

Pour masquer la complexité des problèmes liés à l'organisation des concepteurs de SMA, les chercheurs du domaine proposent un ensemble de modèles qui fournissent des représentations abstraites pour les organisations multi-agents.

## **5- Modèles organisationnels**

Les domaines des modèles organisationnels multi-agents jouent un rôle clé dans la gestion des systèmes multi-agents car ils définissent les rôles, les interactions, les prototypes et la communication entre les agents. Cela est dû aux mécanismes de coordination entre ces agents indépendants. Ces modèles sont utilisés dans des systèmes complexes pour coordonner, réaliser et assurer le fonctionnement optimal du système. Voici quelques-uns des modèles organisationnels AGR, MOISE, JASON

### **5.1- AGR**

Le modèle AGR (Agent, Groupe, Rôle), Il s'agit d'un cadre de base dans un système multi-agents développé pour analyser les interactions au sein des groupes et gérer un système multi-agents. Ce dernier facilite la compréhension des interactions et des comportements des individus qui se trouvent dans des groupes de systèmes complexes, et se concentre sur les rôles et les effets joués par les agents, le figure ci-dessous représentant le modèle AGR :

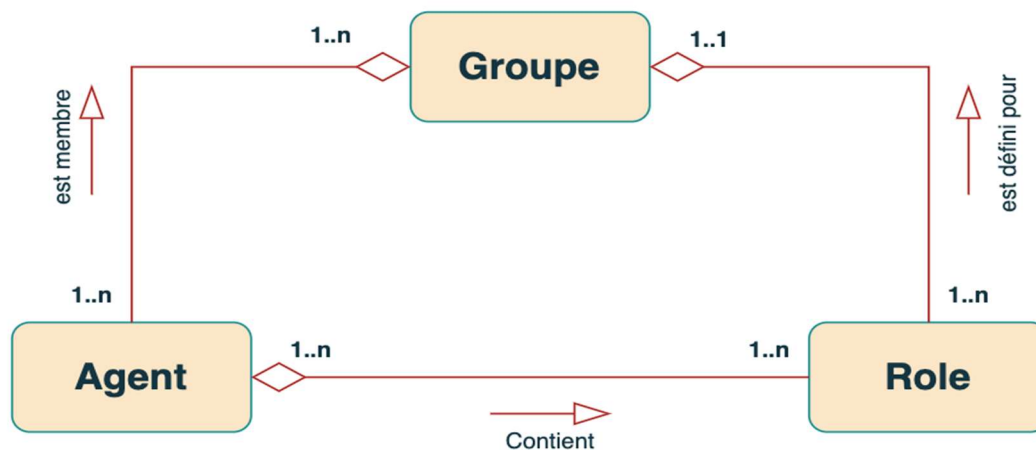


Figure 6 : le modèle Agent/Groupe/Rôle (AGR)

- Les trois concepts de base sont de modèle AGR :

### 5.1.1- Agent

Un agent est une entité active et communicante jouant des rôles au sein des groupes. Un agent peut jouer plusieurs rôles et faire partie de plusieurs groupes. Une caractéristique importante du modèle AGR est qu'aucune contrainte n'est imposée à l'architecture d'un agent ou à ses capacités mentales. Ainsi, un agent peut être aussi réactif, ou aussi intelligent <sup>2</sup>.

### 5.1.2- Rôle

Le rôle est la représentation abstraite d'une position fonctionnelle d'un agent dans un groupe. Un agent doit jouer un rôle dans un groupe, mais un agent peut jouer plusieurs rôles. Les rôles sont locaux aux groupes, et un rôle doit être demandé par un agent. Un rôle peut être joué par plusieurs agents <sup>4</sup>.

### 5.1.3- Groupe

Un groupe est un ensemble d'agents partageant certaines caractéristiques communes. Un groupe est utilisé comme contexte pour un modèle d'activités et pour les organisations de partitionnement. Deux agents peuvent communiquer si et seulement s'ils appartiennent au même groupe, mais un agent peut appartenir à plusieurs groupes. Cette caractéristique nous permettra de définir les structures organisationnelles <sup>4</sup>.

## 5.2- Moise :

Moise (Model of Organization for multi-agent SystEms) est une plateforme organisationnelle pour les systèmes multi agent basée sur des notions comme les rôles, les groupes et liens organisationnels. Il permet à un MAS d'avoir une spécification explicite de son organisation. Cette spécification doit être utilisée à la fois par les agents pour raisonner au sujet de leur organisation et par une plateforme d'organisation qui exige que les agents suivent la spécification [GITHUB].

### 5.2.1- les principes de Moise

Le modèle d'organisation MOISE repose sur le principe de décomposition, notamment en représentant l'organisation en termes d'entités organisationnelles (agents, groupes, rôles) et leurs interactions ce principe permet de décomposer un système complexe en entités plus simples pour la modélisation et la compréhension des relations organisationnelles, Ce Modèle sont basée sur trois des spécifications qui sont la spécifications structurelles, la spécification fonctionnelle et la spécification déontique <sup>5</sup>.

### 5.2.2- Les spécifications de MOISE

Le tableau sous-dessous représente la différence entre ces spécifications :

La spécification	Description
<b>La spécification structurelle</b>	Cette spécification définit les relations entre les agents à travers les concepts de groupes, de rôles et de liens, Les groupes sont composés de sous-groupes et de rôles. Un ensemble de rôles est un ensemble de liens qui définissent les relations entre les rôles [8].
<b>La spécification fonctionnelle</b>	Cette spécification exprime clairement comment le MAS atteint ses objectifs généraux [9].
<b>La spécification déontique</b>	Cette spécification décrit les permissions et les obligations dans une organisation [9].

Tableau 4: Représentation de la spécification de Moise <sup>5</sup>

### 5.3- Jason

Jason est un interpréteur Open Source pour une version étendue d'AgentSpeak un langage de programmation orienté agent et basé sur la logique écrit en Java. Il permet aux utilisateurs de construire des systèmes multi-agents complexes capables de fonctionner dans des environnements précédemment considérés comme trop imprévisibles pour les ordinateurs, Jason est facilement personnalisable et adapté à la mise en œuvre de systèmes de planification réactive selon l'architecture Croyance-Désir-Intention (BDI) <sup>6</sup>.

#### 5.3.1- AgentSpeak

AgentSpeak est un langage de programmation orienté agent. Il est basé sur la programmation logique et l'architecture du modèle logiciel de croyance-désir-intention (BDI) pour les agents autonomes. La langue était initialement appelée AgentSpeak(L), mais est devenue plus populaire comme AgentSpeak, un terme qui est également utilisé pour se référer aux variantes de la langue originale <sup>7</sup>.

#### 5.3.2- les avantage de Les avantage de Jason

- Évolutivité : construire les grands systèmes multi-agents complexes.
- Flexibilité : flexible pour modéliser une grande variété de systèmes multi-agents.
- Réactivité : réagir aux changements dans leur environnement en temps opportun.
- Proactivité : prendre des mesures proactives pour atteindre leurs objectifs.
- Collaboration : la collaborer pour atteindre des objectifs communs.

## 6- Programmation des organisations multi-agents

La programmation des organisations multi-agents est un domaine de développement qui se concentre sur la conception et la mise en œuvre de systèmes composés d'agents ces systèmes sont utilisés dans de nombreux domaines comme la gestion des ressources, les systèmes de transport...etc. Dans cette section nous explorerons le MADKit et JeCaMo

### 6.1- MADKit

La programmation des organisations dans les SMA est une discipline qui vise la modélisation et la simulation des environnements où des agents autonomes interagissent de manière organisée pour atteindre des objectifs spécifiques. Le MADKit (Multi-Agent Development Kit) est un cadre de développement logiciel qui offre des fonctionnalités avancées pour la programmation des organisations dans les systèmes multi-agents. Nous explorerons les caractéristiques, son architecture de MADKit, Et ses principes.

#### 6.1.1- Définition

MADKit est un ensemble de paquets Java qui inclut un noyau agent, ainsi que différentes bibliothèques de messages. L'utilisation du langage Java a été sélectionnée dès les premières étapes en raison de sa portabilité et de la diversité de ses bibliothèques de base. De plus, Java offre des fonctionnalités avancées telles que la sérialisation d'objets et la réflexivité, ce qui renforce les capacités de MADKit <sup>8</sup>.

- Le MADKit est basé sur principes sont :
  1. Architecture à micro noyau.
  2. Agentification systématique des services.
  3. Découplage applicatif entre noyau (l'application d'accueil et l'agent)

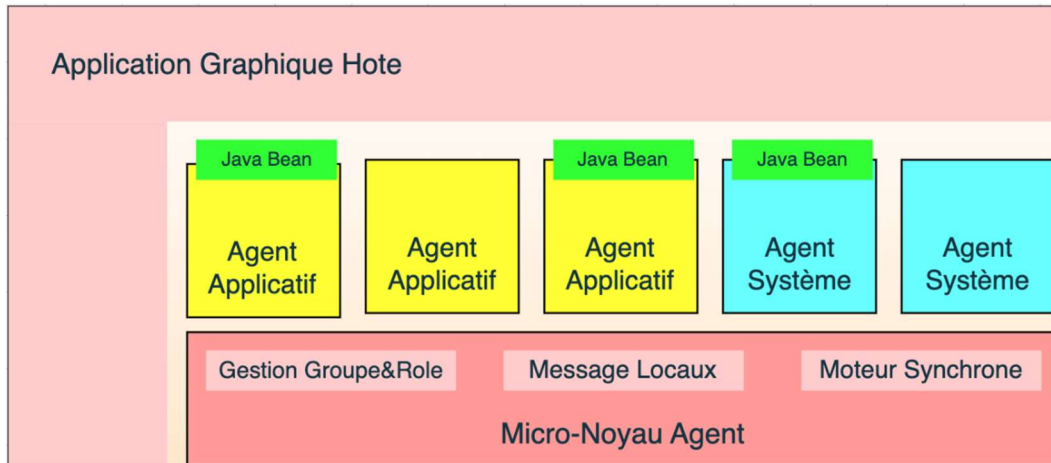


Figure 7 : Architecture générale de MADKit <sup>8</sup>.

### 6.1.2- Architecture de MADKit :

MADKit est construit sur une architecture multi-niveaux qui permet de modéliser les organisations, les agents et les environnements dans un système multi-agent. Il fournit des primitives pour la création, la communication, la coordination et la gestion des agents autonomes. L'architecture modulaire de MADKit permet aux développeurs de concevoir des systèmes complexes et adaptatifs en utilisant des modules réutilisables. L'idée fondamentale de MADKit est d'utiliser la plateforme elle-même autant que possible pour son propre fonctionnement. Tous les services, à l'exception du micro-noyau, sont implémentés sous forme d'agents autonomes. Cette approche vise à maintenir l'uniformité du modèle et à tester les SMA en tant que modèle de programmation. Cela le rapproche des approches méta en architectures objets, bien que nous situons la relation méta entre les agents d'application et les "systèmes" tels que la communication, l'ordonnancement et la surveillance, MADKit se distingue des plateformes d'agents conventionnelles qui se concentrent sur un modèle d'agent ou d'interaction particulier. La taille compacte et les fonctionnalités de base du noyau, ainsi que la neutralité des types d'agents, associées au principe des services agatifiés et du découplage par rapport aux applications "hôtes", permettent en réalité d'obtenir une gamme variée d'environnements d'exécution répondant parfois à des objectifs totalement opposés <sup>9</sup>

### 6.1.3- Fonctionnalités clés de MADKit

MADKit offre un large éventail de fonctionnalités pour la programmation des organisations. Il permet la spécification des rôles, des tâches, des normes et des protocoles de communication au sein de l'organisation. MADKit permet aux développeurs de concevoir des systèmes multi-agents flexibles et évolutifs <sup>10</sup>.

- Voici quelques-unes des fonctionnalités principales de MADKit <sup>10</sup>:
  - Création et gestion des agents.
  - Communication entre les agents.
  - Organisation des agents.
  - Gestion des environnements distribués.
  - Modèles et bibliothèques.
  - Flexibilité et extensibilité.

MADKit est un cadre pour le travail, la création et la gestion de systèmes multi-agents à travers une gamme d'outils, de bibliothèques et de modèles, basés sur sa structure sur les agents, plate-forme de mise en œuvre, organisations organisationnelles, modèles de conception et mécanismes de communication, MADKit fournit une infrastructure complète pour le développement d'applications où il fournit la gestion des agents, la coordination et la structure organisationnelle. Cela contribue à la conception de systèmes robustes, flexibles et distribués qui permettent aux développeurs de logiciels de créer des applications conçues pour plusieurs domaines différents.

## 6.2- JeCaMo

### 6.2.1- Définition

JeCaMo est un cadre efficace dans la programmation multi-agents, Il fournit un haut niveau d'abstraction, car il se compose de trois technologies différents (Jason, Cartago, Moise), où chacune des technologies mentionnées se concentre sur <sup>11</sup>:

- Jason : C'est un langage de programmation pour les agents indépendamment (autonomes).
- Moise : Est un langage pour les organisations multi-agents,
- Cartago : Est un langage de programmation pour les artefacts de l'environnement.

### 6.2.2- Les avantages de JeCaMo

JeCaMo offre de nombreux avantages dont nous mentionnons :

- Évolutivité : Les systèmes JeCaMo peuvent être adapté à un grand nombre d'agents. Car les agents sont légers et peuvent être créés et détruits dynamiquement.
- Flexibilité : les systèmes JeCaMo peuvent être adaptés à différents types de systèmes multi-agents. Il s'agit d'une référence à ce qu'il offre à partir de différents types d'environnements et d'agents.
- Réutilisabilité : Les systèmes JeCaMo fournit un certain nombre de composants réutilisables qui peuvent être utilisés pour construire de nouveaux types de systèmes multi-agents.
- Tolérance aux pannes : Les systèmes JeCaMo fournit une fonction de tolérance aux erreurs qui signifie la continuité du travail des agents en cas de défaillance entre ces agents

La programmation organisationnelle utilisant les systèmes JeCaMo fournit un cadre puissant, efficace et adaptable pour la conception et le développement de systèmes multi-agents. Cela est dû aux nombreuses fonctionnalités offertes pour organiser, coordonner et gérer les connaissances partagées. JeCaMo offre une solution à des

problèmes complexes dans divers domaines d'applications tels que le e-commerce, la robotique et les systèmes autonomes et Transport intelligent, permettant ainsi de relever les défis et de trouver des solutions appropriées.

## **7- Conclusion**

Les organisations dans les SMA représentent un moyen puissant pour coordonner les interactions entre les agents et contrôler leurs comportements. Dans ce contexte, plusieurs concepts ont été proposés dans le cadre des systèmes multi-agents pour représenter les organisations. En plus, plusieurs plateformes et framework ont été développés pour simplifier l'implémentation de ces systèmes. Ce chapitre présente les concepts de base de ce domaine.

# **Chapitre 03 :**

## **L'approche**

### **Proposée**

## 1- Introduction

Après avoir présenté les systèmes multi-agents normatifs, ce chapitre est consacré à la présentation de notre approche proposée pour l'implémentation des concepts organisationnels des systèmes multi-agents normatifs. Etant donné que notre travail est une extension de NorJADE, il propose des concepts de base des organisations comme les rôles et les organisations en NorJADE. En plus, il permet la communication entre les rôles. Cette approche est mise en œuvre à l'aide de plusieurs technologies et outils. Ainsi, nous commencerons ce chapitre par la présentation des outils utilisés. Ensuite, nous allons présenter notre approche avec un étude de cas.

## 2- Les plateformes et les technologies utilisées

Afin de développer notre approche, nous avons utilisé plusieurs technologies et plateforme. Cette section est dédiée à la présentation de ces technologies, à savoir les ontologies, la programmation orientée-aspect et la plateforme NorJADE.

### 2.1- Les ontologies

En informatique, le terme a plusieurs définitions. La définition la plus célèbre est celle de <sup>1</sup>, qui définit l'ontologie comme une spécification explicite de la conceptualisation. Par visualisation, nous entendons un système compréhensible. Par conséquent, une spécification explicite signifie que le modèle doit être spécifié dans un langage non ambigu, lui permettant d'être manipulé à la fois par des machines et par des humains. En d'autres termes, l'ontologie permet de fournir des représentations de concepts dans un langage non ambigu.

C'est pour ces multiples utilisations que l'ontologie est apparue dans plusieurs domaines

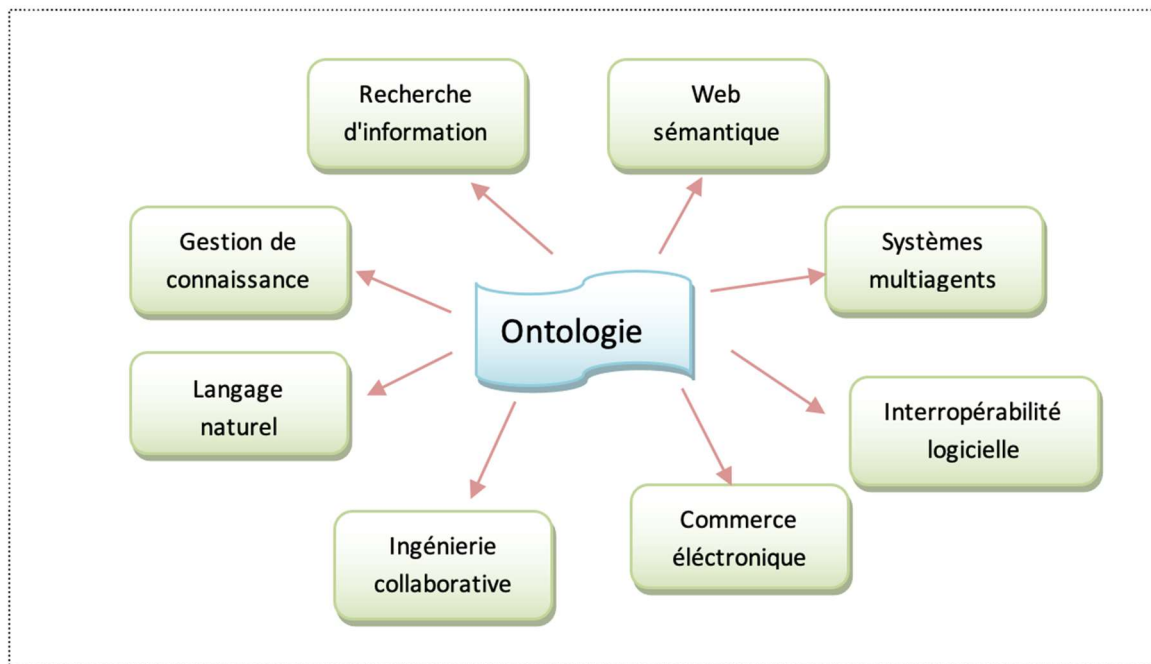


Figure 8 : Les domaines d'utilisations de l'ontologie

- L'ontologie se compose des parties suivantes :
  - **Classes** : ce sont des collections ou des collections Objets avec des propriétés communes.
  - **Les relations** : utilisées pour construire des ontologies car elles représentent des interactions entre des concepts et relient des concepts et des individus les uns aux autres.
  - **Instance** : représente l'objet décrit par le concept
  - **Les Attributs** : propriétés, traits, paramètres que peuvent avoir les objets et les classes.
  
- La classification ontologique est le contenu de la connaissance. Qu'elles représentent, c'est-à-dire le sujet de la conceptualisation <sup>2 3</sup> :
  - **Les ontologies génériques** : permettent de décrire les concepts généraux et elles sont moins abstraits que celles décrites par les ontologies de haut niveau , ( Mikrokosmos, TOVE) .
  - **Les ontologies d'application** : contient toutes les définitions nécessaires pour modéliser les connaissances spécifiques au développement d'une tâche

spécifique. Habituellement, les ontologies d'application combinent des éléments d'ontologies de domaine et d'ontologies générales sélectionnées selon des méthodes spécifiques pour réaliser des tâches cibles.

- **Les ontologies de domaine** : ont pour avantage de permettre une normalisation des concepts dans le cadre du domaine considéré , (KSL serveur).
- **Ontologies de représentation** : Elles proposent un cadre de représentation sans émettre d'hypothèse sur le monde. On les désigne également comme ontologies abstraites ou de haut niveau parce qu'elles permettent de définir des concepts abstraits et peuvent être ré-utilisées pour définir des concepts spécifiques <sup>1</sup>.

Pour développer et exploiter les ontologies, des logiciels spécifiques sont disponibles. Ce sont des éditeurs d'ontologies. Ces éditeurs sont des logiciels qui permettent aux utilisateurs de créer et de modifier des ontologies comme : Protégé, Hozo et Swoop , et plusieurs API sont utilisées pour la manipulation des ontologies (Jena)

## 2.2- la programmation orienté aspect (AOP)

AOP est un acronyme pour Aspect Oriented Programming : la traduction française est Aspect Oriented Programming. AOSD (Aspect Oriented Software Development) est également utilisé. Le développement orienté aspect permet la séparation des préoccupations (SOC). Il est principalement utilisé pour les fonctions horizontales.

La programmation orienté aspect est un style de programmation qui suggère de séparer le code technique du code de travail d'une application pour résoudre les chevauchements techniques ou architecturaux. AOP ne remplace pas la POO, mais la complète plutôt en apportant des solutions élégamment mises en œuvre à certaines de ses limitations ou lacunes. Fonctionnalité. AOP facilite la mise en œuvre des fonctionnalités du navigateur de manière standard. Traditionnellement, ces fonctionnalités ont été partiellement implémentées à l'aide de modèles de conception, de frameworks ou d'outils (générateurs, précompilateurs) sans AOP.

- Le tisseur AspectJ est basé sur les concepts suivants :
  - **Aspects** : modules qui définissent les plugins et leurs points d'activation,
  - **Greffon** (Anglais, advice ) : Un programme à activer à un certain point de l'exécution du système, spécifié par un point de jonction,
  - **Tissage** ou tramage (en anglais, weaving): un système logiciel qui insère statiquement ou dynamiquement des appels à des plugins,
  - **Point d'action** , (en anglais, pointcut) : la position logicielle où l'aspect weaver est inséré dans l'aspect,
  - **Point de jonction** , d'exécution (en anglais, point de jonction) : une position spécifique dans le processus d'exécution du système, où le plug-in est effectivement inséré. Pour clarifier, il n'est pas possible d'insérer des plugins au milieu du code de la fonction, par exemple. D'autre part, cela peut être fait avant, autour, à la place ou après l'appel de la fonction.

## 2.3- NorJade

NorJADE est un framework pour développer des systèmes multi-agents normatifs utilisant la plate-forme très populaire JADE et des technologies bien connues (comme la programmation orientée aspect et les ontologies). Par conséquent, nous présenterons l'ontologie utilisée pour créer le framework NorJADE, puis nous présenterons les aspects utilisés pour améliorer la plate-forme JADE avec les aspects normatifs <sup>4</sup>.

### 2.3.1- NorJADE ontologie

Une ontologie qui comprend tous les concepts importants a été développée pour représenter différents types de critères. La figure 1 représente les concepts utilisés pour définir les normes dans le cadre NorJADE, par exemple, une loi est définie par un agent d'exécution (obligation, recommandation ou interdiction), contrôle le comportement, la validité, est appliquée par l'exécuteur et est appliquée par un organisme de réglementation. Mécanisme (organisation ou application) et a une conséquence (récompense ou punition). Pour l'utiliser, vous devez créer des individus à partir de classes d'existence pour représenter les normes qui contrôlent votre système.

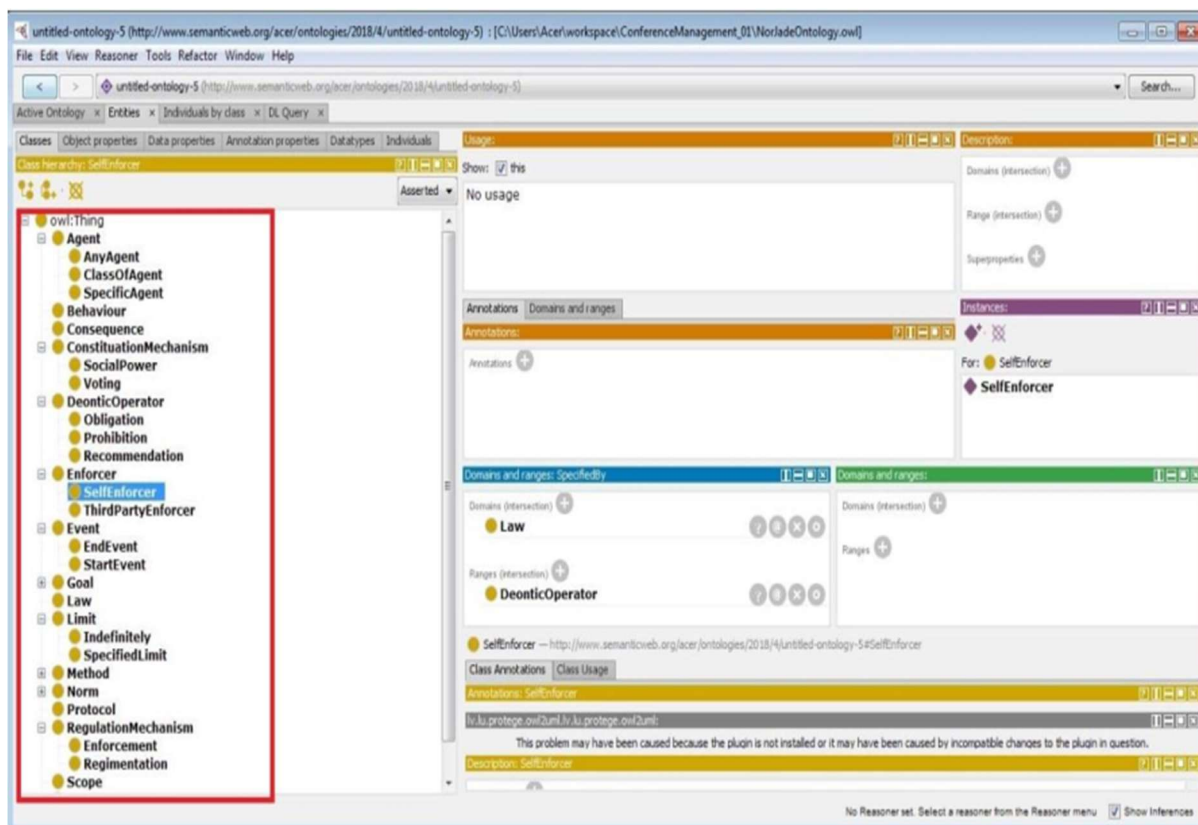


Figure 9 : Concepts utilisés pour spécifier les normes dans NorJADE 4

### 2.3.2- NorJADE Aspects

NorJADE a été développé en utilisant la classe NorJADEOntologyBase qui est utilisée pour le traitement des ontologies afin d'extraire diverses informations sur les normes. Par conséquent, cette classe se compose de certaines requêtes SPARQL qui sont utilisées pour traiter l'ontologie NorJADE afin de collecter des informations sur les critères spécifiés. Ces méthodes sont ensuite utilisées par différents aspects pour vérifier que les comportements mis en œuvre sont conformes aux normes et mettre en œuvre les conséquences de ces normes. NorJADE se compose d'un ensemble des aspects qui sont présentés dans le tableau suivant.

Package	Class / Aspect	Method / structure	Description
NorJADEAspects	Event		<b>Classe utilisée pour définir un événement et inclut certaines méthodes pour manipuler un événement</b>
	ExtendingAgent	ExecutedBehaviours	<b>Un vecteur pour enregistrer les comportements exécutés par un agent</b>
		InterceptedEvent	<b>Un vecteur utilisé pour enregistrer les événements interceptés</b>
		SelfEnforcementApplication	<b>Cette méthode est utilisée pour appliquer une auto-punition/récompense</b>
		ThirdPartyEnforcement	<b>Cette méthode est utilisée pour appliquer une punition/récompense à un tiers</b>
		IsExecutedBehaviour	<b>Cette méthode est utilisée pour vérifier si un comportement est exécuté ou non</b>
		IsConcernedByLaw	<b>Cette méthode vérifie si l'agent est concerné par une loi ou non</b>
		IsValidLaw	<b>Cette méthode vérifie si la loi est valide ou non</b>
	ExtendingBehaviour		<b>Cet aspect est utilisé pour interrompre un comportement qui est régulé par un mécanisme de régimentation</b>
	InterceptBehaviour		<b>Cet aspect est utilisé pour intercepter l'exécution du comportement et le traiter selon le mécanisme de régulation et l'opérateur déontique</b>
	InterceptEvents		<b>Cet aspect permet d'intercepter les événements et de les traiter selon leur type (type de début ou de fin).</b>
	ListOfAgents	Agents	<b>Un vecteur qui est utilisé pour enregistrer des références sur les agents</b>

			<b>afin de les manipuler (par exemple, lorsque nous avons besoin de l'exécution d'une méthode comme récompense de punition)</b>
	LoadNorJadeOntology		An aspect used to load the NorJadeOntology when main function is executed
	ProceduralNorm		<b>Cet aspect est utilisé pour définir les normes procédurales</b>
NorJadeOntology	NorJadeOntologyBase	getLaw	<b>Renvoie une loi associée à un comportement</b>
		getLawProperties	<b>Retourner les propriétés d'une loi</b>
		getEnforcer	<b>Renvoyez l'exécuteur</b>
		getAgentClass	<b>Renvoie la classe d'un agent</b>
		getEventProperties	<b>Renvoyer les propriétés d'un événement</b>
		getBehaviour	<b>Renvoie un comportement associé à un objectif</b>
		getBehaviourName	<b>Renvoie le nom d'un comportement</b>
		getEventId	<b>Renvoyer l'identifiant d'un événement</b>
		getLawValidity	<b>Rétablir la validité d'une loi</b>
		getEvent	<b>Retourner un événement</b>
RewardPunishment	Behaviours to define by users		<b>Décrire la récompense ou la punition à exécuter en cas de violation des normes</b>

Tableau 5: Détails sur les packages, les aspects, les classes et les méthodes 4

### 3- Approche proposée

Dans ce modeste travail, nous avons développé une extension au cadre NorJADE en ajoutant des rôles et des organisations avec possibilité de communication entre les rôles dans les organisations, et l'agent qui joue un rôle est récompensé ou puni selon les critères qui contrôlent son comportement. Bien sûr, cette extension est basée sur le même principe que NorJADE : nous utilisons une ontologie et une programmation orientée aspect pour étendre la plateforme JADE avec les fonctionnalités nécessaires. Dans cette section, nous détaillons cette approche. Tout d'abord, nous introduisons l'ontologie avancée avant d'introduire les aspects.

#### 3.1- Description de l'ontologie

A l'ontologie nous avons ajouté une classe rôle et une classe organisation, et les relations entre eux et les classe NorJade (PlayRole, In , DoneBy , ControleBy ). Comme il est décrit dans Figure 03 : Les relations entre les classes .

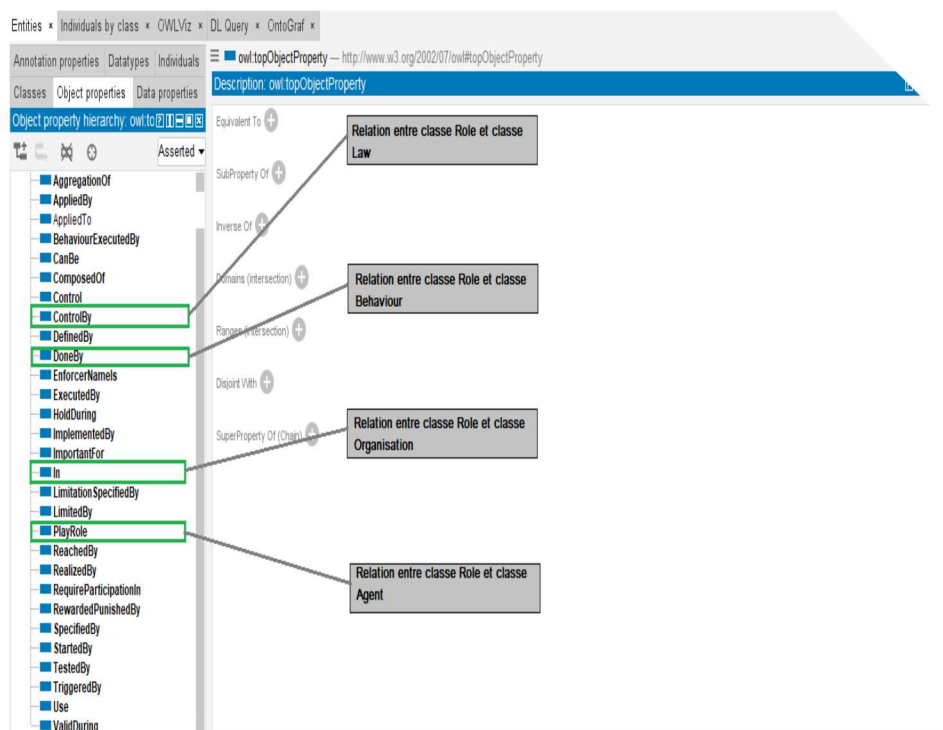


Figure 10: Les relations entre les classes (Object propriétés).

Pour utiliser notre framework, vous devez créer des individus à partir des classes d'existence pour représenter les normes qui contrôlent votre système. Vous devez créer des lois qui contrôlent les rôles .

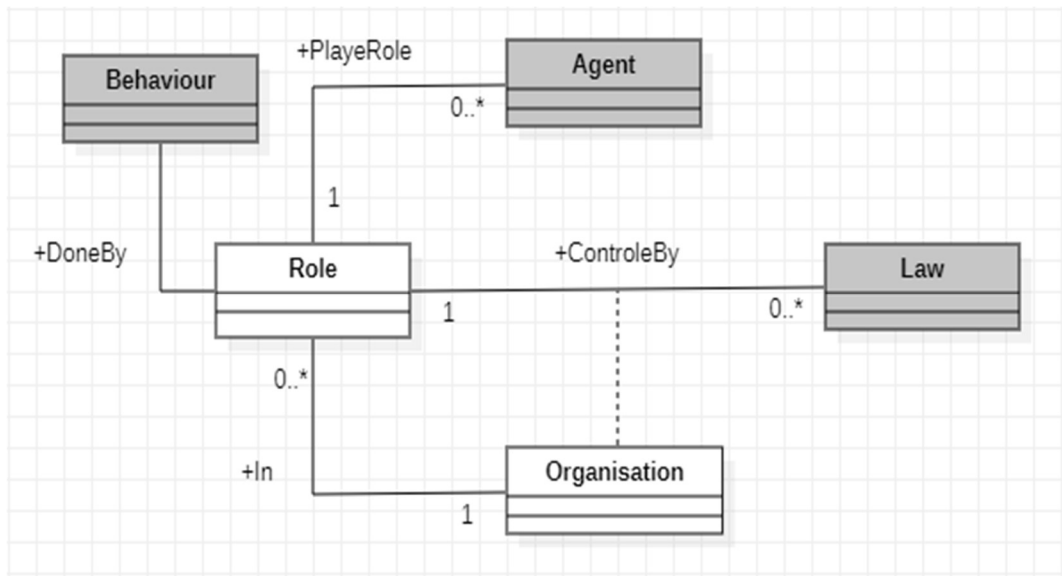


Figure 11: Les notions d'approche proposée dans NorJADE .

### 3.2 Description des aspects et java:

Afin de répondre à l'ontologie en évolution, nous étendons la plate-forme Jade en utilisant la programmation orientée aspect, ajoutons de nouvelles classes Java et étendons les classes standard de la plate-forme de nouvelles façons.

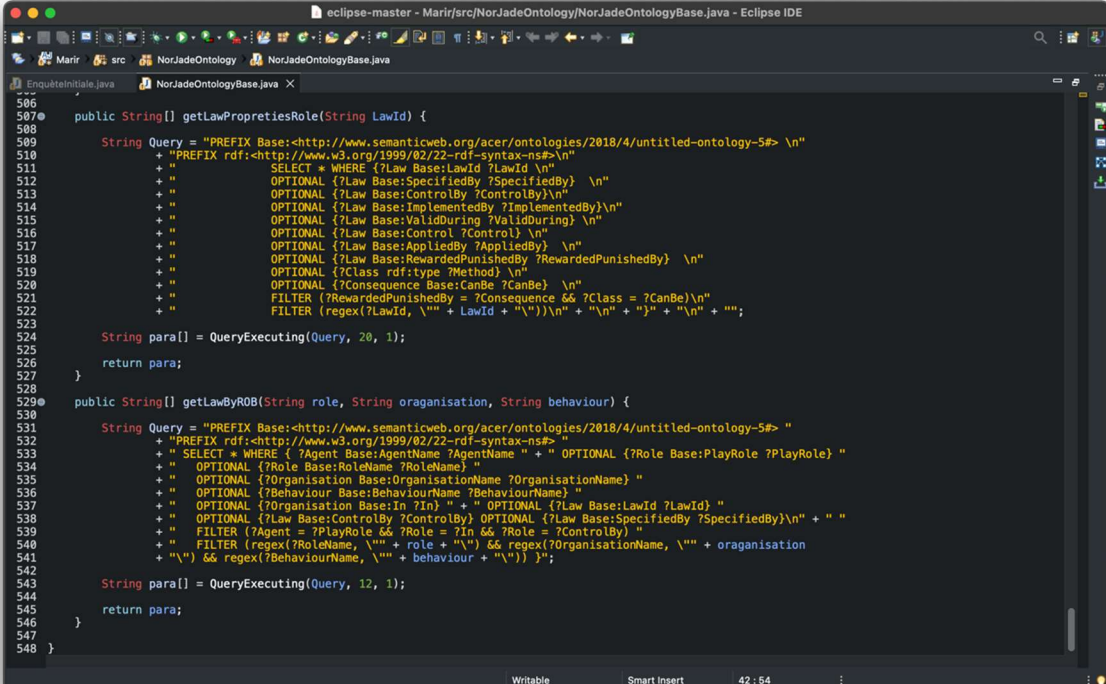
Dans un premier temps, nous avons créé la classe Java " Organisation " pour représenter les organisations dans un système multi-agent modulaire et nous pouvons la mapper comme faisant partie d'une autre organisation. Et la classe " RID " est un identifiant pour les rôles, et la classe " Role " représente le rôle dans un système multi-agent standard où ils peuvent être échangés entre eux et il enregistre tous les rôles qui ont été créés dans la classe " Manager ".

Et nous complétons et étendons les classes standard " NorJade " en utilisant la programmation orientée aspect où nous avons ajouté l'aspect " RIDACLMessage " une variable source de rôle a été définie dans la classe jade " ACLMessage " pour enregistrer l'expéditeur du message et ajouter une table pour enregistrer les récepteurs et aussi nous ajout du l'aspect " ExtendingBehaviour " pour ajouter le

L'approche proposée

rôle qui agit sur le comportement sélectionné. Et on a aussi fait l'aspect " **ExtendingAgent** ", qui sert à définir le rôle de chaque agent, et l'aspect " **InterceptBehaviourRole** ", qui vérifie les normes et les lois spécifiées dans l'ontologie avant chaque comportement réalisé par l'un des rôles et les applique à l'agent qui joue le même rôle.

Nous avons ajouté deux fonctions dans la classe java `NorJadeOntologyBase` définie dans le package `NorJadeOntology` du Framework `NorJADE` qui nous permettent d'obtenir des informations d'Ontology en requête sparql comme indiqué sur la figure suivante :



```
506 public String[] getLawPropertesRole(String LawId) {
507     String Query = "PREFIX Base:<http://www.semanticweb.org/acer/ontologies/2018/4/untitled-ontology-5#> \n"
508     + "PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#> \n"
509     + "SELECT * WHERE {?Law Base:LawId ?LawId \n"
510     + "OPTIONAL {?Law Base:SpecifiedBy ?SpecifiedBy} \n"
511     + "OPTIONAL {?Law Base:ControlBy ?ControlBy}\n"
512     + "OPTIONAL {?Law Base:ImplementedBy ?ImplementedBy}\n"
513     + "OPTIONAL {?Law Base:ValidDuring ?ValidDuring} \n"
514     + "OPTIONAL {?Law Base:Control ?Control} \n"
515     + "OPTIONAL {?Law Base:AppliedBy ?AppliedBy} \n"
516     + "OPTIONAL {?Law Base:RewardedPunishedBy ?RewardedPunishedBy} \n"
517     + "OPTIONAL {?Class rdf:type ?Method} \n"
518     + "OPTIONAL {?Consequence Base:CanBe ?CanBe} \n"
519     + "FILTER (?RewardedPunishedBy = ?Consequence && ?Class = ?CanBe)\n"
520     + "FILTER (regex(?LawId, \"\" + LawId + \"\")\n" + "\"\n\" + \"}\" + \"\n\" + \"\"";
521
522     String para[] = QueryExecuting(Query, 20, 1);
523     return para;
524 }
525
526 public String[] getLawByROB(String role, String oraganisation, String behaviour) {
527     String Query = "PREFIX Base:<http://www.semanticweb.org/acer/ontologies/2018/4/untitled-ontology-5#> "
528     + "PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#> "
529     + "SELECT * WHERE { ?Agent Base:AgentName ?AgentName " + "OPTIONAL {?Role Base:PlayRole ?PlayRole} "
530     + "OPTIONAL {?Role Base:RoleName ?RoleName} "
531     + "OPTIONAL {?Organisation Base:OrganisationName ?OrganisationName} "
532     + "OPTIONAL {?Behaviour Base:BehaviourName ?BehaviourName} "
533     + "OPTIONAL {?Organisation Base:In ?In} " + "OPTIONAL {?Law Base:LawId ?LawId} "
534     + "OPTIONAL {?Law Base:ControlBy ?ControlBy} OPTIONAL {?Law Base:SpecifiedBy ?SpecifiedBy}\n" + " "
535     + "FILTER (?Agent = ?PlayRole && ?Role = ?In && ?Role = ?ControlBy) "
536     + "FILTER (regex(?RoleName, \"\" + role + \"\") && regex(?OrganisationName, \"\" + oraganisation
537     + \"\") && regex(?BehaviourName, \"\" + behaviour + \"\"));
538
539     String para[] = QueryExecuting(Query, 12, 1);
540     return para;
541 }
542 }
```

Figure 12: Montrez ce que nous avons ajouté dans la classe `NorJADEOntologyBase`

## 4- Étude de cas

Afin de valider et d'illustrer notre approche, nous avons d'abord créé un exemple pour illustrer la communication entre deux rôles différents (un étudiant et un

L'approche proposée

agent de sécurité). Comme le montre la figure suivante :

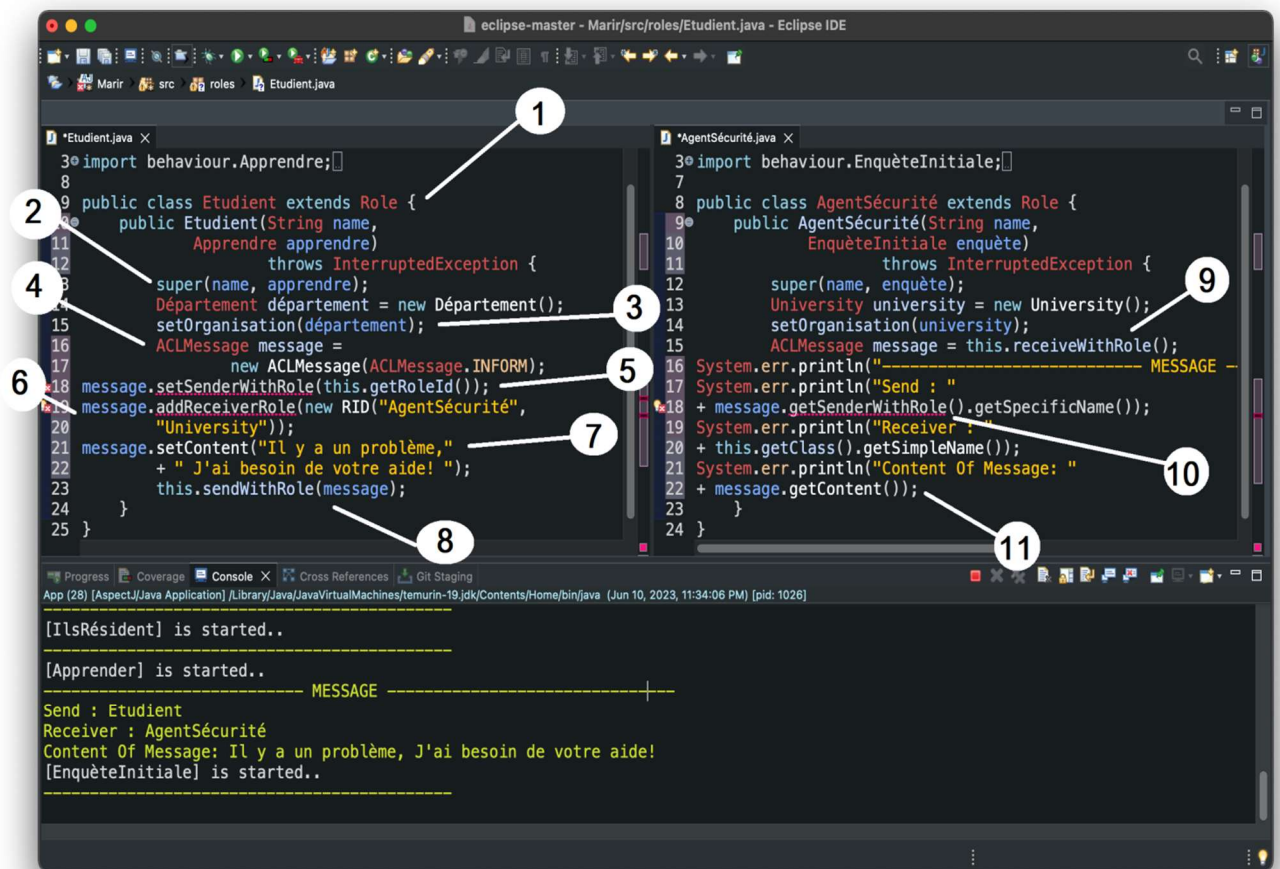


Figure 13: Exemple expliquer communication entre les rôles

- Déterminer le rôle :
  1. Créer une classe qui étend la classe de rôle .
  2. Définir la construction de deux variables, la première pour le nom du rôle et la seconde pour son comportement .
  3. Le rôle peut être ajouté dans une organisation : Tout d'abord, créez un objet d'une classe représentant l'organisation et donnez-lui des paramètres à la fonction `setOrganisation()`.
- Communication entre les rôles :

L'approche proposée

➤ Le côté de l'expéditeur du message :

4. Créez un objet de la classe ACLMessage .
5. Attribuer le rôle de dispatcher en donnant son ID en paramètre à la fonction `setSenderWithRole()` .
6. Ajoutez des destinataires au message en appelant la fonction et en leur donnant un nouvel ID de rôle avec le nom du destinataire et l'organisation à laquelle ils appartiennent.
7. Spécifie le texte du message à ajouter comme dans la communication entre les agents .
8. La transmission est envoyée en la donnant à une fonction `senderWithRole()` .

Le côté destinataire du message :

9. La fonction est appelée et renvoie l'Object transmis .
10. Récupérer la fonction de l'expéditeur à la fonction `getSenderWithRole()` .
11. Connaître le contenu du message à la fonction `getContent()` .

Afin de clarifier le travail des rôles et de punir et récompenser l'agent qui joue le rôle de lui-même , nous avons donné l'exemple illustré dans la figure suivante .

```

Raouf.java
6 public class Raouf extends Agent {
7
8
9     1)
10    2)
11    3)
12    4)
13    5)
14    6)
15
16    protected void setup() {
17        IlsRésident ilsRésident = new IlsRésident();
18        Résident résident =
19            new Résident("Résident", ilsRésident);
20        this.PlayRole(résident);
21        System.err.println("[Agent] "
22            + this.getLocalName() + " is started");
23    }
24
25    public void VérifierLesRésidents() {
26        System.err.println("[Method VérifierLesRésidents]
27            + "-----");
28    }
29
30 }
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

Résident.java
9 public class Résident extends Role {
10     public Résident(String name, IlsRésident ilsRésident) {
11         super(name, ilsRésident);
12         Résidence résidence = new Résidence();
13         setOrganisation(résidence);
14     }
15 }
16

IlsRésident.java
3 import jade.core.behaviours.Behaviour;
4
5 public class IlsRésident extends Behaviour {
6     public void action() {
7         System.out.println("[IlsRésident] is started..");
8         System.err.println("-----");
9     }
10 }
11

```

Console Output:

```

INF-U:
Agent container Main-Container@192.168.1.8 is ready.
[Agent] Raouf is started
[IlsRésident] is started..
-----
[NorJADE Framework] : The behaviour : behaviour.IlsRésident is regulated by the law :CardLoi with the deontic operator : Prohibit.
[NorJADE Framework] : A self punishment will be executed
[NorJADE Framework] : The method : VérifierLesRésidents will be executed by the agent Raouf as Reward/Punishment
[Method VérifierLesRésidents] is started..
-----

```

Figure 14: Exemple montrant punir ou récompenser un agent en fonction du comportement d'un rôle.

- Un agent nommé Raouf est défini et comprend :
  - 1) Créer un objet à partir de la classe Behavior "IlsRésident ".
  - 2) Donner le comportement " IlsRésident " au rôle " Résident " Et c'est déterminé dans l'introspection selon notre approche propose .
  - 3) Déterminez le rôle " Résident " de l'agent joue , Il est également défini en ontologie.
  - 4) Dans la classe Rôle, on définit une organisation à laquelle appartient " Résident " , Il est également défini en ontologie .
  - 5) Dans la classe Behaviour " IlsRésident ", un message est défini pour indiquer que le comportement demandé a été implémenté .

Nous identifié dans l'ontologie une loi appelée " CardLoi "notre unité est une propriété " ControlBy " , De la figure suivante , on remarque que cette loi se punit ou se récompense elle-même « AppliedBy SelfEnforcer » et la conséquence de AccèsInterdit lié à la fonction suivante .

- 6) La fonction VérifierLesRésidents ce sera une punition ou une récompense selon l'agent selon la définition de l'ontologie .

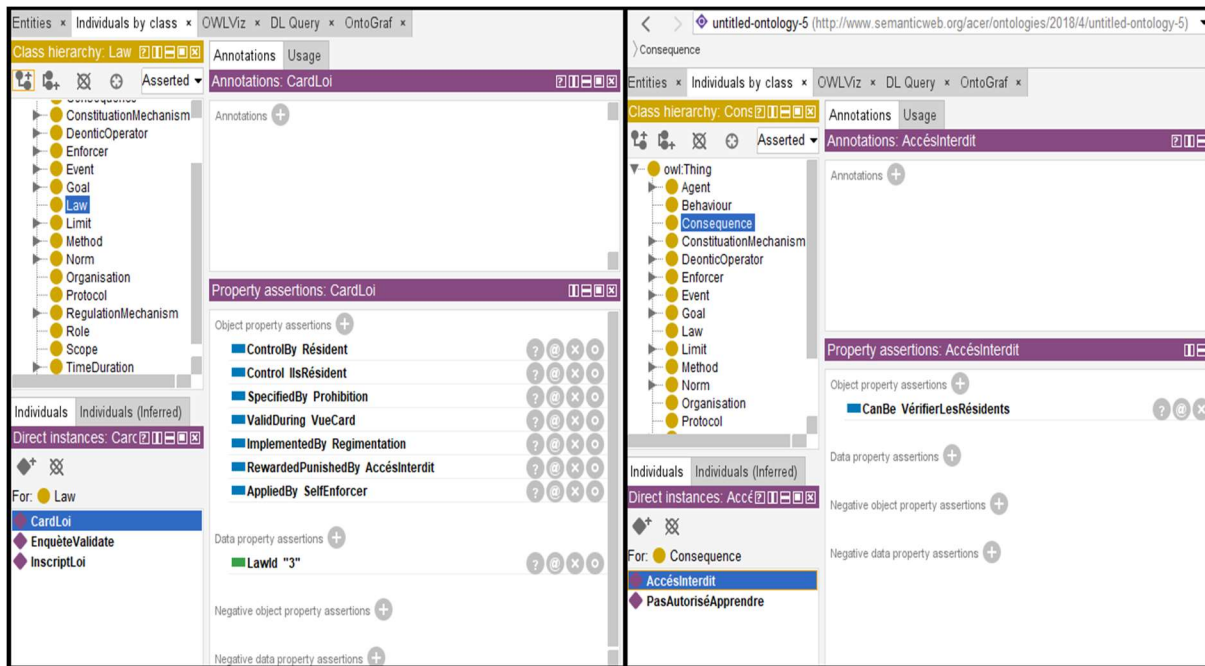


Figure 15: Une image d'ontologie montrant les propriétés de la loi et conséquence .

## 5- Conclusion

Dans ce chapitre nous avons présenté notre approche proposée pour la programmation des concepts organisationnels dans les systèmes multi-agents normatifs. Notre approche qui consiste à une extension de NorJADE permet de l'ajout des organisations, des rôles et la communication entre les rôles. Cette approche est validé par une étude de cas.

## Conclusion générale :

JADE est une plateforme qui permet au programmeur de créer et de contrôler plus facilement des agents, et il a la possibilité de communiquer, et les comportements des agents y sont définis, puis il a été développé en ajoutant des ajouts, et il s'appelait NorJADE.

NorJADE est une extension de NorJADE permet d'implémenter les systèmes multi-agents normatifs. Dans ces systèmes on peut contrôler le comportement de l'agent en vérifiant les règles et normes des normes de comportement dans la pénalisation ou la récompense de l'agent selon le projet l'ontologie.

Dans notre projet, nous avons ajouté des organisations, des rôles et la communication entre eux et entre eux dans une organisation avec leur indépendance de comportement, et les règles et normes qu'ils voient contrôler le comportement des connaissances dans l'ontologie sont atteintes, et l'agent qui joue cela rôle est puni ou récompensé.

Bien entendu, l'ajout des concepts organisationnels à NorJADE permet de développer des systèmes plus réels, mais aussi ouvre les portes à des nouvelles extensions et défis possibles. D'une part, nous envisageons d'étendre le framework NorJADE par des mécanismes permettant de couvrir les aspects dynamiques des organisations multi-agents. D'autre part, nous pensons à tester ce framework par la simulation des systèmes sociaux réels.

## Bibliographie

### Chapitre 01 :

- [1]→ Hannachi , Sibertin-Blanc .Introduction aux Système Multi-Agents . Universite Toulouse.
- [2]→ Ferber J. Les Systèmes Multi Agents: vers une intelligence collective.1995.
- [3]→ Stuart Russell, Peter Norvig. *Intelligence artificielle 3e éd.* Pearson France, 2010 .
- [4]→ Marir.T. Support de cours, Les Systèmes Multi-Agents. 2019 .
- [5]→ (Marir, Mokhati, & Seridi-Bouchelaghem, Do We Need Specific Quality Models for Multi-Agent Systems? - Toward Using the ISO/IEC 25010 Quality Model for MAS, 2014)
- [6]→ Youssfi Mohamed . Signaux Systèmes Distribués et Intelligence Artificielle. 2014 .
- [7]→ Ferber, J. (1995). *Les systemes multi agents*. Paris: InterEditions .
- [8]→ Morin, Edgar. "Les Idées Leur Habitat, Leur Vie, Leurs Mœurs, Leur Organisation." (1991) .
- [9]→ Fox M.S., An organizational view of distributed systems. IEEE Trans. Syst.Man. Univ. Cybern., vol. SMC-II; 1981, pp. 70-80.
- [10]→ Baeijs Christof, Fonctionnalité émergente dans une société d'agents autonomes. Étude des aspects organisationnels dans les SMA réactifs. Thèse de doctorat 1998.
- [11]→ Boella, *Five guidelines for normative multiagent systems*, In Proceedings of JURIX 2009-The 22nd International Conference on Legal Knowledge and Information Systems. IOS Press, 2009.
- [12]→ Javier vazquez-salceda. *The Role of Norms and Electronic Institutions in MultiAgent Systems - The HARMONIA Framework*. Springer Basel AG, 2000.
- [13]→ G. Boella, L. v. d. Torre, H. Verhagen, Introduction to Normative Multiagent Systems, 2006
- [14]→ www.larousse.fr. (2020). Consulté le 6, 6, 2020, sur Dictionnaire online:  
<https://www.larousse.fr/dictionnaire/francais>.
- [15]→ D. Villatoro, "Self-organization in decentralized agent societies through social norms," in *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '11)*, pp. 1373–1374, May 2011.
- [16]→<sup>1</sup> Moamin A. Mahmoud, Mohd Sharifuddin Ahmad, Mohd Zaliman Mohd Yusoff, and Aida Mustapha: A Review of Norms and Normative Multiagent Systems. Hindawi Publishing Corporation The Scientific World Journal, Volume 2014
- [17]→ H. P. Young, "The evolution of conventions," *Econometrica*, vol. 61, no. 1, pp. 57–84, 1993.
- [18]→ D. Villatoro, S. Sen, and J. Sabater-Mir, "Of social norms and sanctioning: a game theoretical overview," *International Journal of Agent Technologies and Systems*, vol. 2, no. 1, pp. 1–15, 2010.
- [19]→ M. J. Piskorski and A. Gorbatai, "Testing Coleman's social-norm enforcement mechanism: evidence from wikipedia," *HBS Working*, vol. 11, no. 55, 2011.
- [20]→ D. G. Lawrence, "Procedural norms and tolerance: a reassessment," *The American PoliticalScience Review*, vol. 70, no. 1, pp. 80–100, 1976.
- [21]→ R. Rubino, A. Omicini, and E. Denti, "Computational institutions for modelling norm-regulated MAS: an approach based on coordination artifacts," in *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems Workshops (AAMAS '05)*, pp. 127–141, 2005.
- [22]→ G. Boella and L. van der Torre, "An architecture of a normative system," in *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '06)*, pp. 229–231, Hakodate, Japan, May 2006.
- [23]→ P. Caire, "A normative multi-agent systems approach to the use of conviviality for digital cities," in *Proceedings of The International Conference on Coordination, Organizations, Institutions, and Norms in Agent Systems III (COIN '07)*, pp. 245–260, 2007.
- [24]→ G. Boella and L. V. D. Torre, "Regulative and constitutive norms in normative multi-agent systems," in *Proceedings of The 9th International Conference on the Principles of Knowledge Representation and Reasoning*, pp. 255–266, Whistler, Canada, 2004.

## Bibliographie

- [25]→ B. Loges, "A norm in the making? The emergence of the "responsibility to protect" (R2P) and UN security council deliberations," in Proceedings of the 7th Pan-European International Relations Conference, Stockholm, Sweden, September 2010.
- [26]→ A. Ahmad, M. Ahmed, M. Z. M. Yusof, M. S. Ahmad, and A. Mustapha, "Resolving conflicts between personal and normative goals in normative agent systems," in 2011 7th International Conference on Information Technology in Asia: Emerging Convergences and Singularity of Forms (CITA'11), Kuching, Sarawak, July 2011.
- [27]→ N. Oren, M. Croitoru, S. Miles, and M. Luck, "Understanding permissions through graphical norms," in Proceedings of the 8th International Conference on Declarative Agent Languages and Technologies (DALT '10), Toronto, Canada, July 2010.
- [28]→ Moamin A. Mahmoud, Mohd Sharifuddin Ahmad, Mohd Zaliman Mohd Yusoff, and Aida Mustapha: A Review of Norms and Normative Multiagent Systems. Hindawi Publishing Corporation The Scientific World Journal, Volume 2014.
- [29]→ C. Hollander and A. Wu, "The current state of normative agent-based systems," *Journal of Artificial Societies and Social Simulation*, vol. 14, no. 2, article 6, 2011.
- [30]→ B. T. R. Savarimuthu, *Mechanisms for norm emergence and norm identification in multi-agent societies [Ph.D.thesis]*,
- [31]→ G. H. V. Wright, "Deontic Logic," *Mind*, vol. 60, no. 237, pp. 1–15.
- [32]→ C. Castelfranchi and R. Conte, *Cognitive and Social Action*, UCL Press, London, UK, 1995.
- [33]→ J. M. Galan and L. R. Izquierdo, "Appearances can be deceiving: lessons learned re-implementing Axelrod's 'evolutionary approach to norms'," *Journal of Artificial Societies and Social Simulation*, vol. 8, no. 3, 2005.
- [34]→ C. Hollander and A. Wu, "The current state of normative agent-based systems," *Journal of Artificial Societies and Social Simulation*, vol. 14, no. 2, article 6, 2011.

## Chapitre 02 :

- [1] Jacques Ferber, Olivier Gutknecht, and Fabien Michel. From Agents to Organizations: An Organizational View of Multi-Agent Systems. LIRMM –University of Montpellier (2004).
- [2] Ferber, J. Les Systèmes Multi Agents : vers une intelligence collective. InterEditions, 1995.
- [3] « Systèmes Multi-Agents Programmer avec MadKit - PDF Téléchargement Gratuit ». <https://docplayer.fr/13597047-Systemes-multi-agents-programmer-avec-madkit.html> (consulté le 25 mai 2023).
- [4] J. Ferber, T. Stratulat, J. Tranier, et T. Fr, « Towards an Integral Approach of Organizations in Multi-Agent Systems », *Handb. Res. Multi-Agent Syst. Semant. Dyn. Organ. Models*, janv. 2009, doi: 10.4018/978-1-60566-256-5.ch003.
- [5] Marion Nicolas. Étude de modèles d'organisation sociale pour les environnements virtuels de formation. Mémoire de Master2 en informatique encadré par Ronan Querrec. 2006.
- [6] Bordini, R. H., Hübner, J. F., and Wooldrige, M. Programming Multi-Agent Systems in AgentSpeak using Jason John Wiley & Sons, 2007.
- [7] « AgentSpeak », *Wikipedia*. 18 mars 2023. Consulté le : 5 juin 2023. [En ligne]. Disponible sur : <https://en.wikipedia.org/w/index.php?title=AgentSpeak&oldid=1145283501>

## Bibliographie

- [8] O. Gutknecht et J. Ferber, « The MadKit Agent Platform Architecture », in *Infrastructure for Agents, Multi-Agent Systems, and Scalable Multi-Agent Systems*, T. Wagner et O. F. Rana, Éd., in *Lecture Notes in Computer Science*, vol. 1887. Berlin, Heidelberg : Springer Berlin Heidelberg, 2001, p. 48-55. doi: 10.1007/3-540-47772-1\_5.
- [9] O. Gutknecht, F. Michel, et J. Ferber, « MadKit: une architecture de plate-forme multi-agent générique ».
- [10] Bonnet, G., & Reynaud, C. (2008). MADKIT: An efficient and scalable multi-agent platform. In *Multi-Agent Systems and Applications V* (pp. 36-51). Springer.
- [11] « JaCaMo Project | Multi-Agent Programming Framework ». <http://jacamo.sourceforge.net/> (consulté le 3 juin 2023).

## Chapitre 03 :

- [1] A Translation Approach to Portable Ontology Specifications, Knowledge Aquisition.
- [2] N. Guarino, Formal Ontology and Information Systems, In *Formal Ontology in Information Systems*, N Guarino (Ed.), IOS Press. pp 3-15, 1998.
- [3] G. van Heijst, G. Schreiber, B. Wielinga, Using explicit ontologies for KBS development, *International Journal of Human-Computer Studies*, 42(2/3), pp 183-292, 1997
- [4] Marir.Toufik , User Guide of NorJADE Framework ,[github.com/MarirToufik/NorJADEFramework](https://github.com/MarirToufik/NorJADEFramework) ,(2019)