

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université 20 Août 1955-Skikda

Faculté des Sciences

Département d'Informatique



**Mémoire de fin d'études en vue de l'obtention du diplôme en  
Master 2**

**Option : Réseaux et Systèmes Distribués (RSD)**

**Thème:**

**Simulation à base d'agents d'un Système de Contrôle  
de Processus Industriels**

**Réaliser par :**

**Gharssallah Mohamed Nadjib**

**Gherairia Mohamed Amine**

**Encadreur :**

**Dr. Nouredine SEDDARI**

**Année 2021/2022**

# REMERCIEMENTS

*Je remercie tout d'abord ALLAH pour m'avoir donné la force et la volonté tout au long de mon travail de fin d'étude car sans lui rien n'aurait pu être.*

*Un grand merci à mon encadreur de mémoire, Mr Seddari Nour-Eddine pour ses précieux Conseils et son Orientation. Ainsi que son support, sa patience et ces efforts durant ce travail que dieu le bénisse.*

*Je tiens à remercier également les membres du jury d'avoir accepté d'évaluer mon travail.*

*Je remercie tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.*

*Enfin, Un grand merci à tous ceux qui m'ont accompagné au cours de toutes ces années, famille et amis.*

*Merci.*

# *Dédicace*

*A ma chère mère,*

*A mon cher père,*

*Qui n'ont jamais cessé, de formuler des prières à mon égard, de me soutenir et de m'épauler pour que je puisse atteindre mes objectifs.*

*A ma famille, mes proches et à ceux qui me donnent de l'amour et de la vivacité.*

*A mon encadreur Mr. Seddari*

*A tous mes amis qui m'ont toujours encouragé, et à qui je souhaite plus de succès.*

*A tous ceux que j'aime.*

*Merci !*

*Gherairia Mohamed Amine*

*A ma chère mère,*

*A mon cher père,*

*Qui n'ont jamais cessé, de formuler des prières à mon égard, de me soutenir et de m'épauler pour que je puisse atteindre mes objectifs.*

*Aux deux être qui me sont les plus chères au monde à mes enfants*

*Israa et Rassim,*

*A ma femme pour m'avoir comblé d'amour et d'affection,  
pour m'avoir soutenu et encouragé dans tous ce que j'ai  
entrepris et avoir fait.*

*A ma famille, mes proches et à ceux qui me donnent de l'amour et de la  
vivacité.*

*A mon encadreur Mr. Seddari*

*A tous mes amis qui m'ont toujours encouragé, et à qui je souhaite  
plus de succès.*

*A tous ceux que j'aime.*

*Merci !*

*Gharsallah mohamed nadjib*

## RÉSUMÉ:

La simulation à base d'agents ou encore Multi-Agents s'impose de plus en plus pour son efficacité et sa simplicité. En effet, ce type de simulation offre, au contraire des autres types, la possibilité de représenter directement les entités simulées, leurs comportements et leurs interactions sans avoir recours aux équations mathématiques. Ce travail est une contribution dans ce sens, le but est de proposer un modèle à base d'agents AUML pour simuler les systèmes complexes industriels. Ce dernier présente le problème de la complexité, qui peut être maîtrisé avec l'approche des Systèmes Multi-Agents(SMA). Notre modèle est validé par une étude de cas du processus de déshydratation. Cette dernière est consolidée par une simulation faite dans plateforme multi-agents JADE.

**MOTS-CLÉS:** Simulation, Systèmes Multi-Agents (SMA), JADE, Processus industriels.

## ABSTRACT :

Simulation based on agents or multi agents is more and more essential for its effectiveness and its simplicity. Indeed, this type of simulation offers, on the contrary of other types, the possibility of directly representing the simulated entities, their behaviours and their interactions without having recourse to the mathematical equations. This work is a contribution in this way; the goal is to build a simulator based on agents of an industrial process. This process as almost all the industrial processes present the problem of the complexity, which can be controlled with the approach Multi-Agents. We used the software tool JADE as platform of development and execution of our simulator, with satisfactory results.

**KEYWORDS:** Simulation, Multi-Agents systems (MAS), JADE, industrial processes.

## ملخص

المحاكاة القائمة على عوامل أو عوامل متعددة ضرورية أكثر فأكثر لفعاليتها وبساطتها. في الواقع ، يقدم هذا النوع من المحاكاة ، على عكس الأنواع الأخرى ، إمكانية التمثيل المباشر للكيانات المحاكاة وسلوكياتها وتفاعلاتها دون اللجوء إلى المعادلات الرياضية. هذا العمل هو مساهمة بهذه الطريقة. الهدف هو بناء جهاز محاكاة يعتمد على عوامل عملية صناعية. هذه العملية مثل جميع العمليات الصناعية تقريبًا تمثل مشكلة التعقيد ، والتي يمكن التحكم فيها من خلال نهج Multi-Agents. استخدمنا الأداة البرمجية JADE كمنصة لتطوير وتنفيذ جهاز المحاكاة الخاص بنا ، مع نتائج مرضية. الكلمات المفتاحية : المحاكاة ، أنظمة الوكلاء المتعددة (MAS) ، JADE العمليات الصناعية.

# Table de Matières

Introduction Générale	1
-----------------------	---

## ***Chapitre 1 : Modélisation et simulation des systèmes***

Introduction	4
<b>1 / Modélisation des systèmes</b>	4
<b>1-1 / Objectif de la modélisation</b>	4
<b>1-2/ Notion de système</b>	4
<b>1-3/ Types de systèmes</b>	4
1-3-1/ Système déterministe	5
1-3-2/ Système probabiliste (stochastique)	5
1-3-3/ Système continu	5
1-3-4/ Système discret (discontinu)	5
1-3-5/ Système mixte	5
<b>1-4/ Notion de modèle</b>	5
<b>1-5/ Les étapes de modélisation d'un système</b>	6
<b>1-6/ Types de modèles</b>	6
1-6-1/ Les modèles physiques	6
1-6-2/ Les modèles abstraits	7
<b>1-7/ Modélisation informatique</b>	7
<b>1-8/ Les outils de modélisation informatique</b>	7
1-8-1/ Le modèle entité / association	7
1-8-2 / Les réseaux de pétri	7
1-8-3 / Modélisation orientée objet	8
1-8-4 / Modélisation orientée agent	8
<b>2/ Simulation des systèmes</b>	9
<b>2-1/ La simulation informatique</b>	9
<b>2-2 / Catégories de simulation informatique</b>	10
2-2-1 / La simulation continue	10
2-2-2 / La simulation discrète	10
2-2-3 / La simulation par agents	11
<b>2-3 / Les avantages de la simulation</b>	11
<b>3 / Domaine d'application de la simulation</b>	11
<b>4./ Buts de la modélisation et de la simulation</b>	12
<b>Conclusion</b>	12

## **Chapitre 02 : Les systèmes Multi-Agents**

<b>Introduction</b>	14
<b>1/L'intelligence artificielle distribuée (IAD)</b>	14
1-1 / Problématique de l'IAD	15
<b>2/Définition d'un agent</b>	15
2-1 / Caractéristiques d'un agent	16
2-1-1/ La situation	16
2-1-2/ L'autonomie	16
2-1-3/ La flexibilité	16
2-1-4/L'interactivité	17
2-2/ Type d'agents	17
<b>3/ Les systèmes Multi-Agents</b>	19
3-1 / Architecture d'un SMA	20
3-1-1 / Le modèle de tableau noir (l'approche centralisée)	20
3-1-2/ Le modèle d'acteurs (contrôle distribué)	21
3-2 / Interaction dans un système Multi-Agents	21
3-2-1/ La coopération	22
3-2-2 / La négociation	22
3-2-3/ La coordination	23
3-3 / Communication dans les systèmes Multi-Agents	23
<b>4 /Les Méthodes de conception des Systèmes Multi-Agents</b>	24
4-1 / GAIA	24
4-2/ Multi-agent Systems Engineering (MaSE)	25
4-3/ ALAADIN	26
4-4/ADELFE	26
4-5 / La méthodologie Ingénias	26
4-6 / La méthodologie Tropos	27
<b>5/ Les plates-formes Multi-Agents</b>	27
5-1 / Jack	28
5-2 / ZEUS	28
5-3 / Agent-Builder	28
5-4 / MADKIT	29
5-5 / JADE	29
<b>Conclusion</b>	29

## **Chapitre 3: Les systèmes Industriels Complexes**

<b>1.Introduction</b>	31
<b>2. La complexité d'un système industriel</b>	32
2.1. Les entrées/sorties d'un système industriel	32
2-1-1 / Les E/S des matières d'œuvre	32
2-1-2 / Les E/S d'information	32
2-1.3 Les E/S d'énergie	32
2-1.4 /Les E/S intempestives	32
<b>3/Processus industriel</b>	33

3-1 /Processus	33
3-2 /Perturbation	33
3.3 / Grandeurs de commande	34
<b>4 / L'informatique industrielle</b>	34
4-1 /Domaines d'application	34
<b>5/Système De Compression Du Gaz Démercurisation :</b>	34
5-1/Présentation du Complexe GL1.K SKIKDA	34
5-2/ Description générale de l'activité	35
5-3 /Description du système	36
5-4/Exploitation	36
5-4-1 Marche normale	36
5-4-2 Adsorbeur de mercure 14-MD01	37
5-4-3 Filtre à gaz traité 14-MD02	37
5-5/Paramètres d'exploitation usuels	37
5-6/ Exploitation provisoire	37
5-6-1/Adsorbeur de mercure	37
5-6-2 / Filtre à gaz traité	38
5.7Démarrage	38
5-8Arrêt	40
5-8-1Arrêt normal	40
5-8-2/Arrêt d'urgence	40
5-9 /Sécurité	41
5-9-1Systèmes critiques d'arrêt de sécurité	41
5-9-2Liste des soupapes de sécurité	41
5-9-3/Équipement de lutte contre l'incendie	41
<b>Conclusion</b>	42

## *Chapitr4 Une Architecture à base d'agent pour la simulation d'un système industriel*

<b>1/Introduction :</b>	43
<b>2/Description de l'architecture proposée :</b>	43
<b>3/Structure des agents de la simulation de systèmes industriels :</b>	44
<b>3-1/Agent l'adsorbeur de mercure 14-MD01</b>	45
3-1-1/Le module de communication :	45
3-1-2/L'interface utilisateur	45
3-1-3/Le module d'exécution :	45
3-1-4/La base de connaissance	45
<b>3-2/Agent de filtre à gaz traité 14-MD02 :</b>	45
3-2-1/Lemoduledecommunication:	45
3-2-2/L'interface utilisateur :	45
3-2-3 /Le module d'exécution :	45
<b>3-2-4/La base de connaissance :</b>	45
<b>3-3/Agent pression gaz :</b>	46

: Le module de communication/1-3-3	46
3-3-2/L'interface utilisateur :	46
3-3-3/Le module d'exécution :	46
3-3-4/Le module de planification et de coordination	46
3-3-5/La base de "Connaissances globales" :	46
3-3-6La base "connaissances individuelles" :	47
3-4/Agent Sécurité :	47
3-4-1/le module d'authentification et d'analyse :	47
3-4-2/Le module de communication :	47
3-4-3/Le module d'exécution :	47
<b>3-4-4/La base de connaissance :</b>	47
4/Exemple de scénario	48
4-1/Démarrage	48
4-2/Arrêt	48
Conclusion :	48

## *Chapitre 5 : La Conception*

<b>1/Introduction :</b>	50
<b>2/ Le langage AUML</b>	50
<b>3/ Le fonctionnement du système:</b>	50
3-1 Agent l'adsorbant de mercure 14-MD01	50
3-2 Agent de filtre à gaz traité 14-MD02	51
3-3Agent pression gaz	51
3-4Agent Sécurité	51
<b>4/ Le diagramme de cas d'utilisation</b>	51
<b>5/ Diagramme de classe</b>	52
5-1 La classe de l'agent filtre à gaz traité 14-MD01	53
5-2 La classe de l'agent filtre à gaz traité 14-MD02	53
5-3 La classe de l'agent pression gaz	54
5-4 Diagramme de classe du système	54
6/ Diagramme de séquence	55
6-1 Diagramme de séquence d'authentification	55
Conclusion	55

## *Chapitre 6 : Implémentation*

<b>Introduction</b>	57
<b>2/Outils de programmation :</b>	57
<b>3/ Présentation des outils</b>	57
3-1 Le langage de programmation Java :	57
3-2 JDK : (Java Development Kit	58
3-3 Eclipse	59
3-4 La plateforme Jade	59
3-4-1 Définition :	59

3-4-2 L'architecture de la plate-forme jade	60
3-4-3 Comportement des agents dans JADE :	61
3-4-4 L'intérêt d'utilisation de JADE dans notre application :	62
4/ Intégration de la plate-forme JADE dans Eclipse :	62
<b>5/Implémentation du système :</b>	63
5-1Présentation des interfaces graphiques :	64
<b>Conclusion</b>	66

### *Table des figures :*

<b>Figure</b>	<b>Titre</b>	<b>Page</b>
1.1	processus de modélisation	6
1.2	Un modèle sur les réseaux de pétri (RDP).	8
1.3	La simulation informatique selon Fishwick, 1997.[7]	9
1.4	Vue d'ensemble de la simulation informatique	10
2.1	Modèle d'agent-réactif	18
2.2	structure d'unagent cognitif.	18
2.3	Modèle d'agent hybride.	19
2.4	Le modèle de tableau noir.	21
2.5	Le modèle d'acteur.	21
2.6	Les modèles de GAIA	25
2.7	Les différentes phases de MaSE[16]	25
3.1	Vuegénérale d'un système de production industriel	31
3.2	:LesE/Sd'un système industriel	33
3.3	VueAérienne du complexe GL1.K	35
3.4	système de démercurisation	36
3.5	Adsorbeur de mercure contourné	38
3.6	Filtre à gaz traité contourné	38
3.7	Étapes de démarrage 1 et 2	39
3.8	Étapes de démarrage 3 et 4	39
3.9	Arrêt normal	40
4.1	Représentation des systèmes industriels basé agent	44
4.2	:Structured' Agent pression fuel-gaz	46
4.3	Structured' Agent Sécurité	47
5.1	Diagramme de cas d'utilisation.	51
5.2	Présentation graphique de l'échange des messages dans la classe agent.	52
5.3	La classe de Agent l'adsorbeur de mercure 14-MD01	53
5.4	La classe de l'agent filtre à gaz traité 14-MD02	53
5.5	La classe de l'agent pression gaz.	54
5.6	Diagramme de classe du système	54
5.7	Les opérateurs AND, XOR et OR en AUML	55
5.8	Diagramme de classe d'authentification.	55
6.1	java	58
6.2	logo d'Eclipse.	59

6.3	logo de jade.	59
6.4	Architecture de la plateforme jade.	61
6.5	Fenêtre pour ajouter une Library.	62
6.6	configurer Jade sur Eclipse	63
6.7	lancer la main	63
6.8	Authentification	64
6.9	Interface de la simulation (groupe de système de démercurisation).	64
6.10	Interface de la simulation (Adsorbent de mercure activé).	65
6.11	.Interface de la simulation (groupe filtration de gaz).	65
6.12	L'interface du simulateur dans état d'arrêt.	66

### *Table d'abréviations :*

Abréviation	Signification
GL1/K	Complexe de Liquéfaction du Gaz Naturel SKIKDA
CP1/K	Complexe Pétrochimique SKIKDA
GN	Gaz Naturel
GNL	Gaz Naturel liquéfié
GPL	Unité de séparation Propane/ Butane
SMA	Système Multi-agents
IAD	Intelligence Artificiel Distribué
TIC	Température Indicator Controller
PC	Partie Commande
SdC	Salle de contrôle/commande
DCS	Distributed Control Système

### *Liste des tableaux*

Tableau	Titre	Page
2.1	Différences entre agents cognitifs et agents réactifs	18
3.1	Variables d'exploitation pour le système de démercurisation	37
3.2	Liste des soupapes de sécurité du système de démercurisation	41
3.3	Liste de l'équipement anti-incendie du système de démercurisation	41
3.4	Liste de l'équipement anti-incendie du système de démercurisation	42

De nos jours le monde connaît des développements et des avancements terribles dans les domaines de la simulation.

La modélisation est la première étape de la simulation. Cette étape passe par l'analyse de comportement et définit les problèmes d'un système ou d'un phénomène quelconque, le modèle numérique n'est qu'une approximation d'un phénomène réel ou hypothétique rédigé pour la simulation.

La simulation est la traduction réelle du phénomène du processus en présence d'un ensemble d'équations - mathématiques - dans les programmes informatiques développés pour l'étude et l'évaluation.

La simulation par ordinateur c'est la création d'un modèle numérique à l'aide des logiciels pour résoudre des problèmes d'une situation existante ou hypothétique ou un problème inattendu, ce qui facilite le processus d'étudier le comportement du système et ça permet la modification sur les hypothèses ou l'influence de comportement des variables du système, et ce type de simulation a connu un grand développement au cours des dernières décennies et est actuellement en vigueur dans les domaines de la recherche pour améliorer la performance d'une usine ou d'une entreprise.

L'activité de supervision et de contrôle est complexe en raison des caractéristiques dynamiques du processus ainsi que le plus grand nombre de composants qui le composent et par conséquent ça va engendrer plus d'interactions et plus de complexité. À la différence des situations statiques où l'environnement ne change que sous l'effet des actions de l'opérateur, la dynamique d'un environnement évolutif impose des situations dynamiques que l'opérateur ne contrôle que partiellement.

Notre travail se situe dans ce cadre, il permet de présenter la conception et l'implémentation d'un simulateur d'un processus industriel, son contrôle et son suivi, ce processus concerne le système de déshydratation (unité 13) au niveau du complexe de liquéfaction du gaz naturel GL1K groupe SONATRACH SKIKDA.

Grace aux systèmes multi-agent, on peut modéliser des systèmes réels dans lesquels des comportements très complexes émergent d'interactions relativement simples et locales entre de nombreux individus différents, donc un système multi-agents est plus naturel pour décrire et simuler un système composé d'entités, il sera particulièrement bien adapté à la description d'un système du point de vue de l'activité de ses constituants, c'est-à-dire lorsque le comportement des individus est complexe (donc difficile à décrire avec des équations).

La simulation multi-agents nous apparaît être la démarche la plus adéquate à la modélisation de ce type de système (les systèmes complexes tels que le procédé du système

de déshydratation du gaz). La simulation multi-agents est un domaine de recherche récent, qui se base sur la notion d'agent.

Ainsi, notre mémoire se compose de six chapitres organisés de la façon suivante :

- **Modélisation et simulation des systèmes** : Ce chapitre constitue une introduction à notre travail par la présentation des notions sur la modélisation et la simulation d'une façon générale.
- **Les systèmes industriels complexes** : Ce chapitre contient quelques concepts de base des systèmes industriels et leur complexité du contrôle.
- **Les systèmes Multi-Agents** : Ce chapitre contient les systèmes Multi-Agent et leur caractéristiques.
- **Le processus de déshydratation du gaz** : dans ce chapitre nous avons exposé les différents composants entrant dans le processus étudié, et son fonctionnement.
- **La conception** : Dans ce chapitre, nous présentons notre modèle par l'utilisation des principes de la méthode AUML.
- **L'implémentation** : Ce dernier chapitre représente la réalisation et l'implémentation du simulateur modèle par l'utilisation de la plate-forme JADE.

# **Chapitre 1**

## **Modélisation et Simulation des Systèmes**

## Introduction :

Dans la vie courante, nous modélisons tous et tout le temps : à chacun des êtres qui nous entourent, qu'il s'agisse d'objets matériels, de personnes ou d'institutions, nous associons une image mentale à un objet qui nous permet d'anticiper son comportement. Nous faisons des simulations pour évaluer les conséquences et prendre les décisions.

Depuis la nuit des temps, l'homme observe les relations, les régularités et les symétries de son univers. Pour décrire, structurer ces relations, ces régularités et ces symétries, en particulier dans le domaine de la physique, on fait appel au langage mathématique. Toutes les lois de la nature sont exprimées sous forme d'équations. Le comportement d'un phénomène, d'un changement d'état, est interprété par des relations.

Ce chapitre constitue une introduction à notre travail par la présentation de quelques notions, qui nous semblent sur la modélisation et la simulation d'une façon générale.

## 1/Modélisation des systèmes :

### 1-1 /Objectif de la modélisation :

L'objectif de la modélisation, dans son sens le plus général, est la connaissance du monde réel. Les sciences dites dures (physique, chimie, biologie, ...) ou les sciences humaines (économie, sociologie, psychologie, ...) proposent des modèles. Ces modèles sont d'abord validés par la confrontation de leurs résultats avec ceux d'expériences réalisables. Une fois validés, ceux-ci ont une valeur prédictive et suggèrent d'autres expériences non réalisées jusqu'à maintenant. Si ces nouvelles expériences corroborent les prédictions du modèle, alors la validation de celui-ci est confirmée, et la connaissance du domaine modélisé est réellement augmentée. efficace. [1]

### 1-2/Notion du système :

Il existe de nombreuses définitions du « système » ; on en a retenu deux qui dégagent les qualités essentielles de cette notion : « **Totalité organisée, faite d'éléments solidaires ne pouvant être définis que les uns par rapport aux autres en fonction de leur place dans cette totalité.** » (F. de SAUSSURE) « **Ensemble d'éléments en interaction dynamique, organisés en fonction d'un but.** » (J. de ROSNAY)

### 1-3/Types de systèmes :

Il existe plusieurs types de systèmes : statique ou dynamique, déterministe ou stochastique, continu ou discret. [2]

## **1-3-1/Système déterministe :**

Un système est dit déterministe s'il se comporte toujours exactement de la même manière lorsqu'il est placé dans des conditions identiques.

## **1-3-2/Système probabiliste (stochastique) :**

Un système qui se comporterait de façons différentes sans cause apparente, et cette absence de cause apparente est appelée le hasard. La formalisation mathématique de tels systèmes est fournie par la théorie des probabilités

## **1-3-3/ Système continu :**

Les changements de l'état du système se font permanences avec le temps (dépend d'une variable liée au temps).

## **1-3-4/ Système discret (discontinu) :**

Le système est caractérisé par des événements qui surviennent à des instants non fixes et causent un changement de l'état jusqu'au prochain événement.

## **1-3-4/Système mixte :**

La séparation entre système continu et discret est en quelque sorte artificielle, car en réalité la plupart des systèmes possèdent des composantes continues et discrètes à la fois. De tels systèmes sont dits mixtes.

## **1-4/Notion de modèle :**

Un « modèle » est la représentation d'un objet du monde réel. En mécanique, les modèles sont des outils permettant de prédire les réponses statiques et dynamiques d'un phénomène qui servent à son interprétation ou à la prédiction de son comportement. Le mécanicien substitue aux objets réels des corps géométriques au comportement établi, leur donne des masses, leur impose des efforts, eux même modélisés, et met en équation, c'est-à-dire transforme un problème réel en un problème mathématique, C'est une construction abstraite qui permet de comprendre le fonctionnement d'un système de référence en répondant

à une question qui le concerne. Représentation simplifiée de ce système, un modèle s'appuie sur une théorie générale et il est exprimé dans un langage spécifique appelé langage de modélisation.

D'après P. Hubert [Hubert, 1969] « **un modèle est défini comme un objet ou une personne à imiter, un exemple ou un archétype. Il est devenu dans le domaine scientifique une construction matérielle ou abstraite "ressemblant" à l'objet modélisé,**

selon un certain nombre de caractéristiques pertinentes eu égard aux données disponibles et à l'«objectif poursuivi ».[3]

## 1-5/ Les étapes de modélisation d'un système :

L'étape de modélisation est une phase essentielle à la simulation. Différents points doivent être abordés :

- Définir l'objectif de la modélisation
- Définir les éléments du système (via la réalisation d'une fonction, ou d'un processus) et les limites du système (les entrées, les sorties).
- Définir les interactions entre ces éléments (hiérarchie).
- Définir la dynamique du système (entités qui circulent entre les éléments, comportement du système au cours du temps).
- Abstraction (choisir les éléments du système pertinents pour l'étude).
- Formalisation, conceptualisation : Modèle mathématique (les lois discret et les lois continues), modèle logiciel (Simulink, Siman Arena), modèle graphique (réseaux de Pétri)[4].

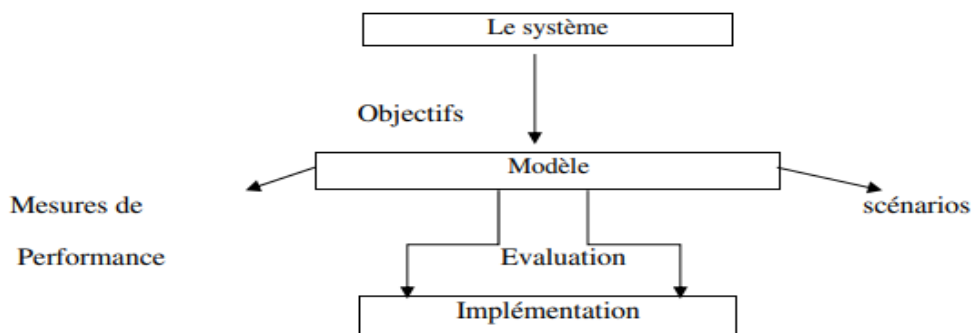


Figure 1.1 : processus de modélisation

## 1-6/ Types de modèles :

On distingue plusieurs catégories de modèles On selon la nature du système étudié :

- Iconiques ou physiques.
- Symboliques ou abstraits.

### 1-6-1/ Les modèles physiques :

Est une représentation matérielle simplifiée et, en général, à une échelle réduite d'une situation problématique traduisent les phénomènes sous forme de représentations concrètes, homothétiques

## **1-6-2/Les modèles abstraits :**

Ils font appel aux langages logicomathématiques. C'est le cas, par exemple, des équations mathématiques utilisées en physique pour décrire le mouvement rectiligne uniformément accéléré d'objets en chute libre.

## **1-7/Modélisation informatique :**

Dans ce modèle, la réalité est représentée par un ensemble de programmes informatiques, qui en s'exécutant décrivent l'évolution de la variation d'un ensemble de variables en fonction du temps.

## **1-8/Les outils de modélisation informatique**

Il existe plusieurs outils pour la modélisation informatique : les systèmes à file d'attente, les réseaux de pétri,...etc. Ces outils sont à la base de différents logiciels de simulation.

### **1-8-1/Le modèle entité / association :**

Le modèle Entité/Association a été proposé par *Chen*, en 1976 pour la modélisation des données et des liens existant entre elles, avec des concepts simples et efficaces. C'est une représentation naturelle du monde réel du système à étudier. Il est bâti autour de trois concepts : Entité, Association et Propriétés et permet une description graphique.

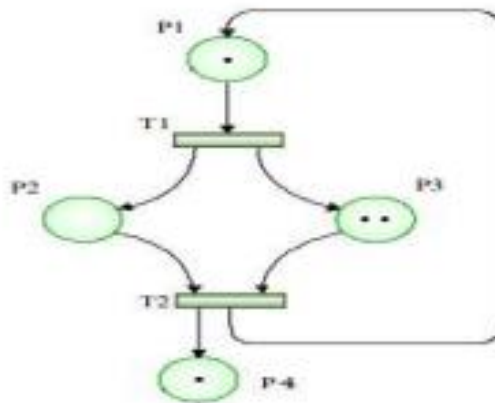
### **1-8-2 /Les réseaux de pétri :**

Les Réseaux de Pétri ont été inventés par Carl Adam Pétri, un mathématicien Allemand, au début des années soixante (1960-1962).

Un RDP (réseaux de pétri) est un graphe biparti (composé de deux types de nœuds) [5]

- Les places ( $P_i$ ) qui permettent de décrire les états du système modélisé. L'ensemble des places est noté  $P = \{P_1, P_2, \dots\}$ .
- Les places peuvent contenir des jetons, représentant généralement des ressources disponibles.
- La distribution des jetons dans les places est appelée le marquage de RDP et chaque place contient un nombre entier (positif ou nul) de jetons.
- Les transitions ( $T_i$ ) qui représentent les changements d'états. L'ensemble de ces transitions est noté  $T = \{T_1, T_2, \dots\}$ .
- Les Places et transitions sont reliées par des arcs orientés.

A chaque arc, on attribue un poids (nombre entier). Par défaut ce nombre est égal à 1.



**Figure 1.2 : Un modèle sur les réseaux de pétri (RDP).**

M0= Marquage initial (c'est la distribution des jetons dans les places)

La modélisation par le Rdp est une méthode descriptive

On n'utilise pas la modélisation par le réseau de pétri car cette méthode donne un modèle graphique sans l'intervenir de l'aspect temporel. Nous basons sur les méthodes analytiques puisque cette méthode est intervenue l'aspect temporel.

### **1-8-3 /Modélisation orientée objet :**

Le modèle orientée objet considère le système comme une collection d'objets dissociés, identifiés et possédant des caractéristiques. Une caractéristique est soit un attribut (une donnée caractérisant l'état de l'objet ou attribut), soit une entité comportementale de l'objet (une fonction ou méthode). La fonctionnalité du système émerge alors de l'interaction entre les différents objets qui le constituent. L'une des particularités de cette approche est qu'elle rapproche les données et leurs traitements associés au sein d'un unique objet. La méthode UML qui se veut actuellement un standard dans les méthodologies orientées objet tire toute sa puissance de ce concept. [6]

### **1-8-4 /Modélisation orientée agent :**

La modélisation OA permet une factorisation (décomposition en sous-systèmes) et une délégation naturelle des tâches, buts et/ou rôles à sous-systèmes multi-agent et leurs agents.

Cette décomposition permet de concevoir plus facilement des systèmes complexes en décomposant le problème en sous-problèmes. Les sous-systèmes sont séparés sur différentes machines et les communications et interactions s'effectuent via des langages de communication qui sont la plupart du temps basés sur la théorie des actes de langage.

## 2 /Simulation des systèmes :

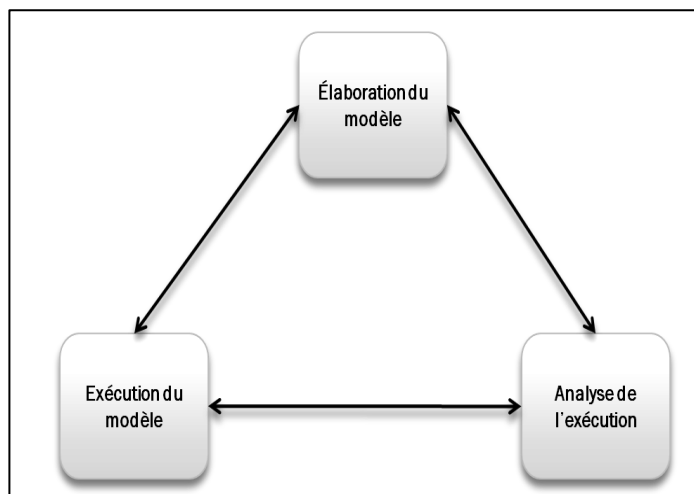
Le terme “simulation” est dérivé du mot latin “SIMULARE” qui veut dire : copier , feindre , faire paraître comme réelle une chose qui ne l’est point. On peut donc très bien dire que simuler le fonctionnement d’un système c’est imiter son fonctionnement au cours du temps en manipulant un modèle. Ceci équivaudrait à la génération d’un historique artificiel des changements d’état du système et l’observation de cet historique pour faire des déductions sur ses caractéristiques de fonctionnement. C’est donc une méthodologie essentiellement pratique qui permet de modéliser aussi bien des systèmes conceptuels (qu’on veut concevoir) que des systèmes existants déjà. Elle peut être utilisée pour

- Décrire et analyser la dynamique d’un système
- Répondre aux questions de type «What If ? » sur le système réel
- Aider à la conception d’un système réel

Pour Fishwick,

*" Computer simulation is the discipline of designing a model of an actual or theoretical physical system, executing the model on a digital computer, and analyzing the execution output" [7]*

.Fishwick illustre par ailleurs ses propos à l’aide du schéma suivant (figure 1.3)



**Figure 1.3 La simulation informatique selon Fishwick, 1997.[7]**

### 2-1/ La simulation informatique :

La simulation informatique est l'un des outils permettant de simuler des phénomènes réels, le but principal de la simulation est de :

- Étudier un système réel de manière à comprendre son fonctionnement interne et/ou à en prévoir son évolution sous certaines conditions.

- Cette étude se fait nécessairement à travers un modèle du système réel qui est utilisé pour réaliser les expérimentations.
- **La théorie de la simulation :**

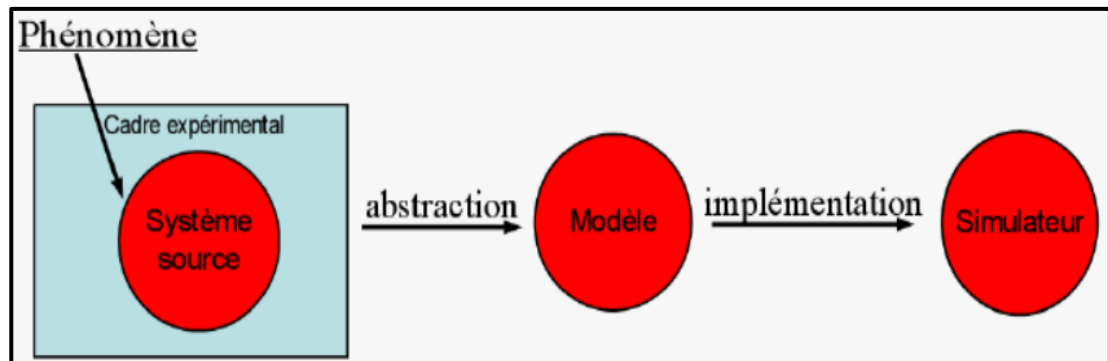


Figure 1.4 Vue d'ensemble de la simulation informatique.

- **Système source :** c'est le système étudié, ou plus généralement le système d'intérêt. Il peut être réel ou virtuel..
- **Cadre expérimental:** définit des observables (des trajectoires d'états observées) pour le systèmes source et le simulateur, qui devront être mis en relation pour statuer sur la validité du modèle, au regard des objectifs de modélisation
- **Modèle :** un modèle est un ensemble de règles, d'équations, de contraintes qui spécifie des relations dynamiques entre des entrées et des sorties
- **Simulateur :** c'est l'entité qui génère le comportement d'entrée sortie décrit par le modèle. Il peut s'agir d'un calculateur, d'un mécanisme physique quelconque ou encore du cerveau humain.

## 2-2 / Catégories de simulation informatique :

On peut distinguer trois catégories de simulations :

### 2-2-1 / La simulation continue :

Dans laquelle l'état du système est modifié en continu au fil du temps sur la base d'un ensemble d'équations différentielles définissant les taux de changement des variables d'état.

### 2-2-2 / La simulation discrète :

Chaque événement se produit à un instant donné dans le temps et marque un changement d'état dans le système. Entre des événements consécutifs, aucun changement dans le système n'est supposé se produire ; ainsi, le temps de simulation peut passer directement à l'heure d'occurrence de l'événement suivant, qui est appelée progression temporelle de l'événement suivant.

## **2-2-3 / La simulation par agents :**

La simulation multi-agents est construite en s'appuyant sur les interactions d'agents évoluant dans un système. L'agent peut être un consommateur, une entreprise, un pays ou bien encore un employé. Un des intérêts de la méthode est de pouvoir travailler avec différents types d'agents aux caractéristiques propres et pouvant varier au cours du temps.

## **2-3 / Les avantages de la simulation :**

- La simulation est souvent moins chère que l'expérimentation et comporte beaucoup moins de risques lorsque l'homme fait partie du système étudié. Les résultats peuvent être obtenus beaucoup plus rapidement.
- La simulation permet d'effectuer des recherches sur un système isolé, en faisant varier les paramètres un à un et en recommençant avec les mêmes conditions initiales.
- La simulation informatique est un outil qui a eu rapidement une bonne réputation dans la conception et d'analyse des systèmes.
- La simulation aide les chercheurs, les ingénieurs et bien d'autres professionnels à prendre dans le temps des décisions intelligentes concernant la conception et l'opération d'un système.
- La simulation est utilisée pour investir une large variété de questions de type "WHAT IF" (QUOI SI) à propos d'un système réel.
- La simulation permet le contrôle du temps.
- La simulation peut être utilisée même si les données considérées sont en quelque sorte superficielle et sommaire.
- La simulation permet d'avoir un aperçu des variables les plus importantes sur la performance d'un système ainsi que les interactions entre elles.

## **3 / Domaine d'application de la simulation :**

Les domaines d'application sont divers. Sont listés ci-dessous quelques classes d'applications et quelques exemples de problèmes typiques rattachés à ces classes [1]:

### ➤ **Systèmes de flux de production**

- Equilibrage de lignes d'assemblage.
- Conception de systèmes de transfert entre des postes.
- Dimensionnement des stocks d'un atelier.
- Comparaison de pilotage.
- Evaluation de la charge prévisionnelle.
- Etude de la synchronisation entre les réceptions des pièces et l'assemblage.

### ➤ **Flux logistiques et systèmes de transport**

- Conception et dimensionnement d'entrepôts.
- Dimensionnement d'une flotte de camions.
- Etude de procédures de contrôle des flux de véhicules en circulation.
- **Production des services**
  - Etude de transactions bancaires.
  - Gestion de restaurants.
- **Systèmes informatiques et télécommunications**
  - Etude de la file d'attente mémoire d'un serveur.
  - Etude des comportements des utilisateurs.
- **Autres classes d'applications**
  - Domaine militaire (coordination des opérations, ...),
  - Gestion d'hôpitaux (personnel, lits, service d'urgence, ...), - la météo, les jeux, ...

## **4 /Buts de la modélisation et de la simulation :**

Plusieurs buts peuvent être réalisés par la simulation :

- Meilleure compréhension du système.
- Mesure de performance du système.
- Dimensionnement/optimisation de systèmes de production (avant/après réalisation).
- Expérimentation du comportement dynamique des systèmes à l'aide de modèles informatiques.
- Réalisation d'un modèle est son utilisation économique (plus simple, moins chère, moins dangereuse, possible, etc.) à la place du système.
- Identification des facteurs critiques (Panne, Blocage).
- Réponse aux questions « que se passe-t-il...si ...? »
- Outil d'aide à la décision en « temps réel » (durant exploitation).
- Outil de formation du personnel (simulateurs de vol, simulateurs de choix...).

## **Conclusion**

Le modèle peut être caractérisé par quelques paramètres physiques, et ils 'adapte aux données expérimentales.

La simulation nous permet de visualiser le comportement d'un système et de l'analyser. Ceci dit, il ne faudrait pas oublier que les résultats obtenus par simulation, dépendent des hypothèses retenues pour construire les modèles.

# **Chapitre 2**

## **Les systèmes multi-agents (SMA)**

### Introduction

L'Intelligence Artificielle Distribuée (IAD) est au carrefour de plusieurs disciplines, telles que la psychologie cognitive, la sociologie, la biologie, et l'informatique. Ces études sont utilisées entre autre pour modéliser et construire des systèmes informatiques dans lesquels les capacités de traitement, de représentation et de raisonnement sont distribuées dans un ensemble de sous- systèmes appelés agents. L'ensemble de ces agents constitue une société appelée Système Multi-Agents. Selon les ressources de chacun, les agents interagissent entre eux pour construire des directives en vue de la résolution de problèmes.

Ce chapitre présente la problématique de l'intelligence artificielle distribuée (IAD), les concepts principaux de la technologie d'agent et des systèmes multi- agent, quelques outils pour le développement de tels systèmes.

### 1/L'intelligence artificielle distribuée (IAD) :

L'Intelligence Artificielle Distribuée (IAD) est née, au début des années 80, de la volonté de remédier aux insuffisances et d'enrichir l'approche classique de l'IA en proposant la distribution de l'expertise sur un groupe d'agents, non soumis à un contrôle centralisé, devant être capables de travailler et d'agir dans un environnement commun et de résoudre les conflits éventuels. En résumé, l'IAD s'intéresse entre autre à la modélisation de comportements intelligents qui sont le produit de l'activité coopérative entre plusieurs agents, d'où la réalisation des systèmes dits

L'idée de base de l'intelligence artificielle distribuée consiste à l'exploitation de plusieurs entités intelligentes pour résoudre un problème complexe. Cependant, la conception de la solution en utilisant ces entités peut être effectuée selon plusieurs façons. Chaque méthode de résolution possède ses propres problèmes et défis en conséquence, l'intelligence artificielle distribuée a été développée en trois pistes différentes :

La Résolution Distribuée de Problèmes (RDP) cette branche consiste à la division du problème en un ensemble de sous problèmes. Puis, on attribue les sous problèmes à des entités différentes afin de les résoudre. Ensuite, nous devons synthétiser les solutions partielles pour trouver la solution finale.

Les Systèmes Multi-Agents (SMA) : Dans cette branche on a un ensemble d'entités dotées de comportements intelligents qui doivent coopérer pour résoudre un problème commun. Chaque entité possède son propre but et ses propres plans.

### 1-1 / Problématique de l'IAD :

Les problèmes que l'IAD s'attache à résoudre, sont les problèmes classiques de l'IA qui ont pris une nouvelle dimension dans le contexte multi-agents et les nouveaux problèmes proprement liés au thème de l'IAD, on peut citer :

- ✓ La modélisation de la connaissance et le problème de sa répartition sur les différents agents regroupés en des sociétés : comment formuler, décomposer, allouer des problèmes et synthétiser les résultats d'un groupe d'agents ?
- ✓ Les problèmes de génération de plans d'actions où il faut prendre en considération la présence d'autres agents. Ces problèmes sont liés au comportement d'un agent au sein d'un groupe. On s'intéresse alors à ses capacités sociales : la répartition des tâches, le partage des ressources, le raisonnement sur les autres agents (pouvoir modéliser leurs connaissances et être en mesure de connaître leurs plans d'actions et de raisonner en fonction de ces plans).
- ✓ La gestion des conflits entre les agents et le maintien de la cohérence des décisions et des plans d'actions.
- ✓ Le problème de la communication : comment permettre la communication et l'interaction entre les agents ? Quel langage et quel protocole faut-il employer ? Une communication dans les univers multi-agents n'est plus une simple tâche d'entrée-sortie, mais doit être modélisée comme un acte pouvant influencer sur l'état des autres agents.
- ✓ Les problèmes spécifiques au groupe d'agents, qui portent sur l'organisation, l'architecture de l'ensemble des agents et les paradigmes de coopération et d'action ;
- ✓ D'autres thèmes de recherche sont présents dans le contexte multi-agents, à savoir, le raisonnement temporel, le raisonnement hypothétique, la représentation de la connaissance imprécise, etc. [8]

### 2 / Définition d'un agent :

La définition de l'agent ne fait pas l'unanimité. En effet, plusieurs définitions peuvent être extraites de la littérature spécialisée. Dans un article scientifique célèbre intitulé (**Is it an Agent, or just a Program? A Taxonomy for Autonomous Agents**), Stan Franklin et Art Graesser présentent la liste de définitions les plus acceptées par les spécialistes de ce domaine.

Un agent peut être défini comme étant :

- Une entité qui perçoit son environnement et agisse sur celui-ci .

- Un système informatique existant dans un environnement dynamique qui peut le percevoir et le contrôler de manière autonome en exécutant les tâches ou en réalisant les objectifs pour lesquels ce système a été conçu.
- Une entité qui exécute continuellement trois fonctions : la perception des conditions dynamiques d'un environnement, l'action afin d'affecter les conditions de cet environnement et le raisonnement afin d'interpréter les perceptions et déterminer les actions .
- Une entité autonome, réelle 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 .
- Un système informatique autonome, réactif et pro-actif.[9]

### **2-1 / Caractéristiques d'un agent :**

#### **2-1-1/ La situation :**

Toutes les définitions notent qu'un agent doit être situé dans un environnement. Un agent peut donc percevoir son environnement et agir sur les changements de cet environnement. Il est important de noter que cette caractéristique ne soit pas intrinsèque pour tous les systèmes informatiques.

#### **2-1-2/L'autonomie :**

Cette caractéristique est, peut-être, la caractéristique la plus importante des agents. On désigne par cette caractéristique la capacité de l'agent d'agir sans l'intervention d'un tiers. En d'autres termes, un agent peut atteindre ses objectifs sans demander l'aide d'un autre agent, système ou utilisateur. En plus, un agent possède le contrôle sur son état et son comportement.

#### **2-1-3/La flexibilité**

La capacité de l'agent de changer son comportement ou sa structure afin d'atteindre ses objectifs. l'agent doit être doté au moins par trois caractéristiques :

- **La réactivité :** la capacité de l'agent de percevoir son environnement et répondre aux différents changements apparus dans le temps adéquat.
- **La pro-activité :** la capacité de l'agent de prendre l'initiative en démontrant des comportements orientés objectifs.
- **La sociabilité :** la capacité de l'agent d'interagir avec les autres agents en utilisant un langage de communication.

### 2-1-4/L'interactivité:

L'agent est social et il rencontre différents types d'interactions (avec d'autres agents, avec un humain ou avec son environnement) qui le conduit à reconsidérer en permanence ses traitements en cours. Cela traduit le fait qu'il s'adapte sans cesse aux changements de son entourage qui pourraient modifier de façon pertinente son comportement à tous les niveaux (objectif, plan, action...etc.)

### 2-2/Type d'agents :

Les systèmes multi-agents sont en général classés en deux principales familles : les systèmes cognitifs et les systèmes réactifs. Contrairement aux systèmes réactifs, les systèmes cognitifs se rapprochent le plus du modèle de sociétés d'experts. Chaque agent cognitif a une représentation explicite de croyances, d'intentions, d'actes de langages, de modèles des autres agents, etc..

#### ➤ **Agents réactifs :**

Les agents réactifs sont basés sur l'intelligence artificielle réactive. Cette école de l'intelligence artificielle est basée sur la possibilité de concevoir des comportements intelligents à partir de comportements simples. En effet, l'intelligence est un phénomène émergent de l'interaction des entités simple. Agents sans intelligence (sans anticipation, sans planification) qui réagissent par stimulus-réponse à l'état courant de l'environnement. Des comportements intelligents peuvent émerger de leur association.

- ✓ Pas de représentation explicite.
- ✓ Organisation implicite/induite
- ✓ Communication via l'environnement.

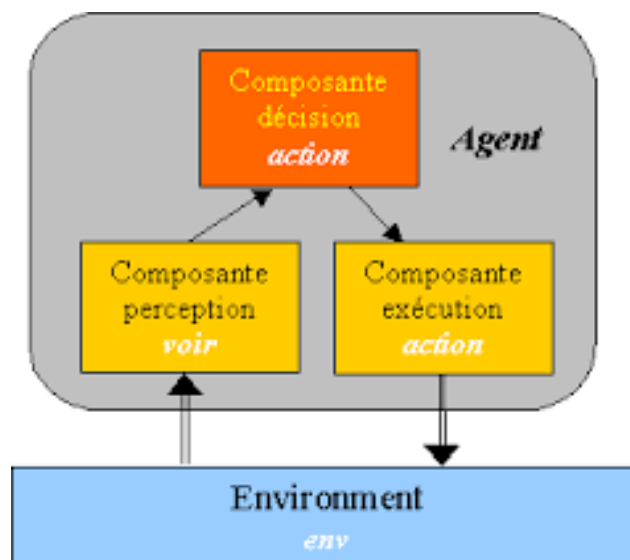


Figure 2.1. Modèle d'agent-réactif.

### ➤ Agents cognitifs :

On parle ici de système d'agents intelligents. Les agents cognitifs sont plus évolués, résultats des recherches menées dans le domaine de l'intelligence artificielle. Ils possèdent une représentation globale de leur environnement et des agents avec lesquels ils communiquent, ils tiennent aussi compte de leurs actions antécédentes. Chaque agent possède une base de connaissances comprenant l'ensemble des informations nécessaires à l'accomplissement de sa tâche, ainsi qu'à l'interaction avec l'environnement et les autres agents.

Les Agents cognitifs caractérisés par :

- ✓ Représentation explicite de l'environnement.
- ✓ Peut tenir compte de son passé.
- ✓ Agents complexes (difficile à implémenter).
- ✓ Petit nombre d'agents.

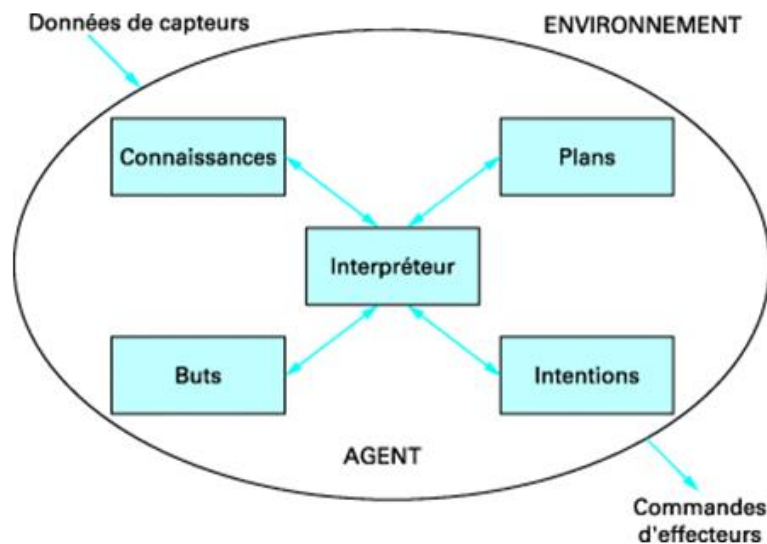


Figure 2.2. structure d'unagent cognitif.

Le tableau suivant résume les différences entre les agents cognitifs et les agents réactifs.

AGENTS COGNITIFS	AGENTS REACTIFS
Représentation explicite de l'environnement	Pas de représentation explicite
Peut tenir compte de son passé	Pas de mémoire locale
Agents complexes	Fonctionnement stimulus/action
Nombre d'agents réduit	Nombre d'agents élevé

Tableau.2.1 : Différences entre agents cognitifs et agents réactifs. [10]

### ➤ Agents hybrides :

La division des agents en deux classes ne représente qu'une vue idéaliste. En fait, ces deux classes ne présentent que les cas extrêmes. En réalité, un agent nécessite des capacités réactives et des capacités cognitives. Ainsi, il est important de combiner les deux approches citées ci-dessus. Un agent hybride est un agent conçu en couches. Les couches inférieures assurent des comportements réactifs. Par contre, les couches supérieures sont responsables de capacités cognitives complexes comme les aspects sociaux. Bien entendu, les couches supérieures manipulent des connaissances alors que les couches inférieures manipulent directement des données. En conséquence, des couches intermédiaires sont responsables de transformer des données en connaissances.

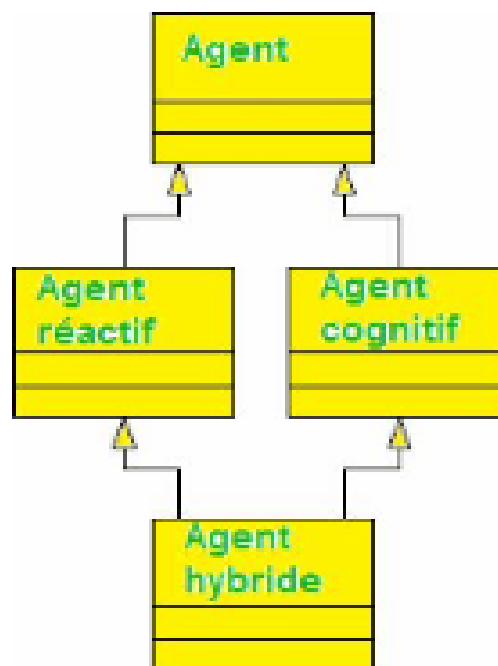


Figure 2.3. Modèle d'agent hybride.

### 3/ Les systèmes Multi-Agents

Un système multi-agent est un système distribué composé d'un ensemble d'agents. Contrairement aux systèmes d'IA, qui simulent dans une certaine mesure les capacités de raisonnement humain, les SMA sont conçus et implantés idéalement comme un ensemble d'agents interagissant, le plus souvent, selon des modes de coopération, de concurrence ou de coexistence [11]

SMA est généralement caractérisé par : [12]

Chaque agent a des informations ou des capacités de résolution de problèmes limitées, ainsi chaque agent a un point de vue partiel :

- ✓ il n'y a aucun contrôle global du système multi agent;
- ✓ les données sont décentralisées;
- ✓ le calcul est asynchrone.

D'autre part, **J.Ferber** donne la définition suivante :

Un système multi-agents est un système composé des éléments suivants :

- Un environnement E : c'est-à-dire un espace disposant généralement d'une métrique.
- Un ensemble d'objets O situés : c'est-à-dire pour tout objet, il est possible, à un moment donné, d'associer une position dans E. Ces objets sont passifs, c'est-à-dire qu'ils peuvent être perçus, créés, détruits et modifiés par les agents
- Un ensemble d'agents A, qui sont des objets particuliers ,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. »

### **3-1 /Architecture d'un SMA :**

Bien que les systèmes multi-agent soient un modèle des systèmes distribués, l'exécution des agents doivent être cohérent. Ainsi, une question fondamentale dans ce domaine consiste au choix d'un type de contrôle de différents agents qui composent le système. Il existe dans la littérature spécialisée deux approches différentes de contrôle des agents :

#### **3-1-1 /Le modèle de tableau noir (l'approche centralisée) :**

Dans cette approche, le système est composé d'un ensemble des agents (appelés de sources de connaissances), un tableau noir qui assurent l'interaction entre les agents et un module de contrôle. Le module de contrôle assure l'exécution de la source de connaissances adéquate. En fait, ce module ordonne les sources de connaissances selon le contenu du tableau noir. Les agents utilisent le tableau noir comme un espace partagé qui assure leur interaction. Ainsi, chaque agent peut accéder au tableau noir pour la lecture ou l'écriture. Les changements dans le tableau noir peuvent déclencher l'exécution d'une source de connaissances. En conséquence, le module de contrôle planifie l'exécution de cette source de connaissances selon son niveau de priorité. La figure 2.4 présente une vue simplifiée de ce modèle

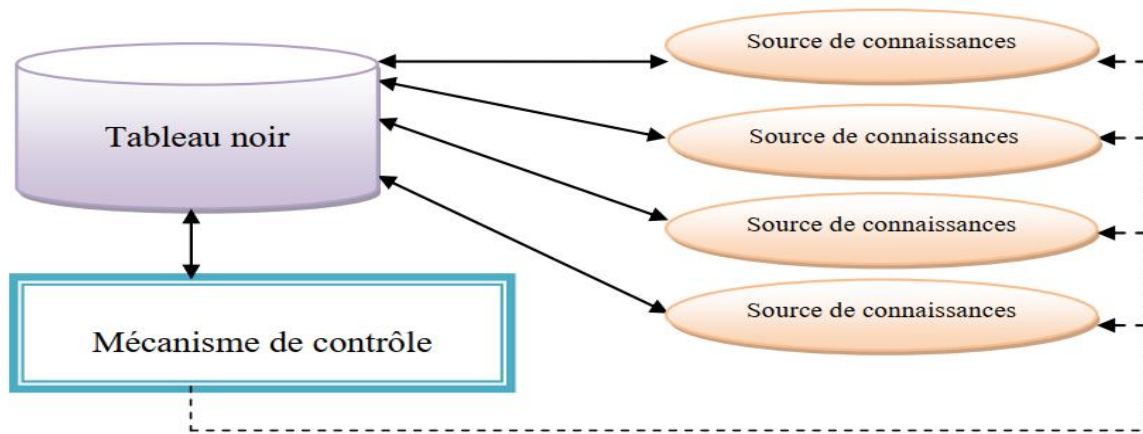


Figure 2.4 : Le modèle de tableau noir.

### 3-1-2/Le modèle d'acteurs (contrôle distribué) :

Le modèle d'acteurs est un autre modèle de l'IAD. Les acteurs sont des entités autonomes qui communiquent par messages asynchrones. Ainsi, il n'existe pas un contrôle centralisé mais chaque acteur possède le contrôle sur lui-même. En conséquence, les acteurs éprouvent de la difficulté d'atteindre des objectifs globaux avec les connaissances locales. La Figure 2.5 montre ce modèle de l'IAD.

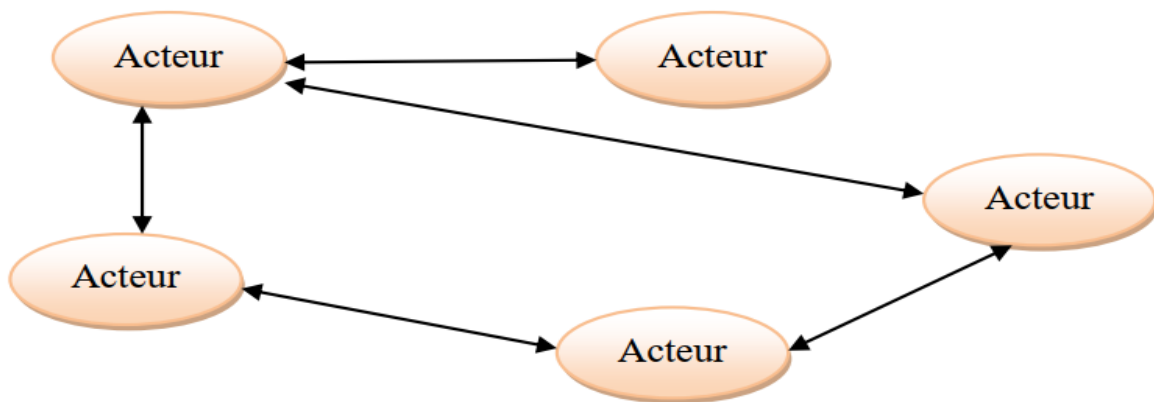


Figure 2.5 : Le modèle d'acteur.

### 3-2 /Interaction dans un système Multi-Agents:

Les agents sont, généralement, caractérisés par un ensemble limité de compétences. En conséquence, l'interaction des agents est une condition importante pour atteindre effectivement les objectifs d'un agent. En fait, un agent isolé n'est qu'une entité handicapée dans le sens où il ne peut pas atteindre ses objectifs. A cause de cette importance de l'interaction des agents, certains auteurs la considèrent comme la raison d'être des systèmes

multi-agents. On ne peut parler d'un système multi-agents en assemblant simplement des agents sans interaction (un système multi-agents est par définition un ensemble d'agents en interaction). Cependant, l'interaction est aussi la source des problèmes dans les systèmes multi-agents.

En général, les interactions sont mises en œuvre par un transfert d'informations entre agents ou entre l'environnement et les agents; soit par perception, soit par communication. L'interaction entre les agents apparaît sous plusieurs modes, qui sont la coopération, la coordination et la négociation.

### **3-2-1 La coopération :**

Les relations entre les agents ne sont pas toujours des relations conflictuelles. Au contraire, les relations des agents peuvent être aussi des relations positives. Par des relations positives, on entend l'augmentation de synergie des agents ou la gestion des interdépendances des tâches de l'agent. En fait, l'existence de ces relations est justifiée par l'inexistence d'un agent qui possède toutes les capacités et les connaissances pour la résolution du problème. Les agents dans ces situations font recours aux techniques de coopération. Généralement, on parle de coopération quand il n'existe pas une situation conflictuelle.

Jacques Ferber définit la coopération entre plusieurs agents comme "une situation dans laquelle, soit l'ajout d'un nouvel agent permet d'accroître les performances du groupe, soit l'action des agents sert à éviter ou à résoudre des conflits potentiels ou actuels".

La situation de coopération apparaît quand les actions de chaque agent satisfont au moins l'une des conditions suivantes:

- ✓ Les agents ont un but en commun et leurs actions tendent à réaliser ce but.
- ✓ Les agents effectuent les actions qui réalisent non seulement leurs propres buts mais aussi ceux des autres

### **3-2-2 / La négociation:**

La négociation est peut-être la méthode de résolution de conflit la plus parlante. L'objectif de la négociation est de trouver une solution qui convienne à tous les agents qui sont en conflits.

On distingue deux types de négociation :

- **La compromission :** pour cela, les agents en question ont besoin d'interagir de nombreuses fois afin de trouver petit à petit des compromis qui aboutissent à une solution envisageable, c'est-à-dire une solution permettant aux agents de continuer vers leurs objectifs. Ces compromis consistent à vérifier l'importance de chaque

contrainte de l'agent et d'abandonner ou mettre de côté les moins importantes. Lorsque les contraintes restantes chez les deux parties sont satisfaisantes, la négociation se termine et le conflit est résolu.

- **La remise en question** : cela ressemble à la compromission, mais on adapte les buts et non les contraintes. Ici, chaque agent cherche à vérifier ses propres objectifs profonds de manière à échanger des informations et intervenir sur ces buts profonds, et non sur des buts intermédiaires. Cela revient à adapter les objectifs intermédiaires, les modifier ou les redéfinir complètement, de manière à garantir l'objectif profond de chaque agent.

### **3-2-3/ Lacoordination:**

Dans les systèmes multi-agents, du fait des interactions quasiment omniprésentes entre les agents générant des organisations plus ou moins complexes selon les systèmes, la notion de coordination apparaît comme une nécessité. En effet, lorsqu'un agent interagit pour une raison qui lui est propre, il doit produire plus d'actions que prévu, des actions qui ne sont pas directement source de satisfaction d'un objectif mais sont des étapes intermédiaires nécessaires à l'accomplissement de ce même objectif. Il y a alors un besoin de coordonner ses actions avec ces agents devenus des collaborateurs.

T. W. Malone définit la coordination comme étant « l'ensemble des activités supplémentaires qu'il est nécessaire d'accomplir dans un environnement multi-agents et qu'un seul agent poursuivant les mêmes buts n'accomplirait pas ». La coordination n'implique pas nécessairement de communication, mais comprend la notion d'influence et le fait que les comportements des agents devant se coordonner doivent s'adapter à un contexte particulier, par exemple deux agents qui doivent éviter une collision mais dont les routes se croisent (Malone 1988). Les agents devant se coordonner doivent donc effectuer un traitement additionnel de l'information. [13]

### **3-3 / Communication dans les systèmes Multi-Agents :**

La communication représente la base de tous les types de l'interaction. Par la communication on entend, la transformation d'une information d'un agent à un ou plusieurs autres agents à l'aide d'un langage articulé ou par d'autres codes. Cependant, la communication n'est pas toujours associée à l'échange explicite des messages. En fait, la communication peut être directe ou indirecte.

Dans la communication directe, les agents changent explicitement des messages. Par contre, dans la communication indirecte, les agents manipulent leur environnement qui représente le

support de la communication. Ce type de communication est appliqué généralement dans les agents réactifs. Ainsi, les fourmis utilisent leurs traces comme moyen de communication. En plus, la communication peut être de type point à point ou de type diffusion. Dans le cas de communication point à point l'émetteur envoie le message à un seul récepteur. Par contre dans le cas de communication par diffusion, un émetteur envoie le message à un ensemble d'agents. Une dernière classification de la communication entre les agents est basée sur l'intentionnalité des agents. La communication peut être un processus intentionnel où les agents ont conscience de l'acte de communication comme elle peut être un incident.

La communication de données, du mode de communication qui peut être :

- ✓ **Direct** : envoi de message
- ✓ **Indirect** : utilisation de tableau noir. Les agents accèdent à une base de données partagée appelée tableau noir dans laquelle les informations sont portées

### **4 / Les Méthodes de conception des Systèmes Multi-Agents :**

Les systèmes multi-agent disposent plusieurs modèles de conception : GAIA, MASE, ALLAADIN...etc.

#### **4-1 / GAIA :**

La méthodologie Gaia est un modèle conceptuel pour l'analyse et la conception des systèmes multi-agent.

GAIA se compose de deux phases principales (figure 2.6) :

- la phase d'analyse qui produit deux modèles : le modèle de rôles et le modèle d'interactions;
- la phase de conception qui transforme les modèles abstraits de la phase d'analyse en modèles moins abstraits. Le processus de conception se base sur les trois modèles : modèle d'agent, modèle de service et modèle de relation[14]

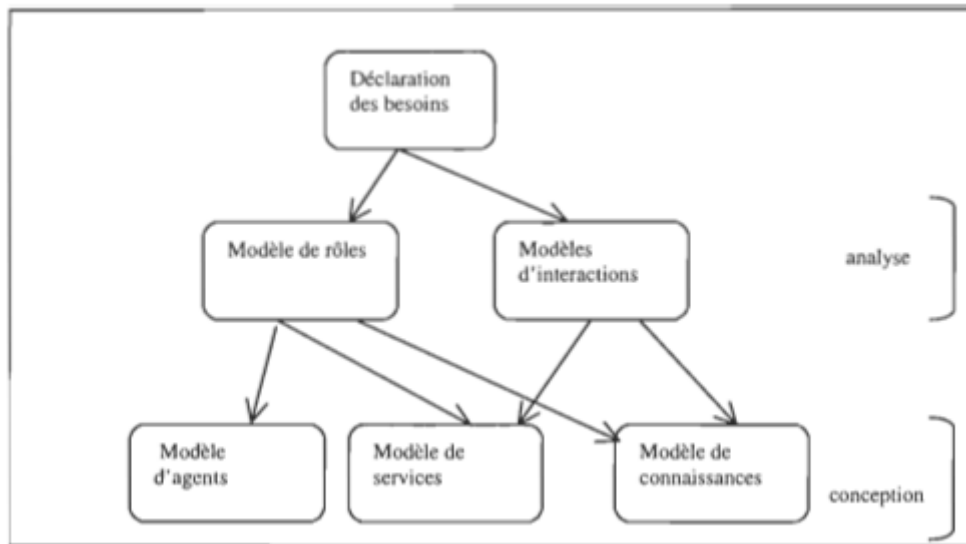


Figure 2.6 : Les modèles de GAIA

### 4-2/ Multi-agent Systems Engineering (MaSE) :

Les auteurs de cette méthodologie définissent un agent comme un objet actif ayant un objectif et un langage de communication. Cette méthodologie utilise les techniques de OMT (Object Modeling Technique) ou de UML (Unified Modeling Language) avec quelques caractéristiques de plus et quelques modifications de la sémantique du paradigme objet pour pouvoir capter les concepts d'agent et les comportements coopératifs des agents. Elle utilise deux langages pour décrire les agents et les SMA : Agent Modeling Language (AgML) et Agent DefinitionLanguage (AgDL). Les différentes phases de MaSE (figure 2.7) sont : Conception du domaine (Domain Level Design); Conception d'agents (Agent Level Design); Conception de composants (Component Design); Conception du Système (System Design). [15]

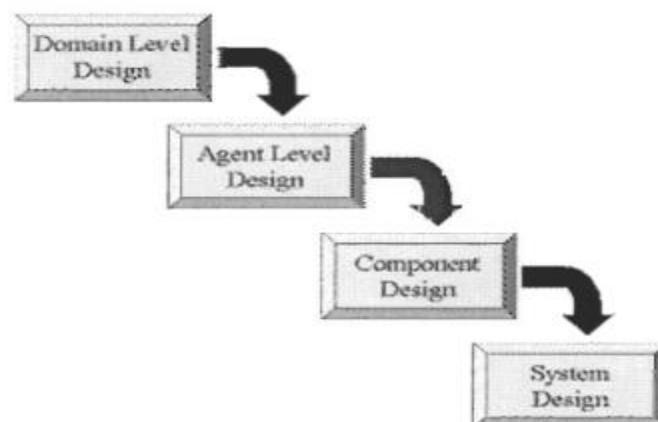


Figure 2.7 : Les différentes phases de MaSE [16]

### **4-3/ ALAADIN :**

Est une organisation centrée méta modèle générique des systèmes multi-agent. Elle définit une structure conceptuelle générale qui sera utilisée dans le développement du système multi-agent. Elle décrit le système multi-agent d'une perspective organisationnelle, jusqu'à l'utilisation du terme d'état mental d'agents (centré agent) ces modèles aussi appelés AGR sont concentré sur trois concepts de base : Agent, Groupe, et Rôle. Un rôle est une représentation abstraite de la position fonctionnelle d'un agent dans un groupe et en même temps, c'est une vue de l'agent assumant ce rôle.

### **4-4/ADELFE :**

Le but de ce projet est de réaliser un Atelier de Développement de Logiciels à Fonctionnalité Emergente (ADELFE). Ces logiciels sont basés sur les techniques des systèmes multi-agents adaptatifs et auto-organiseurs développés à l'IRIT et à ARTAL. Par leur capacité d'adaptation, ces systèmes multi-agents sont aptes à fonctionner dans un environnement à très forte dynamique et à faire face à des imprévus. Ces logiciels sont adaptés pour résoudre des problèmes complexes, distribués pour lesquels un contrôle global est impossible à mettre en œuvre. Avec l'essor des technologies du futur comme Internet, l'évolution des réseaux de machines, ce type de logiciels apporte des solutions aux problèmes de conception de logiciels adaptés à des applications dans ces environnements. Actuellement, il n'existe pas de méthodologie et d'aide à la conception pour ces systèmes adaptatifs. La réalisation de cet atelier de développement consiste à donner :

- une méthodologie de conception ;
- une notation basée sur UML ;
- une plate-forme qui fournira un outil de modélisation visuel, une bibliothèque de composants permettant des simulations et un prototypage rapide.[17]

### **4-5 / La méthodologie Ingénias :**

Dans Ingénias, l'approche générale pour spécifier un système multi-agents consiste à diviser le problème en plusieurs aspects plus concrets qui forment les différentes vues du système. Le but global d'Ingénias est de fournir un ensemble de méthodes et d'outils de développement génériques permettant le développement des systèmes multi-agents. Ingénias est fondée sur la définition de méta-modèles décrivant les différents aspects d'un système multi-agents ainsi que leurs relations. Sur cet aspect, Ingénias s'inscrit dans la continuité de la méthodologie MESSAGE [18] qui a appliqué les techniques de méta- modélisation dans le domaine des systèmes multi-agents. Ingénias a fait évoluer les méta- modèles en incorporant

d'autres concepts. Les modèles sont ensuite construits avec des instances des entités méta-modèles.

Ainsi, le méta-modèle est considéré comme la définition des langages de modélisation pour la description des systèmes multi-agents [19] [20].

### **4-6 /La méthodologie Tropos :**

Tropos[21]est une méthodologie de développement fondée sur les concepts utilisés en ingénierie des besoins. Elle adopte une approche transformationnelle, dans le sens où, à chaque étape, les modèles vont être raffinés de manière itérative par ajout ou suppression d'éléments ou de relations dans les modèles. [22]

### **5/Les plates-formes Multi-Agents :**

Tout comme les méthodologies, il existe une multitude de plates-formes multi-agents dédiées à différents modèles d'agent. Les plates-formes fournissent une couche d'abstraction permettant de facilement implémenter les concepts des systèmes multi-agents. D'un autre coté, elle permet aussi le déploiement de ces systèmes. Ainsi, elles constituent un réceptacle au sein duquel les agents peuvent s'exécuter et évoluer. En effet, les plates-formes sont un environnement permettant de gérer le cycle de vie des agents et dans lequel les agents ont accès à certains services.

Comme le choix d'une plate-forme d'agent a une grande influence sur la conception et la mise en œuvre des agents, FIPA a produit les normes qui décrivent comment une plate-forme d'agent devrait être. Ces normes existent pour assurer une conception uniforme des agents indépendamment de la plate-forme.

Dans ce qui suit, nous présentons quelques plates-formes. Cette liste n'est pas exhaustive. Elle représente cependant les plates-formes les plus utilisées et les plus citées dans la littérature. Pour comparer les différentes plates-formes, nous appliquons un cadre de comparaison qui se base sur les éléments suivants :

- **les types de systèmes visés :** certaines plates-formes sont conçues pour développer des applications dans des domaines spécifiques tel que la simulation, les systèmes distribués, etc.
- **les modèles d'agents supportés :** certaines plates-formes imposent un modèle d'agent spécifique (agent BDI, agent collaboratif, etc.).

- **les méthodologies** : certaines plates-formes proposent une méthodologie de développement, d'autres plates-formes sont utilisées par certaines méthodologies orientées agent.
- **le langage d'implémentation appliqué** : généralement, le langage d'implémentation qui est adopté par les plates-formes est Java. Nous étudierons les classes abstraites qui sont fournies par la plate-forme, qui aident à architecturer les applications.

### 5-1 /Jack :

Jack est décrit comme étant un environnement pour construire, exécuter et intégrer des systèmes multi-agents commerciaux, écrits en Java et utilisant une approche orientée composants.

La particularité de Jack est sa forte orientation vers la programmation agent, ce qui mène à une grande versatilité, l'architecture des agents pouvant aller du comportement simplement réactif au BDI complet, à l'aide de l'architecture fournie ou non. D'après [Ricordel 2001], la documentation fournie est très technique, et ne couvre pas les aspects méthodologiques, en fait comme nous l'avons déjà signalé la conception des SMA est encore mal appréhendée, particulièrement pour les aspects de l'analyse et de la conception. Le déploiement manque également de support pour cette plate-forme.[23]

### 5-2 /ZEUS :

Zeus est un environnement complet qui utilise une méthodologie appelée « rôle modeling » pour le développement de systèmes collaboratifs. Les agents possèdent trois couches. La première couche est celle de la définition où l'agent est vu comme une entité autonome capable de raisonner en termes de ses croyances, ses ressources et de ses préférences. La seconde couche est celle de l'organisation. Dans celle-ci, il faut déterminer les relations entre les agents. La dernière couche est celle de la coordination. Dans celle-ci, on décide des modes de communication entre les agents, protocoles, coordination et autres mécanismes d'interactions. [23]

### 5-3 /Agent-Builder :

Agent-Builder est une suite intégrée d'outils permettant de construire des agents intelligents. Cette plate-forme est adaptée pour tous types de systèmes. L'élaboration du comportement des agents se fait à partir du modèle BDI et du langage AGENT-0. KQML est utilisé comme langage de communication entre les agents. AgentBuilder est composé d'une interface graphique et d'un langage orienté agent permettant de définir des croyances, des engagements

et des actions. Il permet également de définir des ontologies et des protocoles de communications inter-agents.

### 5-4 / MADKIT :

MadKit (Multi agent development Kit), est une plate-forme multi-agents, implémentée en java et a l'originalité d'être basé sur un modèle organisationnel « modèle AGR » plutôt qu'une architecture d'agent ou un modèle d'interaction spécifique. L'utilisation de groupes et de rôles associés à des agents est mis en œuvre tant en tant qu'outil de modélisation et de conception pour les développeurs de systèmes multi-agents, que de principe d'architecture de la plate-forme elle-même. [24]

### 5-5 / JADE :

La plate-forme JADE (Java Agent DEvelopmentframeworkestcertainement celle qui est la plus utilisée par la communauté des systèmes multi-agents. JADE permet de développer et d'exécuter des applications distribuées basées sur le concept d'agents et d'agents mobiles. Elle est compatible à la plate-forme FIPA.

Elle est implémentée en JAVA et possède trois modules principaux :

- DF « Directory Facilitator » fournit un service de « pages jaunes » à la plate-forme.
- ACC « Agent Communication Channel » gère la communication entre les agents.
- AMS « Agent Management System » supervise l'enregistrement des agents, leur Authentification, leur accès et l'utilisation du système. [25]

## Conclusion

Les systèmes multi-agents permettent de mieux décomposer un problème complexe en un ensemble d'entités autonomes et interagissant entre eux afin d'atteindre un objectif donné. Ainsi, on peut maîtriser la complexité des problèmes complexes en élaborant des modèles basés SMA qui seront mis en œuvre sous forme de simulateurs logiciels permettant de faire évoluer un groupe d'agents dans le temps.

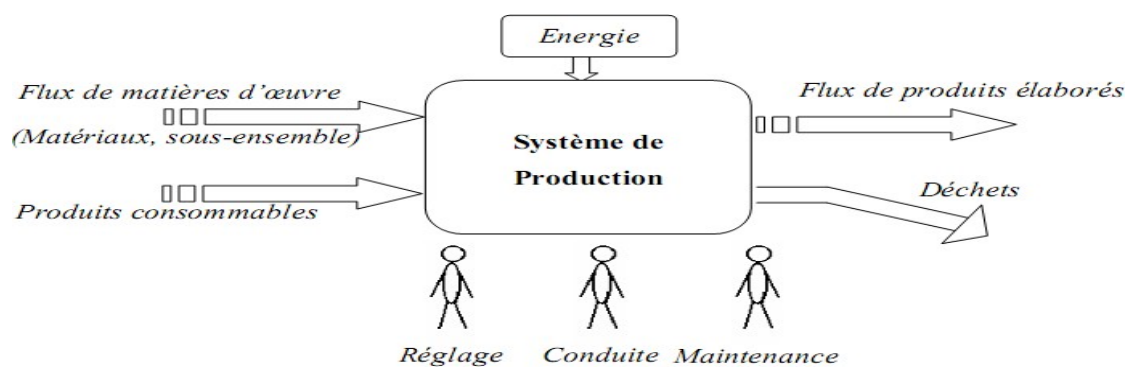
Actuellement le domaine informatique peut être le domaine le plus actif. Parmi les nouveautés, la technologie Multi-agent c'est ouvrir la voie des changements radicaux dans une large échelle des sciences qui sont liées aux plusieurs domaines tel que l'intelligence artificielle distribuée, l'économie, les systèmes distribués...etc.

# **Chapitre 3**

## **Système Industriel Complexe**

### 1/Introduction :

Les systèmes industriels qui résultent d'un processus d'ingénierie basé sur le cycle en V entrent typiquement souvent dans cette catégorie. La réalisation de tels systèmes industriels fait en effet appel à des centaines d'ingénieurs issus de nombreuses spécialités différentes. Dans l'industrie automobile, un projet "véhicule" représente ainsi en moyenne une charge de 1.500 hommes-années de travail, répartie sur 4 ans, fait intervenir 30 à 50 corps de métiers différents et met en jeu des budgets de l'ordre du milliard d'euros. Dans le contexte des systèmes d'information, les grands projets d'infrastructure informatique s'étalent souvent sur plusieurs années en impliquant des équipes de plusieurs centaines d'informaticiens à temps complet et des budgets de plusieurs centaines de millions d'euros. La conduite de ce type de projets pose donc aujourd'hui des difficultés majeures car personne n'arrive plus à appréhender des systèmes de ce type dans leur globalité.



**Figure3.1:Vuegénéraled'unystème deproductionindustriel**

### 2/La complexité d'un système industriel :

La complexité des systèmes industriels reste certes une notion floue et subjective, mais elle correspond à une réalité industrielle forte : elle caractérise les systèmes dont la maîtrise de la conception, de la maintenance et de l'évolution pose des problèmes importants, liés à leur taille et au nombre de technologies utilisées, qui rendent l'ensemble difficile à appréhender. En cela, les systèmes industriels complexes se distinguent d'autres systèmes techniquement compliqués mais dont les difficultés de conception peuvent être résolues par un seul ingénieur talentueux.

Une des raisons majeures de la complexité de ces systèmes industriels vient de ce qu'ils sont souvent eux-mêmes des systèmes de systèmes et qu'ils résultent donc de l'intégration de nombreux autres systèmes industriels relativement complexes.

Les systèmes complexes peuvent être définis comme des systèmes comportant un grand nombre de composants, relativement simples mais interdépendants. La compréhension et la prévision du comportement de tels systèmes peut être difficile à anticiper ou à comprendre même lorsque le comportement de chacun des composants constitutifs est simple. L'hétérogénéité des composants et l'éventuelle présence de boucles de rétroaction rendent plus difficile encore la prédiction de leurs comportements, imprévisibilité encore accentuée en présence de décisions humaines dans les systèmes dynamiques.[26]

Notre application concerne la simulation d'un système de Compression du gaz d'alimentation au niveau de ce groupe (Complexe GNL1/K) et Ce chapitre présente la définition, l'organisation, et les caractéristiques principales d'un système industriel complexes et présente l'architecture et le fonctionnement de Compression du gaz tout en présentant les différents composants matériels entrant dans le cadre de notre étude

### **2-1/ Les entrées/sorties d'un système industriel:**

Tout système traite des entrées et élabore des sorties, les E/S apparaissent sous trois formes : matière, énergie et information.

#### **2-1-1 / Les E/S des matières d'œuvre :**

La fonction du système est le traitement de ces matières d'œuvre pour les faire passer d'un état initial à un état final. Ainsi, sa tâche consiste en un ensemble de transformations.

#### **2-1-2 / Les E/S d'information :**

Ce sont des signaux échangés aux niveaux interne et externe du système. Les informations échangées sont essentielles pour le dialogue homme/machine ou pour la communication avec l'extérieur (vers d'autres systèmes). Il est inconcevable qu'un système quel qu'il soit ne puisse pas communiquer avec son environnement. Cette tâche doit donc exiger quel que soit le système.

#### **2-1.3 Les E/S d'énergie :**

Plusieurs sources sont utilisées par les équipements industriels pour la mise en œuvre des processus physiques, La sortie d'énergie correspond soit à la sortie principale lorsqu'elle est matière d'œuvre, Soit à des énergies résiduelles ou perdues lors de la conversion énergétique ou par frottement.

#### **2-1.4 / Les E/S intempestives :**

Ce sont des grandeurs qui produisent des effets indésirables sur l'état de fonctionnement normal du système, ils s'agit de perturbations (défaut d'une matière d'œuvre, parasites

énergiques...), et de sorties de type déchets causant des nuisances (projections, bruits, fuites...). Souvent, la nature même de ces entrées est une source importante de difficultés pour l'établissement de modèle de fonctionnement et de décision fiable. Pour cela, ces entrées doivent être contrôlées de telle sorte qu'elles ne puissent pas causer de nuisances énormes dans les systèmes de production.

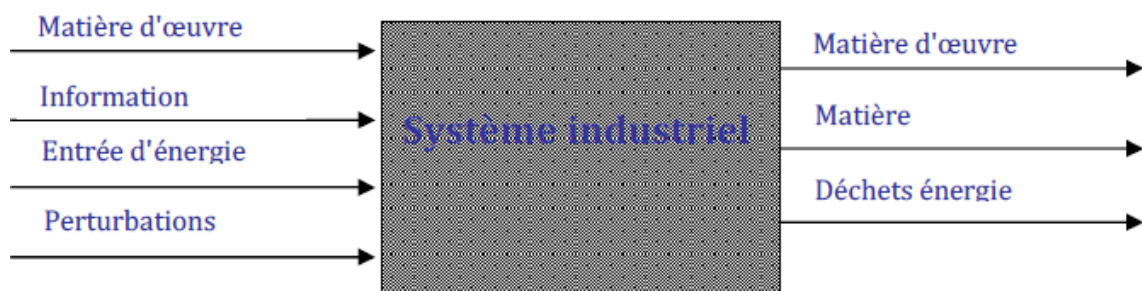


Figure 3.2: Les E/S d'un système industriel

### 3/Processus industriel :

Un processus (dit encore procédé ou processus) industriel est l'ensemble des opérations permettant de transformer des matières premières en biens de consommation.

#### 3-1 /Processus :

Dans un environnement industriel, un processus est considéré comme une séquence d'actions ou une action qui est conditionnée sur une entrée et produit un signal de sortie, les données d'entrée peuvent être une perturbation ou une amplitude d'ordre. La notion de processus introduit une notion dynamique qui peut expliquer l'évolution d'un phénomène ou d'un système dans le temps, qui comprend une partie arbitraire, car un processus complexe peut être décomposé en plusieurs processus d'interaction plus basique. Le système de contrôle de processus Les processus de fabrication ou de production sont souvent appelés systèmes de contrôle de processus, ou l'automatisation de systèmes de contrôle de processus industriels ordres.

#### 3-2 Perturbation

Les perturbations sont des signaux d'entrée agissant sur le procédé, mais qui ne sont pas connus ou sur lesquelles on ne désire pas agir.

### **3.3 / Grandeurs de commande**

Les grandeurs de commande sont des signaux d'entrée qui agissent aussi sur le procédé sur lesquelles les valeurs sont imposées au processus.

### **4 / L'informatique industrielle :**

L'informatique industrielle couvre l'ensemble des techniques de conception d'analyse et de programmation de systèmes à base d'interfaçage de l'informatique avec de l'électronique, électrotechnique, mécanique, robotique, etc., à vocation industrielle. Elle concerne l'utilisation de l'outil informatique pour la fabrication de produits industriels, du bureau d'études (conception assistée par ordinateur) à leur production (fabrication assistée par ordinateur, automatique, robotique) en passant par la logistique, la gestion des stocks, etc.[27]

Une autre définition courante de l'informatique industrielle est qu'elle concerne les programmes informatiques dont les variables représentent en quelque sorte des grandeurs physiques ; par exemple la température et la pression d'un réservoir, la position d'un bras de robot, etc.

#### **4-1 / Domaines d'application :**

Les domaines d'application de l'informatique industrielle sont très vastes, par exemple :

- l'aviation (appareil de mesure, tableaux de bord, ...).
- L'aérospatiale (pilotage des engins spatiaux).
- L'automobile (assistance à la conduite, système de freinage, ...).
- La robotique, les alarmes, instrumentation, médicale, téléphonie.
- Terminaux de paiement pour carte bancaire ...
- Conception assistée par ordinateur, fabrication assistée par ordinateur.
- Système de contrôle, supervision et commande de processus, qui nous intéresse dans notre travail.

### **5/Système De Compression Du Gaz Démercurisation :**

#### **5-1/Présentation du Complexe GL1.K SKIKDA :**

Le complexe GL1.K, l'un des principaux pôles d'hydrocarbures d'Algérie, est situé à environ quatre kilomètres à l'est de la ville de Skikda, wilaya de Skikda située au nord-est de l'Algérie. Il est implanté dans la zone industrielle de Skikda située dans le golfe de Stora.

Sa capacité annuelle de production est de 13.2 millions de M3 de gaz naturel liquéfié et une capacité de stockage de 196 000 M3 de GNL.



Figure 3.3: Vue Aérienne du complexe GL1.K

### 5-2/ Description générale de l'activité :

Le complexe GL1.K comprend les installations de réception du gaz naturel produit par les champs gaziers d'HASSIR'MEL qui se situent à 550 km au sud d'Alger.

Après traitement dans les trains de liquéfaction, le gaz liquéfié est stocké à la température d'environ  $-162^{\circ}\text{C}$  dans trois bacs cryogéniques d'une capacité totale de 196 000 m<sup>3</sup> avant d'être exporté.

Le complexe de liquéfaction du gaz naturel comprend actuellement: 3 unités de liquéfaction :

- Une unité de stockage et de chargement de GNL;
- Une unité de traitement, de stockage et d'expéditions de GPL;
- Une unité de production d'isobutane

Les unités annexes/utilités (stockage gazoline, réseau de torche...). À la sortie du gisement de HASSI R'MEL, le gaz naturel (GN) est un mélange d'hydrocarbures dont la teneur en méthane est supérieure à 80%. Il contient également de l'éthane (C<sub>2</sub>), du propane (C<sub>3</sub>), du butane (C<sub>4</sub>), du pentane (C<sub>5</sub>) et des traces d'hydrocarbures lourds (C<sub>6</sub>+).

Le GN peut aussi contenir d'autres constituants tels que de l'hydrogène (H<sub>2</sub>), de l'azote (N<sub>2</sub>), du gaz carbonique (CO<sub>2</sub>), des vapeurs d'eau, ainsi que des impuretés sous forme de poussières.

Les objectifs principaux du complexe de Skikda sont de : traiter et liquéfier le gaz naturel, qui sera destiné à l'exportation. d'extraire les sous-produits « nobles » présents dans le GN, à

savoir l'éthane, le propane, le butane et le pentane.

### **5-3 /Description du système :**

Le système de démercuration est situé en amont du système de la tour de lavage 15-MD01 (Volume I, section 3-1) et reçoit du gaz d'alimentation du système de déshydratation (Volume 1, section 2-2). Le gaz d'alimentation descend à travers l'adsorbeur de mercure 14-MD01, qui contient un lit de charbon actif imprégné de soufre qui réduit le niveau de mercure dans le gaz à moins de 5 ng/Nm<sup>3</sup>. Si les niveaux de mercure ne sont pas réduits, les unités en aval et les équipements en aluminium risquent une défaillance mécanique due à la corrosion par le mercure. Le lit de l'adsorbeur de mercure a une durée de vie garantie d'au moins 9 ans.

Le gaz passe alors à travers le filtre à gaz traité 14-MD02 qui enlève 99,5% des particules entraînées de 10 microns et plus. Il est rare d'avoir à changer les éléments du filtre. Une vanne de by pass est installée autour du filtre à gaz traité pour faciliter le remplacement des éléments du filtre.



**Figure 3.4: système de démercuration**

### **5-4/Exploitation :**

#### **5-4-1 Marche normale :**

Ce chapitre décrit les principes de fonctionnement du système de démercuration. Consulter le paragraphe 3.2 pour une liste des variantes d'exploitation usuelles de ce système.

### **5-4-2 Adsorbteur de mercure 14-MD01 :**

Le gaz d'alimentation est fourni par le système de déshydratation et circule de haut en bas dans l'adsorbteur de mercure 14-MD01. L'adsorbteur de mercure est opéré manuellement. L'opérateur doit vérifier les chutes de pression à travers le lit comme indiqué par 14-PDI1003 pour déterminer quand il faut remplacer ou inspecter le lit adsorbant de mercure. Des pressions différentielles anormalement élevées peuvent indiquer que le lit est encrassé et doit être remplacé. Un by pass est installé autour de l'adsorbteur de mercure pour permettre à l'usine de fonctionner pendant le remplacement du lit.

### **5-4-3 Filtre à gaz traité 14-MD02 :**

Après avoir traversé l'adsorbteur de mercure 14-MD01, le gaz d'alimentation passe par un filtre à gaz traité 14-MD02 qui retient 99,5% des particules entraînées de 10 microns et plus. Ce sont des particules de poussière formées par l'érosion du charbon actif adsorbant et qui sont causées par la charge initiale ou l'attrition (frottement) pendant l'exploitation. Le filtre à gaz traité est opéré manuellement. L'opérateur doit vérifier les chutes de pression à travers le filtre comme indiqué par 14-PDI1005 pour déterminer quand il faut inspecter et entretenir le filtre. Un by pass est installé autour du filtre à gaz traité pour permettre à l'usine de fonctionner pendant que les éléments du filtre sont nettoyés et changé.

### **5-5/Paramètres d'exploitation usuels :**

Le tableau suivant donne les variables opératoires usuelles pour le système de démercurisation.

<b>d'identification</b>	<b>Description</b>	<b>Valeur normale</b>	<b>Unités</b>
I4-FDI-1003	Indicateur de pression différentielle de l'adsorbteur de mercure	0.1	bar
I4-FDI-1005	Indicateur de pression différentielle du filtre à gaz traité	0.1	bar

**Tableau 3.1 Variables d'exploitation pour le système de démercurisation**

### **5-6/ Exploitation provisoire :**

#### **5-6-1/Adsorbteur de mercure :**

Quand le lit de l'adsorbteur de mercure 14-MD01 est encrassé et doit être remplacé, il est procédé à l'isolement de l'adsorbteur. Pendant l'isolement de l'adsorbteur de mercure, le gaz

d'alimentation passe par le by pass du système et s'écoule directement dans la tour de lavage 15-MD01.

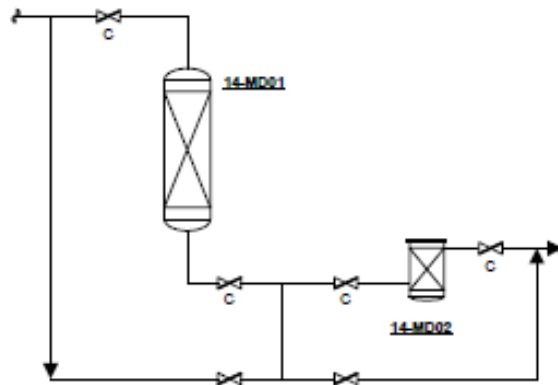


Figure 3.5 Adsorbeur de mercure contourné

### 5-6-2 / Filtre à gaz traité :

Lorsque les éléments du filtre à gaz traité 14-MD02 sont bouchés et doivent être nettoyés ou changés, le filtre est isolé. Pendant la période d'isolement du filtre à gaz traité, le gaz d'alimentation passe par un système de by pass et s'écoule directement vers la colonne d'épuration.

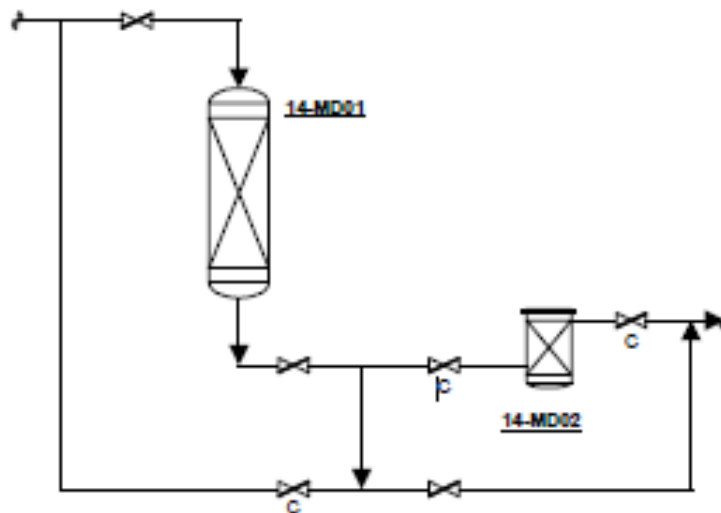


Figure 3.6 : Filtre à gaz traité contourné

### 5.7 Démarrage :

Cette section donne une vue d'ensemble des étapes nécessaires pour la mise en service du système de démercuration. Pour les besoins de cette section, il faut supposer que les conditions suivantes sont présentes. Les utilités sont en service, le gaz d'alimentation est

disponible, les unités en aval sont prêtes pour recevoir le gaz d'alimentation et le système est inerté et prêt au démarrage.

1. Pressuriser progressivement le système de démercuration (à une allure qui ne dépasse pas 1 bar par minute) à la pression du gaz d'alimentation en utilisant le by pass d'admission de 2'' autour de la vanne d'isolement principale sur la ligne NG-13023-36.
2. Chasser l'azote déplacé vers la torche chaude en utilisant le by pass de la 14-PRV-1009. Vérifier toutes les brides, presse-étoupes, joints d'étanchéité et connexions d'instrument pour prévenir les fuites.

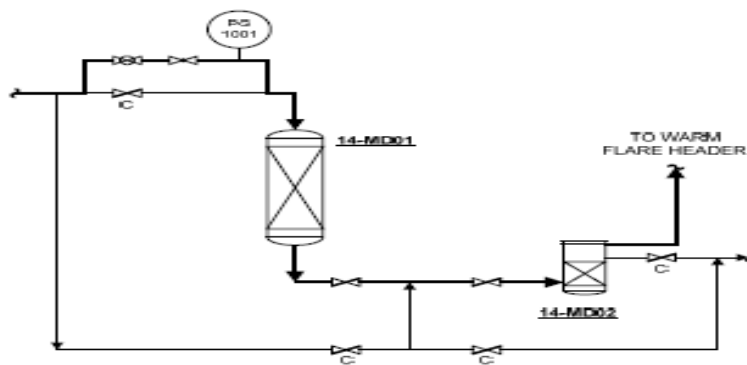


Figure 3.7 Étapes de démarrage 1 et 2

1. Quand la pression est équilibrée de part et d'autre de la vanne d'isolement principale, la vanne d'isolement 36'' en aval du filtre à gaz traité 14-MD02 peut être ouverte.
2. S'assurer que les vannes de by pass de l'adsorbant de mercure 14-MD01 et du filtre à gaz traité 14-MD02 sont fermées.

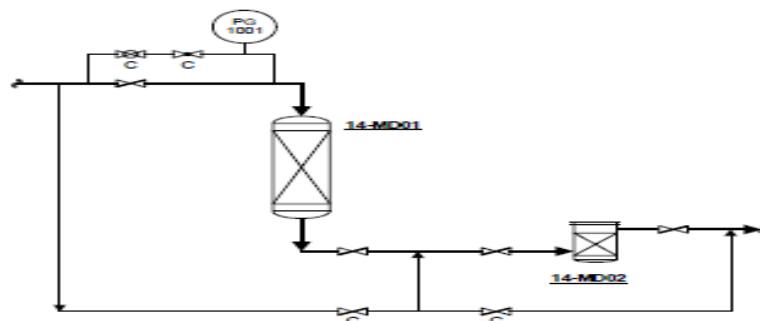


Figure 3.8 Étapes de démarrage 3 et 4

### 5-8 Arrêt :

#### 5-8-1 Arrêt normal :

Les étapes suivantes sont basées sur un arrêt programmé du système de démercuration.

1. Fermer la vanne d'admission d'isolement et les vannes de by pass de 2'' en amont de l'adsorbeur de mercure 14-MD01.
2. Fermer la vanne d'isolement des effluents du filtre à gaz traité 14-MD02.
3. S'assurer que les vannes de by pass de l'adsorbeur de mercure 14-MD01 et du filtre à gaz traité 14-MD02 sont fermées.
4. S'assurer que les vannes d'isolement de la ligne NG-14001-36'' entre 14-MD01 et 14-MD02 sont ouvertes.
5. Dépressuriser le système progressivement vers la torche chaude à travers le by pass manuel de 14-PVR-1009. Limiter l'allure de dépressurisation à 1 bar par minute.
6. La dépressurisation est terminée quand la pression indiquée sur 14-PG-1004 est voisine de celle de l'atmosphère.

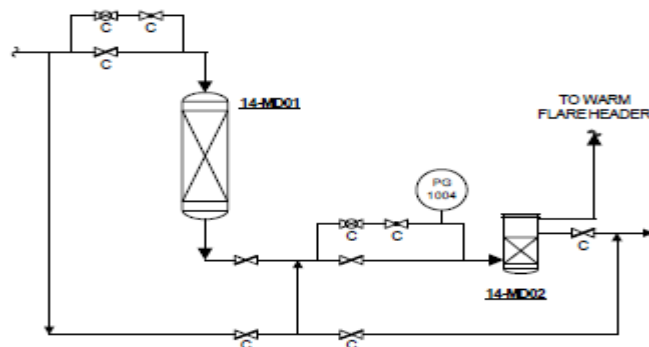


Figure 3.9 Arrêt normal

#### 5-8-2/Arrêt d'urgence :

En cas d'urgence nécessitant un arrêt du système de démercuration, le système doit être arrêté conformément aux procédures ci-dessus.

- Perte de gaz d'alimentation

Les pertes de gaz d'alimentation ne créent pas de problème dans le système de démercuration parce que le système reste statique jusqu'à ce que les conditions soient stabilisées et que le débit de gaz soit rétabli.

### 5-9 /Sécurité :

#### 5-9-1 Systèmes critiques d'arrêt de sécurité :

Sans objet.

#### 5-9-2 Liste des soupapes de sécurité :

La liste des soupapes de sécurité du système de démercuration comprend toutes les soupapes associées au système

Numéro de vanne	Description/emplacement	Pression de tarage (barg)
14-PRV-1000A/B	Gaz d'alimentation/refoulement adsorbant de mercure vers torche chaude	75.00
14-PRV-1009	Effluents/surpression filtre à gaz traité vers torche chaude	75.00

**Tableau 3.2. Liste des soupapes de sécurité du système de démercuration**

#### 5-9-3 Équipement de lutte contre l'incendie :

La liste des équipements de lutte contre l'incendie du système de démercuration comprend les équipements anti incendie associés à se système

Description	Quantité
<b>BBF – Tamis moléculaire / Déshydratation</b>	
63-SP-008 Moniteur eau d'incendie	1
63-SP-005 Couverture anti-feu	1
63-SP-010 Dévidoir à flexible incendie	2
63-SP-011 Armoire lance à eau 4 directions	1
63-SP-024, Moniteur d'eau de réseau d'extinction incendie monté sur prise d'eau 4 directions	1
63-SP-050 Moniteur à eau d'incendie (Élevé)	1
<b>BBH – Décarbonatation</b>	

**Tableau 3.3. Liste de l'équipement anti-incendie du système de démercuration**

Description	Quantité
63-SP-005 Couverture anti-feu	1
63-SP-010 Dévidoir à flexible incendie	1
63-SP-011 Armoire lance à eau 4 directions	2
63-SP-024 Moniteur d'eau de réseau d'extinction incendie monté sur prise	4
63-SP-050 Moniteur à eau d'incendie (Élevé)	1

**Tableau 3.4 Liste de l'équipement anti-incendie du système de démercurisation**

### **Conclusion**

Les systèmes industriels complexes sont caractérisés, d'une part, par la nature matérielle de ses composants, d'autre part, par la richesse des interactions mises en jeu. Les composants sont en effet en interaction entre eux, mais aussi avec l'environnement matériel du système. Ils peuvent le percevoir et agir sur lui, tout comme l'environnement agit sur eux. Dans ce chapitre nous avons identifié en particulier les systèmes d'informatiques industriels et traité le processus de COMPRESSION DU GAZ au niveau du complexe GL1/K.

Ceci représente la première étape dans la démarche de la réalisation d'un notre simulateur

**Chapitre 4**  
**Une Architecture à base d'agent pour la**  
**simulation d'un**  
**système industriel**

## 1/Introduction :

Devant la diversité de champs d'activités des systèmes industriels, il est difficile de trouver une architecture qui peut être qualifiée de référence. Une architecture générique comportant tous les concepts nécessaires pour modéliser et assurer les activités de cycle de vie, pourrait être un noyau pour une diversité de système industriel, c'est-à-dire, quelque soit le secteur d'activité étudié, le concepteur peut utiliser cette architecture comme référence pour développer une place de le système industriel en adoptant les concepts de base de cette architecture suivant les spécification de ce secteur.

Les systèmes multi agents sont devenus un paradigme dominant dans le domaine de développement des systèmes distribués complexes. L'importance des concepts de ce paradigme est due à son aptitude à la modélisation des connaissances et des systèmes complexes, distribués, coopératifs et intelligents. Ces aspects rendent intéressante l'adoption d'une approche multi agents pour l'établissement d'un système industriel.

Les mécanismes de coordination et de communication préconisés par l'approche agent fournissent des solutions élégantes. L'idée est d'utiliser les concepts des systèmes multi agents pour assurer les différentes activités de cycle de vie d'un système industriel et ainsi, adapter les solutions fournies pour résoudre les différents problèmes rencontrés durant l'établissement d'un système industriel.

Dans ce chapitre, nous allons proposer une architecture générique basés agent comportant tous les concepts nécessaires pour assurer le processus de cycle de vie d'un système industriel.

Nous commençons la description de notre approche par la motivation de choix de cette approche. Ensuite, nous présentons la description de l'architecture proposée et les structures internes des différents agents, ainsi que leurs rôles dans le système industriel.

## 2/Description de l'architecture proposée :

L'approche proposée dans ce travail est basée sur simulation dans système industriel complexes. Pour rendre clair et explicite un ensemble d'informations largement implicites.

L'architecture définis dans ce travail est une architecture basée agent, conçue pour supporter simulation dans système industriel complexes c'est-à-dire, que l'architecture proposée est sensée contenir tous les concepts nécessaires pour assurer toutes les activités liées au industriel.

L'architecture exprime la structure fondamentale du système industriel à analyser et à concevoir. Elle définit l'ensemble de composants et modules fonctionnels décrits en termes de leurs comportements et interfaces, ainsi que la façon d'interaction de ces composants afin

d'accomplir correctement l'objectif d'une simulation dans système industriel. Donc, une description architecturale est principalement requise pour la spécification de la structure du système.

Dans le contexte de notre étude, les composants fonctionnels, correspondent aux différents agents constituant la simulation dans système industriel et à leur modules internes.

Nous allons présenter dans ce qui suit les spécifications des différents composants, ainsi que les concepts liés à leur fonctionnement.

L'idée est de modéliser système industriel comme étant un système multi agents distribués, où chaque agent est chargé d'exécuter une étape de la simulation. D'un point de vue organisationnel, système industriel est un regroupement d'entreprise, et d'individus. Ces entités représentent les nœuds d'une infrastructure gérée par un coordinateur. D'un point de vue conceptuel, l'ensemble des individus de la place de système est modélisé en tant qu'un SMA. Chaque individu est représenté par un agent autonome.

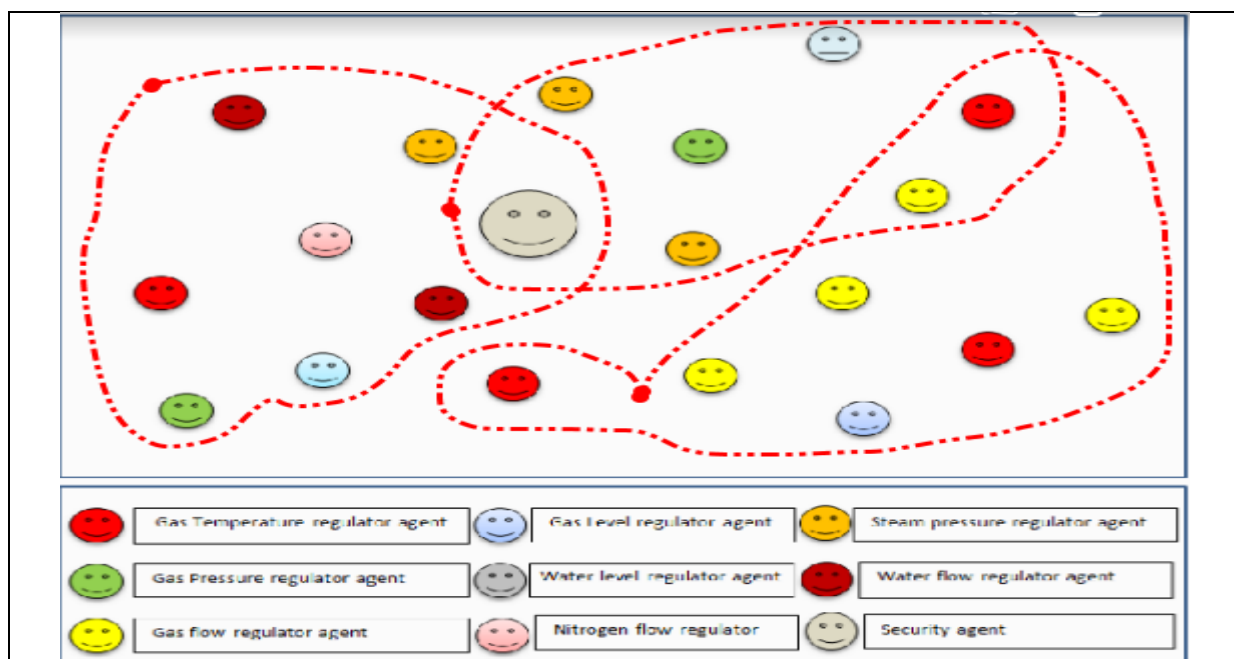


Figure 4.1: Représentation des systèmes industriels basés sur des agents

### 3/ Structure des agents de la simulation de systèmes industriels :

Nous allons décrire dans ce qui suit la structure et les fonctionnalités des différents Agents et qui sont :

### **3-1/Agent l'adsorbeur de mercure 14-MD01 :**

Cet agent responsable de réduit le niveau de mercure dans le gaz à moins de 5 ng/Nm<sup>3</sup>. il peut communiquer avec filtre à gaz traité 14-MD02 et Agent pression gaz

#### **3-1-1/Le module de communication :**

Le module de communication est responsable de l'interaction de l'agent avec les autres agents du système pour obtenir, envoyer, recevoir des informations. Par exemple Recevoir le niveau de mercure.

#### **3-1-2/L'interface utilisateur :**

Ce module permet une interaction avec l'utilisateur, afin que ce dernier puisse interagir avec notre système.

Cette interface fournit les outils nécessaires à l'utilisateur pour comprendre le niveau de mercure en gaz à travers des simulations

Le résultat de Régulateur le niveau de mercure de GNAffiché sur cette interface.

#### **3-1-3/Le module d'exécution :**

Dans ce module, le traitement de la recommandation se fait selon l'algorithme déterminé.

#### **3-1-4/La base de connaissance:**

Dans ce module est stockée toute la connaissance de l'agent

### **3-2/Agent de filtre à gaz traité 14-MD02 :**

Cet agent filtre à gaz traité qui retient 99,5% des particules entraînées de 10 microns et plus. il peut communiquer avec Agent l'adsorbeur de mercure 14-MD01 et Agent pression gaz

#### **3-2-1/Le module de communication:**

Le module de communication est responsable de l'interaction de l'agent avec les autres agents du système pour obtenir, envoyer, recevoir des informations.

#### **3-2-2/L'interface utilisateur :**

Ce module permet une interaction avec l'utilisateur, afin que ce dernier puisse interagir avec notre système.

Cette interface fournit les outils nécessaires à l'utilisateur pour comprendre le processus de filtre à gaz traité à travers des simulations

#### **3-2-3 /Le module d'exécution :**

Dans ce module, le traitement de la recommandation se fait selon l'algorithme déterminé.

#### **3-2-4/La base de connaissance :**

Dans ce module est stockée toute la connaissance de l'agent

## 3-3/Agent pression gaz :

Cet agent contrôle la pression gaz/HP vers 14-MD01 et 14-MD02., il peut communiquer avec tous les agents de notre système. Et communiquer aussi avec l'interface graphique afin de donner une représentation visuelle de l'Agent pression gaz.

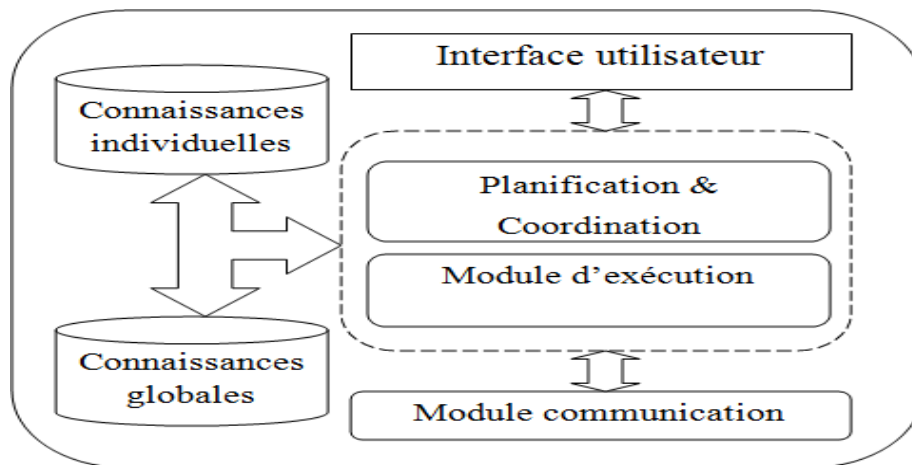


Figure 4.2: Structure d'Agent pression fuel-gaz

### 3-3-1/Le module de communication :

Le module de communication est responsable de l'interaction de l'agent avec les autres agents du système pour obtenir, envoyer, recevoir des informations. Par exemple envoyer la valeur de pression.

### 3-3-2/L'interface utilisateur :

Ce module permet une interaction avec l'utilisateur, afin que ce dernier puisse interagir avec notre système.

Cette interface fournit les outils nécessaires à l'utilisateur pour comprendre le processus de pression d'alimentation en gaz à travers des simulations

Le résultat de Régulateur de pression fuel-gaz/HP vers la turbine 01-MJ01-GT. Affiché sur cette interface.

### 3-3-3/Le module d'exécution :

Dans ce module, le traitement de la recommandation se fait selon l'algorithme déterminé.

### 3-3-4/Le module de planification et de coordination :

Il est responsable de la gestion de la coopération et la formulation des offres afin de répondre aux sous-buts (les parties de la décomposition du but global) annoncés par l'agent Intégrateur.

### 3-3-5/La base de "Connaissances globales" :

Contient les informations concernant l'état de l'interaction dont laquelle il participe ainsi

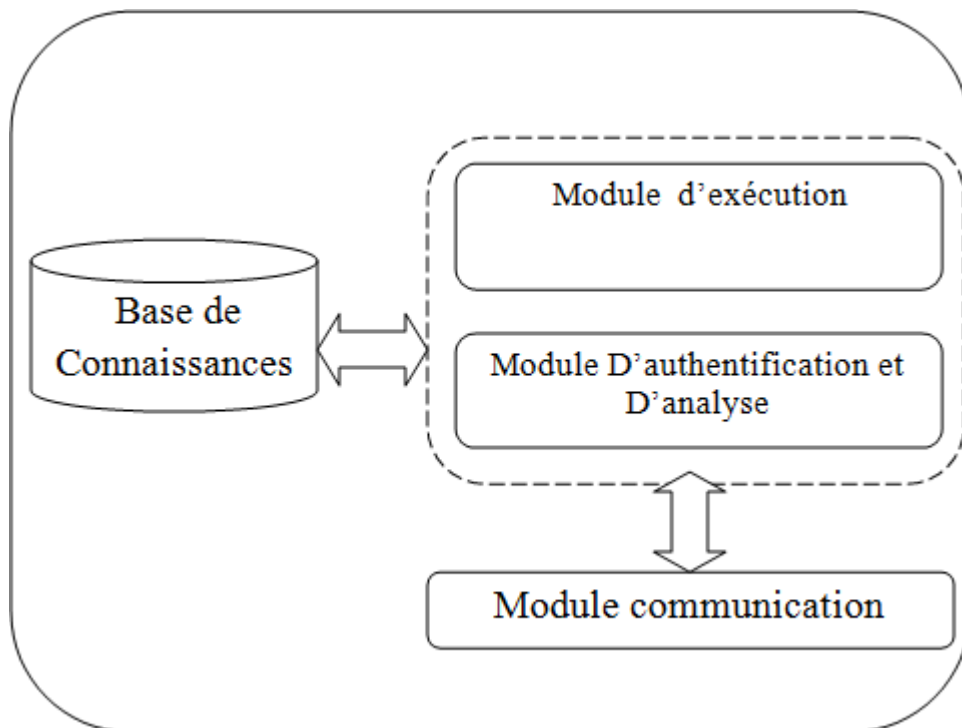
quelareprésentationdesautresagentsintervenantdanscetteinteraction.

### **3-3-6 La base "connaissances individuelles" :**

Contientdesinformationssurl'agentlui-même,c'est-à-diresescapacitéset sescompétences,l'étatetlachargedetravailactuelle,... etc.

### **3-4/Agent Sécurité :**

Cet agentresponsabled'arrêtd'urgence du système de démercurisation, il peutcommuniquer avec tous les agents de notre système.



**Figure4 .3:Structured'AgentSécurité**

#### **3-4-1/le module d'authentification et d'analyse :**

Le rôle de ce module est d'authentifier le utilisateur et d'analyser les données. Le module se sert des informations stocké dans le registre d'information, la même chose pour l'analyse des données.

#### **3-4-2/Le module de communication :**

Le module de communication est responsable de l'interaction de l'agent avec les autres agents du système pour obtenir, envoyer, recevoir des informations.

#### **3-4-3/Le module d'exécution :**

Dans ce module, le traitement de la recommandation se fait selon l'algorithme déterminé.

#### **3-4-4/La base de connaissance :**

Dans ce module est stockée toute la connaissance de l'agent.

## **4/Exemple de scénario**

### **4-1/Démarrage**

Ce chapitre décrit les étapes nécessaires pour la mise en service du le système de démercurisation. Concernant ce chapitre du manuel, il est supposé que les conditions suivantes sont satisfaites :

- Pressuriser progressivement le système de démercurisation (à une allure qui ne dépasse pas 1 bar par minute) à la pression du gaz d'alimentation en utilisant le by pass d'admission de 2'' autour de la vanne d'isolement principale sur la ligne NG-13023-36.
- Chasser l'azote déplacé vers la torche chaude en utilisant le by pass de la 14-PRV-1009. Vérifier toutes les brides, presse-étoupes, joints d'étanchéité et connexions d'instrument pour prévenir les fuites

### **4-2/Arrêt**

Les étapes suivantes sont basées sur un arrêt programmé du système de démercurisation gaz d'alimentation. L'arrêt de ce système se fait en totale coordination avec celui des systèmes en aval.

1. Fermer la vanne d'admission d'isolement et les vannes de by pass de 2'' en amont de l'adsorbeur de mercure 14-MD01.
2. Fermer la vanne d'isolement des effluents du filtre à gaz traité 14-MD02.
3. S'assurer que les vannes de by pass de l'adsorbeur de mercure 14-MD01 et du filtre à gaz traité 14-MD02 sont fermées.
4. S'assurer que les vannes d'isolement de la ligne NG-14001-36'' entre 14-MD01 et 14-MD02 sont ouvertes.
5. Dépressuriser le système progressivement vers la torche chaude à travers le by pass manuel de 14-PVR-1009. Limiter l'allure de dépressurisation à 1 bar par minute.
6. La dépressurisation est terminée quand la pression indiquée sur 14-PG-1004 est voisine de celle de l'atmosphère .

## **Conclusion :**

Afin de maîtriser la complexité de notre système, nous avons le décomposé en un ensemble d'entités autonomes qui interagissent entre eux afin de répondre aux besoins des clients.

Dans le chapitre suivant, on va présenter l'implémentation de notre système, et les outils nécessaires qui seront utilisés afin de réaliser l'application

**CHAPITRE 5**  
**conceptio Aumlll**

## **1/Introduction :**

Toute application informatique nécessite une phase de conception afin d'aboutir à un produit final de qualité et dans les délais attendus. Dans ce chapitre nous introduisons le formalisme AUML qui s'appuie sur UML qui est une méthode de génie logiciel utilisée pour les développements en langages orientés objets. Elle est déjà largement utilisée par la communauté des concepteurs objet et son succès continue de croître.

Ce chapitre concerne la présentation du modèle Multi-Agents que nous avons adopté pour modéliser notre système.

## **2/ Le langage AUML :**

Le langage UML (Unified Modeling Language, ou langage de modélisation unifié) a été pensé pour être un langage de modélisation visuelle commun, et riche sémantiquement et syntaxiquement. Il est destiné à l'architecture, la conception et la mise en œuvre de systèmes logiciels complexes par leur structure aussi bien que leur comportement. L'UML a des applications qui vont au-delà du développement logiciel, notamment pour les flux de processus dans l'industrie. Il est centré sur l'interaction entre les agents, qui a pour but les objectifs suivants :

- ❖ Faire progresser l'état de l'art de la modélisation orientée agent.
- ❖ Permettre aux développeurs de mieux comprendre comment modéliser des applications orientées agent

<https://www.lucidchart.com/pages/fr/langage-uml>

## **3/ Le fonctionnement du système:**

Notre système permet de simuler le fonctionnement du processus de système de démercurisation, de gaz ce dernier élimine une grande quantité de mercure contenue dans le gaz d'alimentation..

Notre modélisation multi-agent de simulateur contient les agents suivants :

### **3-1 Agent l'adsorbeur de mercure 14-MD01**

Cet agent responsable de réduire le niveau de mercure dans le gaz à moins de 5 ng/Nm<sup>3</sup>. il peut communiquer avec le filtre à gaz traité 14-MD02 et l'Agent pression gaz

# CHAPITRE 5 conceptio Aumlll

## 3-2 Agent de filtre à gaz traité 14-MD02

Cet agent filtre à gaz traité qui retient 99,5% des particules entraînées de 10 microns et plus. il peut communiquer avec Agent l'adsorbeur de mercure 14-MD01 et Agent pression gaz

## 3-3 Agent pression gaz :

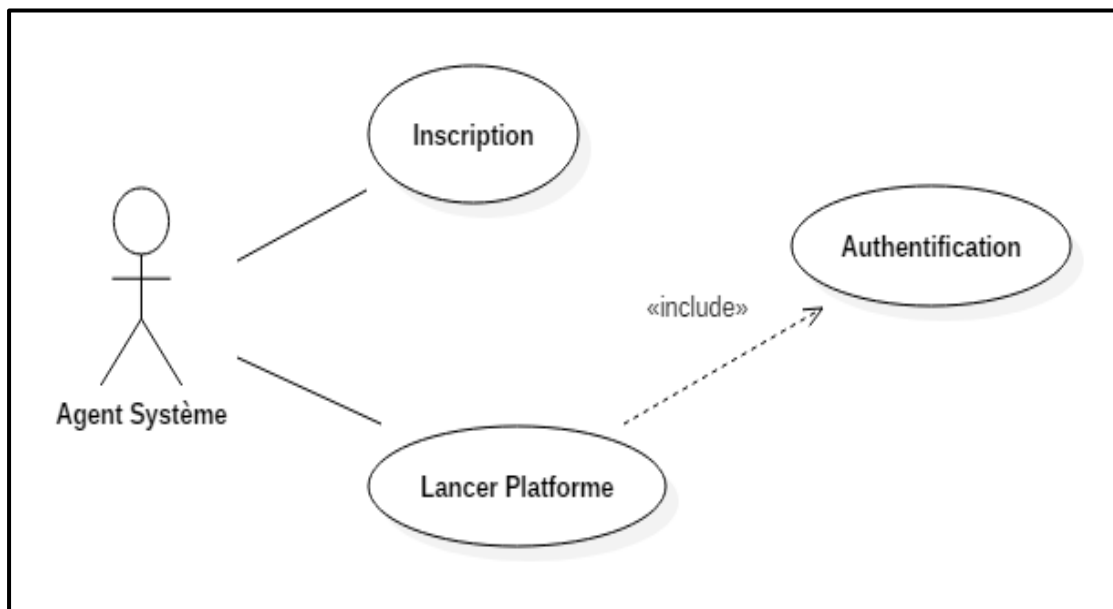
Cet agent contrôle pression gaz/HP vers 14-MD01 et 14-MD02., il peut communiquer avec tous les agents de notre système. Et communiquer aussi avec l'interface graphique afin de donner une représentation visuelle de l'Agent pression gaz.

## 3-4 Agent Sécurité :

Cet agent responsable d'arrêt d'urgence du système de démercurisation, il peut communiquer avec tous les agents de notre système.

## 4/ Le diagramme de cas d'utilisation :

Les diagrammes de cas d'utilisation sont utilisés pour définir des cas d'utilisation du système et donnent une analyse du système. Ils représentent les cas d'utilisation, les acteurs, et les relations qui existent entre les acteurs et les cas d'utilisation.



**Figure 5.1 Diagramme de cas d'utilisation.**

## 5/ Diagramme de classe :

Les diagrammes de classes sont utilisés pour représenter le point de vue statique. Ainsi, le diagramme de classes dans AUML a pour but de représenter un système entier d'agents, et non simplement des agents individuels ou fortement regroupés.

Une classe d'agent représente un agent ou un groupe d'agents pouvant jouer un rôle ou avoir un comportement déterminé, une classe d'agent comporte :

- ❖ Description de la classe d'agent et des rôles.
- ❖ Description de l'état interne.
- ❖ Actions, méthodes et services fournis.
- ❖ Messages échangés.
- ❖ Description de la classe d'agent et des rôles : (Nom de la classe agent/ role1,role2,...) un agent d'une classe donnée peut avoir plusieurs rôles.
- ❖ Description de l'état interne : Définition de variables d'instance qui reflètent l'état de l'agent.
- ❖ Actions, méthodes et services fournis : elles sont définies comme dans UML, avec éventuellement des pré-conditions, post-conditions ou invariants.
- ❖ Envoi et réception de messages: description des messages émis et reçus par l'agent
- ❖ en précisant les protocoles.



**Figure.5.2. Présentation graphique de l'échange des messages dans la classe agent.**

# CHAPITRE 5 conceptio Aumlll

## 5-1 La classe de l'agent filtre à gaz traité 14-MD01:

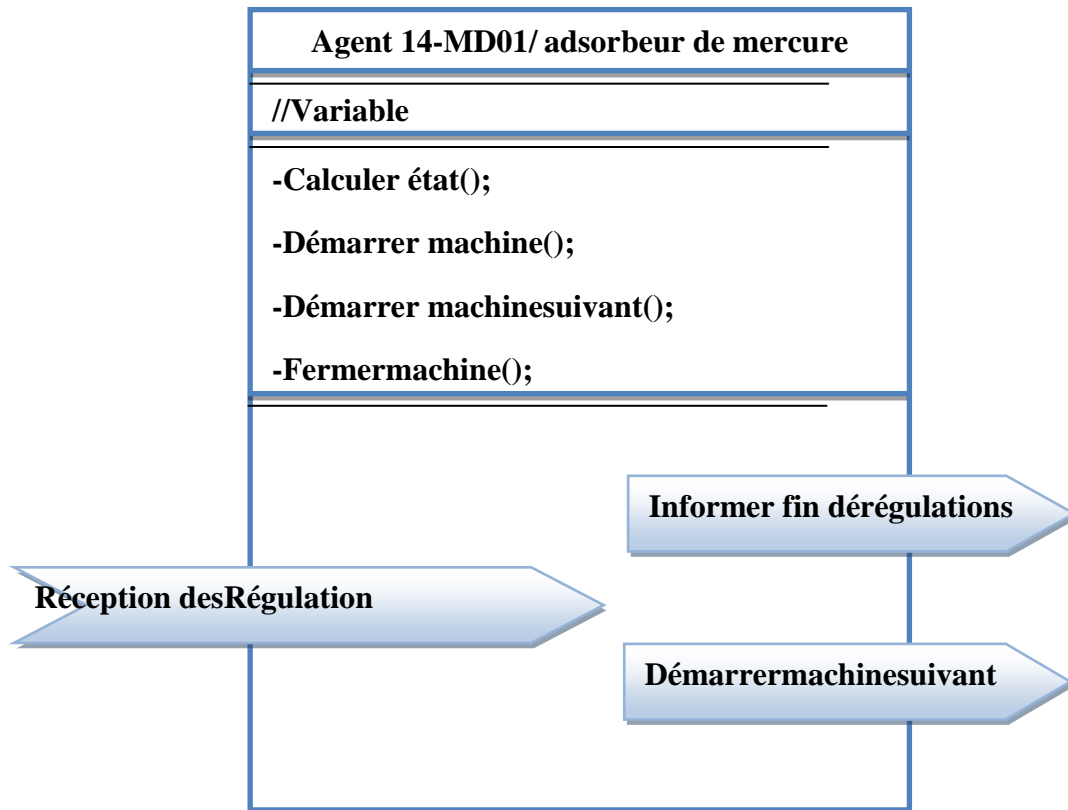


Figure.5.3. La classe de Agent l'adsorbeur de mercure 14-MD01

## 5-2 La classe de l'agent filtre à gaz traité 14-MD02:

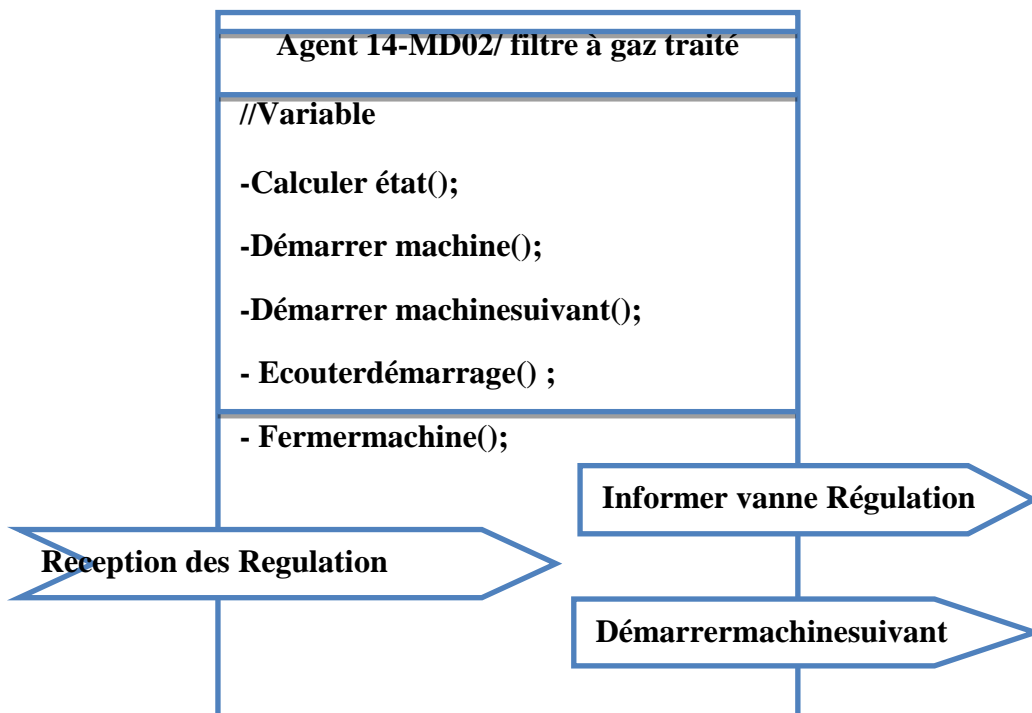


Figure.5.4. La classe de l'agent filtre à gaz traité 14-MD02

# CHAPITRE 5 conceptio Aumlll

## 5-3 La classe de l'agent pression gaz :

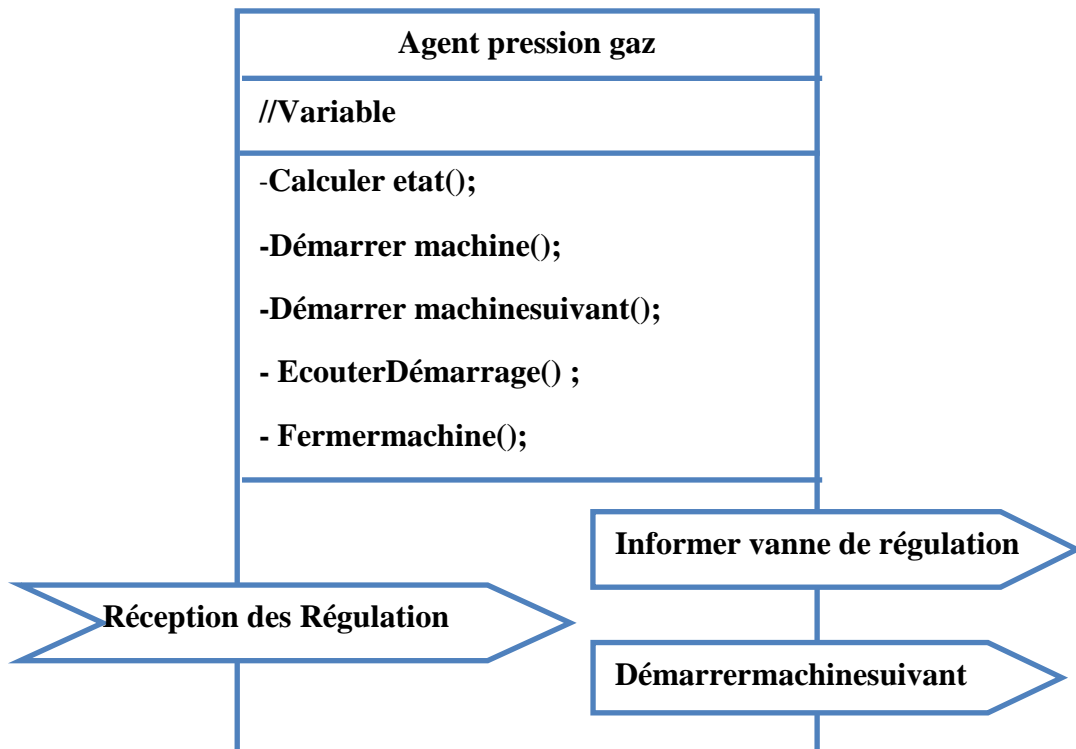


Figure 5.5. La classe de l'agent pression gaz.

## 5-4 Diagramme de classe du système :

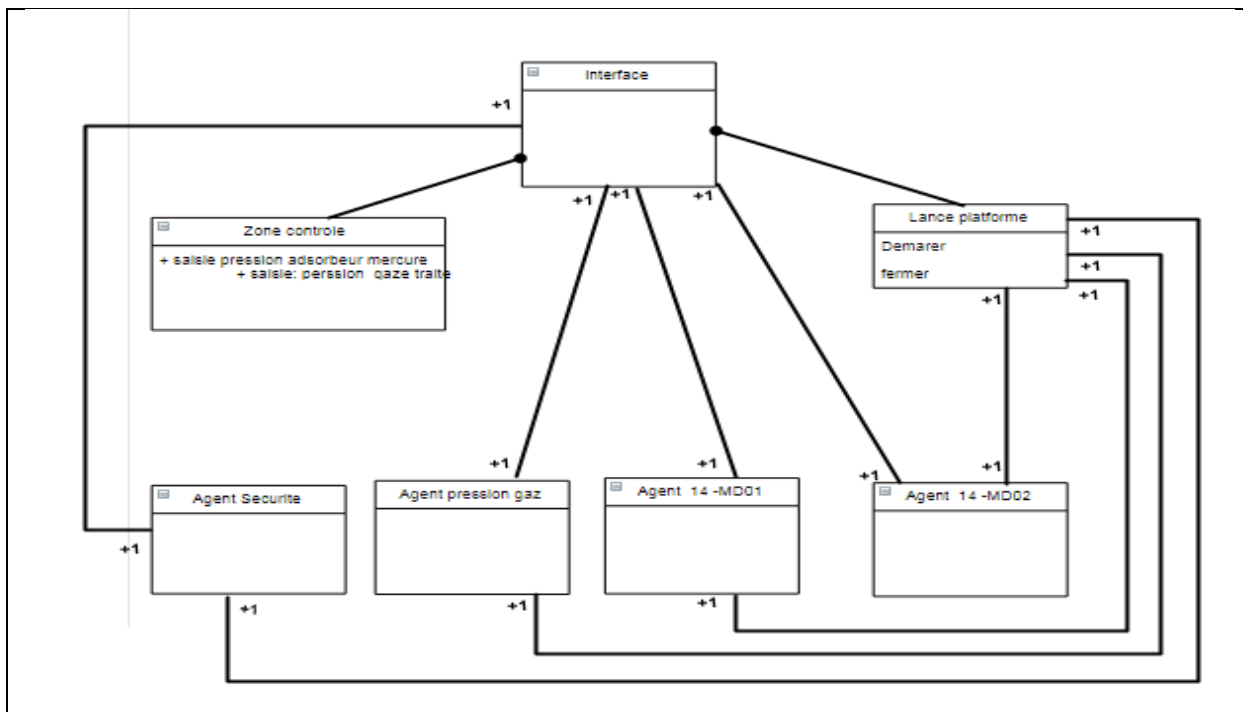


Figure 5-6 Diagramme de classe du système

# CHAPITRE 5 conceptio Aumlll

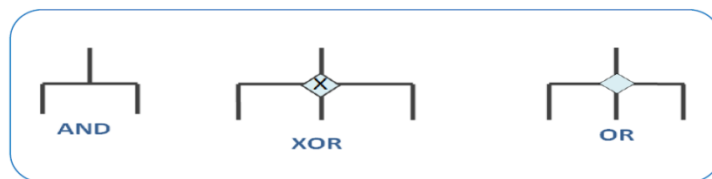
## 6/ Diagramme de séquence :

Les diagrammes de séquence permettent de modéliser le comportement dynamique du système. En effet, les diagrammes de séquence dans les systèmes multi agents sont des schémas qui expriment l'échange de messages entre les agents, donc les déférences qui existent entre le diagramme de séquence en UML et celui en AUML sont : [28]

a) Dans le diagramme de séquence AUML il existe une spécification plus riche du rôle d'un agent.

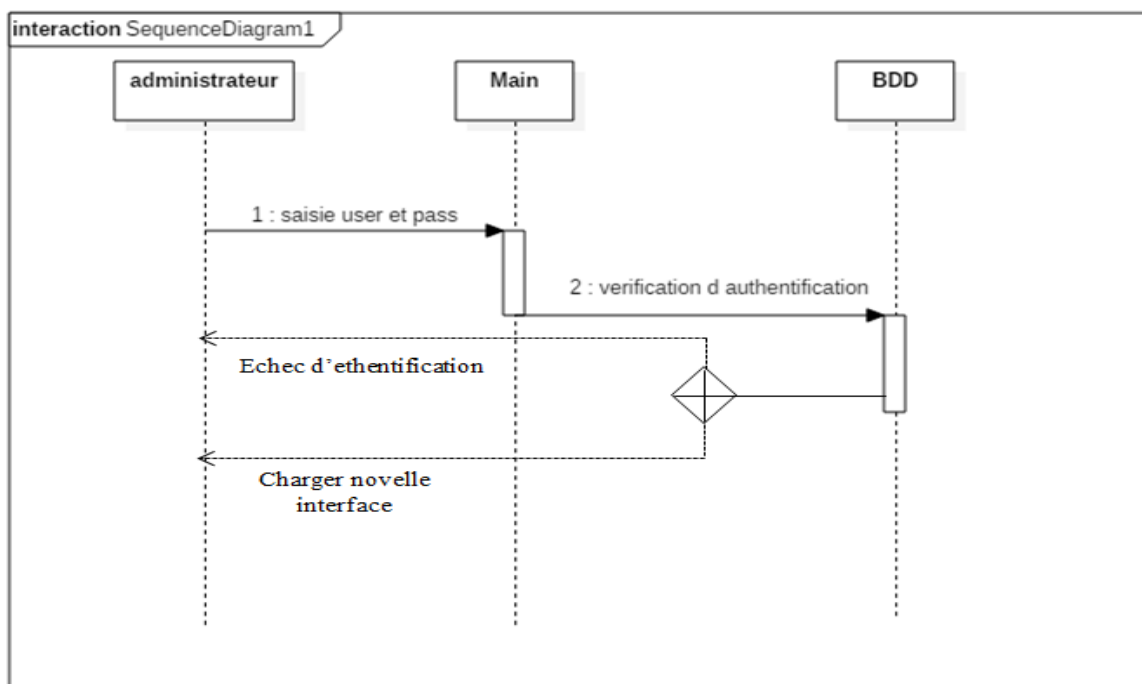
b) L'ajout de nouveaux types de branchements dans les diagrammes de séquence AUML afin de prendre en compte l'indéterminisme du comportement d'un agent.

La figure.25 montre les trois types de branchement: branchement AND (émissions concurrentes), branchement OR (0 ou plusieurs émissions à la fois) et branchement XOR (1 seule émission parmi plusieurs candidates).



**Figure 5.7. Les opérateurs AND, XOR et OR en AUML.**

### 6-1 Diagramme de séquence d'authentification :



**Figure.5.8. Diagramme de classe d'authentification.**

## CHAPITRE 5 conceptio Aumlll

### Conclusion :

Afin de maîtriser la complexité de notre système, nous avons le décomposé en un ensemble d'entités autonomes qui interagissent entre eux afin de répondre aux besoins des clients.

**CHAPITRE 6**  
**Implementation.**

# CHAPITRE 6 Implementation

## 1/ Introduction :

Après avoir présenté notre modélisation de l'approche que nous avons proposée suite à une recherche approfondie, nous passons à la réalisation d'un exemple illustratif en utilisant des outils que nous jugeons performant pour notre solution, l'ensemble de ses outils forme notre environnement de développement dont les caractéristiques seront citées.

## 2/Outils de programmation :

Pour réaliser notre application on a utilisé trois outils de développement :

- ❖ Le langage Java : notre travail est articulé sur ce langage.
- ❖ IDE Eclipse : est un environnement de développement intégré facilitant la tâche de la programmation.
- ❖ Plate-forme JADE : c'est l'outil le plus utilisé. car il a servi à la création du système multi-agents et à l'établissement de la communication entre agents, et vu l'importance de cet outil nous allons l'explicitier encore plus.

## 3/ Présentation des outils :

### 3-1 Le langage de programmation Java :

Java est un langage de programmation et une plate-forme informatique qui ont été créés par Sun Microsystems en 1995. Beaucoup d'applications et de sites Web ne fonctionnent pas si Java n'est pas installé et leur nombre ne cesse de croître chaque jour. Java est rapide, sécurisé et fiable.

- **Pourquoi utiliser Java ?**
  - Java fonctionne sur différentes plates-formes (Windows, Mac, Linux, Raspberry Pi, etc.)
  - C'est l'un des langages de programmation les plus populaires au monde
  - Il est facile à apprendre et simple à utiliser
  - Il est open-source et gratuit
  - Il est sécurisé, rapide et puissant
  - Il a un énorme support communautaire (des dizaines de millions de développeurs)
- Java est un langage orienté objet qui donne une structure claire aux

# CHAPITRE 6 Implementation

programmes et permet de réutiliser le code, réduisant ainsi les coûts de développement



Figure 6.1 : java

## 3-2 JDK : (Java Development Kit)

Java SE JDK est le kit de développement JAVA officiel édité par Oracle. Java est un langage de programmation orienté objet très populaire, qui permet la réalisation d'applications performantes. Les programmes JAVA peuvent être exécutés sur n'importe quel système d'exploitation équipé d'une machine virtuelle JAVA. Pour y parvenir, le langage est compilé sous forme de bytecode qui est ensuite interprété par la machine virtuelle Java.

Le JDK comprend plusieurs outils, parmi lesquels :

- **Javac** : le compilateur Java.
- **Java** : un interpréteur d'applications (machine virtuelle).
- **applet viewer** : un interpréteur d'applets.
- **jdb** : un débogueur.
- **javap** : un décompilateur, pour revenir du bytecode au code source.
- **javadoc** : un générateur de documentation.
- **jar** : l'éditeur d'archives Java.

Dans notre travail, La version utilisée pour l'implémentation est jdk 8.2.

## 3-3 Eclipse

Eclipse IDE est un environnement de développement intégré libre (le terme Eclipse désigne également le projet correspondant, lancé par IBM) extensible, universel et polyvalent, permettant potentiellement de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Eclipse IDE est principalement écrit en Java (à l'aide de la bibliothèque graphique SWT, d'IBM), et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions.

La spécificité d'Eclipse IDE vient du fait de son architecture totalement développée autour de la notion de plug-in (en conformité avec la norme OSGi) : toutes les fonctionnalités de cet atelier logiciel sont développées en tant que plug-in.



**Figure 6.2 : logo d'Eclipse.**

## 3-4 La plateforme Jade

### 3-4-1 Définition :

JADE est une plateforme qui permet de construire des systèmes multi-agents (SMA), créée par le laboratoire TILAB, entièrement implémenté en JAVA. En tant que solution intégrée, programmer avec JADE signifie que dans un environnement de développement unique, vous définissez le code et la base de données. Cette structure aide les développeurs (et ceux qui débutent dans le code) à créer et à démontrer des prototypes fonctionnels complets extrêmement rapidement.



**Figure 6.3 : logo de jade.**

- **Pourquoi JADE ?**

La clé est l'intégration. Si une modification est apportée à la définition de la base de données, par exemple, tous les éléments de code qui utilisent cette partie de la base de données sont identifiés et si possible modifiés. Il n'y a pas d'administration de base de données distincte impliquée.

L'utilisation du concept orienté objet dans l'ensemble du produit réduit le besoin de traduire le monde réel en une définition plus abstraite, avec laquelle les logiciels de base de données conventionnels fonctionnent.

Il y a aussi une équipe de développeurs passionnés derrière JADE chez Jade Software qui organise des cours, des groupes d'utilisateurs et une feuille de route ouverte à laquelle tous les utilisateurs peuvent contribuer. Lorsque vous utilisez JADE, vous faites partie d'une communauté mondiale d'utilisateurs.

### **3-4-2 L'architecture de la plate-forme jade**

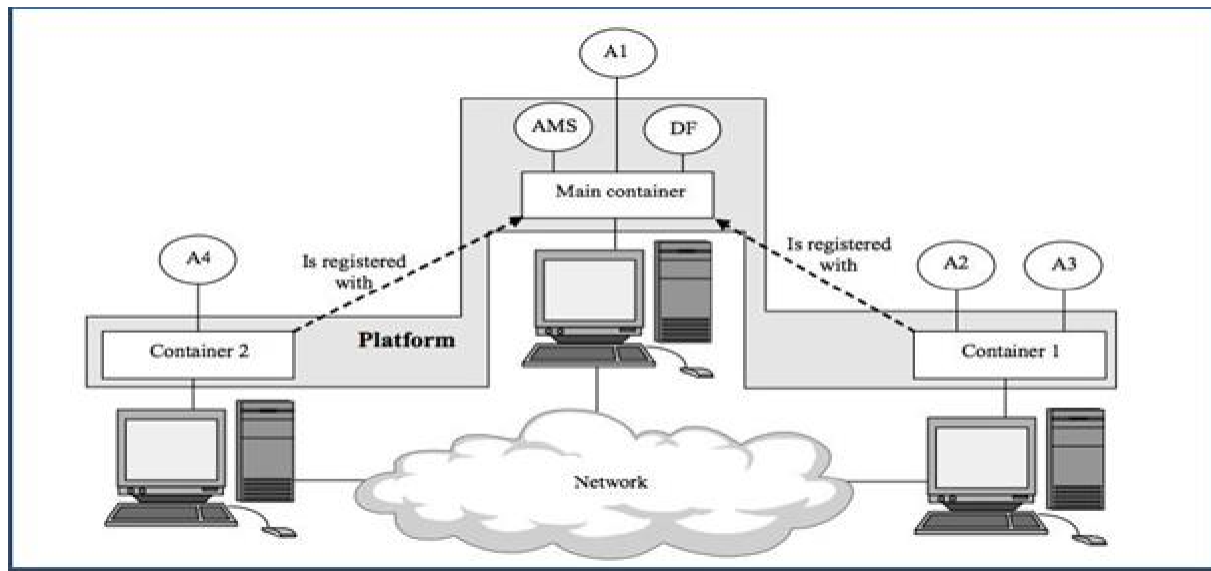
La figure 6.4 représente les principaux éléments architecturaux de JADE. Une application basée sur JADE est constituée d'un ensemble de composants appelés Agents ayant chacun un nom unique. Les agents exécutent des tâches et interagissent en échangeant des messages. Regardez le didacticiel de programmation JADE pour une explication détaillée sur la façon de

développer des agents et de leur faire exécuter des tâches et communiquer. Les agents vivent au-dessus d'une plate-forme qui leur fournit des services de base tels que la livraison de messages. Une plateforme est composée d'un ou plusieurs Containers. Les conteneurs peuvent être exécutés sur différents hôtes, réalisant ainsi une plate-forme distribuée. Chaque conteneur peut contenir zéro ou plusieurs agents. Par exemple, en référence à l'image ci-dessous, le conteneur "Container 1" dans l'hôte Host 3 contient les agents A2 et A3. Même si dans certains scénarios particuliers ce n'est pas toujours le cas, vous pouvez considérer un conteneur comme une JVM (donc, 1 JVM ==> 1 conteneur ==> 0 ou plusieurs agents). Un conteneur spécial appelé Main Container existe sur la plate-forme. Le conteneur principal est lui-même un conteneur et peut donc contenir des agents, mais se distingue des autres conteneurs par

1. Il doit être le premier conteneur à démarrer sur la plate-forme et tous les autres conteneurs s'y inscrivent au moment de l'amorçage

## CHAPITRE 6 Implementation

2. Il comprend deux agents spéciaux : l'AMS qui représente l'autorité dans la plateforme et est le seul agent capable d'effectuer des actions de gestion de la plateforme telles que démarrer et tuer des agents ou arrêter l'ensemble de la plateforme (les agents normaux peuvent demander de telles actions à l'AMS). Le DF qui fournit le service Pages Jaunes où les agents peuvent publier les services qu'ils fournissent et trouver d'autres agents fournissant les services dont ils ont besoin.



**Figure 6.4 : Architecture de la plateforme jade.**

### 3-4-3 Comportement des agents dans JADE :

JADE utilise l'abstraction Comportement pour modéliser les tâches qu'un agent peut exécuter et les agents instancient leurs comportements selon leurs besoins et leurs capacités. De point de vue de la programmation concurrente, un agent est un objet actif, ayant un thread de contrôle. JADE utilise un modèle de programmation concurrente "un thread-par-agent" au lieu d'un modèle "un thread-par-comportement" pour éviter une augmentation du nombre de threads d'exécution exigés sur la plateforme d'agents. Ceci signifie que, pendant que les agents différents s'exécutent dans un environnement multi-threads de préemption, deux comportements d'un même agent sont planifiés coopérativement.

Donc, le développeur d'agents doit étendre la classe Agent et implémenter les tâches spécifiques de l'agent par une ou plusieurs classes Comportement, les instancier et les ajouter à l'agent.

La classe Agent représente une superclasse commune pour tous les agents définis par

# CHAPITRE 6 Implementation

l'utilisateur, du point de vue du programmeur, la conséquence est qu'un agent JADE est simplement une classe Java qui étend la classe de base Agent. Cela permet à l'agent d'hériter un comportement fondamental caché (qui traite toutes les tâches liées à la plate-forme, telles que l'enregistrement, la configuration, la gestion à distance, etc.

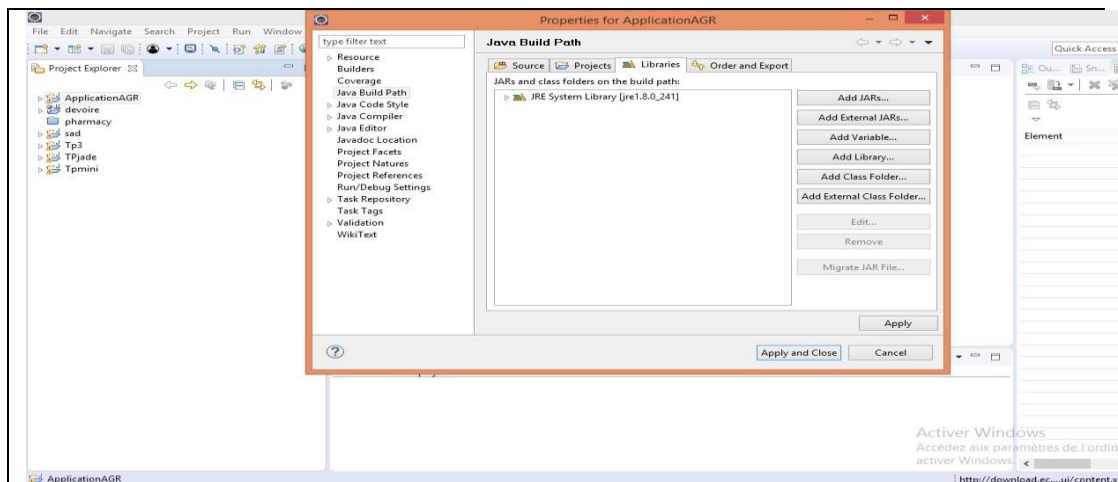
## **3-4-4 L'intérêt d'utilisation de JADE dans notre application :**

Ces critères sont naturellement en relation avec nos attentes d'un système multi agents et des caractéristiques des agents. Ainsi, la plate-forme JADE a la propriété suivante : les agents peuvent communiquer entre eux. Cette communication n'est pas un simple échange de messages mais un dialogue. Il ne s'agit pas pour un agent de répondre mécaniquement à un message mais de pouvoir analyser plusieurs possibilités de réponses avant de réagir.

## **4/ Intégration de la plate-forme JADE dans Eclipse :**

Pour la configuration de jade avec Eclipse, procédons de la manière suite :

1. Démarrer Eclipse, et Extrayez la liste des outils du projet et cliquez sur propriétés et allez à JAVA Build Path comme suite :



**Figure 6.5 : Fenêtre pour ajouter une Library.**

2. Cliquez sur le bouton 'Add External JARs' pour ajouter la bibliothèque jade.jar, puis validez en appuyant sur 'Apply and Close'.
3. Pour configurer Jade sur Eclipse, cliquez sur la liste des outils de bouton 'Run' et accédez aux configurations de run. Nous écrivons en Java application dans Main et dans (x)=arguments. Comme suit :

# CHAPITRE 6 Implementation

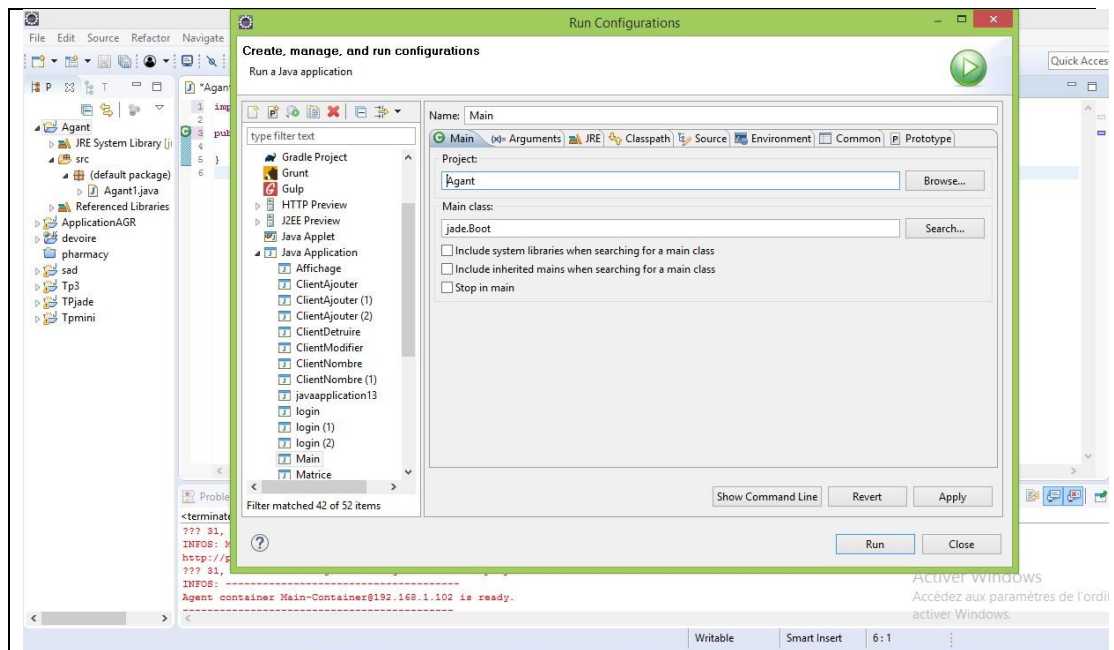


Figure 6.6 : configurer Jade sur Eclipse

## 4. Cliquez sur bouton pour exécuter configurer dans agent

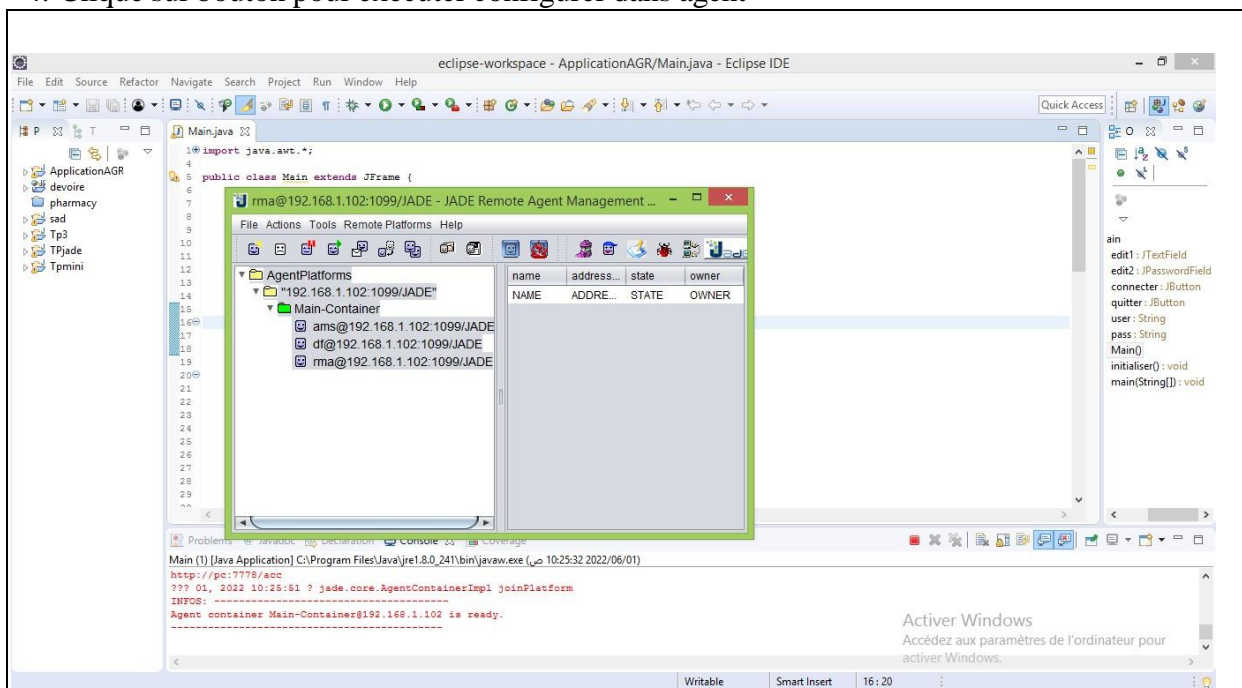


Figure 6.7 : lancer la main

## 5/Implémentation du système :

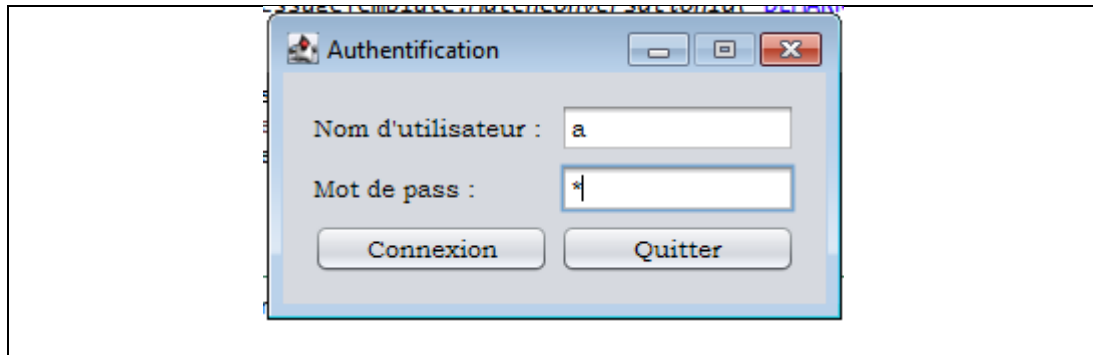
Pour illustrer la simulation multi-agents nous allons suivre un exemple pour fournir des captures d'écrans qui décrivent les différentes étapes de la simulation.

# CHAPITRE 6 Implementation

## 5-1Présentation des interfaces graphiques :

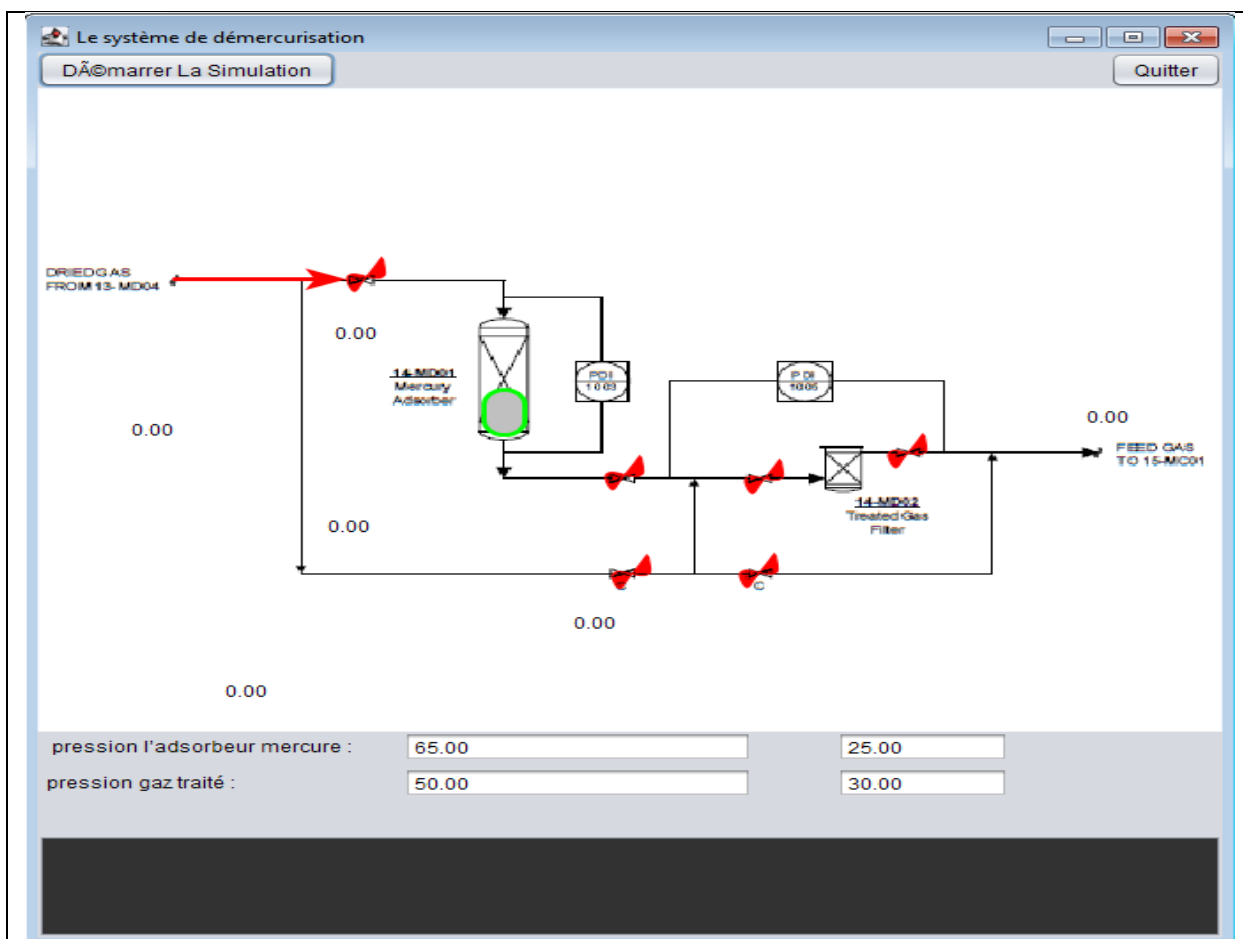
a. Exécution graphique :

➤ Authentification



**Figure 6.8 Authentification**

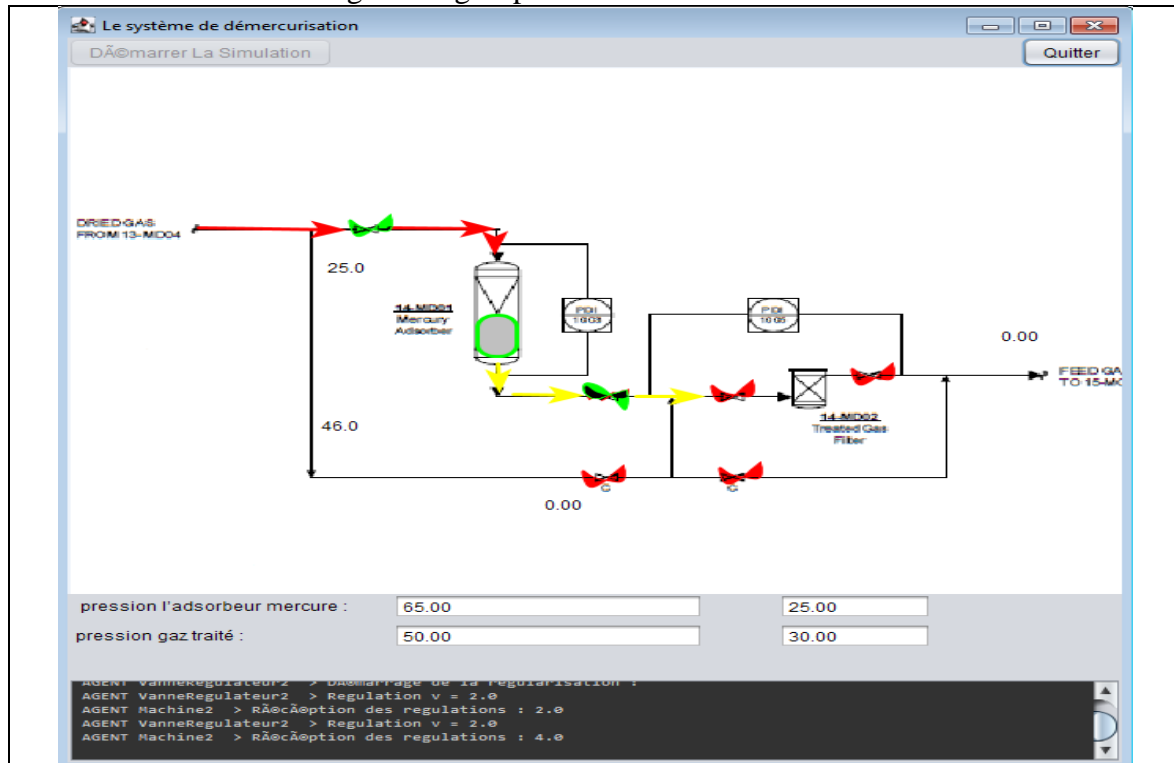
➤ Lancement du agents dugroupe de Le système de démercuration:



**Figure 6.9. Interface de la simulation (groupe de système de démercuration).**

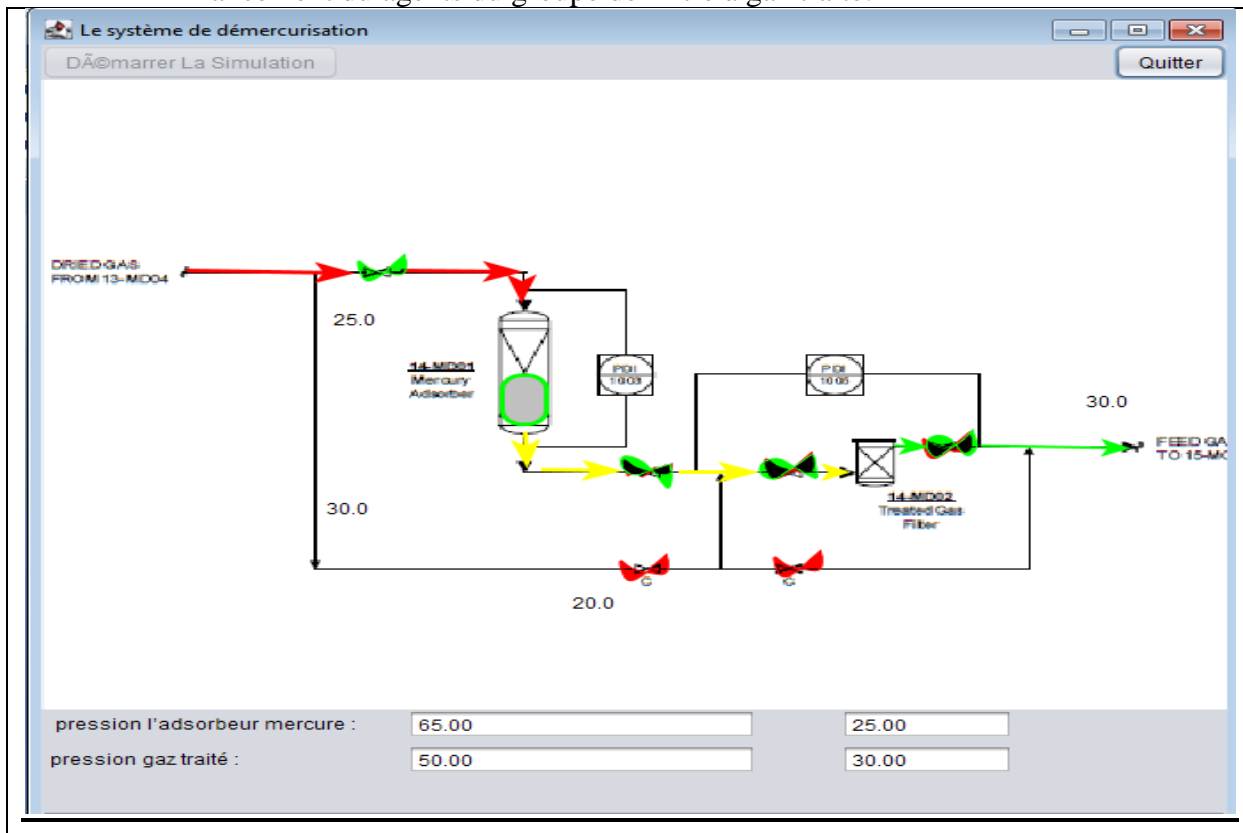
# CHAPITRE 6 Implementation

- Lancement du agents dugroupe de Adsorbeur de mercure:



**Figure 6.10 Interface de la simulation (Adsorbeur de mercure activé).**

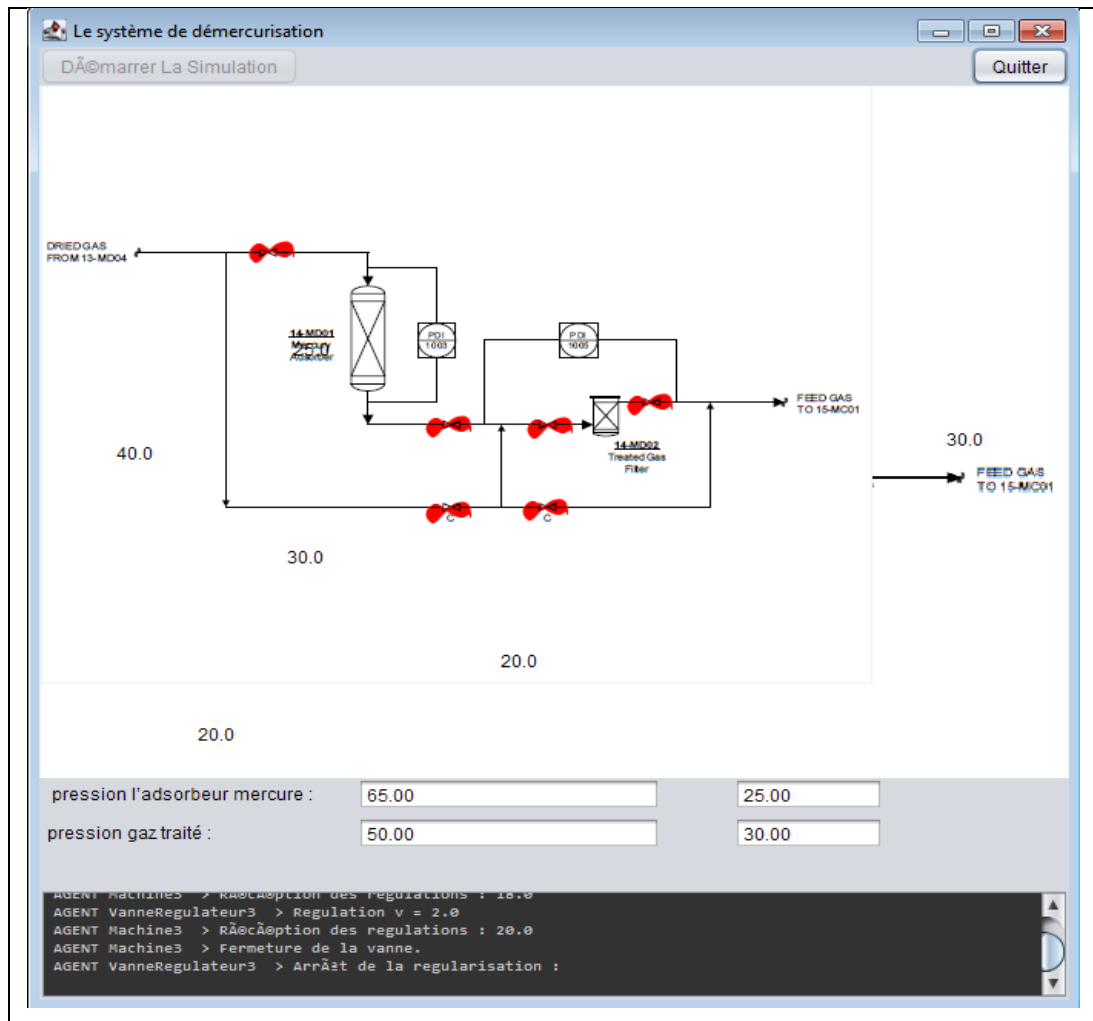
- Lancement du agents du groupe de Filtre à gaz traité:



**Figure6.11 .Interface de la simulation (groupe filtration de gaz).**

# CHAPITRE 6 Implementation

- L'interface du simulateur dans état d'arrêt :



**Figure 6.12 L'interface du simulateur dans état d'arrêt.**

## Conclusion :

Nous avons utilisé la plateforme jade afin de créer et d'implémenter notre système Le système de démercuration., nous avons constaté que cette plateforme est appropriée dans le domaine de simulation informatique et permet de faciliter la tâche aux développeurs débutants dans ce domaine.

# Conclusion générale

## Conclusion générale

A l'inverse des approches traditionnelles qui traitent de systèmes à composantes en nombres restreint, les systèmes complexes ne peuvent pas être approchés uniquement par l'analyse de leurs constituants.

Les interactions entre les différentes parties et les propriétés qui en émergent sont aussi importantes.

Le tout n'étant pas la somme des parties. Dans ce travail, nous avons considéré que la dynamique globale d'un système complexe peut être modélisée par un ensemble d'entités autonomes qui interagissent dans un environnement commun. Cette dimension est prise en compte dans la modélisation et la simulation Multi-Agents qui offre aujourd'hui une alternative très séduisante aux représentations classiques des systèmes complexes.

De plus, l'approche Multi-Agents n'est pas restreinte à un domaine particulier et le paradigme qu'elle définit peut être utilisé pour modéliser toutes sortes de systèmes. C'est pourquoi, de par son aspect novateur et interdisciplinaire, la simulation Multi-Agents représente à n'en pas douter un paradigme de modélisation qui possède un énorme potentiel dans le cadre de l'étude des systèmes complexes.

Nous nous sommes efforcés d'appliquer et d'approfondir ces notions au cours de modeste travail. La considération principale est d'aboutir à effectuer un découpage (agentification) optimal faisant passer de l'espace réel vers de l'espace virtuel (monde modélisé).

Finalement, nous avons soutenu notre approche par un exemple d'application, dans le cas de la régulation d'un processus industriel complexe et nous avons construit un simulateur industriel de ce système

# Bibliographie

- [1] **Robert Valette** Qu'est ce qu'un bon modèle LAAS-CNRSF- 31077 Toulouse Cedex 4  
Version Juillet 1999
- [2] **Jean-Christophe POUSSIN** « Notions de système et de modèle » Agronome ORSTOM,  
Laboratoire d'Économie rurale, INA-PG, 78850 Thiverval Grignon
- [3] [Hubert, 1969] **P. Hubert**, *Le Glossaire International d'Hydrologie*, novembre, 1996.
- [4] **Professeur Jean-Louis Boimond** : « simulation systèmes de production réseaux de petri siman  
–arena », thèse de doctorat, [http://laris.univ-angers.fr/\\_attachments/jean-louis-article2/Cours%2520de%2520simulation.pdf?download=true](http://laris.univ-angers.fr/_attachments/jean-louis-article2/Cours%2520de%2520simulation.pdf?download=true)
- [5] « **Les réseaux de petri** », université de caen/ basse- Normandie (U.F.R : sciences caen),  
master 1A mention EEA parcours AEII, 20 juin 2006.
- [6] **Pascal Roques and Franck Vallée**. UML en action. Eyrolles, 2e édition, 2003
- [7] [Fishwick, 1997] **P. A. Fishwick**, Computer simulation : *growth through extension*.  
*Transactions of the Society for Computer Simulation International*, 14(1), 13–23, 1997.
- [8] [Agha Actor]: *a model of concurrent computation in distributed système* , MTT Press,  
1986
- [9] **Dr. Toufik MARIR** , Les Systèmes Multi-Agents, Université Larbi Ben M'hidi – Oum El  
Bouaghi, 2017
- [10] [Reichgelt, 1990] **H. Reichgelt**, "Different styles of agents architectures", *Proceedings  
of the 1st belief representation and agent architectures workshop*, Gallier J.R. (ed), pp29-39,  
Cambridge, UK, mai 1990
- [11] **M.N.Huhns**. *Distributed artificial intelligence*. pitman Publishing Morgan Kaufman,  
1987.
- [12] [B.Chaib-draa], « *Agent et systeme multi-agent , universite* », novembre , 1999.
- [13] **Malone, T. W. 1988**. *What is coordination theory*, à National Science Foundation  
Coordination Theory Workshop, MIT.

# Bibliographie

- [14] **Zambonelli F., Jennings N. R, and Wooldridge M.**(2000), «*Organisational Abstractions for the Analysis and Design of Multi-Agent Systems*» Proc. Ist Int. Workshop on Agent-Oriented Software Engineering, Limerick, Ireland, 127-141.
- [15] **Roberto A Flores-Mendez;** Towards the standardization of Multi-Agent Systems Architectures : An Overview. ACM Crossroads' special issue on Intelligent Agents, summer 1999.
- [16] **Scott A. Deloach.,** Multiagent Systems Engineering A Methodology And Language for Designing Agent Systems. 1999.
- [17] **Projet National : ADELFE** <https://www.irit.fr/ADELFE/2000> – 2003
- [18] **[Caire Giovanni]** [et al.] Agent Oriented Analysis Using Message/UML [Revue]. - [s.l.] : Agent-Oriented Software Engineering II, 2001.
- [19] **Pavón Juan et Gómez-Sanz Jorge J** Agent oriented software engineering with INGENIAS [Revue]. - [s.l.] : 3rd International Central and Eastern European Conference on Multi-Agent Systems, 2003.
- [20] **Fernández Rubén Fuentes, Gómez-Sanz Jorge J.** et Pavón Juan Towards requirements elicitation in multi-agent systems [Revue]. - [s.l.] : 17th European Meeting on Cybernetics and Systems Research, 2004
- [21] **M. Cossentino, S. Gaglio, L. Sabatucci, and V. Seidita.** *The PASSI and Agile PASSI MAS Meta-Models Compared with a Unifying Proposal. In Proceeding of the CEEMAS' 05 Conference*, pages 183–192., Budapest, Hungary, sept 2005.
- [22] **M. Cossentino, S. Gaglio, A. Garro, and V. Seidita.** *Method fragments for agent design methodologies : from standardization to research. International Journal on Agent Oriented Software Engineering (IJAOSE), 2007.*
- [23] **J. C. Collis, D. T. Ndumu, H. S. Nwana, and L.C. Lee.** *The ZEUS agent building toolkit. BT Technology Journal, 16(3), 1998.*
- [23] **P. Busetta, R. Ronnquist, A. Hodgson, and A. Lucas.** *Jack intelligent agents - components for intelligent agents in java. AgentLink News Letter, 1999.*

# *Bibliographie*

[24] **J. Ferber and O. Gutknecht.** *Madkit :A generic multi-agent platform. In 4 th International Conference on Autonomous Agents.,2000.*

[25] **F. Bellifemmine, A. Poggi, and G. Rimassa.** *JADE - A FIPAcpliant agent framework. In 4th International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agents, 1999.*

[26]<https://open-organization.com/>

[27]<https://www.univ-tlse3.fr/>