



Université 20 août 1955 – SKIKDA

Faculté de Sciences

Département d'informatique



Mémoire

En vue de l'obtention du diplôme de

Master

Filière: Génie Logiciel Avancé Et Applications

Thème

Simulation Basée SMA d'un Système Industriel

Réaliser par :

- Braham abd rezzak-
- Kouicem Akram

Encadré par :

Dr. Seddari Noureddine

Année Universitaire: 2021/2022

Remerciement

Avant tout nous remercions Allah le tout puissant qui nous adonné la force, la patience et le courage pour qu'on puisse accomplir ce modeste travail.

Nous remercions très sincèrement notre enseignant d'atelier Dr. Seddari Noureddine pour son soutien moral, son aide, son orientation ainsi que sa précieuse disponibilité...

Nous tenons ici à représenter nos sincères remerciements à nos parents, nos frères et soeurs que dieu les protège pour leur amour, leur patience et leur soutien durant toutes nos années

Nous exprimons notre sincère reconnaissances ainsi qu'enotre profonde gratitude à toute personne qui a contribué de près ou de loin à l'achèvement de ce travail ; nos enseignants, nos amis, nos collègues et toute la promotion de 2022

Enfin, Nous adressons également nos remerciements à toutes les personnes qui de diverses façons et à différents moments, nous ont apporté leur aide,

Akram, Abd rezzak

Dédicace

En premier mot remercions Dieu le tout puissant qui nous a réunis et aidés pour présenter notre fait. Nous dédions notre travail, comme preuve de respect, de gratitude, et de reconnaissance :

A nos très chers parents qui nous ont soutenus tout au long de notre cursus scolaire (moralement et matériellement), A nos frères et sœurs ainsi qu'à nos amis dont le soutien a été permanent et inconditionnel. A Tous nos proches qu'on n'a pas cités mais qui sont dans nos cœurs.

Résumé

La modélisation et la simulation basées sur les agents (SGAA) est une nouvelle approche de modélisation des systèmes autonomes et des agents en interaction. Et plus populaire pour son efficacité et sa simplicité. Il constitue une approche dans le domaine de la modélisation de systèmes complexes. En effet, ABMS offre, contrairement à autres types de simulations, la possibilité de représenter directement les entités simulées, leurs comportements, et leurs interactions sans avoir à recourir à équations mathématiques. Ce travail est une contribution dans ce sens, le but est de proposer un modèle basé sur un agent pour simuler un système industriel. Ce dernier présente le problème de complexité, qui peut être maîtrisé avec l'approche Multi-Agent Systems (MAS).

Notre modèle est validé par une étude de cas du processus de déshydratation du gaz naturel. Ce dernier est consolidé par une simulation réalisée sur la plateforme multi-agents JADE (Java Agent Développement Framework). Mots-clés : Modélisation et simulation basées sur les agents (ABMS); Système industriel; Systèmes multi-agents (MAS); Plateforme multi-agents JADE. La modélisation et simulation basées sur les agents (ABMS) est une nouvelle approche de modélisation des systèmes autonomes et des agents en interaction. Cette méthode devient de plus en plus et plus populaire pour son efficacité et sa simplicité. Il constitue une approche dans le domaine de la modélisation de systèmes complexes. En effet, ABMS offre, contrairement à autres types de simulations, la possibilité de représenter directement les entités simulées, leurs comportements, et leurs interactions sans avoir à recourir à équations mathématiques. Ce travail est une contribution dans ce sens, le but est de proposer un modèle basé sur un agent pour simuler un système industriel.

Ce dernier présente le problème de complexité, qui peut être maîtrisé avec l'approche Multi-Agent System (MAS). Notre modèle est validé par une étude de cas du processus de déshydratation du gaz naturel. Ce dernier est consolidé par une simulation réalisée sur la plateforme multi-agents JADE (Java Agent DEvelopment Framework).

Mots-clés: Modélisation et simulation basées sur des agents (ABMS); Système industriel; Systèmes multi-agents (MAS); Plateforme multi-agents JADE.

Abstract

Agent-based modeling and simulation (ABMS) is a new approach to modeling autonomous systems and interacting agents. This method is becoming more and more popular for its efficiency and simplicity. It constitutes an approach in the field of modeling complex systems. Indeed, ABMS offers, contrary to other types of simulations, the possibility of directly representing the simulated entities, their behaviors, and their interactions without having to recourse to mathematical equations. This work is a contribution in this sense, the goal is to propose an agent-based model to simulate an industrial system.

The latter presents the problem of complexity, which can be mastered with the Multi-Agent Systems (MAS) approach. Our model is validated by a case study of the natural gas dehydration process. The latter is consolidated by a simulation made in the multi-agent platform JADE (Java Agent DEvelopment Framework).

Keywords: Agent-based modeling and simulation (ABMS); Industrial system; Multi-Agent Systems (MAS); Multi-agent platform JADE.

Table des matières

Titre	Page
Remerciement	
Dédicace	
Résumé	
Introduction générale	1
Chapitre 1: Modélisation Et Simulation Des Systèmes Complexes	
1. Introduction	4
2. Les systèmes complexes	4
2.1 Types de systèmes	5
2.1.1 Les systèmes complexes naturels	5
2.1.2. Les systèmes complexes artificiels	6
2.1.3 Système déterministe	6
2.1.4 Système probabiliste (stochastique)	6
2.1.5 Système continu	6
2.1.6 Système discret (discontinu)	6
2.1.7 Système mixte	7
3. Modélisation et simulation	7
3.1 Modélisation informatique	7
3.2 Notion de modèle	7
4.Outils de modélisation	9
4.1. Une file d'attente	10
4.2 Les réseaux de pétri	11
4.3 Modélisation orientée objet	11
4.4 Simulation informatique	12
4-5 Vérification informatique	14
5. Conclusion	15

Chapitre 2: Système Multi-agent	
1. Introduction	17
2. La notion d'agent	17
2.1. Définition :	17
2.2/ Environnement	18
2.3 Architectures d'agents	19
2.3.1 Agent basé sur la logique	20
2.3.2 Agent réactif	20
2.3.2.1 Agents à reflex simple	20
2.3.2.2 Agents conservant une trace du monde	20
2.3.3 Agent délibératif	20
2.3.4 Agent BDI (<i>Belief-Desire-Intention</i>)	21
2.3.5 Caractéristiques d'un système multi-agents	22
3. Les systèmes Multi-Agents (SMA)	24
3.1 Ddéfinition	24
3.2. De l'Intelligence Artificielle à l'Intelligence Artificielle Distribuée	24
3.3 L'intelligence artificielle distribuée	26
4. Plateformes multi agents	27
5. Les applications des SMA	28
6. Conclusion	29
Chapitre 03 : Description De System A Simuler	
1. Introduction	31
2. Description du système	33
2.1 Pré-refroidissement	33
2.2 Lavage	33
3. Exploitation	35
3.1 Marche normale	36

3.1.1. Refroidisseur de gaz d'alimentation/ propane HP 15-MC01	36
3.1.2. Refroidisseur de gaz d'alimentation/ propane MP 15-MC02	36
3.1.3. Refroidisseur de gaz d'alimentation/ propane BP 15-MC03	36
3.1.4. Colonne d'épuration 15-MD01	36
3.1.5. Refroidisseur de fond de la colonne d'épuration 15-MC16	36
3.1.6. Rebouilleur de la colonne d'épuration 15-MC15	37
3.1.7. Ballon de reflux de la colonne d'épuration 15-MD02	37
3.1.8. Pompes de reflux de la colonne d'épuration 15-MJ01/A	37
3.1.9. Paramètres d'exploitations normales	38
3.1.10. Exploitations temporaires	39
3.2. Démarrage	40
3.3. L'approche proposée	42
3.4. Structure des agents du système de déshydratation	42
4. Conclusion	44
Chapitre 04 : Implémentation	
1. Introduction	46
2. Outils de programmation	46
3. Présentation des outils :	46
3.1. Le langage de programmation Java :	46
3.2 JDK : (Java Development Kit)	47
3.3 NetBeans	47
3.4 La plateforme Jade	48
3.4.1 Définition	48
3.4.2 L'architecture de la plate-forme jade :	48
3.4.3 Comportement des agents dans JADE :	49
3.4.4 L'intérêt d'utilisation de JADE dans notre application :	50
3.5 Intégration de la plate-forme JADE dans Netbeans :	50
4. Implémentation du système	52

5.Conclusion	53
Conclusion Générale	55
Bibliographies	57

Table des Figures

<i>Titer</i>	<i>Page</i>
<i>Figure: 1.1.</i> Vue générale du système complexe.	5
<i>Figure: 1.2.</i> L'activité de modélisation	9
<i>Figure: 1.3.</i> Typologie de la modélisation graphique des réseaux de Pétri.	11
<i>Figure: 1.4.</i> Simulation informatique selon MICHEL	13
<i>Figure 2.1</i> L'agent et son environnement	19
<i>Figure 2.1:</i> Architecture d'un agent BDI	22
<i>Figure 2.2 :</i> Le système HearsayII.	26
Figure 3.1: Détails Colonne D'expiration 15-Md01	32
<i>Figure 3.1:</i> Refroidisseurs de gaz d'alimentation	33
<i>Figure 2.3 :</i> La température du ballon de reflux	35
Figure 3.4: structure organisationnelle du systeme	43
Figure 4.1. logo de java.	46
Figure 4.2. logo de netbeans.	47
Figure 4.3. Architecture de la plateforme jade.	49
Figure 4.4. Fenêtre pour ajouter une Library.	50
Figure 4.5. Boite de saisie du nom de Library.	51
Figure 4.6. Fenêtre de parcourir le fichier JAR.	51
Figure 4.7. Interface de la simulation	53

Listes des tableaux

<i>Titer</i>	<i>Page</i>
Tableau 01 : Variables opératoires pour le système de la colonne d'épuration	38

Introduction Générale

L'activité de Modélisation et Simulation (M&S) joue un rôle de plus en plus primordial dans l'étude des systèmes (naturels ou artificiels) complexes.

Le but de la modélisation est de construire un modèle afin de résoudre un problème d'analyse ou de conception. Un modèle est une représentation simplifiée du comportement observable et de la structure d'un système réel. La résolution de ces problèmes, appliquée sur le modèle, permet d'obtenir une solution dans le monde du modèle. Cette solution est interprétée dans le monde réel, et devient la solution pour le système réel.

La simulation Multi-Agents nous apparaît être la démarche la plus adéquate à la modélisation de ce type de systèmes (système colonne d'épuration de gaz). La simulation Multi-Agents est un domaine de recherche relativement récent, qui se base sur la notion d'agent. Un système Multi-Agents est naturellement adapté pour décrire et simuler un système composé d'entités, il permet au modèle de paraître plus proche de la réalité.

De plus, un système Multi-Agents est particulièrement bien adapté à la description d'un système du point de vue de l'activité de ses composantes, c'est-à-dire lorsque le comportement des individus est complexe (difficile à décrire avec des équations).

De plus, la modélisation est plus facilement interprétable par un observateur humain, car la description par un SMA est plus naturelle que par de simples processus; la validation par un expert en sera donc facilitée, car il pourra facilement se rapporter au monde réel.

Un SMA est composé de plusieurs sous-systèmes autonomes appelés agents dont chacun a une activité et des informations propres. Le comportement général d'un SMA est lié à l'activité combinée de l'ensemble de ses agents et la réalisation d'une tâche peut alors impliquer plusieurs entités.

Pour implémenter notre simulateur de colonne d'épuration de gaz par les systèmes Multi-Agents, nous avons adopté la plateforme de développement JADE.

Notre mémoire se compose de six chapitres organisés de la façon suivante:

1. **Modélisation et simulation:** ce chapitre constitue une introduction à notre travail par la présentation des notions sur la modélisation, la simulation et la vérification d'une façon générale.
2. **System multi agent:** Les différentes définitions relatives au concept « agent » et « systèmes multi-agents », leurs méthodes d'apprentissage et un état d'art sur les travaux réalisés dans l'apprentissage des systèmes multi-agents est présenté.

3. **Colonne d'épuration de gaz** : ce chapitre constitue des informations pour les procédés industriels, les régulateurs.
4. **L'implémentation** : Ce dernier chapitre présente la réalisation et l'implémentation du notre système en utilisant la plateforme JADE

CHAPITRE 1 :
MODELISATION ET SIMULATION DES SYSTEMES
COMPLEXES

1. Introduction

La démarche de modélisation et de simulation est une approche scientifique qui a pour but de produire de la connaissance sur des phénomènes naturels ou artificiels. Cette démarche est utilisée pour comprendre, prédire voire contrôler des objets ou des phénomènes qu'ils soient déjà existants ou encore au stade de la conception, et testent et ajustent en temps réel les produits dans un environnement virtuel proche des contraintes réelles de production

La simulation de procédés permet aux industriels d'une part d'améliorer l'efficacité et la rentabilité d'un procédé existant et d'autre part de concevoir et de simuler une nouvelle unité de production.

La modélisation permet, à partir de faits expérimentaux, de construire un outil mathématique reliant les sorties d'un système à ses entrées. On entend par système une structure physique prise de manière isolée, un cristalliseur, un réacteur chimique, un atelier... représentent autant d'exemples. Ses entrées sont les paramètres qui agissent sur son comportement. Elles peuvent être contrôlées ou non. Ses sorties sont les paramètres résultants des valeurs données aux entrées.

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

2. Les systèmes complexes

Dans la littérature il existe plusieurs définitions sur les systèmes complexes. Dans cette section, nous allons présenter quelques définitions sur l'état de l'art sur les systèmes complexes.

Un système est un ensemble qui se constitue par une unité cohérente et autonome d'objets réels ou conceptuels. On peut dire qu'un système est un organisme conçu pour atteindre un objectif ou ensemble d'objectifs avec moyen d'interactions mutuelles et dynamiques.

- D'après la définition du ¹un système complexe: Les systèmes complexes sont des systèmes qui comportent plusieurs parties en interactions avec une capacité de générer des nouvelles qualités de comportement collectif macroscopique, dont les phénomènes sont la formation spontanée de structures différentes que ce soit temporelles, spatiales ou fonctionnelles
- Une autre définition de la part du ²: un système complexe est défini comme un système caractérisé par des propriétés bien déterminé et se basant sur des concepts qui couvre

les différentes sciences.

- D'après un système complexe est la nouvelle approche qui décrit les relations entre les entités dont le but d'offrir des comportements collectifs aux systèmes. De plus, il permet de déterminer la manière d'interaction par la construction des relations avec son environnement.³

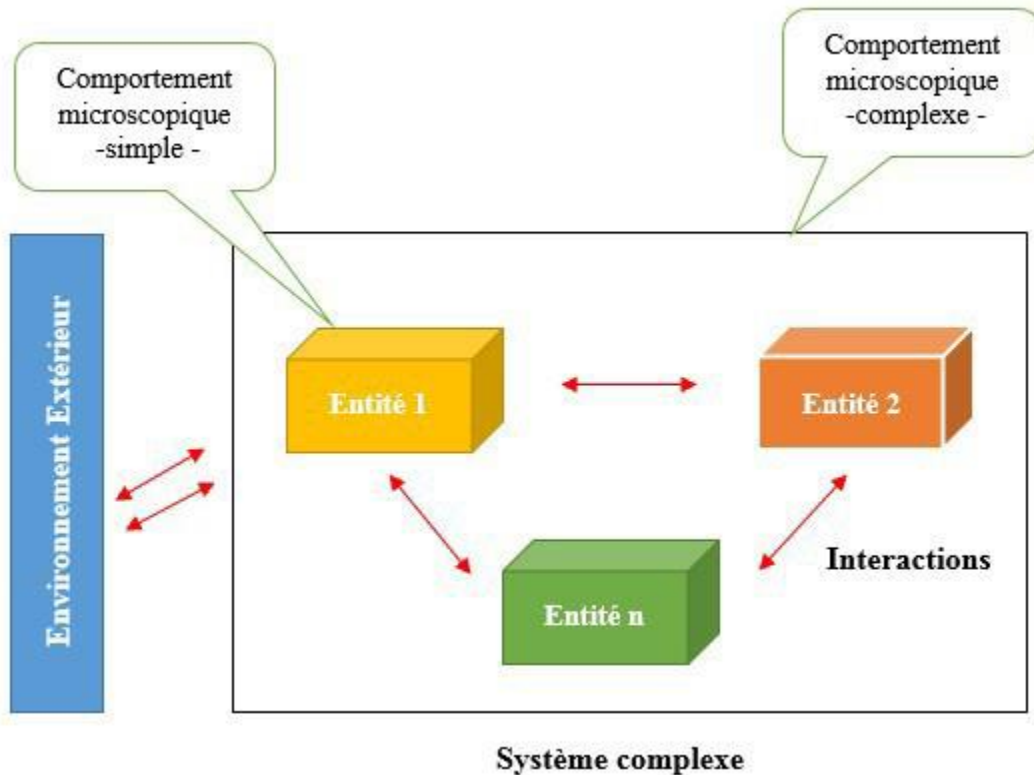


Figure 1.1. Vue générale du système complexe.

2.1 Types de systèmes

Les systèmes complexes sont conçus pour un but d'étudier et analyser les nouvelles situations qui couvrent tous les types de phénomènes. Les systèmes complexes donnent des solutions aux problèmes reliés à la nature tels que les maladies, production de protéines, etc...ou des systèmes artificiels. Les types de systèmes complexes sont cités comme suit

2.1.1 Les systèmes complexes naturels: ils existent plusieurs systèmes complexes dédiés aux problèmes naturels parmi ces derniers nous citons :

- Écologie : organisation des espèces, études des séismes et des volcans, études des rivières etc.

- Biologie : Anatomie humaine, système nerveux et hormonal chez l'homme etc.
- Physique et chimie : mouvement, mécanique quantique, chimie générale, chimie organique etc.
- Économie : Transaction dans le marché, monnaie, Banques, production/consommation etc.
- Transport : Trafic routier, Embouteillage, déplacement des véhicules etc.

2.1.2. Les systèmes complexes artificiels:

plusieurs projets sont adoptés pour un but d'analyser un système complexe telle que :

- **Projet Simdelta** : un système multi-agents conçu pour simuler la pêche sur le Delta central de Niger .⁴
- **Projet Manta** : une plateforme qui simule des organisations sociales des fourmis.⁵
- **Projet Rivage** : un modèle qui décrit les processus physiques ruissellement et d'infiltration d'eau qui peuvent conduire éventuellement à la formation de mares, déravis ou chemins d'eau

2.1.3 Système déterministe

Il fonctionne de manière prévisible, l'interaction entre les différentes parties est connue avec exactitude. Si l'on possède une description de l'état du système à un moment donné, le prochain état du système peut être donné exactement et sans erreur.

2.1.4 Système probabiliste (stochastique)

Il est décrit en termes de comportement probable (qui dépend du hasard). Une certaine marge d'erreur accompagne toujours la prédiction de ce que fera le système.

2.1.5 Système continu

Les changements de l'état du système se font en permanence avec le temps (dépend de variables liées au temps).

2.1.6 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.

2.1.7 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.

3. Modélisation et simulation

La modélisation est l'acte de construire un modèle (pompe, compresseur, vanne ect.)

- Une simulation est le processus d'utilisation d'un modèle pour étudier le comportement et les performances d'un système réel ou théorique.
- Dans une simulation, les modèles peuvent être utilisés pour étudier les caractéristiques existantes ou proposées d'un système.
- But de la modélisation et de la simulation
- Résout les problèmes du monde réel de manière sûre et efficace
- Fournit une méthode d'analyse importante qui est facilement vérifiée, communiquée et comprise.
- Fournit des solutions précieuses en donnant un aperçu clair des systèmes complexes.
- Environnement sans risque Il fournit un moyen sûr de tester et d'explorer différents scénarios (et si). L'effet du changement des niveaux de personnel dans une usine peut être vu sans mettre la production en danger. Prenez la bonne décision avant de faire des changements dans le monde réel.
- Économisez du temps et de l'argent Les expériences virtuelles avec des modèles de simulation sont moins coûteuses et prennent moins de temps que les expériences avec des actifs réels. Les campagnes marketing peuvent être testées sans alerter la concurrence ni dépenser inutilement de l'argent.
- Visualisation Les modèles de simulation peuvent être animés en 2D/3D, ce qui permet de vérifier, communiquer et comprendre plus facilement les concepts et les idées. Les analystes

et les ingénieurs gagnent en confiance dans un modèle en le voyant en action et peuvent clairement démontrer les résultats à la direction.

- Aperçu de la dynamique Contrairement aux analyses basées sur des feuilles de calcul ou des solveurs, la modélisation par simulation permet d'observer le comportement du système dans le temps, à n'importe quel niveau de détail. Colonne de distillation Dynamique avec changement de débit/composition d'alimentation. Précision accrue Un modèle de simulation peut capturer beaucoup plus de détails qu'un modèle analytique, offrant une précision accrue et des prévisions plus précises. Cependant, la précision est liée à la bonne solution de simulation.

3.1 Modélisation informatique

La modélisation informatique consiste en la construction d'un ensemble de programmes qui décrivent la structure ou architecture du système réel et l'agencement des différents composants de ce système ainsi que les interactions entre ces composants. L'exécution de ces programmes sur un ordinateur pour des valeurs des données déterminées conduit à un ensemble d'états de ces programmes et abouti à des ensembles de résultats. Cette dernière étape est appelée simulation du système

3.2 Notion de modèle

D'une façon générale Un modèle est une représentation d'un système réel (physique, économique, humain,...etc) réalisé dans le but de mieux étudier ce système est expliquer certains de son comportement.

D'après P. Hubert ⁶« un modèle et 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 ».

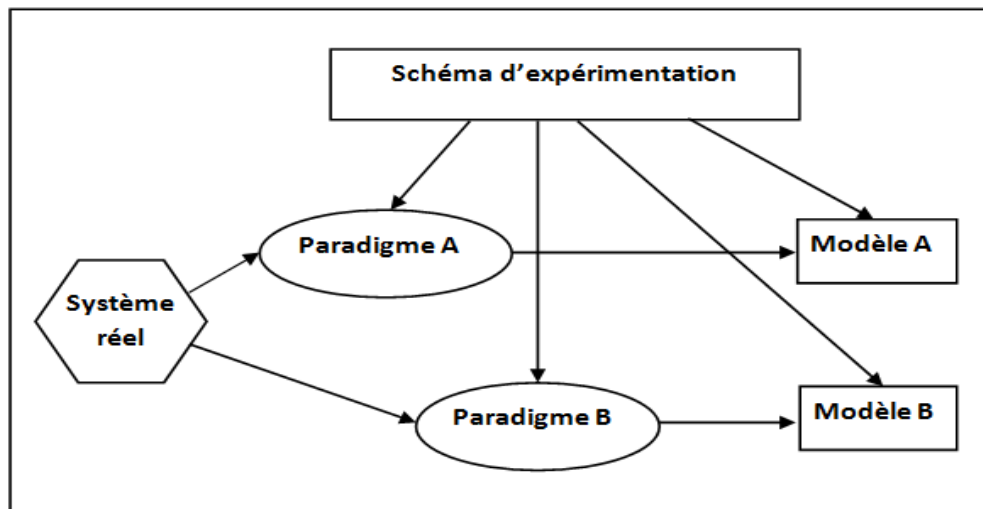
Selon J. Ferber⁷: « Un modèle, en science, est une image stylisée et abstraite d'une portion de réalité».

Pour P.A Fishwick:⁸ « Modéliser c'est décrire la réalité sous la forme d'un système dynamique, à l'aide d'un langage de description, à un certain niveau d'abstraction ».

Selon A. Pavé:⁹ « Un modèle est une représentation symbolique de certains aspects d'un objet ou d'un phénomène du monde réel ».

Parfois, il est impossible d'étudier le système directement du fait qu'il soit inaccessible (système solaire), trop couteux, il change trop rapidement (tir nucléaire) ou lentement (mouvement d'une comète), dans ce cas, l'étude est faite sur un modèle, c'est-à-dire un deuxième système conforme à l'original est aussi parfait que l'étude l'exige.

Un modèle résulte des fonctions et actions menées sur le système réel ou une de ses parties qui intéressent le domaine d'étude. Ces fonctions et actions sont menées dans un cadre expérimental et sont exprimées à l'aide de formalismes et de paradigmes selon une ou plusieurs méthodologies. La notion de paradigme se substitue ici à la notion de vérité dans le cadre général. La figure 1.2 represented activate de modulization.



*Figure : 1.2. L'activité de modélisation*¹⁰

En théorie des systèmes, on distingue, généralement deux points de vues complémentaires l'un de l'autre: Un point de vue structurel qui a trait à l'architecture même du système considéré et un point de vue comportemental qui considère les évolutions du système dans le temps.

4.Outils de modélisation

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.

4.1. Une file d'attente

On s'intéresse maintenant aux réseaux fermés de files d'attente [5, 21, 15]. Une file est constituée d'une file d'attente et d'un ou plusieurs serveurs. Un client qui arrive dans une file patiente dans la file d'attente jusqu'à être servi par un serveur puis quitte la file.

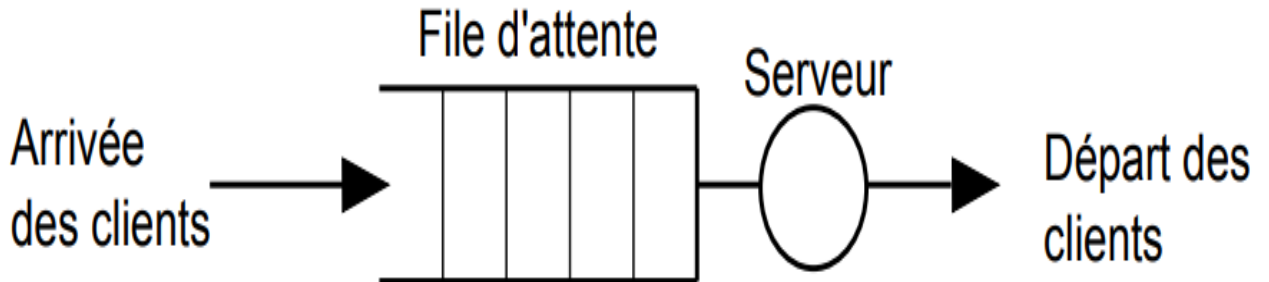


Figure : 1.3. File simple à un serveur.

L'arrivée des clients est décrite par un processus stochastique. On appelle capacité d'une file le nombre maximal de clients que la file d'attente peut contenir. Si une file a atteint sa capacité maximale, tout nouveau client arrivant dans la file est rejeté. La capacité d'une file peut être finie ou infinie. Une discipline de service détermine l'ordre dans lequel les clients reçoivent le service. Par exemple, la discipline FIFO (First In First Out) implique que l'on serve les clients par ordre d'arrivée. La loi des arrivées ainsi que le temps de service sont modélisés par une loi de probabilité. La plus couramment utilisée est la loi exponentielle qui induit ainsi un processus de Poisson (on note ainsi M pour "markovien"). La notation de Kendall décrite ci-dessous, normalise la description d'une file.

Un réseau de files d'attente est un ensemble de files interconnectées dans lequel les clients peuvent être dirigés après avoir été servis d'une file vers une autre. On dit qu'un réseau est ouvert si les clients peuvent arriver de l'extérieur du réseau et en ressortir. Dans le cas opposé, si les clients ne sont autorisés ni à quitter le réseau ni à y rentrer on parle alors de réseau fermé. C'est ce type de réseaux que nous étudierons par la suite. Parmi les réseaux fermés, nous en distinguerons deux types: les réseaux monoclases et multiclases. Dans un réseau multiclasse les clients se partagent le réseau mais chaque client appartient à une classe et est servi puis dirigé en fonction de sa classe. Dans un réseau monoclasse on ne fait pas de distinction entre les clients.

4.2 Les réseaux de pétri

Le réseau de pétri est un modèle mathématique à variables discrètes, créés en 1962 par Carl Adam Pétri. Il se représente par un graphe biparti (composé de deux types de noeuds) orienté (composé d'arc(s)) reliant des places et des transitions (les noeuds). Deux places ne peuvent pas être reliées entre elles, ni deux transitions. Les places peuvent contenir des jetons (pouvant par la suite prendre différentes valeurs), représentant généralement des ressources disponibles.

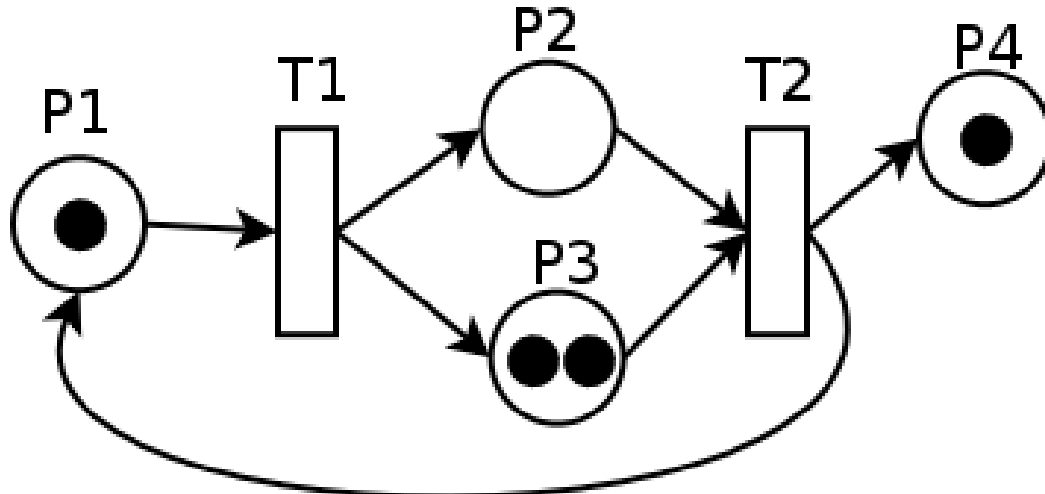


Figure: 1.3. Typologie de la modélisation graphique des réseaux de Pétri.

4.3 Modélisation orientée objet:

La modélisation orientée objet (MOO) est la construction d'objets à l'aide d'une collection d'objets qui contiennent des valeurs stockées des variables d'instance trouvées dans un objet. Contrairement aux modèles orientés enregistrement, les valeurs orientées objet sont uniquement des objets.

L'approche de modélisation orientée objet crée l'union du développement d'applications et de bases de données et les transforme en un modèle de données et un environnement de langage unifiés. La modélisation orientée objet permet l'identification et la communication des objets tout en prenant en charge l'abstraction, l'héritage et l'encapsulation des données.¹¹



Figure: 1.3. Construire une architecture de données orientée métier

4-4 Simulation informatique

Le terme simulation vient du latin «simulatio» qui signifie: apparence et imitation. Si nous nous en tenons, dans notre cadre, au terme d'imitation, la simulation est donc l'imitation d'un phénomène du monde réel. Mais quel besoin l'homme a-t-il d'imiter les phénomènes du monde réel ? ¹². La réponse à cette question n'est pas unique, mais, comprendre le passé, les situations présentes ou prédire et mieux planifier les états futurs d'un système sont les objectifs partagés par de nombreux outils d'analyse et de simulation .

Définition: La simulation est la programmation et la manipulation d'un modèle sur un ordinateur, ainsi que l'analyse des résultats, et elle est largement utilisée pour le développement de nouvelles architectures de communication et de protocoles de réseau ¹³. Nous appellerons temps de simulation le temps de calcul utilisé pour simuler une expérience, l'une des étapes de la simulation est la description d'un système qui consiste à:

- Identifier les éléments (entités).
- Déterminer les attributs.
- Spécifier les activités (processus à l'origine des changements).

Cette deuxième définition donne un aperçu global et intuitif de ce qu'est un processus de simulation par ordinateur.

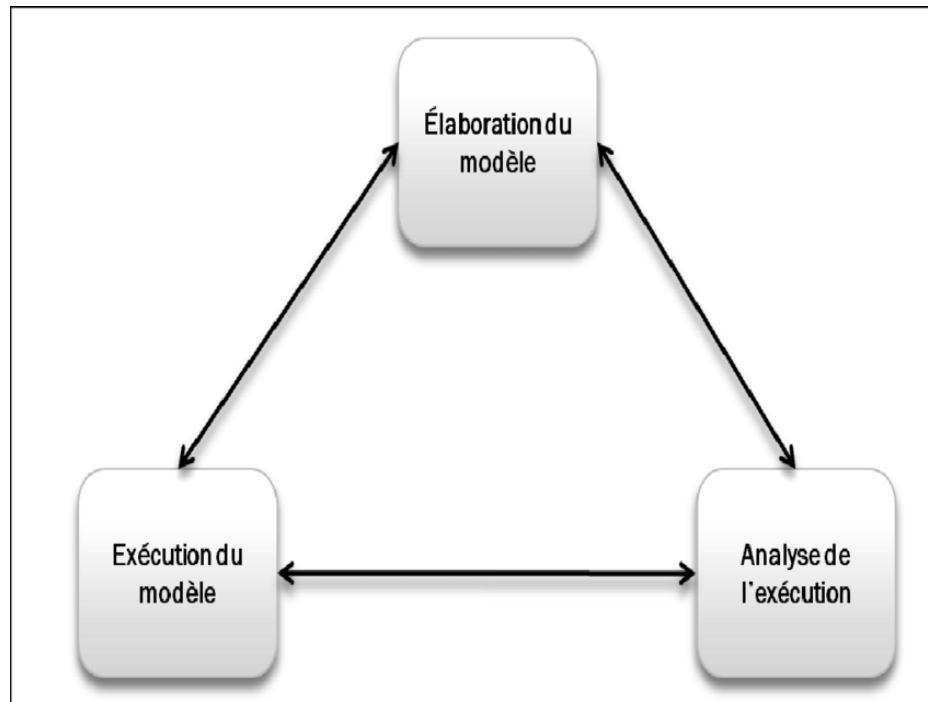


Figure: 1.4. Simulation informatique selon MICHEL¹⁴.

MICHEL définit ainsi cette discipline comme un processus itératif non linéaire composé de 3 tâches fondamentales, fortement interdépendantes :

- Elaboration du model.
- L'exécution du modèle sur ordinateur.
- L'analyse de l'exécution du modèle et des résultats obtenus.

Encore une fois, cette définition fait clairement apparaître l'importance du modèle, et donc de son élaboration, dans la conception d'une simulation. Par ailleurs, il est intéressant de remarquer que cet énoncé s'attache principalement à définir quelles sont les tâches fondamentales qui sont associées à cette démarche scientifique. En effet, au-delà des objectifs sous-tendus par ce type d'approche, il est dans un deuxième temps important de définir les différentes phases qui caractérisent sa mise en œuvre

A- Types

En fonction du type d'évènements dans la simulation, nous distinguons deux types de systèmes de simulation: les systèmes discrets et les systèmes continus.

B- Systèmes de simulation discrets

Il s'agit d'un système dans lequel les variables liées à la simulation ne changent d'état qu'en plusieurs points de l'axe du temps. Ces systèmes sont également appelés: systèmes de simulation à événements discrets.

C- . Systèmes de simulation continus

Les variables de ces systèmes peuvent changer d'état à tout moment pendant la simulation.

D- Concepts liés à la simulation

- **Evènement:** L'évènement est caractérisé par la date à laquelle l'état du système a changé (date de l'évènement). Il existe des évènements internes et externes (également appelés évènements endogènes et exogènes).
- **Activité:** Dans chaque évènement, des objets associés participent à l'opération, ces opérations initiées (ou terminées) sur chaque évènement sont appelées activités. Toute activité est limitée à son début et sa fin par évènement.
- **Processus:** Un processus est le groupement d'une séquence d'évènements dans l'ordre chronologique dans lequel ces évènements se produiront. Comme les évènements peuvent initier des activités, un processus peut inclure aussi des activités.

4-5 Vérification informatique:

En informatique, la validation de données est un processus permettant de s'assurer que les données ont subi un tri adéquat afin de ne garder que les données de qualité. Ainsi les données deviennent correctes et utiles. Le processus utilise des routines souvent nommées « règles de validation » ou bien « contraintes de validation » qui vérifient l'exactitude, la signification et la sécurité des données qui sont entrées dans le système de validation. Les règles peuvent être mises en œuvre via des fonctionnalités automatisées d'un dictionnaire de données ou par l'inclusion d'un logiciel doté d'une logique de validation.¹⁵

La validation des données est destinée à fournir certaines garanties bien définies pour l'adéquation, la précision et la cohérence pour tout type de saisie utilisateur dans une application ou un système automatisé. Les règles de validation des données peuvent être définies et conçues à l'aide de diverses méthodologies et être déployées dans différents contextes.

Des règles de validation des données peuvent être définies, conçues et déployées, par exemple:

Contextes de définition et de conception:¹⁶

- Dans le cadre de la phase de collecte des exigences dans une ingénierie logicielle ou la conception d'une spécification logicielle
- Dans le cadre d'une phase de modélisation des opérations BPM Départements :
- comme un ensemble de programmes ou de routines de logique métier dans un langage de programmation
- comme un ensemble de procédures stockées dans un système de gestion de base de données Pour les applications métier, la validation des données peut être définie via des règles déclaratives d'intégrité des données ou des règles métier basées sur des procédures. Les données non conformes à ces règles affecteront négativement l'exécution des processus métier. Par conséquent, la validation des données doit commencer par la définition du processus métier et un ensemble de règles métier au sein de ce processus. Les règles peuvent être collectées via l'exercice de capture des exigences.

5. Conclusion

Le modèle peut être caractérisé par quelques paramètres physiques, et il s'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 02 :

systeme multi-agent

1. Introduction:

Historiquement, les systèmes multi agents ont été appliqués pour la résolution de problèmes d'intelligence artificielle dans les années 70. En effet, à cette époque, l'évolution des domaines d'application de l'intelligence artificielle a investi des univers complexes et hétérogènes tels que l'aide à la décision, la reconnaissance et l'interprétation des formes, la conduite et le contrôle de processus industriels, etc... L'approche classique en IA qui s'appuyait sur une centralisation de l'expertise au sein d'un système unique a connu ses limites. Les chercheurs se sont tournés vers la distribution de l'expertise et sa décentralisation sur un ensemble d'agents. Ces agents doivent être capables de travailler et d'agir dans des environnements communs, ils doivent pouvoir communiquer, gérer des conflits éventuels, négocier des situations, Ceci a conduit à la naissance de nouvelles notions en IA telles que la coopération, la coordination d'actions, la négociation et l'émergence. Ce processus de distribution de l'intelligence est connu sous le nom d'intelligence artificielle distribuée (IAD) et le système d'agents sous le vocable système multi agents (SMA). Ce système se positionne au carrefour de plusieurs domaines qui sont : la programmation, l'intelligence artificielle et les systèmes répartis (distribués). Ce domaine représente, en quelque sorte ; le résultat de la rencontre de l'intelligence artificielle des Systèmes Distribués.

2. La notion d'agent :

2.1. Définition :

Etymologiquement, le mot **agent** est inspiré du latin "agere" qui signifie agir, donc littéralement, un agent est une entité qui agit.¹⁷ Il n'existe pas encore un consensus sur la définition du terme agent. Ceci étant dû à la relative jeunesse du domaine. En plus du fait que diverses communautés scientifiques revendiquent ce terme avec des problématiques qui leur sont propres. Même si ces différences sont complémentaires et conduites à se rencontrer à terme, sans oublier que la définition

d'agent peut varier selon le type d'application pour laquelle il a été conçu. Jacques Ferber¹⁸ définit l'agent comme étant : "une entité autonome physique ou abstraite qui est capable d'agir sur elle-même et sur son environnement, qui dans un univers Multi Agents peut communiquer avec d'autres agents et dont le comportement est la conséquence de ses observations, de ses connaissances, et des interactions avec les autres agents",

2.2/ Environnement :

L'environnement est un aspect très important dans l'étude des systèmes multi agents, puisque dans cet environnement ils vivent, ils le perçoivent à travers ses capteurs et agit sur lui à travers ses effecteurs.

Donc un agent doit avoir une bonne modélisation de son environnement assez complète pour pouvoir agir efficacement. Mais elle ne peut aucunement être complète due à la nature des environnements et des problèmes à résoudre avec tels systèmes.

Bien que les environnements sont différents, et il y a beaucoup de critères qui entre dans leurs propriétés, il est difficile d'avoir une liste exhaustive de leurs caractéristiques. Une classification des environnements est celle de Russell et Norvig¹⁹

- **Accessible versus inaccessible.** Un environnement est accessible si l'agent peut avoir une vue complète, exacte et actualisée sur l'état de l'environnement. Et la plupart des environnements réels ne sont pas accessible dans ce sens.
- **Déterministe versus non déterministe.** Un environnement est déterministe si toute action sur cet environnement a un seul effet, c à d qu'il n y a pas une ambiguïté sur l'état qui résulte après l'achèvement d'une action.
- **Statique versus dynamique.** Un environnement est statique s'il est supposé inchangé sauf par les actions de l'agent. Par contre un environnement dynamique peut avoir des changements au-delà de la portée de l'agent.
- **Discret versus continu.** Un environnement discret est un environnement où les perceptions de l'agent et ses actions sont au nombre limité à l'inverse d'un environnement continu.

A travers cette classification on peut considérer un environnement idéal comme étant accessible, déterministe, statique et discret, mais en réalité que les environnements existants possèdent toute ces caractéristiques d'une manière plus au moins différente.

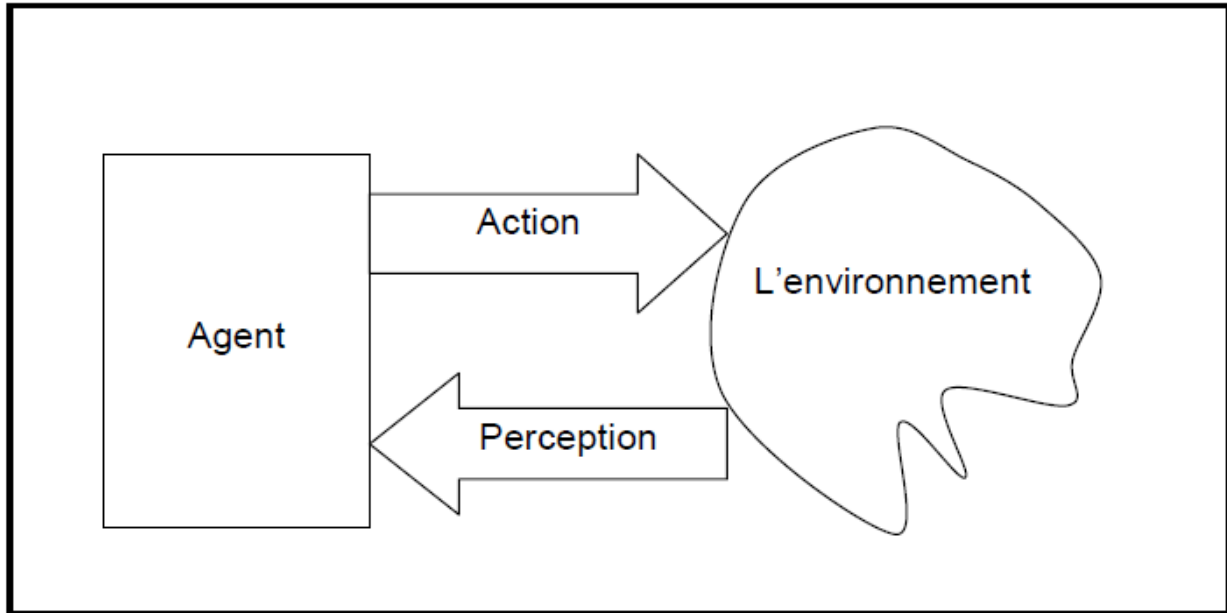


Figure 2.1 L'agent et son environnement

On peut classer les agents selon leurs interactions avec son environnement entre des agents *purement communicants* et des agents *purement situés*,

Un agent *purement communicant* est un agent qui n'a pas une perception de son environnement, et il n'agit pas directement sur cet environnement, comme par exemple il s'agit des agents mobiles.

Un agent *purement situé* est quant à lui différent des autres agents par le fait qu'il ne possède pas une perception des autres agents, ces agents ne communiquent pas directement avec les autres agents mais indirectement à travers ses perceptions et actions sur l'environnement²⁰

2.3 Architectures d'agents

Un agent se caractérise essentiellement par la manière dont il est conçu et par ses actions, en d'autres termes par son architecture et par son comportement. L'architecture correspond à un point de vue de concepteur, qui peut se résumer par la façon d'assembler les différentes parties d'un agent de manière qu'il accomplisse les actions que l'on attend de lui?

Suivant la façon dont un agent prend les décisions de ses réactions on peut distinguer plusieurs types d'agents.

L'architecture d'un agent caractérise ainsi sa structure interne, c'est-à-dire le principe d'organisation qui sous-tend l'agencement de ses différents composants

2.3.1 Agent basé sur la logique

Cette architecture est basée sur une représentation symbolique de l'environnement, ainsi que des actions que l'agent doit entreprendre. Puis une suite de déductions logiques qui permettent de faire la liaison entre les perceptions (qui sont déjà représentées symboliquement), et les actions qui conviennent.

2.3.2 Agent réactif

Un agent réactif est un agent qui réagit aux changements de l'environnement en suivant des règles de correspondance directes entre un ensemble de perceptions et un ensemble d'actions.

On peut trouver deux modèles de conception d'agents réactifs :

2.3.2.1 Agents à reflex simple

Ce type d'agent réagit seulement en fonction de ses perceptions actuelles, il utilise des règles de la forme : **Si** condition **Alors** action.

2.3.2.2 Agents conservant une trace du monde

A la différence de type précédent ce type d'agent considère en plus de la perception des capteurs, des informations sur comment le monde évolue, et l'impact des actions de l'agent sur son environnement sont aussi prises.

Un tel agent doit avoir un mécanisme qui (en fonction de l'historique de son environnement) permet de distinguer deux situations ayant des perceptions identiques, mais qui sont en réalité différentes

Par exemple, si un radar détecte un missile et que l'instant d'après, il le perd de vue, du à un obstacle, cela ne signifie nullement qu'il n'y a plus de missile. Dès lors, l'agent en charge du contrôle doit tenir compte de ce missile dans le choix de ses actions et ce, même si le radar ne détecte plus le missile. Le problème que nous venons de mentionner survient parce que les capteurs de l'agent ne fournissent pas une vue complète du monde. Pour régler ce problème, l'agent doit maintenir des informations précédentes sur l'état du monde, et utilise ses informations lors de la prise des prochaines actions.

2.3.3 Agent délibératif

Un agent délibératif est un agent qui fait une suite de délibération pour choisir ses actions, soit en se basant sur les buts, soit en utilisant une fonction d'utilité.

2.3.4 Agent BDI (*Belief-Desire-Intention*)

BDI est l'acronyme de Belief-Desire-Intention qui signifie en français de croyances, désires, intention. Donc le comportement d'un agent BDI est basé sur ces trois concepts. Un agent BDI est composé de :

- Un ensemble de croyances courantes : représentent les informations que l'agent a maintenant sur son environnement.
- Une fonction de révision de croyances : permet de mettre à jour les croyances de l'agent en fonction de ses croyances précédentes et de l'état actuel de l'environnement.
- Une fonction de génération des options : elle détermine les options disponibles actuellement à l'agent (ses désires, ou encore les choix des actions à entreprendre).
- Un ensemble d'options courantes : qui sont les désires ou les alternatives possibles pour l'agent.
- Une fonction de filtre : c'est le processus de délibération de l'agent qui en fonction de croyances, désires, et intentions actuelles de l'agent génère les nouvelles intentions.
- Un ensemble d'intentions courantes : C'est le centre d'attention actuel de l'agent, c'est les objectifs qu'il essaye d'atteindre.
- Une fonction de sélection des actions : cette fonction détermine les actions à effectuer en se basant sur les intentions actuelles de l'agent.

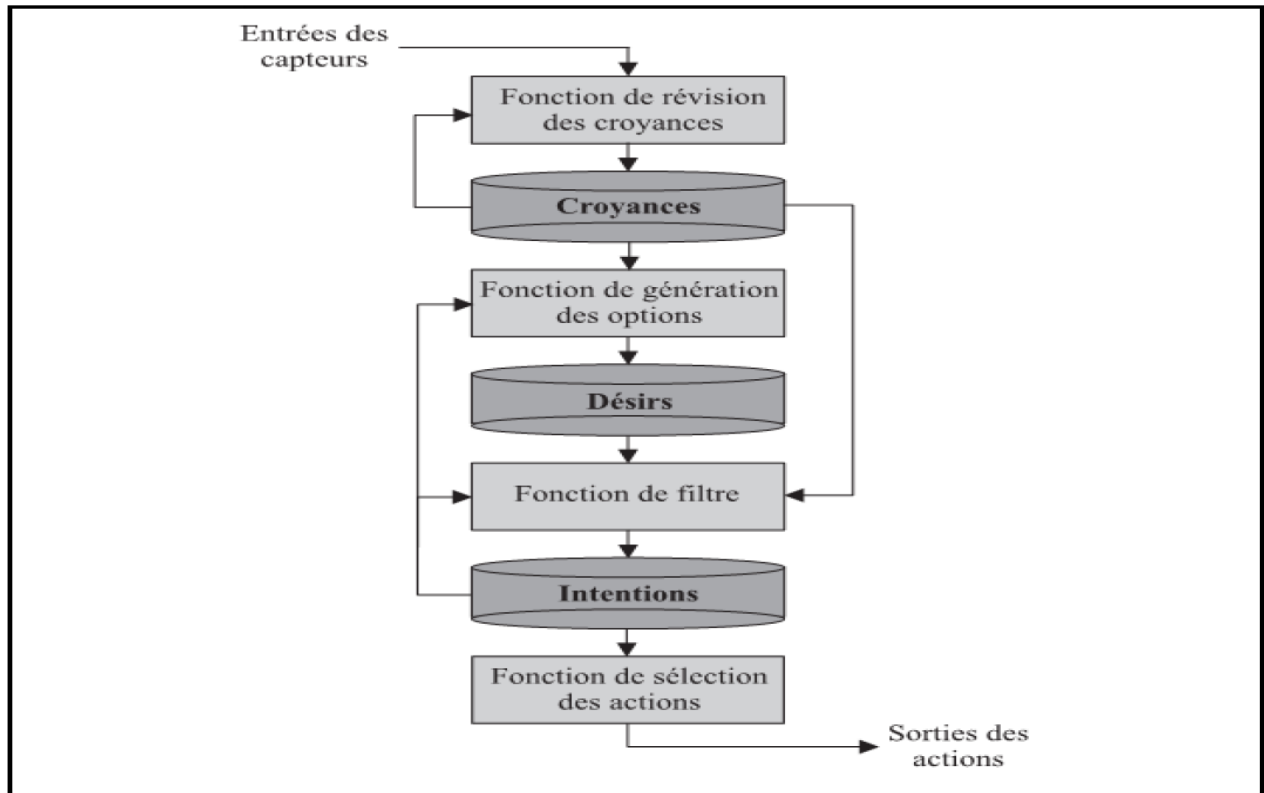


Figure 2.1: Architecture d'un agent BDI

2.3.5 Caractéristiques d'un système multi-agents

D'après²¹ un système Multi-agents est caractérisé par les faits suivants :

- Chaque agent accède à un ensemble d'informations et dispose de capacités de résolution de problèmes limitées, ainsi chaque agent dispose d'une vue partielle de l'environnement dans lequel il évolue et des autres agents du système.
- Il n'y a pas de contrôle global du système multi-agents,
- Les données sont décentralisées,

De même d'autres caractéristiques peuvent être liées aux agents ²²:

- Distribution : L'élément de base du système est l'agent. le système est modulaire.
- Autonomie : un agent prend ses propres décisions en fonction de ses buts, de ses objectifs et en fonction de ses capacités et connaissances. Son activité est permanente et il est en évolution permanente du point de vue apprentissage.

- Décentralisation : les agents agissent et réagissent de façon indépendante, il n'y a pas de décisions centrales valables pour tout le système.
- Échange de connaissances : les agents sont capables de s'échanger des informations entre eux en utilisant des langages et des protocoles plus ou moins élaborés.
- Interaction : les agents ont des influences les uns sur les autres ce qui permet d'engendrer des comportements menant à la réalisation de buts communs.
- Organisation : les structures matérielles et logicielles des agents créent des relations entre les agents et le réseau de ces relations forme une organisation qui peut évoluer au cours du temps.
- Situation dans un environnement : L'environnement dans lequel évoluent les agents crée les agents une source de données, de contraintes et d'incertitudes. C'est un lieu d'actions et d'influences entre agents. L'évolution du SMA est la combinaison des évolutions des agents et de l'environnement.
- Ouverture : le système est en continuelle mutation et évolution, de nouveaux agents peuvent être créés d'autres détruits dans le SMA ou encore certains peuvent être modifiés.
- Émergence : Dans un SMA, la notion d'émergence est primordiale. Cette notion peut être vue comme étant la résultante des comportements et actions individuels des agents qui ne possèdent qu'une vue partielle du système et dont le comportement global mène à la résolution d'un problème distribué.
- Adaptation : Spécifier le but global du système est une tâche difficile et ardue. Cependant, le système adapte son comportement à l'environnement en cours de fonctionnement, et offre une robustesse de ce comportement, à défaut d'une optimisation.
- Délégation : Le contrôle de l'application est délégué aux agents du système.
- Personnalisation : Un agent du système peut être amené à jouer un rôle qui lui est propre.
- Intelligibilité : Les SMA sont considérés comme des outils permettant de modéliser des systèmes en proposant leurs découpages en entités 'maîtrisables' et en proposant des applications pour les appréhender de façon simple et naturelle.

3. Les systèmes Multi-Agents (SMA)

3.1 Définition

Une des principales propriétés de l'agent dans un SMA est celle d'interagir avec les autres agents. Ces interactions sont généralement définies comme toute forme d'action exécutée au sein du système d'agents et qui a pour effet de modifier le comportement d'un autre agent. Elles permettent aux agents de participer à la satisfaction d'un but global.

Cette participation permet au système d'évoluer vers un de ses objectifs et d'avoir un comportement intelligent indépendamment du degré de complexité des agents qui le composent. En général, les interactions sont mises en œuvre par un transfert d'informations entre agents ou entre les agents et leur environnement, soit par perception, ou bien par communication. Par la perception, les agents détiennent la connaissance de changement de comportement d'un tiers au travers du milieu. Par la communication, un agent fait un acte délibéré de transfert d'informations vers un ou plusieurs autres agents.

Le degré de complexité des connaissances nécessaires pour traiter les interactions dépend des capacités cognitives (de raisonnement) de l'agent et du fait que l'agent a ou non une connaissance de l'objectif du système global. En effet, un agent qui poursuit un objectif individuel au sein du système comme c'est le cas pour les agents dits réactifs ne focalise pas son énergie pour interagir avec les autres même s'il y est amené. Par contre, un agent qui participe à la satisfaction du but global du système tout en poursuivant un objectif individuel, va passer une partie de son temps à coopérer ou à se coordonner avec les autres agents. Pour cela, il doit posséder des connaissances sociales qui modélisent ses croyances sur les autres agents.

3.2. De l'Intelligence Artificielle à l'Intelligence Artificielle Distribuée

La création des machines intelligentes est l'un des rêves de l'humanité depuis l'antiquité. Il existe des diverses traces qui montrent cet énoncé dans l'Égypte ancienne ou dans l'Iliade. Cependant, la plupart de ces traces ne prennent qu'une forme mythologique. La déduction formelle proposée par Aristote (le syllogisme) peut être considérée comme le premier pas vers le raisonnement automatique. Cette idée consiste à la possibilité de valider la véracité d'une conclusion à partir de deux prémisses. Par exemple, si on a les prémisses «Tous les hommes sont mortels » et « Mohamed est un homme » on peut alors conclure que « Mohamed est mortel ». Sans doute, le raisonnement formel est l'une des questions fondamentales de l'intelligence artificielle.

L'invention des ordinateurs ne représente qu'un moyen pour la réalisation de ce rêve. En fait, après l'invention des ordinateurs plusieurs chercheurs supposent que ce rêve puisse être réalisé si on peut traiter les symboles en utilisant les ordinateurs. Cependant, plusieurs objections ont été posées et touchent parfois même la nature de l'intelligence. En fait, plusieurs philosophes considèrent l'intelligence comme une caractéristique humaine (caractéristique liée à l'esprit humaine) irréalisable par les machines.

En 1950, Alain Turing propose dans un célèbre article, un test empirique pour dépasser le débat de la faisabilité de réalisation des machines intelligentes. Ce test est connu par « le test de Turing ». Ainsi, une personne peut tester la machine via une interface. La personne chargée de test pose des questions à la machine. La machine est considérée comme intelligente, si on ne peut pas distinguer ses réponses de réponses d'un être humain.

En 1956, le domaine de l'intelligence artificielle est officiellement né dans la conférence de Dartmouth avec l'objectif principal de faire faire de la machine ce que l'homme peut faire. Ce domaine est un domaine pluridisciplinaire. Il englobe l'informatique, la philosophie, la psychologie, la robotique et la linguistique. Ce domaine est, à la fois, un domaine d'ingénierie et de science. D'une part, l'objectif c'est la réalisation d'une machine intelligente (l'ingénierie). D'autre part, l'objectif est de réaliser des modèles permettant la compréhension et la validation des théories des sciences cognitives (la science).

Les participants à la conférence de Dartmouth ont été très optimistes. Ils ont estimé que la réalisation des machines intelligentes est une question de quelques années. A titre exemple, en 1970 Minsky (l'un des fondateurs de l'IA) a annoncé que la réalisation d'une machine avec une intelligence générale d'un être humain ordinaire va prendre de trois à huit ans. Cependant, la complexité de la tâche a montré la fausseté de ces estimations. Par exemple, Newell et Simon déclarent en 1958 qu'après dix ans le champion de monde en jeu d'échec sera un ordinateur. En fait, il faut attendre jusqu'à l'année 1997 pour réaliser « Deep Blue » (le vainqueur de Kasparov – champion du monde en jeu d'échec).

Vers la fin des années soixante-dix, un nouveau courant est apparu. Il consiste à exploiter les progrès des systèmes distribués dans l'intelligence artificielle afin de pallier la complexité des problèmes de l'IA. Ainsi, le domaine de l'intelligence artificielle distribuée (IAD) est apparu.

3.3 L'intelligence artificielle distribuée

A la fin des années soixante-dix, un système appelé « Hearsay II » a été développé au sein de l'université Carnegie-Mellon pour la reconnaissance automatique de la parole. Ce système est basé sur l'utilisation d'un tableau noir. Comme il est présenté dans la Figure 2.2, ce système est composé d'un ensemble de sources de connaissances (SC) qui partagent le tableau noir. Ce dernier est organisé en plusieurs niveaux. En fait, un problème de reconnaissance automatique de la parole se caractérise par la manipulation des connaissances de niveaux d'abstraction différents (signal, entité lexical, entité syntaxique,... etc.). En plus, chaque source de connaissances est spécialisée dans une tâche spécifique (comme la segmentation, la création des syllabes, la construction des mots, ...etc). Donc, chaque source de connaissances manipule des connaissances d'un niveau spécifique et génère des connaissances pour d'autres niveaux. Les changements effectués sur un niveau du tableau noir peuvent activer certaines sources de connaissances. Ces dernières doivent être insérées à une liste d'ordonnancement. Ensuite, un ordonnanceur choisit la source de connaissance la plus prioritaire pour être exécuter.

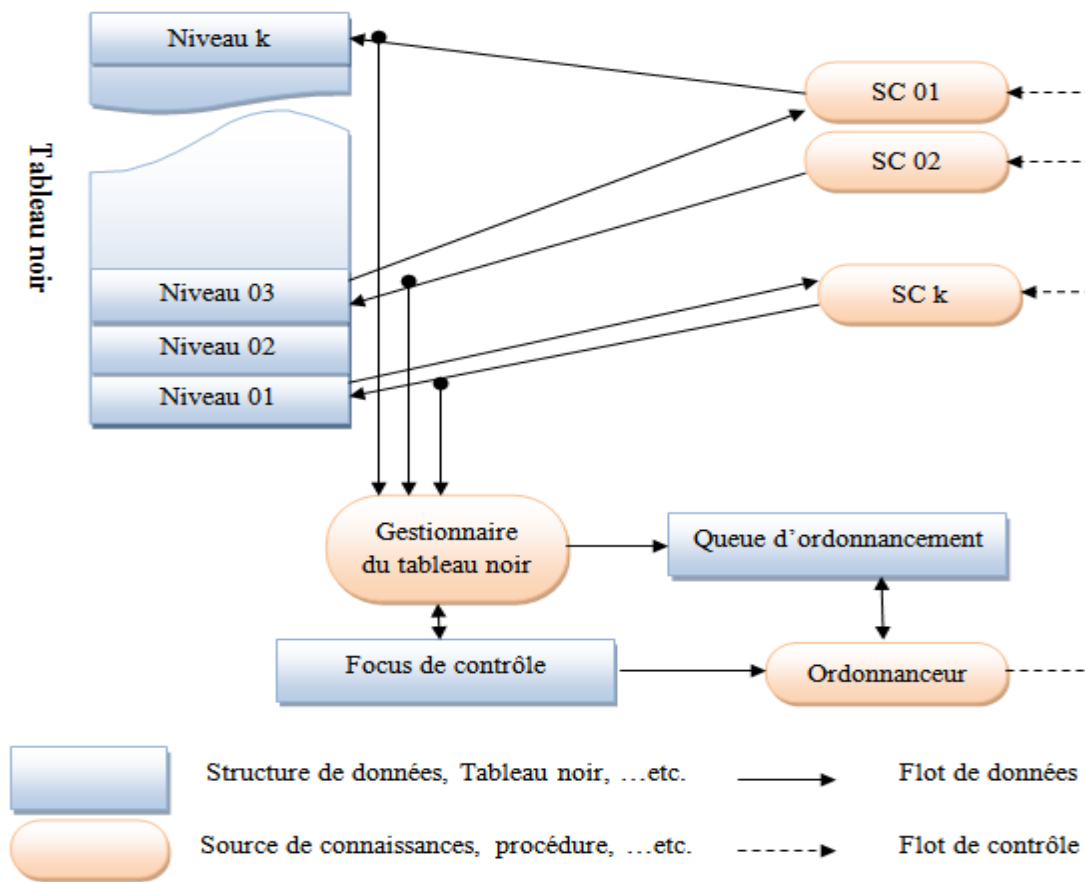


Figure 2.2 : Le système HearsayII.

Par contre au système de l'IA classique, ce système est composé d'un ensemble de sources de connaissances distribuées. En fait, le système Hearsay II est l'un des premiers systèmes de l'IAD. En conséquence, nous pouvons définir l'intelligence artificielle distribuée comme une branche de l'intelligence artificielle classique qui s'intéresse à des comportements intelligents produisant par l'activité de plusieurs entités. Il est important de noter que cette branche ne traite pas les problèmes de bas-niveau de parallélisme qui sont des problèmes au coeur du domaine des systèmes parallèles et distribués.

4. Plateformes multi agents

Il y a deux grands groupes de méthodologies pour le développement orienté agent. Le premier groupe étend ou adapte les méthodologies orientées objet pour mettre en oeuvre les caractéristiques des agents. Beaucoup de méthodologies orientées objet sont utilisées pour développer des systèmes multi agents dans le domaine industriel avec succès, Cependant les extensions et les modifications des méthodologies orientées objet doivent prendre en compte les différences qui existent entre les objets et les agents. Pour le deuxième groupe, il part des méthodologies qui ont été dédiées à la modélisation des systèmes à bases de connaissances.

On peut aussi trouver un certain nombre de plateformes dédiés au développement de systèmes multi agents, dont certain logiciel libre, la plus connue est JADE.

- **JADE** (Java Agent Development Framework - Bellifemine, Poggi, Rimassa, 1999) est une plate-forme multi-agents développée en Java par CSELT (Groupe de recherche de Gruppo Telecom, Italie) qui a comme but la construction des systèmes multi-agents et la réalisation d'applications conformes à la norme FIPA. ²³JADE comprend deux composantes de base : une plate-forme agents compatible FIPA et un paquet logiciel pour le développement des agents Java. Le but de JADE est de simplifier le développement des systèmes multi-agents en conformité avec la norme FIPA pour réaliser des systèmes multi-agents interopérables.

-La plate-forme ZEUS

ZEUS (Nwama e.a., 1999)²⁴ est une plate-forme multi-agents conçue et réalisée par British Telecom (Agent Research Programme of BT Intelligent Research Laboratory) pour développer des applications collaboratives. ZEUS est écrit dans le langage Java et il est fondé sur les travaux de la FIPA

- La plate-forme MADKIT

MADKIT (Madkit, 2003) est une plate-forme développée par le Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM) de l'Université Montpellier II.

MADKIT est libre pour l'utilisation dans l'éducation. Il est écrit en Java et est fondé sur le modèle organisationnel Alaadin. Il utilise un moteur d'exécution où chaque agent est construit en partant d'un micro-noyau. Chaque agent a un rôle et peut appartenir à un groupe. Il y a un environnement de développement graphique qui permet facilement la construction des applications.²⁵

- La plate-forme SWARM

SWARM (Minar e.a., 1996) est une plate-forme multi-agents avec agents réactifs. L'inspiration du modèle d'agent utilisé vient de la vie artificielle. SWARM est l'outil privilégié de la communauté américaine et des chercheurs en vie artificielle. L'environnement offre un ensemble de bibliothèques qui permettent l'implémentation des systèmes multi-agents avec un grand nombre d'agents simples qui interagissent dans le même environnement. De nombreuses applications ont été développées à partir de SWARM qui existe aujourd'hui implémenté en plusieurs langages (Java, Objective-C).²⁶

5. Les applications des SMA

Les systèmes multi-agents sont convenables, généralement, pour le développement **des systèmes complexes et les systèmes répartis et hétérogènes** où la solution est le résultat de **l'interaction de plusieurs entités**. En conséquence, plusieurs domaines sont considérés comme des domaines d'application idéals. Bien entendu, il est difficile de citer tous les domaines d'application des systèmes multi-agents. En fait, nous avons choisi de citer dans ce cours les domaines d'application dans lesquels des applications réelles ont été vu le jour. Ainsi, les systèmes multi-agents ont été appliqués dans les domaines suivants :

- **La gestion des processus métier** : les décisions prises dans les grandes entreprises sont basés sur des informations provenant de plusieurs sources. Le défi réel est de prendre des décisions correctes avec des informations incomplètes ou obsolètes. Plusieurs chercheurs trouvent que les systèmes multi-agents sont idéaux pour travailler dans ces conditions. Le système ADEPT est développé pour ce but. Ce système est composé d'un ensemble d'agents qui modélisent les rôles et les départements d'une entreprise. Bien entendu, chaque agent offre un ensemble de services. Un agent peut demander des services d'autres agents à travers un processus de négociation permettant de fixer le coût, les délais et la qualité du service demandé. Un processus de négociation terminé avec succès représente un engagement entre les parties de négociation.

- **Le domaine médical :** le domaine médical en générale, et les soins intensifs chirurgicaux particulièrement se caractérisent par plusieurs attributs favorisant l'utilisation des systèmes multi-agents. Les systèmes de soins intensifs sont organisés en équipes avec plusieurs experts qui coopèrent et partagent de l'information afin d'atteindre leurs objectif. En plus, le niveau d'expertise des intervenants n'est pas le même. En fait, certains intervenants n'ont pas les compétences nécessaires pour traiter des situations données. Par exemple, les infirmières ne peuvent pas interpréter les données rassemblées. Le système **GUARDIEN** modélise parfaitement ces systèmes. Il est composé d'un ensemble d'agents permettant la perception des données, le raisonnement, le contrôle et le transfert des commandes aux effecteurs.
- **Les systèmes d'informations coopératifs :** actuellement, les informations sont répartis partout (dans les réseaux locaux ou le réseau Internet) avec des structures et des natures hétérogènes. Les systèmes d'information coopératifs (SIC) sont des systèmes capables de rechercher, filtrer et fusionner les informations. Ces systèmes peuvent être exploités dans le domaine financière (trouver des prix d'action, des informations de bourse,...etc.), le domaine touristique (trouver les itinéraires pour une destination donnée, des informations sur les sites d'hébergement, les moyens de transports, ...etc.), les chaînes logistiques,...etc. Dans le domaine financier, le système multi-agents **Warren** est un exemple typique. Ce système est composé d'un ensemble d'agents qui collectent des informations de web et de présenter ces informations aux usagers.

5. Conclusion

Dans ce chapitre, nous avons présenté les concepts d'agent et de système multi-agents ainsi que leurs caractéristiques. De plus, nous avons étudié les méthodologies de conception des SMA et les plateformes permettant de les mettre en œuvre. Il faut noter que l'on dénote un besoin insistant de la standardisation et de l'uniformisation de ces méthodologies et plateformes. De plus, les équipes de chercheurs ont axé leurs recherches sur des sujets précis et pointus, ce qui s'est reflété sur les outils et environnements qu'ils ont développés.

Dans le chapitre suivant, nous allons présenter la description de system à simuler.

CHAPITRE 03 :
DESCRIPTION DE SYSTEM A SIMULER

1. Introduction

Un procédé industriel est un procédé de nature mécanique ou chimique destiné à synthétiser des produits chimiques, en grande quantité et dans des conditions acceptables.

Dans ce chapitre, nous allons présenter les concepts de base qui constituent presque tous les procédés industriels, ensuite nous allons étudier un système industriel de Colonne d'épuration de gaz

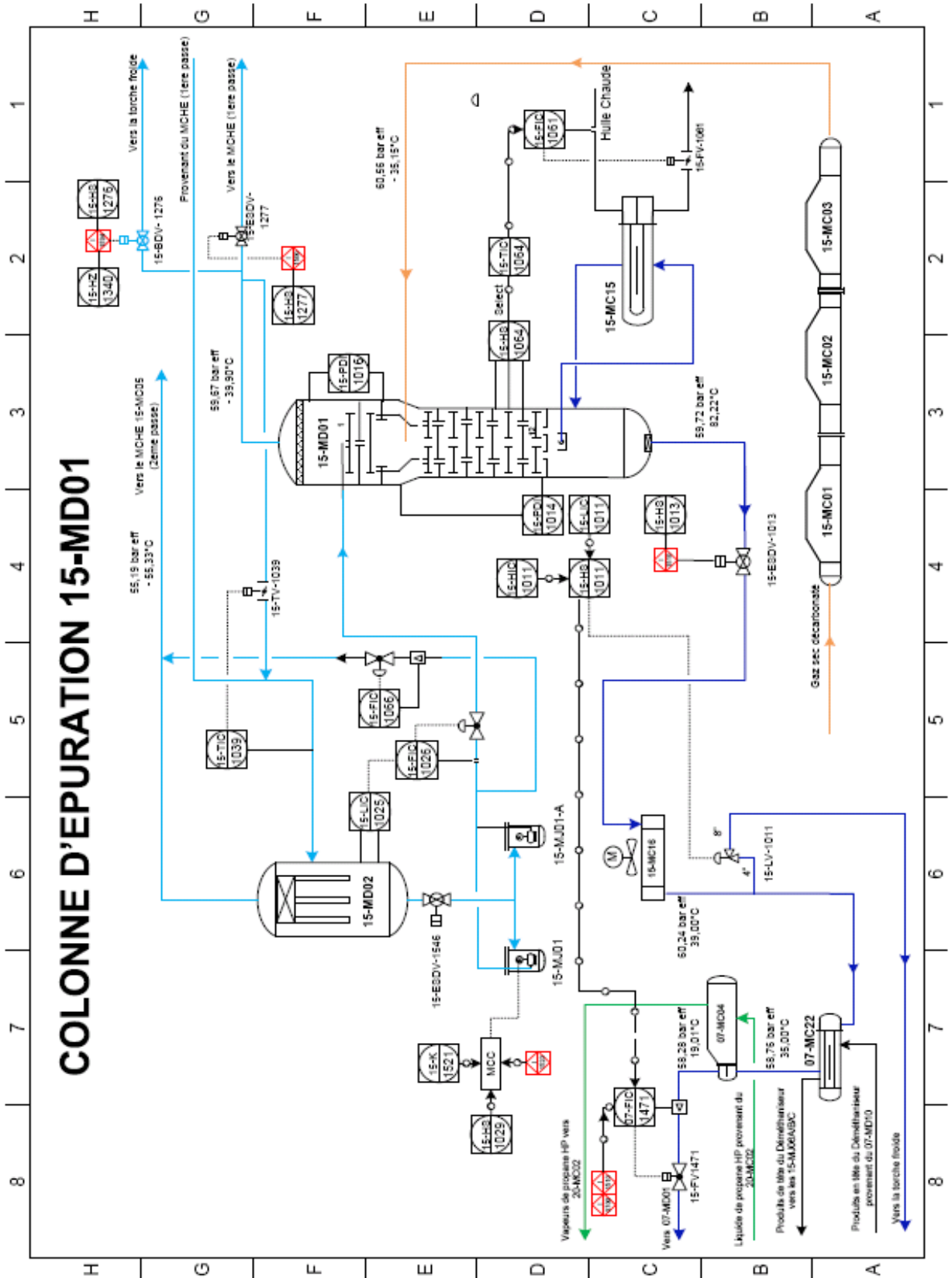


Figure 3.1: Détails Colonne D'epuration 15-Md01

2. Description du système

2.1 Pré-refroidissement

Le gaz d'alimentation venant du système de démercurisation à un débit de 813 049 kg/h se mélange avec le gaz de recyclage venant du système compresseur de recyclage du BOG dont le débit est de 27 172 kg/h.

Ce mélange de gaz circule à travers les refroidisseurs de gaz d'alimentation propane HP, MP, BP 15-MC01/2/3 à un débit de 840 221 kg/h à 22°C et 62 bar eff. Ceci permet de baisser sa température à -35°C et condenser partiellement le gaz d'alimentation avant d'entrer dans la colonne d'épuration 15-MD01.

Ces refroidisseurs sont des échangeurs de type Kettle à tubes renforcés du fait qu'un débit biphasique est généré au moment où il traverse ces tubes.

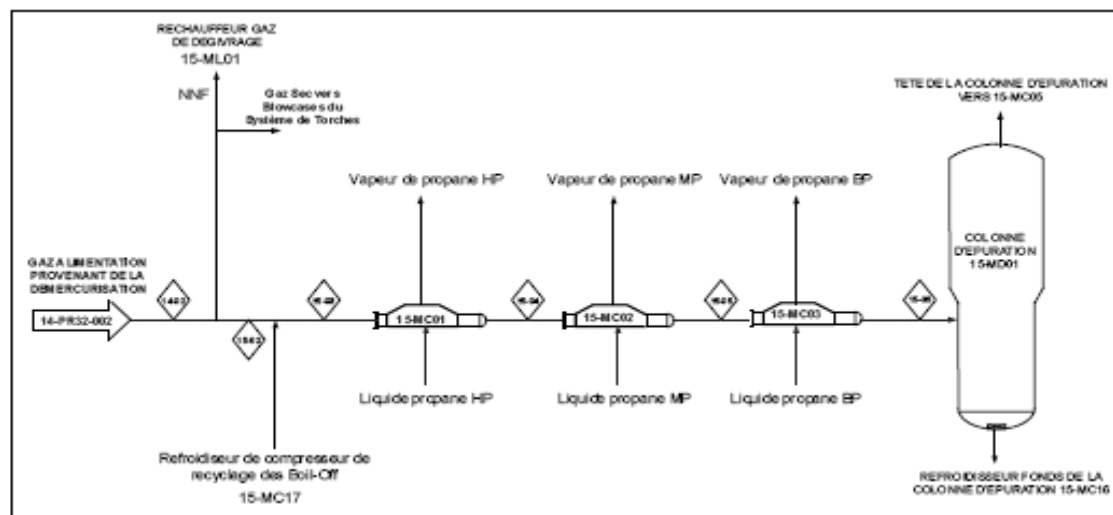


Figure 3.3: Refroidisseurs de gaz d'alimentation

2.2 Lavage

La colonne d'épuration 15-MD01 élimine les hydrocarbures lourds et tous les aromatiques contenues dans le gaz d'alimentation, et ce afin d'éviter le colmatage dans l'échangeur principal 15-MC05.

La colonne d'épuration augmente les taux d'éthane et de propane qui sont utilisés comme réfrigérants pour le procédé de liquéfaction, et aide à produire un GNL selon les spécifications commerciales.

Le GN venant du refroidisseur de GN/propane BP 15-MC03 entre dans la colonne d'épuration 15-MD01 à un débit normal de 840 221kg/h à -35°C et à 60,6 bar eff.

Le rebouilleur de la colonne d'épuration 15-MC15 utilise de l'huile chaude pour le réchauffage. La colonne d'épuration est équipée d'un plateau de soutirage pour diriger le liquide vers le rebouilleur de la colonne d'épuration. Un débit d'hydrocarbure mélangé (venant du rebouilleur de la colonne d'épuration) est renvoyé vers la colonne en-dessous du plateau de soutirage à partir duquel il monte à travers les cheminées et les plateaux à clapets.

Il y a deux courants qui quittent la colonne d'épuration; le courant de fond et le courant de tête. Le courant de fond, constitué d'hydrocarbures lourds et d'aromatiques, est refroidi dans le refroidisseur de fond de la colonne d'épuration 15-MC16. Il est ensuite dirigé vers l'unité 07 de fractionnement.

L'unité de fractionnement sépare ce débit de ses composants d'éthane, de propane, de butane, d'isopentane et de condensat (gazoline naturelle). Le courant de tête de la colonne d'épuration est mélangé avec du GPL récupéré du fractionnement.

Le courant mélangé est ensuite envoyé vers le bout chaud du 15-MC05, et en ressort en deux-phases. Enfin, il alimente le ballon de reflux de la colonne d'épuration 15-MD02. Le ballon de reflux sépare la phase liquide et la renvoie vers la colonne d'épuration en reflux grâce aux pompes de reflux de la colonne d'épuration 15-MJ01/A. La phase vapeur venant du 15-MD02 revient au milieu du 15-MC05 à - 55°C en tant que constituant principal de la production de GNL.

La température du ballon de reflux est contrôlée par le 15-TIC-1039 qui dévie une partie du courant de tête de la colonne d'épuration allant vers le 15-MC05.

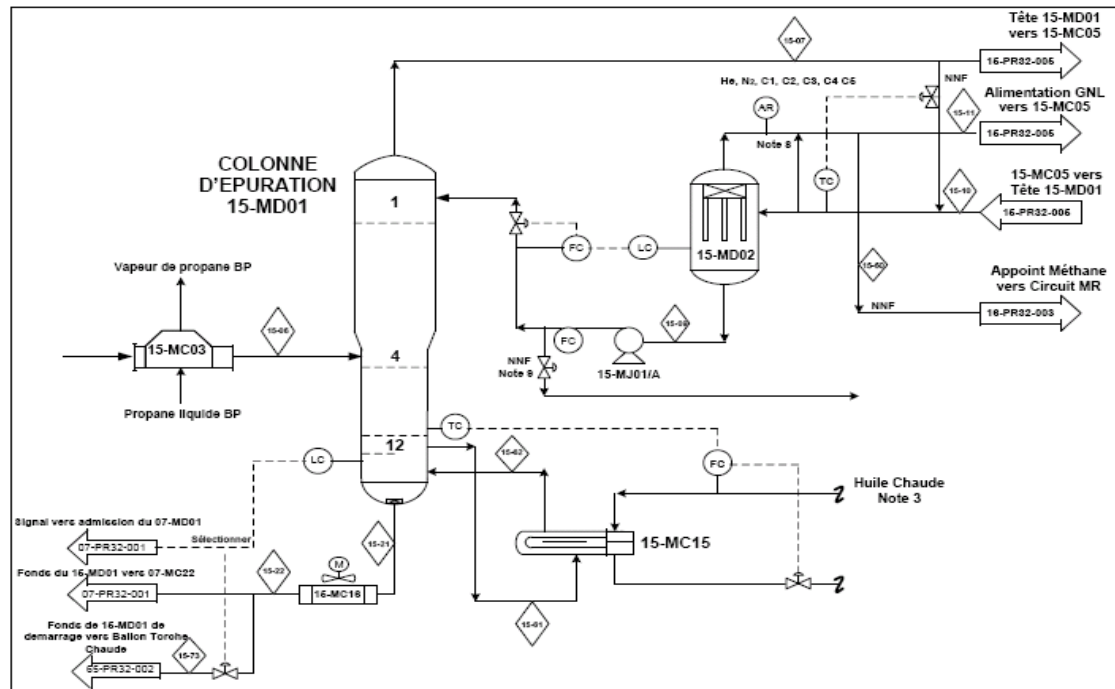


Figure 4.3. La température du ballon de reflux

3. Exploitation

3.1 Marche normale

3.1.1. Refroidisseur de gaz d'alimentation/propane HP 15-MC01

Le refroidisseur de gaz d'alimentation/propane HP 15-MC01 est un refroidisseur de type Kettle qui utilise le propane (HP à 2,7 bar eff) comme refroidisseur sur le côté calandre.

Le niveau du propane HP, côté calandre, est contrôlé par le 15-LIC-1005 qui règle la vanne d'admission de propane. La vapeur de propane HP sortante du côté calandre du 15-MC01 alimente le ballon d'aspiration de propane HP 16-MD04.

3.1.2. Refroidisseur de gaz d'alimentation/propane MP 15-MC02

Le refroidisseur de gaz d'alimentation/propane MP 15-MC02 utilise le propane (MP à 1,0 bar eff) comme refroidisseur sur le côté calandre.

Le niveau du propane MP, côté calandre, est contrôlé par le 15-LIC-1007 qui règle la vanne d'admission de propane. La vapeur de propane MP sortante du côté calandre du 15-MC02 alimente le ballon d'aspiration de propane MP 16-MD03.

3.1.3. Refroidisseur de gaz d'alimentation/propane BP 15-MC03

Le refroidisseur de gaz d'alimentation/propane BP 15-MC03 utilise le propane (BP à 0,2 bar eff) comme refroidisseur du côté de la calandre.

Le niveau du propane BP, côté calandre, est contrôlé par le 15-LIC-1020 qui règle la vanne d'admission de propane. La vapeur de propane BP sortante du côté calandre du 15-MC03 alimente le ballon d'aspiration de propane BP 16-MD02.

3.1.4. Colonne d'épuration 15-MD01

Le commutateur sélecteur 15-HS-1011 permet à l'opérateur de choisir entre le contrôle de niveau en cascade et le contrôle de niveau direct des produits de la colonne d'épuration.

Quand le commutateur 15-HS-1011 est sur OFF, le régulateur de niveau 15-LIC-1011 maintient le niveau dans la colonne d'épuration en réglant le point de consigne du régulateur de débit 07-FIC-1471 qui agit sur la vanne de contrôle 15-LV-1471. Le liquide est drainé de la colonne vers le refroidisseur de fond 15-MC16 puis alimente le réchauffeur des produits de tête du déméthaniseur 07-MC22 et ensuite alimente le refroidisseur d'alimentation du déméthaniseur 07-MC20 de l'unité 07. La vanne de débit 07-FV-1471 règle le débit du courant de fond qui quitte le 07-MC20.

Quand le commutateur 15-HS-1011 est sur ON, le régulateur 15-FIC-1471 est permuté du mode cascade et mis en contrôle de point de consigne localement. Le régulateur de débit 07-FIC-1471 contrôle la vanne 07-FV-1471. Ensuite, le régulateur 15-LIC-1011 contrôle la vanne 15-LV-1011 afin de maintenir le niveau de la colonne d'épuration en purgeant le liquide vers la torche froide.

La température de la colonne d'épuration est contrôlée par la cascade 15-TIC-1064/ 15-FIC-1061. Cette cascade permet le réglage de la température et ce, en contrôlant le débit de l'huile chaude vers le rebouilleur de la colonne d'épuration 15-MC15. Trois indicateurs de température de la colonne (15-TI-1064, 15-TI-1434 et 15-TI-1435) indiquent les températures des différents plateaux, et sont transmises au commutateur sélecteur 15-HS-1064. Le signal sélectionné est transmis au 15-TIC-1064 pour contrôler la température.

3.1.5. Refroidisseur de fond de la colonne d'épuration 15-MC16

Le refroidisseur de fond de la colonne d'épuration 15-MC16 est un refroidisseur de type ailette-ventilateur qui refroidit les produits de fond de la colonne d'épuration 15-MD01 de 82°C à 39°C. Il est constitué d'une batterie contenant trois (03) ventilateurs à vitesse-fixe. Les trois (03) ventilateurs sont démarrés et arrêtés manuellement par l'opérateur afin de maintenir une température normale de 39°C.

Le nombre de ventilateurs nécessaires à la réalisation de ce refroidissement varie en fonction de la température ambiante.

3.1.6. Rebouilleur de la colonne d'épuration 15-MC15

Le rebouilleur de la colonne d'épuration 15-MC15 renvoie un courant partiellement vaporisé vers le fond de la colonne d'épuration 15-MD01.

La quantité de débit d'huile chaude vers le rebouilleur détermine la température de la colonne d'épuration. La cascade 15-TIC-1064/15-FIC-1061 agit sur le débit d'huile de réchauffage.

3.1.7. Ballon de reflux de la colonne d'épuration 15-MD02

Le courant de tête de la colonne d'épuration 15-MD01 alimente le bout chaud de l'échangeur principal 15-MC05 et revient vers le ballon de reflux de la colonne d'épuration 15-MD02. Il est ensuite partiellement condensé. En cas de faible débit, une injection du GPL, en tête de la colonne d'épuration en amont du bout chaud, permet de purifier les liquides et rétablir le débit de reflux.

La température du ballon de reflux est contrôlée par le 15-TIC-1039 en permettant à la vapeur de tête de by-passer le 15-MC05 à travers la vanne de contrôle 15-TV-1039.

Le niveau du ballon de reflux est contrôlé par le régulateur de niveau 15-LIC-1025 à travers le régulateur de débit de reflux 15-FIC-1026. Ce régulateur de débit règle le débit de reflux renvoyé vers la colonne d'épuration 15-MD01 grâce à la pompe de reflux de la colonne d'épuration 15-MJ01/A.

Le débit de reflux vers la colonne d'épuration peut aussi être réglé par le 15-FIC-1066 et ce, en déviant le débit venant de l'aval des pompes 15-MJ01/A vers l'alimentation du bout froid du 15-MC05 venant de la tête du ballon de reflux 15-MD02. Ceci empêchera le recyclage du GPL vers le fractionnement.

L'opérateur peut aussi utiliser la commande manuelle 15-HIC-1251 pour envoyer le liquide du fond du ballon de reflux vers le collecteur liquide torche froide et utiliser la 15-HIC-1023 pour dégazer les produits de tête de ce ballon vers la torche froide, ceci pendant le démarrage seulement.

3.1.8. Pompes de reflux de la colonne d'épuration 15-MJ01/A

Il y a deux pompes de reflux, dont une en stand-by (de secours). Les pompes sont munies d'une ligne bypass à débit minimum et d'un régulateur de débit 15-FIC-1027 qui règle la vanne 15-FV-1027 afin de contrôler le débit de liquide renvoyé vers le ballon de reflux de la colonne d'épuration 15-MD02.

3.1.9. Paramètres d'exploitations normales

Le tableau suivant représente toutes les variables opératoires pour le système de la colonne d'épuration.

Tableau 01 : Variables opératoires pour le système de la colonne d'épuration

Equipement No.	Description	Valeur Typique	Unité
15-AT-1018	Analyseur des produits de tête du ballon de reflux de la colonne d'épuration:		
	He (Hélium)	0,20	% mole
	N ₂ (Azote)	6,53	% mole
	CO ₂ (Dioxyde de carbone)	0,01	% mole
	C ₁ (Methane)	86,64	% mole
	C ₂ (Ethane)	5,66	% mole
	C ₃ (Propane)	0,76	% mole
	IC ₄ (Isobutane)	0,10	% mole
	NC ₄ (Butane)	0,11	% mole
	IC ₅ (i-Pentane)	0,00	% mole
	NC ₅ (Pentane)	0,00	% mole
15-FIC-1026	Régulateur de débit de reflux de la colonne d'épuration 15-MD01	115 147	kg/h
15-FIC-1027	Régulateur de débit mini de la pompe de reflux de la colonne d'épuration	115 147	kg/h
15-FIC-1061	Régulateur de débit d'huile chaude du rebouilleur de la colonne d'épuration	478 350	kg/h
15-FIC-1066	Régulateur de débit de reflux vers le bout froid du 15-MC05	0	kg/h

Tableau 01 : Variables opératoires pour le système de la colonne d'épuration

Equipement No.	Description	Valeur Typique	Unité
15-LIC-1005	Régulateur de niveau du refroidisseur de propane HP 15-MC01	50	%
15-LIC-1007	Régulateur de niveau du refroidisseur de propane MP 15-MC02	50	%
15-LIC-1011	Régulateur de niveau de la colonne d'épuration 15-MD01	50	%
15-LIC-1020	Régulateur de niveau du refroidisseur de propane BP 15-MC03	50	%
15-LIC-1025	Régulateur de niveau du ballon de reflux 15-MD02	50	%
15-TIC-1039	Régulateur de température du ballon de reflux de la colonne d'épuration 15-MD02	-55	°C
15-TIC-1064	Régulateur de température de la colonne d'épuration 15-MD02	42	°C

3.1.10. Exploitations temporaires

Avant le démarrage du système déméthaniseur, le niveau des produits de fond de la colonne d'épuration 15-MD01 est contrôlé par le commutateur sélecteur du DCS 15-HS-1011 qui est positionné pour un contrôle de niveau direct avec le 15-LIC-1011, en déviant le débit des produits de fond.

Ces produits de fond sont déviés vers le collecteur de la torche froide à travers la vanne de contrôle 15-LV-1011 qui est placée en aval du 15-MC16.

Lorsque le compresseur d'alimentation 01-MJ01 est hors service, le GN est fourni par by-pass à travers le réchauffeur 01-MC02. Dans ces conditions, le débit devrait être de 62% (506482 kg/h) du débit normal, à 39,5 bars eff.

Paramètres de fonctionnement de la colonne d'épuration 15-MD01 en cas d'arrêt du compresseur 01-MJ01.

- Débit d'admission: 508 884 kg/h; 33, 39 bar eff; -35°C.
- Débit de sortie des produits de fond : 68 337 kg/h ; 33,59 bar eff ; 34°C.
- Débit de sortie des produits de tête : 492 720 kg/h ; 33,39 bar eff ; -39°C.
- Reflux: 47 902 kg/h ; 29, 09 bar eff ; -56°C.
- Tête du ballon de reflux de la colonne d'épuration 01-MD02 vers l'échangeur principal 15-MC05 : 456 122 kg/h ; 29,09 bar eff ; -56°C.

3.2. Démarrage

Ce chapitre décrit les étapes nécessaires pour la mise-en-service du système de la colonne d'épuration. Concernant ce chapitre du manuel opératoire, il est supposé que les conditions suivantes sont satisfaites :

- Les utilités et les systèmes suivants sont en service : décarbonatation, sécheurs, démercurisation, réfrigération propane, circuit MR et huile chaude.
- Le gaz d'alimentation est disponible.
- Les unités en aval sont prêtes pour recevoir le gaz.
- Le système est inerté, séché et prêt pour le démarrage.

Comme le système de la colonne d'épuration est déjà mis en service, la charge du système d'huile chaude de l'unité 58 va augmenter dès la mise en marche du 15-MC15.

La charge de l'unité 16 de la réfrigération de propane va augmenter dès la mise en marche des refroidisseurs du gaz d'alimentation 15-MC01, 15-MC02 et 15-MC03.

1. Confirmer que le système de réfrigération de propane est en service, qu'il est en recirculation totale, et que les niveaux de propane dans les refroidisseurs de propane HP/MP/BP 15-MC01/02/03 sont à une valeur normale. Confirmer que les régulateurs de niveau 15-LIC-1005, 15-LIC-1007 et 15-LIC-1020 sont placés en mode automatique et que les refroidisseurs doivent être prêts pour recevoir le gaz d'alimentation.
2. S'assurer que la vanne d'isolement du gaz d'alimentation 15-ESDV-1196 est fermée par le commutateur manuel 15-HS-1196.
3. S'assurer que la vanne d'isolement des produits de fond 15-ESDV-1013 de la colonne d'épuration 15-MD01 est fermé par le commutateur manuel 15-HS-1013.
4. Confirmer que le régulateur de débit d'huile chaude 15-FIC-1061 du rebouilleur de la colonne d'épuration 15-MC15 est en manuel et fermé.
5. Confirmer que la vanne d'isolement des produits de fond 15-ESDV-1546 du ballon de reflux de la colonne d'épuration 15-MD02 est fermée par le commutateur manuel 15-HS-1546.

6. S'assurer que l'échangeur principal 15-MC05 est isolé.
7. Utiliser le commutateur manuel 15-HS-1196 pour ouvrir la vanne d'isolement 15-ESDV-1196 afin de permettre au gaz d'alimentation de circuler lentement vers les refroidisseurs de propane HP/MP/BP 15-MC01/02/03 et vers la colonne d'épuration. Le GN sort de la tête de la colonne à travers le by-pass 15-TV-1039 et alimente le 15-MD02 puis en ressort à travers la 15-HV-1023 vers le collecteur de la torche froide.
8. Démarrer la mise-en-froid du 15-MC05 en suivant les instructions APCI.
9. Etablir le niveau dans la colonne d'épuration. Selon l'alimentation, ce liquide peut ne pas être disponible jusqu'à ce que le bout chaud du 15-MC05 soit refroidi et en service. Il sera peut être nécessaire d'injecter du GPL.
10. Disposer l'huile chaude vers la 15-FV-1061, ouvrir la vanne en contrôle manuel et admettre progressivement l'huile chaude sur le côté tube du rebouilleur de la colonne d'épuration 15-MC15. Placer le régulateur de débit 15-FIC-1061 en automatique et augmenter graduellement la température "sortie" du rebouilleur. Pendant que la température "sortie" du rebouilleur augmente, placer la cascade 15-TIC-1064/ 15-FIC-1061 en automatique.
11. Dès que le niveau dans la colonne d'épuration est suffisant, disposer les produits de fond de la colonne d'épuration vers le collecteur de la torche froide à travers la vanne 15-ESDV-1013. Mettre le régulateur de niveau 15-LIC-1011 des produits de fond de la colonne d'épuration en automatique et sélectionner le 15-HS-1011 pour le contrôle de niveau direct.
12. Avec le commutateur 15-HS-1277, ouvrir la vanne 15-ESDV-1277 pour assurer une circulation de la tête de la colonne d'épuration vers le bout chaud du 15-MC05.
13. Contrôler le niveau dans le 15-MD02 et quand un niveau minimum est établi, ouvrir la vanne 15-ESDV-1546. Démarrer la pompe 15-MJ01/A avec un débit minimum de recirculation sous contrôle du régulateur de débit au refoulement 15-FIC-1027.
14. Etablir le débit de reflux vers la colonne d'épuration avec la cascade 15-LIC-1025/ 15-FIC-1026.
15. Placer le régulateur de température 15-TIC-1039 en automatique avec un point de consigne de -53°C.
16. Contrôler les températures, les niveaux et les pressions des équipements jusqu'à ce que le débit d'alimentation soit entièrement établi à 840 000 kg/h.

3.3. L'approche proposée

La méthode proposée dans ce travail est basée sur l'utilisation d'une architecture à base d'agents pour modéliser le système de déshydratation du gaz naturel. Pour modéliser le système de déshydratation du gaz naturel.

L'architecture définie dans ce travail est basée sur des systèmes multi-agents (MAS), conçus pour simuler le fonctionnement de différents systèmes. Multi-agents (MAS), conçue pour simuler le fonctionnement des différents sècheurs qui composent le système de déshydratation. Sècheurs qui composent le processus de déshydratation.

L'architecture exprime la structure fondamentale du système de déshydratation à modéliser et à simuler. Elle définit l'ensemble de composants fonctionnels décrits en termes de comportements et d'interfaces, ainsi que la manière dont ces composants sont décrits. Comportements et interfaces, ainsi que la façon dont ces composants interagissent pour l'objectif de simulation d'un système de déshydratation du gaz naturel. De gaz naturel. Par conséquent, une description architecturale est principalement requise pour la spécification de la structure du système.

L'idée est de modéliser le système de déshydratation du gaz naturel comme un système multi-agent distribué. Système multi-agents distribué.

D'un point de vue organisationnel, le processus de déshydratation est un ensemble d'éléments passifs dans le sens où ces éléments forment la partie opérative du système (électrovannes, pompes, réservoirs, turbocompresseurs, chaudières, ventilateurs,...), ainsi que d'éléments actifs dans le sens où ces éléments forment la partie contrôle (régulateurs, contrôleurs, serveurs,...), ainsi que des éléments d'interaction et de communication (tuyaux, fils, signaux de commande, signaux de contrôle, alarmes, synoptiques,...) et autres éléments de mémorisation, de stockage, et de traitement de l'information

D'un point de vue conceptuel, tous les éléments du système de déshydratation du gaz naturel sont modélisés. déshydratation du gaz naturel sont modélisés comme un MAS. Chaque élément actif élément actif est représenté par un agent autonome.

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

3.4. Structure des agents du système de déshydratation

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

- **Agent 15-AT-1018** : Analyseur des produits de tête du ballon de reflux de la colonne d'épuration.
- **Agent 15-FIC-1026** : Régulateur de débit de reflux de la colonne d'épuration 15-MD01.

- **Agent 15-FIC-1027** : Régulateur de débit mini de la pompe de reflux de la colonne d'épuration.
- **Agent 15-FIC-1061** : Régulateur de débit d'huile chaude du rebouilleur de la colonne d'épuration.
- **Agent 15-FIC-1066** : Régulateur de débit de reflux vers le bout froid du 15-MC05.
- **Agent 15-LIC-1005** : Régulateur de niveau du refroidisseur de propane HP 15-MC01.
- **Agent 15-LIC-1007** : Régulateur de niveau du refroidisseur de propane MP 15-MC02.
- **Agent 15-LIC-1011** : Régulateur de niveau de la colonne d'épuration 15-MD01.
- **Agent 15-LIC-1020** : Régulateur de niveau du refroidisseur de propane BP 15-MC03.
- **Agent 15-LIC-1025** : Régulateur de niveau du ballon de reflux 15-MD02.
- **Agent 15-TIC-1039** : Régulateur de température du ballon de reflux de la colonne d'épuration 15-MD02.
- **Agent 15-TIC-1064** : Régulateur de température de la colonne d'épuration 15-MD02.

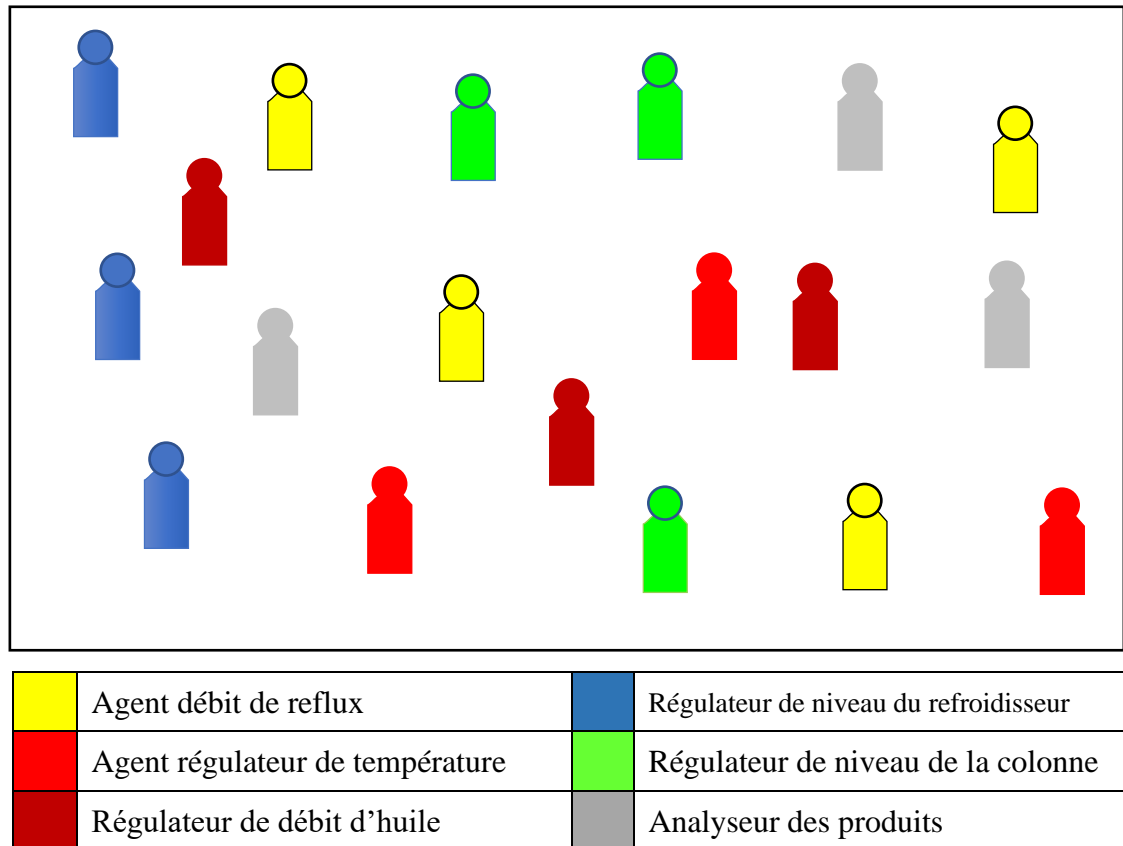


Figure 3.4: structure organisationnelle du systeme

4. Conclusion:

Ce chapitre a traité le Colonne d'épuration de gaz, nous avons donné une idée sur le fonctionnement d'une chaudière. Ceci représente la première étape dans la démarche de réalisation de notre simulateur. Le prochain chapitre concerne la présentation du modèle Multi-Agents que nous avons adopté pour modéliser notre système.

CHAPITRE 4 :

Implémentation

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 Netbeans : 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 :



Figure 4.1. logo de java.

Java est à la fois un langage de programmation informatique orienté objet et un environnement d'exécution informatique portable créé par James Gosling et Patrick Naughton employés de Sun Microsystems avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld. [29]

Il est caractérisé par sa portabilité, multiplateformes, orienté objet ainsi que la richesse de sa bibliothèque.

3.2 JDK : (Java Development Kit)²⁷

Le Java Development Kit, communément appelé JDK, est le kit de développement de base que propose gratuitement la société Oracle. Le Kit de développement 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 NetBeans : ²⁸



Figure 4.2. logo de netbeans.

NetBeans est un projet open source ayant un succès et une base d'utilisateur très large. Sun Microsystems a fondé le projet open source NetBeans en Juin 2000 et continue d'être le sponsor principal du projet.

Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS.

Aujourd'hui, deux projets existent: L'EDI NetBeans et la Plateforme NetBeans.

L'EDI NetBeans (Environnement de Développement Intégré) : est un environnement de développement - un outil pour les programmeurs pour écrire, compiler, déboguer et déployer des programmes. Il est écrit en Java - mais peut supporter n'importe quel langage de programmation. Il y a également un grand nombre de modules pour étendre l'EDI NetBeans.

La Plateforme NetBeans : c'est également une plateforme. Il vous est possible de créer votre propre application Awt ou Swing, basée sur la plateforme NetBeans.

3.4 La plateforme Jade :²⁹

3.4.1 Définition :

Jade est un logiciel libre distribué par TILab en Open Source avec une licence LGPL. Jade a pour but de simplifier le développement des systèmes multi-agents tout en fournissant un ensemble complet de services et d'agents conformes aux spécifications FIPA.



Figure 4.3. logo de jade.

3.4.2 L'architecture de la plate-forme jade :

L'architecture de JADE est basée sur la coexistence de plusieurs Machines Virtuelles (VM) Java et la communication se fait par la méthode RMI (Remote Method Invocation) de Java. Jade inclut tous les composants obligatoires qui contrôlent un Système Multi-Agents (SMA). Ces composants sont :

- AMS (Agent Management System) qui fournit le service de nommage (pour assurer par exemple que chaque agent possède un identifiant unique dans la plateforme) et qui représente l'autorité de la plateforme (par exemple il est possible de créer/arrêter des agents en envoyant des requêtes à l'AMS)
- DF (Directory Facilitator) qui fournit un système de pages jaunes qui permet aux agents de retrouver les agents fournisseurs de services.
- ACC (Agent Container Controller) qui est l'environnement dans lequel les agents peuvent vivre.

Toute la communication entre agents est exécutée par messages ACL (Agent Communication Language). Pour chaque agent, JADE crée et contrôle une file d'attente des messages entrants.

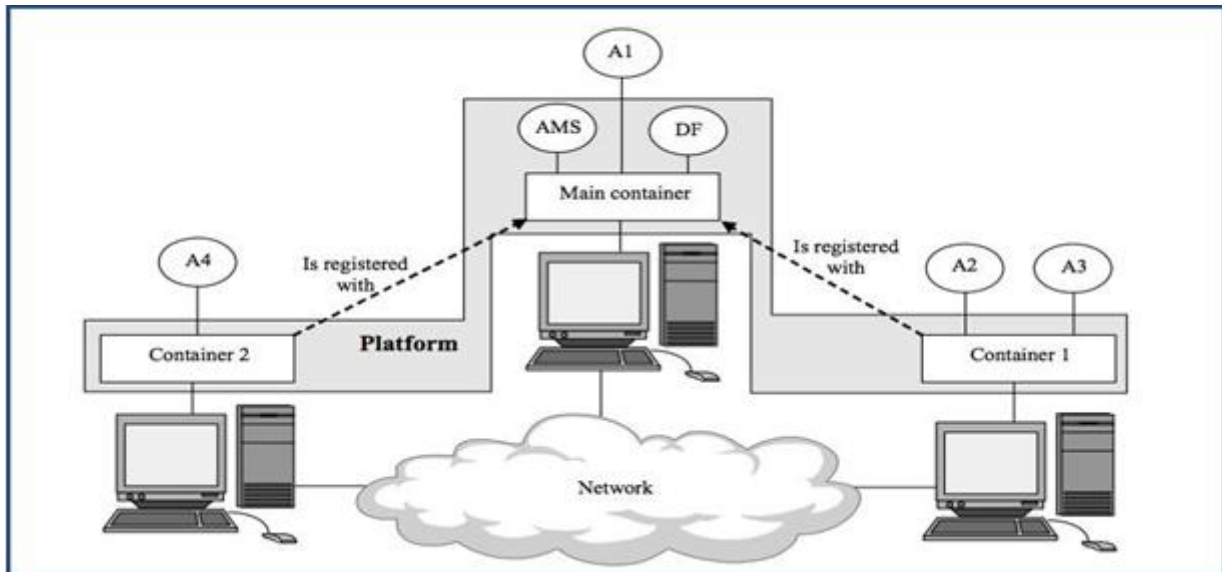


Figure 4.3. Architecture de la plateforme jade.

La plate-forme d'agents peut être répartie sur plusieurs serveurs. Une seule application Java, et donc une seule machine virtuelle de Java (JVM), est exécutée sur chaque serveur. Chaque JVM est un conteneur d'agents qui fournit un environnement complet pour l'exécution d'agents et permet à plusieurs agents de s'exécuter en parallèle sur le même serveur. Cet environnement profite de la technologie multi-thread offerte par Java. En effet chaque agent sera exécuté dans un thread différent. Tous les comportements (behaviors) qui seront associés à un agent en particulier seront exécutés dans le même thread à moins de le spécifier explicitement.

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 plate-forme 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 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.

3.5 Intégration de la plate-forme JADE dans Netbeans :

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

1. Démarrer NetBeans et à partir du menu Outils, sélectionner Ant Library comme suite :

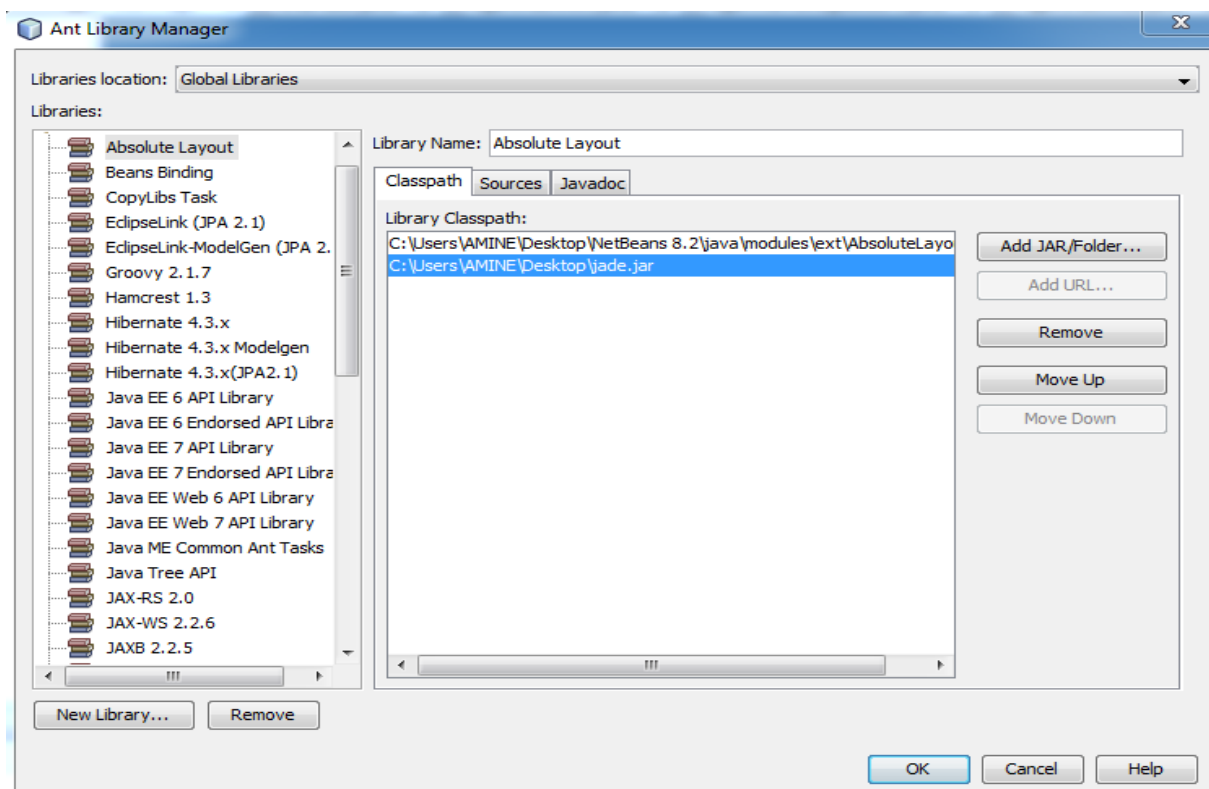


Figure 4.4. Fenêtre pour ajouter une Library.

2. Cliquez sur le bouton new library dans la boîte de Ant Library et saisissez le nom de library par exemple JADE et validez en appuyant sur Ok.

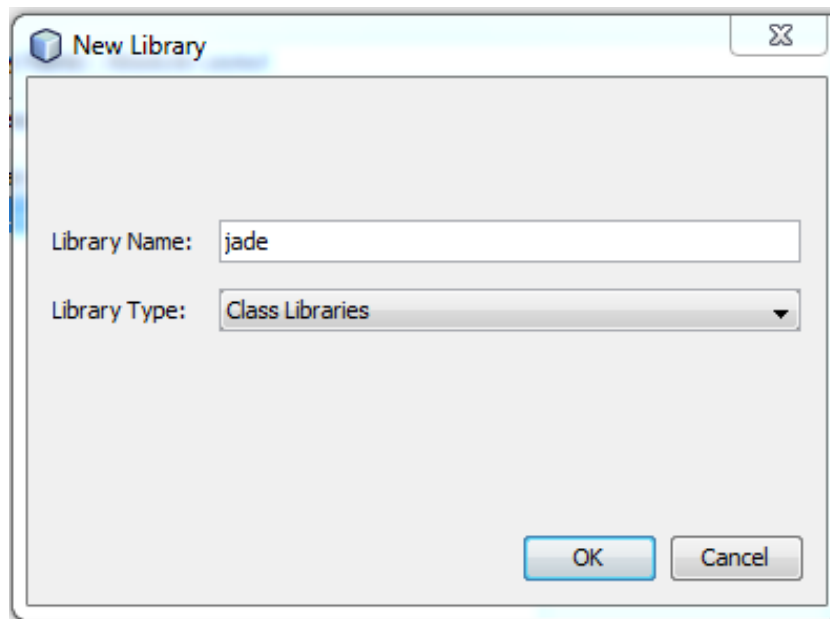


Figure 4.5. Boîte de saisie du nom de Library.

3. Cliquez sur le bouton Add JAR/Folder pour parcourir les jar de plate-forme jade dans le fichier lib c'est jade.jar et cliquez sur le bouton Add JAR/Folder, ensuite validez en appuyant sur OK.

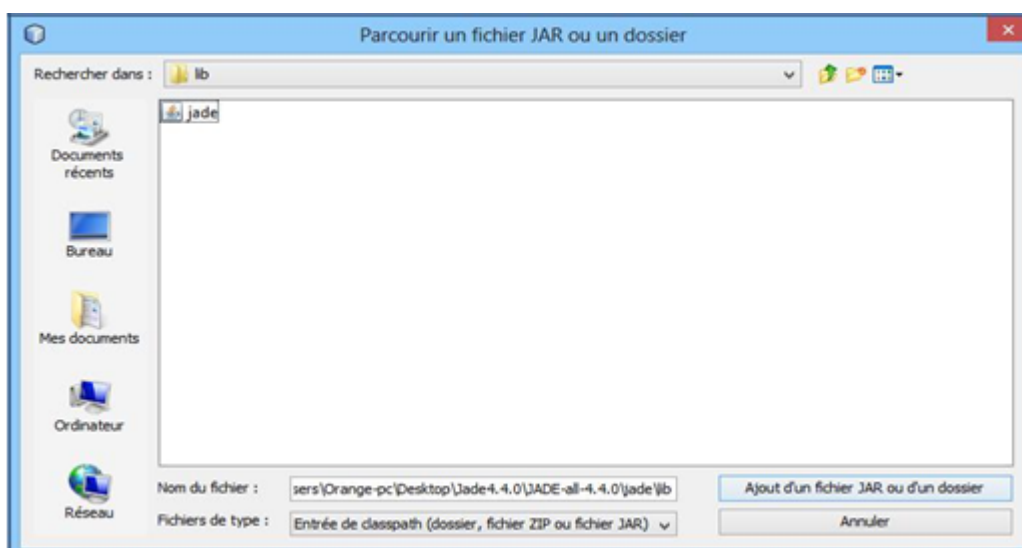



Figure 4.6. Fenêtre de parcourir le fichier JAR.

4. Si on a besoin de bibliothèques de la plate-forme jade on clique avec le bouton droit de la souris  sur dans le projet puis on sélectionne le libraire puis clique sur Ok.

4. Implémentation du système :

Pour illustrer la simulation multi-agents nous allons suivre un exemple pour fournir des captures d'écran.

3.1 Présentation d'interface graphique :

Le simulateur a été développé en utilisant la plateforme JADE afin de bénéficier de la puissance des outils avancés offerts par cette plateforme et pour les raisons suivantes :

- Jade gère la communication entre les agents et offre des interfaces de gestion des agents.
- Le mode de communication direct (avec messages) est supporté par la plate-forme et un large éventail de bibliothèques.
- La plate-forme Jade répond à plusieurs fonctionnalités et offre un large éventail de bibliothèques.

L'interface principale de notre simulateur est présentée dans la figure 4.7, Elle représente la simulation du système de déshydratation et permet à l'utilisateur de piloter le fonctionnement du système et de l'utiliser.

l'utilisateur de piloter le fonctionnement du système et de l'alimenter en chargeant différentes entrées de données.

Les interfaces graphiques du simulateur fournissent de nombreux outils permettant à l'opérateur de réguler et de surveiller les changements dans le système de déshydratation.

Grâce à l'interface de connaissance affichée, l'utilisateur observe les étapes du système et peut agir sur les parties d'entrée du processus en saisissant les valeurs des seuils de fonctionnement et observer le comportement du simulateur en conséquence.

Tel qu'il est prévu, le simulateur offre une gamme particulière de fonctionnalités permettant à l'utilisateur de suivre les étapes du système. De fonctionnalités permettant à l'utilisateur de suivre l'évolution du système de déshydratation et d'être prêt, ainsi, à connaître son fonctionnement.

De même, une telle machine permet de prévoir de nombreux cas de figure.

cas : ceux qui sont acceptables et ceux qui sont interdits. qui sont interdits. D'autres options seront encore en fonction des besoins de l'opérateur.

Exécution graphique :

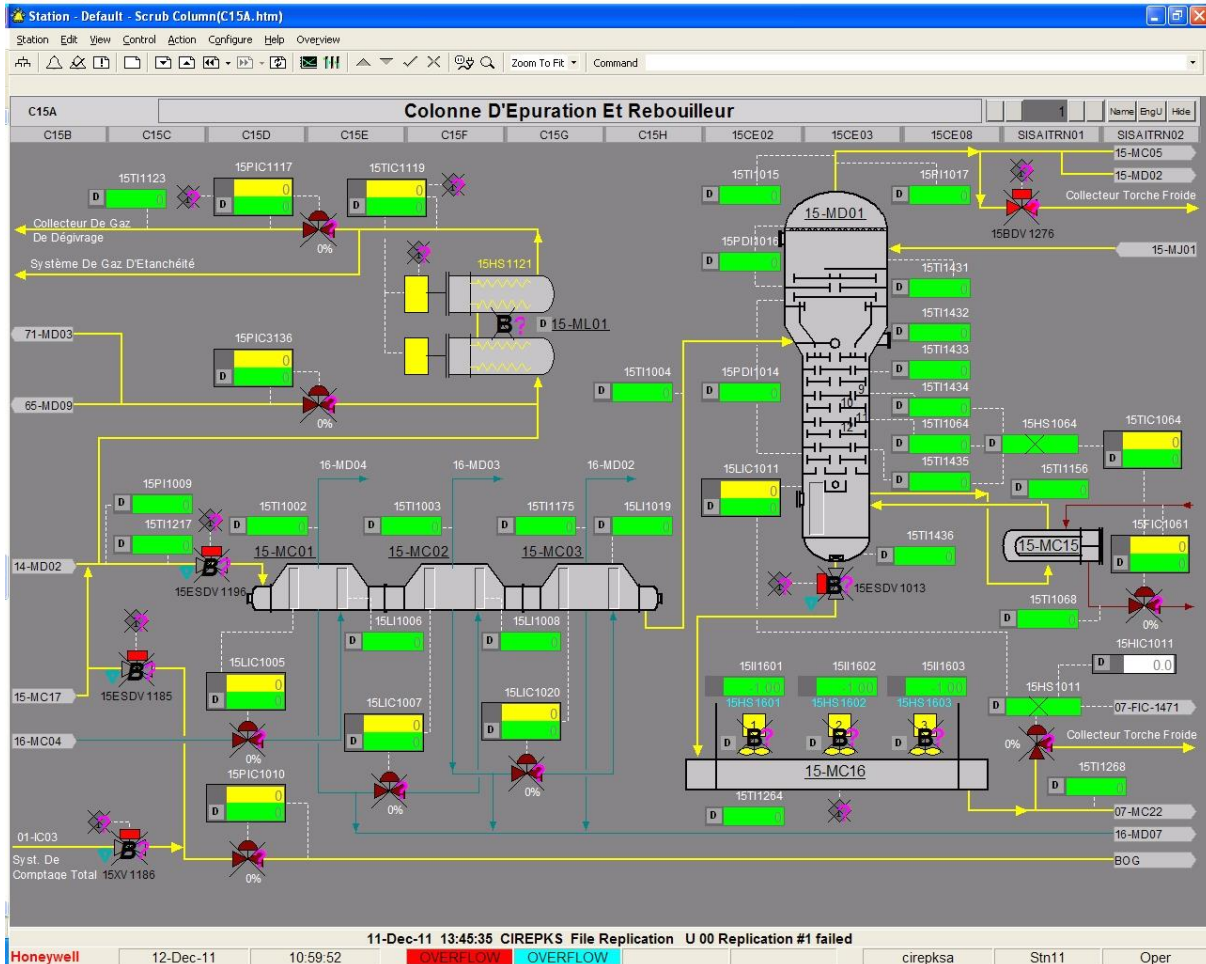


Figure 4.7. Interface de la simulation

5.Conclusion :

Nous avons utilisé la plateforme jade afin de créer et d'implémenter notre système de déshydratation., 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

Les systèmes multi-agents normatifs présentent un domaine de recherche très vaste qui permet d'intégrer les normes et les lois dans les systèmes multi-agents. Cette intégration offre plusieurs avantages tels que la rationalité et la flexibilité. Dans ce travail, nous avons proposé une approche pour l'implémentation de coordination pour les systèmes multi-agents normatifs.

En fait, ce travail est une extension du plateforme JADE proposé pour la programmation des systèmes multi-agents normatifs pour supporter les aspects collectifs de ces systèmes. En conséquence, notre approche suit le même principe de JADE : description des normes en utilisant une ontologie et une manipulation de cette dernière à l'aide des extensions écrites sous forme d'aspects. Cette approche offre plusieurs avantages. D'une part, elle est simple et réutilisable. En fait, il suffit de réutiliser le code des aspects avec un développement d'une ontologie afin d'utiliser notre approche. En plus, elle est extensible. Il est possible d'étendre cette approche par des mécanismes plus sophistiqués. Finalement, cette approche augmente la flexibilité des agents parce que les interactions seront une partie indépendante des codes des agents.

Notre approche est validée en utilisant l'exemple de système de gestion de conférences. Particulièrement, nous avons ciblé la partie de distribution des papiers de ce système (dispatching). Ce système est exécuté en utilisant trois scénarios qui montrent les avantages de notre approche.

Bien entendu, cette approche reste ouverte sur plusieurs extensions. D'une part, il faut étendre cette approche pour supporter les différents mécanismes de coordination qui ne sont pas implémentés en Jade. Pour atteindre cet objectif, il faut tout d'abord proposer un langage pour la description des protocoles de coordination en utilisant les ontologies. D'autre part, nous croyons que l'intégration des normes constitutives dans cette étape est importante afin de permettre une mise-à-jour automatique et au cours d'exécution de différents protocoles

Bibliographies

- 1 - Aziz-Alaoui, M., & Bertelle, C. (Eds.). (2009). From system complexity to emergent properties. Springer Science & Business Media. Knowledge Management, Organizational Intelligence and Learning, and Complexity- Vol.
- 2 - Aziz-Alaoui, Modélisation et Analyse des Systèmes Complexes, support de cours Master 2 Mathématiques & Applications, Université du Havre, 2018.
- 3 - Bar-Yam, Y. (2002). General features of complex systems. Encyclopedia of Life Support Systems (EOLSS), UNESCO, EOLSS Publishers, Oxford, UK, 1.
- 4 - Aziz-Alaoui, Modélisation et Analyse des Systèmes Complexes, support de cours Master 2 Mathématiques & Applications, Université du Havre, 2018.
- 5 - Déchets plastiques dans les océans : "La deuxième plus grande catastrophe écologique dans le monde" estime le navigateur Yvan Bourgnon » [archive], sur Franceinfo, 5 mars 2019 (consulté le 18 mars 2019)
- 6 - P. Hubert, Le Glossaire International d'Hydrologie, novembre, 1996.
- 7 - J.Ferber, Les systèmes multi-agents : vers une intelligence collective, InterEditions, ISBN : 2- 72-96-0572-X, 1995.
- 8 - P.A. Fishwick, Simulation Model Design and Execution. Digital Worlds, Prentice Hall, 1995.
- 9 - A. Pavé, Modélisation en biologie et en écologie, Aléas, 1994.
- 10 - G. Quesnel, Approche formelle et opérationnelle de la multi-modélisation et de la simulation des systèmes complexes, Laboratoire d'Informatique du Littoral, 2006.
- 11- [https://fr.theastrologypage.com/object-oriented-modeling#:~:text=La%20mod%C3%A9lisation%20orient%C3%A9e%20objet%20\(MOO\)%20est%20la%20construction%20d',objet%20sont%20uniquement%20des%20objets.](https://fr.theastrologypage.com/object-oriented-modeling#:~:text=La%20mod%C3%A9lisation%20orient%C3%A9e%20objet%20(MOO)%20est%20la%20construction%20d',objet%20sont%20uniquement%20des%20objets.)
- 12 - S.FOURNIER, «Intégration de la dimension spatiale au sein d'un modèle multi-agents à base de rôles pour la simulation : Application à la navigation maritime,» pour obtenir le grade de Docteur, Université de Rennes 1, 2005.
- 13 - A.KUMAR&all, «Simulators for Wireless Networks : A Comparative Study,» International Conference on Computing Sciences, 2012.
- 14 - F.MICHEL, «Formalisme, outils et éléments méthodologiques pour la modélisation et la simulation multi-agents,» Thèse pour obtenir le diplôme de doctorat, Université Montpellier II, 2004.

- 15 - Michael Schäfer, « Methodology for Data Validation 1.0 (Handbook), revised edition June 2016 » [archive], sur CROS - European Commission, 7 juin 2016 (consulté le 22 février 2020)
- 16 - Mahesh Chand et David Talbot, « Data Components in Visual Studio .NET », dans Applied ADO.NET: Building Data-Driven Solutions, Apress, 2003 (ISBN 978-1-59059-073-7, lire en ligne [archive]), p. 33–71
- 17 - Drogoul, A. (1993). De la Simulation Multi-agents à la Résolution Collective de Problèmes, Thèse de doctorat, Université Paris VI.
- 18 - Ferber, J. (1995). Les Systèmes Multi-Agents : vers une intelligence collective. InterEditions.
- 19 - M. Wooldridge : Introduction to MultiAgent Systems. John Wiley and Sons, 2002. p. 18
- 20 - J. Ferber : Les systèmes multi-agents - Vers une intelligence collective. InterEditions, 1995.
- 21 - Chaib-draa et I. Jarras, «Aperçu sur les systèmes multi agents»,Série Scientifique, Montréal, 2002.
- 22 - Boissier, O., Gitton, S., and Glize, P. (2004). Caractéristiques des systèmes et des applications. In des Techniques Avancées, O. F., editor, Systèmes Multi-Agents (Série ARAGO), volume 29, pages 25–54.
- 23 - Fabio bellifemine, Giovanni Caire, Dominic Greenwood, Developing Multi agent systems with JADE. Willey series in agent technologie 2007.
- 24 - B. Azvine, N. Azarmi , and D. Nauck, (2000), intelligent systems and soft computing: prospects, tools and applications, volume 1804 of LNCS. Springer- Verlag.
- 25 - Plate-forme MADKIT, 2003 <http://www.madkit.org/>.
- 26 N. Minar The swarm simulation system: a toolkit for building multi-agent simulations,
- 27 - <https://www.oracle.com/java/technologies/downloads/>
- 28 - <https://www.oracle.com/tools/technologies/netbeans-ide.html>
- 29 - Jade.tilab.com