



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTERE DE L'ENSEIGNEMENT SUPERIEURE ET DE LA RECHERCHE  
SCIENTIFIQUE

**UNIVERSITE IBN KHALDOUN - TIARET**

# MEMOIRE

Présenté à :

**FACULTÉ MATHÉMATIQUES ET INFORMATIQUE  
DÉPARTEMENT D'INFORMATIQUE**

Pour l'obtention du diplôme de :

**MASTER**

Spécialité : Génie Informatique

Par :

**BELFIRAR Hassina  
ADDA Fatiha**

Sur le thème

---

**Approche multi-agents pour l'ordonnancement des tâches  
Dans un Workflow distribué**

---

Soutenu publiquement le / / 2019 à Tiaret devant le jury composé de :

Mr HATTB N  
Mr GOISMI Mohamed  
Mr DAOUD M. A

Président  
Encadreur  
Examineur



*Dédicace*

*Je dédie ce modeste travail à :*

*A mes chers **parents***

*Qui ont la cause de la lumière de mes  
jours, la source de mes forces, la flamme  
de mon cœur, ma vie et mon bonheur.  
Que dieu leur procure bonne santé et  
longue vie.*

*A mes frères et chers sœurs bien  
sûr.*

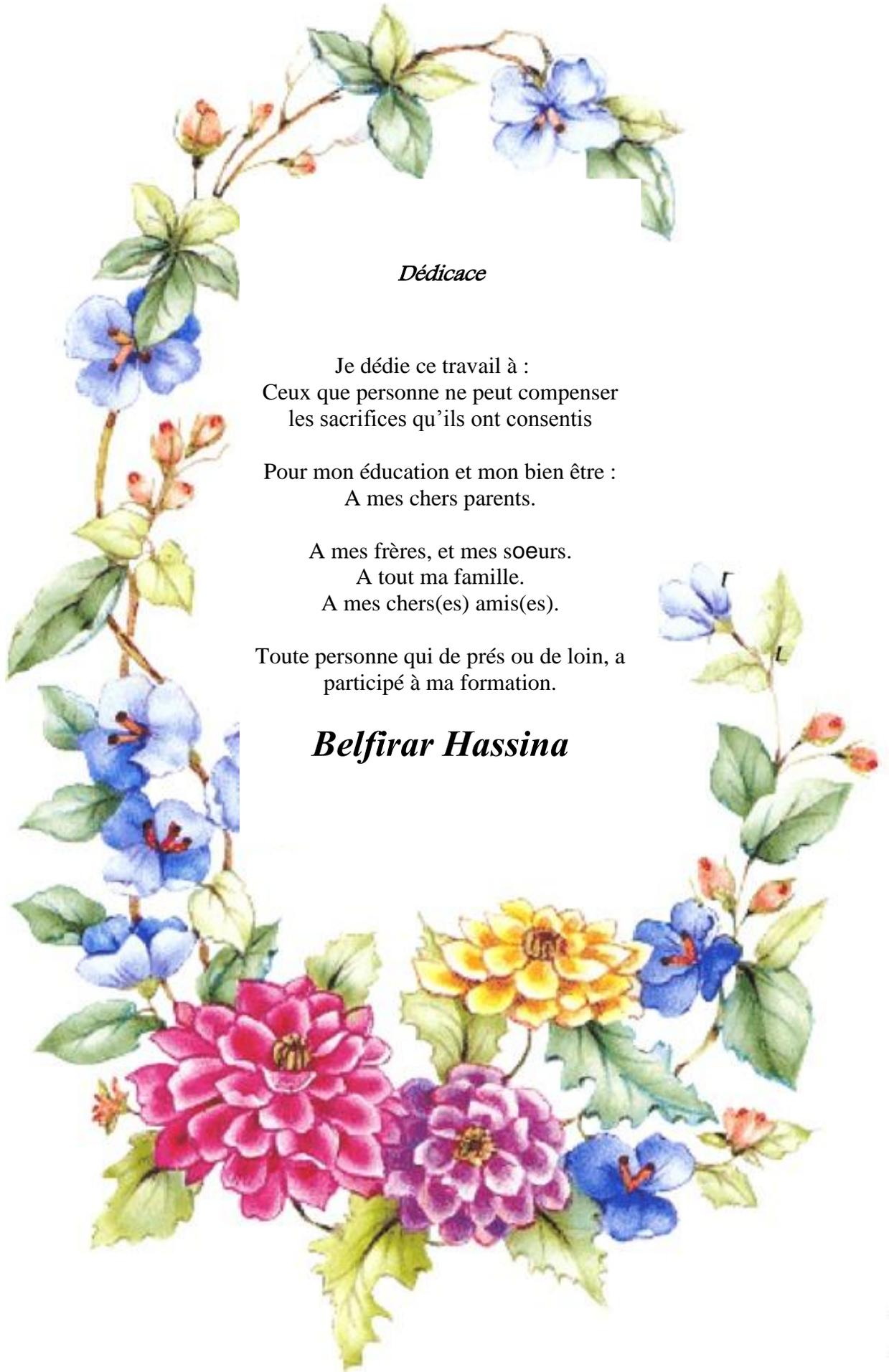
*Sans oublié toute ma famille **ADDA**. Et  
**abdaAllah**,*

*A mon binôme **H.Belfirar***

*A tous mes Amis et spécialement à  
(*nassira, saliha, djamila, theldja, fatdhila  
karima*).*

*A tous ceux que j'aime et qui m'aiment.*

***Adda fatiha***



*Dédicace*

Je dédie ce travail à :  
Ceux que personne ne peut compenser  
les sacrifices qu'ils ont consentis

Pour mon éducation et mon bien être :  
A mes chers parents.

A mes frères, et mes sœurs.  
A tout ma famille.  
A mes chers(es) amis(es).

Toute personne qui de près ou de loin, a  
participé à ma formation.

***Belfirar Hassina***

## **REMERCIEMENTS**

*Nous tenons tout d'abord à remercier Dieu le tout-puissant et Le Miséricordieux, qu'il est nous donné la vie et la force ainsi que le courage et l'audace pour dépasser toutes les difficultés.*

*En second lieux Nous tenons Un grand merci à nos parents pour leurs conseils ainsi que leur soutien inconditionnel, à la fois moral et économique, qui nous permis de réaliser les études que nous voulions de ce mémoire.*

*Nous exprimons nos remerciements à mon encadreur Mr. GOISMI Mohamed et tous les enseignants qui ont participé à notre formation depuis le début de nos études universitaires jusqu'au jour d'aujourd'hui. Nous remercions particulièrement les membres du Jury qui nous font honneur en acceptant d'examiner et de juger notre travail.*

*Enfin nous voudrions exprimer nos remerciements envers les amis et collègues qui nous ont apporté leur soutien moral et intellectuel.*

## Résumé

Depuis quelques années, les entreprises sont entrées dans l'ère de la collaboration informatisée et pour rester compétitives, elles doivent constamment améliorer leur efficacité opérationnelle. L'accessibilité à un volume croissant d'informations et l'intégration de solutions logicielles variées créent de nouvelles exigences par rapport aux outils de gestion de la collaboration. Le workflow est un concept permettant de modéliser et de gérer des ces procédures industrielles ou administratives, impliquant plusieurs acteurs appelés agents situant d'un environnement distribué nommé système multi-agents. L'objectif de ce mémoire est de développer une application workflow coopératif et administratif à l'aide d'un système multi-agent pour un ordonnancement distribué dans un workflow traitant la procédure des demandes de congé dans une entreprise.

**Mots clés** : Ordonnancement, workflow, gestion des congés, Systèmes multi-agents, AUML, JADE.

## Abstract

Since a few years, companies have entered the era of computerized collaboration and to remain competitive, they must constantly improve their operational efficiency, Accessibility to a growing volume of information and the integration of various software solutions create new requirements for collaboration management tools, The workflow is a concept for modeling and managing these industrial or administrative procedures, involving several actors called agents locating a distributed environment called multi-agent system, The purpose of this thesis is to develop a cooperative and administrative workflow application using a multi-agent system for a distributed scheduling in a workflow processing the procedure of requests for leave in a company.

**Keywords**: Scheduling, Workflow, Leave Management, Multi-Agent Systems, AUML JADE.

## ملخص

منذ بضعة سنوات دخلت الشركات عصر التعاون المحوسب و لكي تبقى قادرة على المنافسة يجب أن تحسن باستمرار كفاءتها التشغيلية. إن الوصول إلى حجم متزايد من المعلومات وتكامل حلول البرمجيات المختلفة يخلق متطلبات جديدة لأدوات إدارة التعاون. سير العمل هو مفهوم لنمذجة و إدارة هذه الإجراءات الصناعية أو الإدارية، التي تنطوي على العديد من الجهات الفاعلة تسمى نظام متعدد الوكلاء . الغرض من هذه الأطروحة هو تطوير تطبيق سير عمل تعاوني وإداري باستخدام نظام متعدد الوكلاء لجدولة موزعة في سير عمل معالجة إجراء طلبات الإجازة في الشركة.

## Table des matière

<b>Introduction générale .....</b>	<b>11</b>
<b>Chapitre I : WORKFLOW</b>	
I.1. Introduction .....	13
I.2. Historique de workflow .....	13
I.3. Concepts clés .....	13
I.3.1. <i>PROCESSUS METIER</i> .....	13
I.3.2. <i>MODELE DE PROCESSUS</i> .....	13
I.3.3. <i>TACHE</i> .....	13
I.3.4. <i>NŒUD</i> .....	14
I.3.5. <i>TRANSITIONS</i> .....	14
I.3.6. <i>EVENEMENT</i> .....	14
I.4. Définitions de workflow .....	14
I.4.1. <i>CYCLE DE VIE D'UN WORKFLOW</i> .....	15
I.4.2. <i>SYSTEME DE GESTION DE WORKFLOW</i> .....	16
I.4.3. <i>LE MODELE DE REFERENCE POUR LE WORKFLOW</i> .....	16
I.4.4. <i>TYPES DE WORKFLOW</i> .....	17
I.4.4.1. Workflow d'administration .....	17
I.4.4.2. Workflow de production.....	18
I.4.4.3. Workflow de collaboration ou coopératif .....	18
I.4.4.4. Workflow ad-hoc.....	19
I.4.5. <i>CONCEPTS DE BASE DE WF</i> .....	19
I.4.6. <i>LES TYPOLOGIES DES APPLICATIONS DE WORKFLOW</i> .....	20
I.4.6.1. Une typologie technique .....	20
I.4.7. <i>REGROUPEMENT DES WORKFLOWS</i> .....	21
I.4.8. <i>MODELISATION DES WORKFLOW</i> .....	22
I.4.8.1. L'aspect fonctionnel .....	22
I.4.8.2. L'aspect comportemental .....	22
I.4.8.3. L'aspect informationnel (donnée).....	22
I.4.8.4. L'aspect organisationnel.....	22
I.5. Les Domaines de l'entreprise intéressés par une application de WorKflow .....	22
I.6. Les Grandes Fonctions d'une Application WorKflow .....	23
I.7. Impacts du workflow .....	23
I.7.1. <i>GAINS TANGIBLES</i> .....	23
I.7.2. <i>GAINS INTANGIBLES</i> .....	24
I.8. Les Avantages et les inconvénients de WorKflow .....	24
I.8.1. <i>LES AVANTAGES</i> .....	24
I.8.2. <i>LES INCONVENIENTS</i> .....	25
I.9. Conclusion.....	25
<b>Chapitre II : Généralités sur l'ordonnancement</b>	
II.1. Introduction .....	27
II.2. Définitions.....	27
II.3. Les domaines concernés de l'ordonnancement.....	27
II.4. L'objectif de l'ordonnancement.....	28
II.5. Les notions fondamentales de l'ordonnancement.....	28

II.6. Les étapes de l'ordonnancement .....	<b>29</b>
II.7. Méthodes de résolution des problèmes d'ordonnancement .....	<b>29</b>
II.7.1. <i>MODELES DE PLANIFICATION DE TACHES D'UN PROJET</i> .....	29
II.7.1.1. La méthode PERT .....	29
II.7.1.2. Le Diagramme de Gantt .....	32
II.7.2. <i>LES APPROCHES CLASSIQUES (CENTRALISEES)</i> .....	34
II.7.2.1. Les méthodes de résolution exactes .....	34
II.7.2.2. Les algorithmes approchés .....	35
II.7.3. <i>LES APPROCHES ROBUSTES</i> .....	38
II.7.4. <i>ORDONNANCEMENT DISTRIBUE</i> .....	39
II.7.4.1 Mécanismes de coopération .....	39
II.7.4.2 Systèmes d'ordonnancement distribués .....	40
II.8. Ordonnancement à l'aide de systèmes multi-agents .....	<b>41</b>
II.9. Conclusion.....	<b>41</b>
<b>Chapitre III : Les Systèmes Multi-Agents</b>	
III.1 Introduction.....	<b>43</b>
III.2. <i>POSITIONNEMENT HISTORIQUE</i> .....	43
III.3. Qu'est-ce qu'un agent ? .....	<b>43</b>
III.3.1. <i>L'ARCHITECTURE D'AGENT</i> .....	46
III.3.2. <i>UNE COMPARAISON AVEC LES OBJETS</i> .....	46
III.4. Qu'est-ce qu'un système multi-agents (SMA) ?.....	<b>48</b>
III.4.1 <i>LES SYSTEMES MULTI-AGENTS</i> .....	48
III.4.2 <i>DEFINITION: SYSTEME MULTI-AGENTS (SMA)</i> .....	48
III.4.3. <i>LES INTERACTIONS ET LES AGENTS</i> .....	49
III.4.3.1. La communication .....	50
III.4.4. <i>LES LANGAGES DE COMMUNICATION</i> .....	51
III.4.4.1. La théorie des actes de langage .....	51
III.4.5. <i>SEMANTIQUE DES MESSAGES</i> .....	54
III.4.6. <i>LA COOPERATION</i> .....	54
III.5. Outils & plateformes de développement des SMA .....	<b>55</b>
III.5.1. <i>LA NORME FIPA POUR LES SYSTEMES MULTI-AGENTS</i> .....	55
III.6. domaine d'application.....	<b>56</b>
III.7. Conclusion .....	<b>58</b>
<b>Chapitre IV : Implémentation</b>	
IV.1. Introduction .....	<b>60</b>
IV.2. Type d'agents développés .....	<b>60</b>
IV.3. Pourquoi la plateforme JADE ?.....	<b>61</b>
IV.3.1. <i>JADE</i> .....	61
IV.3.2. <i>UNE PLATEFORME JADE</i> .....	61
IV.4. modèle de workflow .....	<b>62</b>
IV.6. Présentation de l'application .....	<b>65</b>
IV.6. Conclusion .....	<b>73</b>
<b>Conclusion générale</b> .....	<b>74</b>

<b>PERSPECTIVE .....</b>	<b>75</b>
<b>Références bibliographiques.....</b>	<b>76</b>

## Liste des figures

<b>Figure I. 1:</b> un événement des nœuds.....	14
<b>Figure I. 2:</b> Cycle de vie d'un workflow.....	15
<b>Figure I. 3:</b> Le modèle de référence du workflow.....	16
<b>Figure I. 4:</b> Typologie fonctionnelle des applications de workflow.....	21
<b>Figure II. 1:</b> les domaines de l'ordonnancement.....	27
<b>Figure II. 2:</b> les composants d'une tâche.....	30
<b>Figure II. 3:</b> Les types des tâches en PERT.....	31
<b>Figure II. 4:</b> le diagramme de GANTT.....	33
<b>Figure III. 1:</b> Différents types d'agents.....	45
<b>Figure III. 2:</b> L'architecteur minimale d'un agent.....	46
<b>Figure III. 3:</b> La structure d'un message KQML.....	52
<b>Figure III. 4:</b> La structure d'un message FIPA ACL.....	53
<b>Figure III. 5:</b> Le modèle de référence pour une plate-forme multi-agents FIPA.....	55
<b>Figure IV. 1:</b> Architecture distribuée de l'application.....	60
<b>Figure IV. 2:</b> Plateforme globale de jade.....	61
<b>Figure IV. 3:</b> le modèle de workflow.....	63
<b>Figure IV. 4:</b> Diagramme de séquence entre les agents de système.....	64
<b>Figure IV. 5:</b> Fenêtre principal de l'agent serveur.....	65
<b>Figure IV. 6:</b> Fenêtre de l'agent serveur.....	66
<b>Figure IV. 7:</b> Formulaire pour créer un compte client.....	66
<b>Figure IV. 8:</b> Liste des agents clients existe dans le serveur.....	67
<b>Figure IV. 9:</b> L'agent client « belfirar ».....	67
<b>Figure IV. 10:</b> L'agent client « adda ».....	68
<b>Figure IV. 11:</b> L'agent client « Goismi ».....	68
<b>Figure IV. 12:</b> SNIFER, contrôleur des messagers échangés entre les agents du système.....	69
<b>Figure IV. 13:</b> Envoie du message à l'agent client « agent».....	70
<b>Figure IV. 14:</b> Réception du message de l'agent Serveur.....	70
<b>Figure IV. 15 :</b> demande de conge rempli par le demandeur.....	71
<b>Figure IV. 16:</b> la liste de demande congé.....	71
<b>Figure IV. 17:</b> la réponse de remplaçant l'agent client 'Goismi'.....	72
<b>Figure IV. 18:</b> la réception de message de confirmation de la demande par l'agent serveur.....	72
<b>Figure IV. 19:</b> Liste des demandes confirmée.....	73

## Liste des tableaux

<b>Tableau II. 1:</b> les tâches d'un projet dans méthodes PERT. ....	33
<b>Tableau II. 2:</b> La déférence entre méthode exacte et méthode approchée. ....	39
<b>Tableau III. 1:</b> Approche orientée objet (AOO) versus approche orientée agent (AOA).....	48

## Introduction générale

La notion d'organisation dans une entreprise a toujours été dépendante de la notion de tâche ou de celle d'échange d'informations. Or, cette dernière décennie a connu l'émergence et l'utilisation généralisée de la messagerie électronique et surtout la mise en place de plus en plus systématique d'outils d'aide au travail coopératif. Le type workflow qui est un concept permettant de modéliser et de gérer ces tâches industrielles ou administratives, impliquant plusieurs acteurs, documents et actions. Il consiste en des modèles de travail permettant de coordonner les activités de chaque participant et d'assurer leur parfaite interconnexion en s'appuyant sur les systèmes d'informations existants. Mais l'exécution de ces tâches a créé un problème de besoin d'adaptation, de flexibilité et de coopération entre pour réaliser les différents processus. Pour remédier à ce problème nous devons utiliser une technologie permettant de nous fournir des solutions adéquates à notre problème, une première solution pour fournir un espace d'interaction et de communication entre ces acteurs, une deuxième, pour la coopération entre eux et une dernière, pour leur donner plus d'autonomie dans la prise de décision. Cette technologie ce n'est que celle des systèmes multi-agents (SMA) ressemblant à des organisations distribuées dont les acteurs sont physiquement éloignés (notion d'entreprise distribuée ou virtuelle). Cette notion d'organisation d'acteurs distribués est très étudiée dans le domaine de l'Intelligence Artificielle distribuée (IAD).

Ce mémoire a pour objectif de présenter les systèmes multi-agents pour un ordonnancement distribué des tâches dans un workflow traitant la procédure des demandes de congé dans une entreprise.

Le présent mémoire s'articule autour de quatre chapitres faisant un tour sur le domaine du workflow et les agents et des méthodologies d'ordonnancement distribué. Nous présentons notre travail réalisé et son implémentation illustré dans un exemple fourni.

- **Le premier chapitre** sera consacré au domaine des workflow ; nous aborderons les différents concepts liés à ce domaine, les différents types de workflow existants.
- **Le deuxième chapitre** traitera la problématique de l'ordonnancement des tâches où nous présenterons un bref aperçu, après avoir défini la terminologie que nous allons adopter. Nous intéresserons d'abord aux méthodes exactes, puis aux méthodes approchées.
- **Le troisième chapitre** présentera de manière générale le concept d'agent et les systèmes multi-agents, nous y expliquons ce qu'est un agent, un SMA ainsi que les interactions entre eux. Puis la planification des méthodologies d'ordonnancement d'un SMA est détaillée.
- **Le quatrième chapitre** sera consacré pour présenter l'implémentation du système multi-agent (SMA) qui va nécessiter l'utilisation d'une plate-forme multi-agent Jade.

Finalement, ce manuscrit se terminera par une conclusion générale sur le travail réalisé ainsi que quelques perspectives pour les futurs travaux.

Chapitre I

WORKFLOW

## Chapitre I : WORKFLOW

### I.1. Introduction

Dans ce chapitre nous allons présenter le workflow. Cette technologie pluridisciplinaire introduite dans le domaine des organisations résulte de la combinaison de plusieurs domaines tels que le génie logiciel, la modélisation de l'entreprise, la composition des services web, etc...

Pour clarifier ce concept nous allons commencer par une définition donnée par WfMC<sup>1</sup> une des organisations internationales de normalisation dans ce domaine nous présentons aussi le modèle de référence le plus dominant et les types des workflow existants.

### I.2. Historique de workflow

L'industrie de l'imagerie électronique et de la gestion de la production assistée par ordinateur a été la première à réclamer une technologie qui permette l'automatisation des procédures de travail dont étaient réalisées à la main.

À partir de l'année 1975 et jusqu'à 1985, la nouvelle technologie dite de *workflow* a connu un essor important en mettant en place un système capable d'automatiser au mieux les flux de travail.

Or, le regain d'intérêt pour le génie logiciel au début des années 1990 a permis de relancer les recherches concernant les systèmes Workflow afin de mettre en place des systèmes plus simples à utiliser [1].

### I.3. Concepts clés

#### I.3.1. Processus métier

Processus métier est un ensemble de procédure et d'activités plus au moins liées qui réalisent collectivement un objectif métier [02,03], en général au sein d'une structure organisationnelle définissant des rôles et des relations fonctionnelles. Un processus métier peut être entièrement inclus dans une organisation simple ou peut s'étendre sur plusieurs organisations.

Un Processus métier peut combiner des activités automatiques et des activités manuelles (WFMC).

#### I.3.2. Modèle de processus

C'est la représentation d'un processus métier dans une forme qui supporte des manipulations automatiques comme la modélisation et l'exécution par un système de gestion de workflow une telle définition consiste en un réseau d'activité, en des critères pour indiquer le démarrage et la terminaison du processus ,ainsi que des informations sur les activités comme les participants, les applications et les données permettant la mise en œuvre des processus.

#### I.3.3. Tâche

Un processus métier est constitué d'un ensemble de tâches liées par des transitions [04]. Une tâche peut être manuelle ou automatique. Pour s'exécuter, elle utilise des ressources humaines et/ou machines. Quand une ressources et requise, la réalisation de l'activité est attribuée à un participant.

---

<sup>1</sup>WfMC : Workflow Management Coalition

- 1) **Tâche automatique** : une activité qui s'exécute sur un ordinateur et qui est entièrement contrôlée par le système de gestion de processus.
- 2) **Tâche manuelle** : une activité non automatisée qui reste en dehors du contrôle du système de gestion de processus.

### I.3.4. Nœud

Action interne à l'application mais n'ayant pas d'interaction ni avec l'utilisateur ni avec un élément externe.

### I.3.5. Transitions

Elles sont utilisées pour relier tous ces nœuds, représentant le changement d'un nœud à un autre au sein de l'instance.

Une transition est un point dans l'exécution d'une instance de processus ou une activité se termine et une autre démarre.

Une transition peut être inconditionnelle (la terminaison de l'activité précédente déclenche le démarrage de l'activité (ou les activités) suivante(s) ou conditionnelle (ce déclenchement est gardé par une condition logique).

### I.3.6. Évènement

Chaque nœud possède trois événements déclencheurs :

1. **onNodeEnter** : déclenchée à l'entrée du nœud.
2. **onNodeLeave** : déclenchée à la sortie du nœud courant.
3. **on transition** : déclenchée lors de la transition du nœud suivant un événement n'est pas lié qu'à une action, mais peut déclencher plusieurs actions.

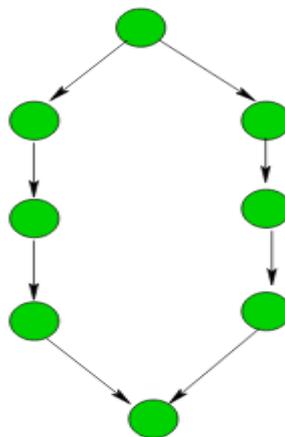


Figure I. 1: un événement des nœuds.

## I.4. Définitions de workflow

- Un Workflow est un flux d'informations au sein d'une organisation, par exemple en transmettant automatiquement des documents entre des personnes. On appelle « Workflow » (traduisez littéralement "flux de travail") la modélisation et la gestion informatique de l'ensemble des tâches à accomplir et des différents acteurs impliqués dans la réalisation d'un processus métier (aussi appelé processus opérationnel). Le terme

de Workflow pourrait donc être traduit en français par Gestion électronique des processus métier. De façon plus pratique le Workflow décrit le circuit de validation des tâches à accomplir entre les différents acteurs d'un processus, les délais, les modes de validation et fournit à chacun des acteurs les informations nécessaires pour la réalisation de sa tâche. Pour un processus de publication en ligne par exemple, il s'agit de la modélisation des tâches de l'ensemble de la chaîne éditoriale.

- Le workflow est l'automatisation totale ou partielle de l'exécution de processus métier. L'exécution au cours de laquelle des documents, des informations et des tâches passent d'un participant à un autre pour exécuter des actions précises selon des règles prédéfinies. Selon [05].

- Un workflow est un ensemble d'actions (étapes) s'enchaînant dans un ordre prédéfini. Ces actions peuvent s'enchaîner en fonction des conditions, d'interactions avec d'autres workflows ou en fonction d'interactions humaines. Les actions sont appelées également des activités. Une activité est un composant réutilisable représentant une étape d'un workflow.

- Un workflow est une représentation d'un procédé métier dans un format interprétable par la machine. Il est constitué d'un réseau d'activités et de dépendances entre elles, des critères pour spécifier le démarrage et la terminaison d'un procédé et des informations sur les activités individuelles (participants, applications, données informatiques associées etc.).

Les activités et les procédés ont des données en entrée et en sortie. Ces données sont représentées comme des ensembles d'éléments de données, appelés conteneurs.

#### I.4.1. Cycle de vie d'un workflow

Le cycle de vie d'un workflow (Figure I.2) est composé essentiellement de deux étapes [06].

- 1) La première étape est consacrée à la modélisation (ou l'édition) d'un processus. C'est-à-dire à la conception d'un schéma de processus. On la retrouve également sous le vocabulaire anglais "Build time". Dans la plupart des Systèmes de gestion Workflow (SGWf), un environnement graphique est proposé pour cette étape.
- 2) La deuxième étape est consacrée à l'instanciation d'un schéma de processus appelé "Cas" et à l'exécution de celui-ci. L'exécution peut faire appel à l'utilisateur et/ou à des applications externes. Cette étape est connue sous le vocabulaire anglais de "Run time". Le composant logiciel chargé de cette étape s'appelle un moteur de workflow.

La figure suivante représente le cycle de vie d'un workflow [07].

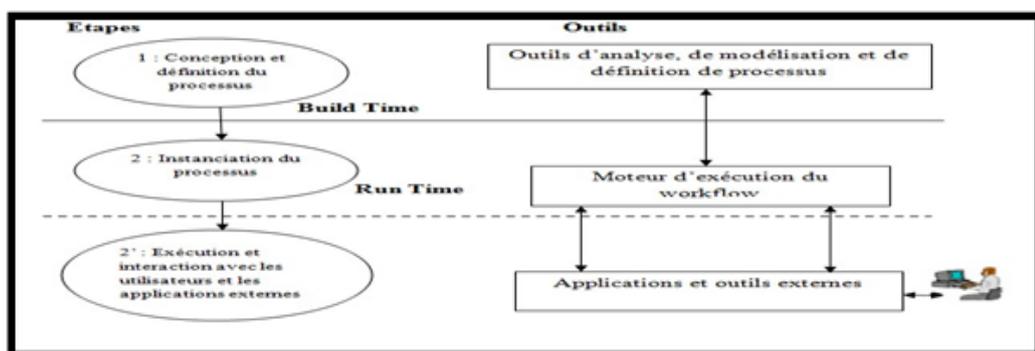


Figure I. 2: Cycle de vie d'un workflow.

Pour que les WFs<sup>2</sup> puissent être utilisés de manière efficace, il est important qu'il intègre les capacités suivantes [08] :

- Supporter les changements des modèles de processus.
- Permettre la surveillance de l'exécution des processus.
- Permettre la distribution des processus à travers des domaines d'affaires.
- Supporter l'assignation des ressources et des étapes des processus.

#### I.4.2. Système de gestion de workflow

Un système de gestion de Workflow (SGWF)<sup>3</sup> se définit comme un système qui définit implémente et gère l'exécution du processus workflow à l'aide d'un environnement logiciel fonctionnant avec un ou plusieurs moteurs de workflow. Il est capable d'interpréter la définition d'un processus, de gérer la coordination des participants et d'appeler des applications externes dont l'ordre d'exécution est pré défini dans une représentation informatique de la logique de ces procédures.

Le SGWF d'un WF joue le même rôle qu'un système de gestion de base de données, il représente l'infrastructure de support d'un WF et de ses processus métiers.

#### I.4.3. Le modèle de référence pour le workflow

Pour permettre l'émergence de standards dans le monde des systèmes des workflows des éditeurs de logiciels, des laboratoires de recherches et des utilisateurs de ces systèmes ont créé le consortium Workflow Management Coalition (WfMC). L'objectif de cette association est la promotion et le développement des systèmes de workflows. Pour cela ils ont réalisé un glossaire afin d'unifier la terminologie, ils ont défini un modèle de référence tel qu'il est noté dans la (figure I.3) centré autour du moteur d'exécution. [09]. Le modèle de référence identifie les composants de base, ainsi que les interfaces qui permettent l'interopérabilité entre les différents produits workflow. Ces composants sont illustrés dans la figure ci-dessous.

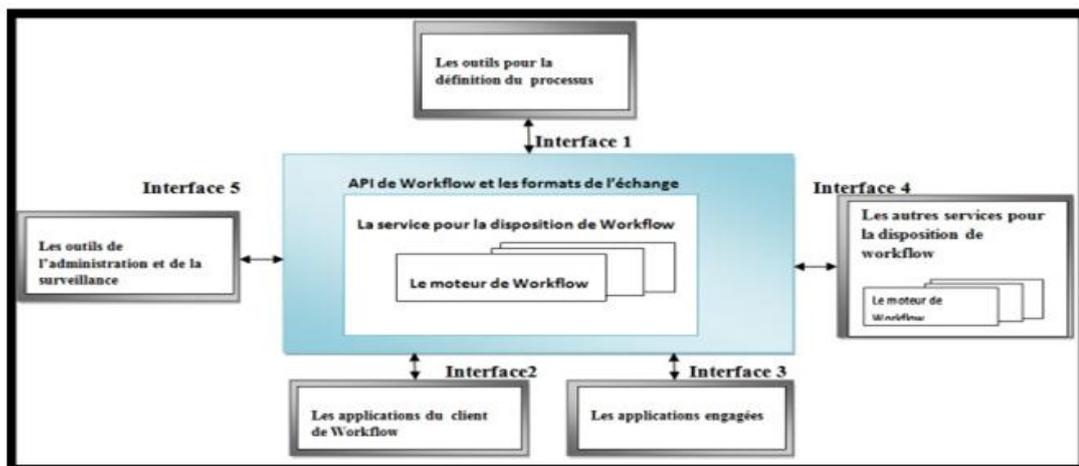


Figure I. 3: Le modèle de référence du workflow.

<sup>2</sup>WFs : Workflow

<sup>3</sup>SGWF :Système de Gestion de Workflow

- 1) **Le service pour la disposition de workflow** : est un service logiciel composé d'un ou plusieurs moteurs Workflow de même type qui servent à définir, gérer et exécuter des procédures Workflow.
- 2) **Le moteur de workflow** : est un service logiciel qui fournit tout ou partie de l'environnement d'exécution d'un workflow.
- 3) **L'Interface 1** : fournit le lien entre les outils de création et de modification de workflows et le moteur de workflow.
- 4) **L'Interface 2** : permet la communication entre les applications du client de workflow et le moteur de workflow.
- 5) **L'Interface 3** : permet d'invoquer des applications bien déterminées d'une activité donnée afin d'exécuter des tâches spécifiques.
- 6) **L'Interface 4** : permet l'interopérabilité et l'échange de travail entre plusieurs systèmes de gestion de workflow autonomes.
- 7) **L'Interface 5** : permet l'interconnexion entre outils d'administration et de surveillance et le moteur de workflow. Elle est divisée en deux parties :
  - a) les fonctions de système de gestion de workflow et les fonctions de cheminement de workflow.
  - b) Les outils d'administration et de surveillance peuvent auditer par exemple les temps d'attente, de complétion et d'exécution ainsi que le routage.

#### I.4.4. Types de workflow

D'après [10, 11] il y a quatre catégories complémentaires de workflow, qui sont :

##### I.4.4.1. Workflow d'administration

Un WF d'administration permet de gérer des procédures administratives dont les règles de déroulement sont bien établies et connues par chacun au sein de l'entreprise [13, 14]. Il est caractérisé par la circulation de documents/formulaires à travers l'entreprise.

( Par exemple, demande de remboursement de frais de missions, demande d'inscription à une formation, etc.).

De ce fait, il aide à transformer la circulation de documents papiers en circulation de documents électroniques. Les workflows d'administration ne sont pas d'une très grande valeur ajoutée pour une entreprise d'une part, et d'autre part, ils sont assez statiques si l'on considère leur degré de répétition élevé.

- 1) **Processus concernés** : les processus de soutien de l'entreprise.
- 2) **Caractéristiques** : il s'agit d'automatiser, suivant des procédures prédéfinies, la manipulation de formulaires électroniques en remplacement des imprimés. En effet, ces formulaires ont pour objectif de simplifier les procédures répétitives. Ils limitent la circulation du papier. Un formulaire s'accompagne généralement de la gestion du circuit<<route>> qu'il doit emprunter en cas de réussite (validation) ou échec (rejet).

De par la nature des informations transmises, le niveau de sécurité requis peut nécessiter l'utilisation de signatures électroniques. Lorsqu'un acteur utilise un circuit au travers d'un message ou d'un formulaire, il doit être capable de savoir où situer le message dans le circuit et ce, à tout moment.

**Exemples** : traitement des frais de déplacement, traitement des demandes de congés  
traitement des prêts...

#### **I.4.4.2. Workflow de production**

Un workflow de production permet de gérer les processus de production dans l'entreprise. Il constitue généralement le cœur de métier de l'entreprise et sa valeur ajoutée dépend énormément de la qualité de ce workflow. C'est l'efficacité de ce type de workflows qui assure à l'entreprise des avantages compétitifs [15,16,17] (par exemple workflow de livraison pour une entreprise de vente en ligne, procédé de gestion de prêts dans une banque, workflow de gestion des demandes de dommages et intérêts au sein d'une compagnie d'assurance, workflow de gestion de la chaîne de production chez un constructeur automobile, etc.).

- 1) **Processus concernés** : les processus opérationnels, répétitifs et critiques pour la performance globale de l'entreprise ou de l'unité organisationnelle qui en est responsable. Ce sont des processus inhérents aux métiers de base de l'entreprise.
- 2) **Caractéristique** : un workflow de production et caractérisé par un cadre procédural formel qui s'applique à toutes les activités et à tous les rôles impliqués dans l'accomplissement d'un processus donné. Les procédures sont définies par des circuits de documents prédéfinis et formalisés en application de règles particulières. Ces applications workflow sont généralement intégrées aux applications de production mises en œuvre dans les métiers correspondants. L'interfaçage avec les applications existantes est alors nécessaire.

**Exemples** : le traitement des contrats d'assurance. Le commerce électronique. Le traitement des achats...

#### **I.4.4.3. Workflow de collaboration ou coopératif**

Un workflow de collaboration [17, 18, 19,20] permet de gérer la conscience et la collaboration de groupe lors d'un projet de travail créatif (par exemple, conception logicielle, réalisation d'un plan de bâtiment, montage d'un projet, etc.). Il est caractérisé par une forte valeur ajoutée au sein de l'entreprise, mais par un faible degré de répétition. Il se distingue des autres types de workflows par le grand nombre de participants qui interagissent en permanence pour la réalisation du projet en commun. La complexité de ce type de procédé réside dans la difficulté de modélisation de la méthodologie de travail à suivre surtout qu'il faut prévoir, pendant le déroulement du workflow, l'arrivée de nouveaux participants, la création de nouvelles activités à intégrer et à gérer, etc.

- 1) **Processus concernés et exemples** : le processus de recherche et développement le processus de conception et de lancement d'un nouveau produit ou d'un nouveau service, le processus d'innovation de produits, le processus de planification stratégique...
- 2) **Caractéristiques** : les applications de workflow de type coopératif allient la complexité des processus et la souplesse organisationnelle attendue des utilisateurs. Les membres d'un groupe modélisent le processus de travail, fixent les règles, exploitent directement l'application et peuvent faire évoluer le processus et ses règles de gestion en fonction des évolutions des modes opératoires [17].

#### I.4.4.4. Workflow ad-hoc

Un workflow ad-hoc [21] représente une classe de workflow destinés à des situations spécifiques où la logique de déroulement à suivre est définie durant l'exécution. Il forme une solution hybride rassemblant des caractéristiques d'administration, de production et de collaboration. Ce type de workflow gère des situations uniques ou causées par des exceptions. Généralement, ce type de workflow ne possède pas de structure prédéfinie l'étape ultérieure à suivre est déterminée essentiellement par les participants au workflow. Par exemple, si un coordinateur envoie une note d'information aux membres de son équipe, ces derniers ont le choix de faire ce qu'ils veulent avec cette note. Ils peuvent éventuellement la renvoyer à d'autres personnes qu'elle peut intéresser.

- 1) **Processus concernés** : les procédures d'exception, occasionnelles voire uniques. Ces processus pourraient, dans certains cas, représenter des enjeux critiques pour la performance de l'entreprise mais ils sont le plus souvent liés à des routines administratives. Contrairement aux workflow de production. Ils sont caractérisés par plus difficiles à définir en détail.
- 2) **Caractéristiques** : Les applications (interfaçage nécessaire) sont plutôt des outils bureautiques tels que tableurs et traitements de texte que des applications de gestion plus lourdes.

**Exemples:** La gestion de correspondance institutionnelle exigeant, parfois, des révisions de recrutement d'une compétence particulière, rédaction collective d'un rapport d'expertise, enquêtes et sondages...

#### I.4.5. Concepts de base de WF

Les concepts de base de workflow sont expliqués par la métaphore de "3R" (Routes Rules, Rôles) de Ronni Marshak [22].

- 1) **Le routage organise la dynamique des processus**

Le routage des documents, des informations, ou des tâches a été la première grande fonction du workflow. Ce routage désigne les itinéraires d'un workflow c'est-à-dire les chemins que prennent les différents résultats d'une activité à l'autre d'un rôle à l'autre et donc, d'un participant à l'autre. Ces chemins peuvent être totalement ou partiellement spécifiés à l'avance. Ou alors aucun chemin n'est prédéfini, l'ordonnancement des activités n'est défini qu'au moment de l'action. L'optimisation d'un workflow est réalisée lorsqu'on spécifie l'ordonnancement des interdépendances fondamentales mais qu'on laisse quand même une petite marge de manœuvre à l'acteur afin de rester flexible.

- 2) **Les règles formalisent la coordination**

La gestion des règles de coordination des activités est complémentaire au routage car l'itinéraire d'un processus dépend des règles qui définissent à la fois la nature des informations et leurs modalités de transport d'une personne à l'autre. Ces règles peuvent être simples, compliquées ou complexes, mais ce qui est sûr c'est qu'elles sont indispensables au workflow.

- 3) **Les rôles accomplissent des activités**

Finalement, il faut gérer les différents rôles, le troisième "R" de la métaphore. Il s'agit de gérer les différentes personnes qui accomplissent les tâches et qui communiquent entre elles. Ici, il est important de noter qu'en fait, on ne gère pas des personnes en tant qu'individus mais en tant que rôles, c'est-à-dire des fonctions. Ces rôles ne sont pas nécessairement des personnes, car les tâches ne

sont pas nécessairement réalisées par des personnes. Router un courrier électronique, faire des calculs complexes sont des exemples de tâches qui peuvent facilement être automatisées par des programmes.

#### **I.4.6. Les typologies des applications de Workflow**

Il existe plusieurs typologies d'applications de Workflow. L'étude des critères techniques permet d'établir une classification basée sur la technologie dominante (messagerie ou base de données). Celle des critères permet d'établir une classification fondée sur les services proposés par les systèmes de gestion de Workflow.

Ces différentes typologies sont utiles pour comprendre ce qu'est le workflow, elles sont également une aide à décision par rapport à une problématique de terrain [15].

##### **I.4.6.1. Une typologie technique**

Consiste à faire la distinction entre les solutions de workflow reposant sur l'exploitation d'une messagerie, celles reposant sur l'exploitation d'une base de données et celles reposant sur les deux à la fois. C'est une classification utile au moment du choix d'un logiciel de workflow. En effet, la technologie détermine les implications d'intégration à l'infrastructure réseau de l'organisation comme aux autres applications existantes appelées par les activités de workflow.

###### **I.4.6.1.1. Le workflow fondé sur une messagerie**

De façon générale, On associe l'automatisation du workflow, au routage automatique de documents. Le routage repose sur une messagerie pour acheminer les documents d'une personne à une autre. Le workflow par routage imite le circuit papier : une personne réalise une action sur un document, qui est ensuite envoyé à la personne suivante. L'inconvénient est qu'une fois envoyé, le document ne peut être consulté que par la personne à qui il est destiné [17].

On distingue deux familles parmi les applications de workflow fondées sur une messagerie :

- 1) les applications<<communicantes>>qui intègrent des fonctionnalités de messageries. les fonctionnalités des messageries (notifications, accusés de réception règles archivage, signatures électroniques) suffisent dans de nombreux cas à créer et gérer des workflows<<légers>>.
- 2) les applications<<communicantes>> qui intègrent des fonctionnalités de workflow. Ces produits sont dotés de fonctionnalités capables de gérer les flux les plus couramment utilisés dans les transactions liées au métier de l'entreprise.

C'est le cas, par exemple, des gestionnaires de formulaires. Ils intègrent des fonctionnalités de messagerie et de workflow. Les utilisations peuvent envoyer des formulaires, les transférer en fonction d'un circuit prédéfini et suivre leur progression dans le circuit [13].

###### **I.4.6.1.2. Le workflow fondé sur une base de données**

Les systèmes de workflow s'appuyant sur une base de données permettant aux utilisateurs de consulter une base de suivi pour vérifier l'état des documents.

Le moteur de workflow s'exécute sur la partie serveur. Le client ne gère que l'interface graphique. Le serveur stocke et gère les bases de données contenant les modèles de workflow, les carnets d'adresse des utilisateurs et des groupes. Le serveur gère aussi le moteur de workflow. Ce moteur est responsable de l'activation du workflow, du suivi, des

notifications, des règles. Il peut s'interfacer à d'autres moteurs de transport pour transférer les objets sur un trajet défini. [13], Ce modèle présente plusieurs avantages :

- 1) Du fait qu'elle réside sur un serveur, la base de données est soumise aux processus de celui-ci capables d'initier une action sans activité particulière de la part de l'utilisateur ou d'une condition externe.
- 2) Le document reste accessible aux autres personnes pendant le déroulement du processus.
- 3) La gestion du workflow est facilitée. Le serveur peut contrôler des instances spécifiques du processus tout en conservant des statistiques consolidées du processus global, ces dernières assurant une gestion et planification améliorées du workflow [17].

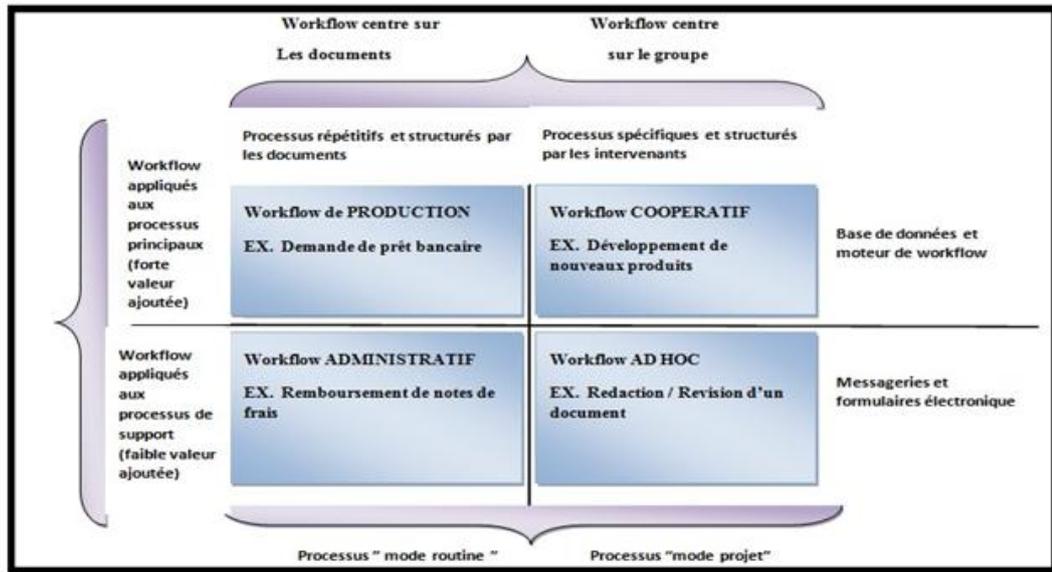


Figure I. 4: Typologie fonctionnelle des applications de workflow.

#### I.4.7. Regroupement des workflows

Une première manière de regrouper les workflows est la suivante :

- 1) les workflows documentaires (rédaction, validation, traduction de documents).
- 2) les workflows métiers, représentant un processus transverse à l'entreprise (concerne Plusieurs entités organisationnelles). Il s'agit le plus souvent de procédures rattachées à une division opérationnelle de l'entreprise. Cette procédure peut être qualifiée de procédure métier, processus métier, procédure opérationnelle ou Business Process. Cela désigne un ensemble d'activités qui s'enchaînent de manière chronologique pour atteindre un objectif, généralement délivrer un produit ou un service, dans le contexte d'une organisation de travail (ex : une entreprise, administration, etc.). Ainsi, on qualifie de Business Process (processus métier) l'ensemble des activités et de procédures qui permettent collectivement la réalisation d'un objectif métier. Le workflow correspond à l'automatisation de ce Business Process.

### **I.4.8. Modélisation des workflow**

Un système workflow est modélisé selon plusieurs aspects, selon [12].

#### **I.4.8.1. L'aspect fonctionnel**

L'aspect fonctionnel concerne l'identification des activités des processus que l'on souhaite modéliser. Il permet d'établir la hiérarchie des activités, c'est-à-dire d'exprimer de possibles décompositions en termes de sous-processus. Enfin, le modèle fonctionnel doit aussi représenter le flux de données associées aux activités et les interdépendances de données entre les activités (data flow).

#### **I.4.8.2. L'aspect comportemental**

Il correspond à la dynamique du processus. Le comportement s'exprime par la modélisation d'un contrôle de flux entre les activités. Ce dernier permet d'indiquer la chronologie de l'exécution des activités, leur flux (séquentiel ou parallèle), les points de synchronisation entre activités ou au contraire, les points de disjonction. De plus, il doit représenter les événements qui permettent de déclencher les activités. Notons pour souligner l'importance de ce modèle, qui permet l'exécution du Workflow. L'aspect comportemental est également appelé aspect de coordination.

#### **I.4.8.3. L'aspect informationnel (donnée)**

Cet aspect concerne l'ensemble des informations et des données qui sont associées aux activités. Il décrit en détail les relations qui existent entre les données, leur type et leur structure.

#### **I.4.8.4. L'aspect organisationnel**

Il concerne la description de l'organisation des acteurs de l'entreprise. Le modèle organisationnel peut refléter fidèlement l'organigramme de l'entreprise, c'est-à-dire la décomposition hiérarchique de celle-ci en départements et services ou bien décrire des unités organisationnelles dans lesquelles on identifie des acteurs. Selon la méthode choisie, la description est plus ou moins détaillée et permet d'établir des liens hiérarchiques entre les acteurs ainsi que des relations entre unités organisationnelles ou départements. Toutefois, quelle que soit la méthode retenue, la description des rôles associés aux différentes activités reste invariante. Les rôles créent l'interface entre le modèle organisationnel et les modèles représentant les activités.

### **I.5. Les Domaines de l'entreprise intéressés par une application de Workflow**

Il y a plusieurs domaines dans l'entreprise qui peuvent être intéressés par une application de Workflow [1] dont on peut citer :

- 1) L'organisation.
- 2) La collaboration.
- 3) La production.
- 4) La gestion des ressources humaines.
- 5) La gestion de relation client.

Le Workflow est Adapté de plus en plus aux procédures bien définies à savoir:

- 1) Traitement des demandes de congé ou de mutation
- 2) des lettres de réclamation des clients
- 3) préparation du budget annuel.

- 4) préparation des instructions de demande de crédit
- 5) Préparation des contrats avec les fournisseurs et partenaires.

## I.6. Les Grandes Fonctions d'une Application Workflow

Il y a plusieurs fonctions d'une Application Workflow [1] on peut citer :

- 1) Le Workflow permet à l'administrateur de l'application de gérer les descriptions de l'organisation, attribuer les droits aux participants, et produire les statistiques d'exploitation.
- 2) Une représentation complète de l'organisation comprenant : les groupes les acteurs, les rôles des acteurs dans l'organisation.
- 3) Un outil d'analyse et de simulation des procédures avant leur mise en exploitation.
- 4) Tenir à jour pour chaque utilisateur une liste de tâches à accomplir.
- 5) Intégrateur au service de l'utilisateur.
- 6) Il rend les informations et documents utiles à chaque tâche accessible immédiatement par simple sélection des icônes qui les représentent à l'écran.

## I.7. Impacts du workflow

Les différents avantages et bénéfices rencontrés lors de l'introduction d'un système de workflow peuvent être de deux natures. Soit ils sont mesurables donc tangibles, soit ils sont moins « palpables », mais contribuent tout autant à l'amélioration significative de la qualité du travail effectué [23].

### I.7.1. Gains tangibles

Du côté des gains tangibles [23]. Nous retrouvons les éléments suivants :

- 1) **Réduction des coûts opérationnels** : les organisations utilisant des systèmes de workflow constatent une diminution des coûts de transition.
- 2) **Amélioration de la productivité** : les opérations routinières et répétitives peuvent être automatisées réduisant ainsi significativement le temps d'exécution du processus .De plus, le travail peut être effectué 24h/24, ceci étant un facteur vital pour les multinationales et les entreprises effectuant des transactions commerciales par le biais d'Internet.
- 3) **Processus plus rapides** : deux facteurs expliquent le gain de temps des processus gérés par des systèmes de workflow .le premier, est dû à l'automatisation des opérations routinières. Le deuxième concerne les activités « manuelle » où nécessitant une intervention humaine. celles –ci, peuvent souvent être effectuées parallèlement (en tous cas pour une partie d'entre elles). Le workflow permet dans ce cas, grâce à une coordination efficace et une attribution des activités à plusieurs acteurs, de faire progresser le processus nettement plus rapidement.

### I.7.2. Gains intangibles

Ces gains [23] sont les suivants :

Service amélioré : grâce à la rapidité de gestion des demandes de la clientèle ainsi qu'à une meilleure information sur l'état d'avancement de celle-ci, le service rendu aux clients s'en trouve amélioré.

- 1) **Amélioration des conditions de travail des employés** : Les tâches répétitives et peu gratifiantes peuvent être automatisées, libérant de cette façon le personnel pour des activités plus intéressantes.
- 2) **Facilitation du changement** : Les entreprises peuvent constamment, grâce aux systèmes de workflow, redéfinir et automatiser leur processus.
- 3) **Augmentation de la qualité** : Suite aux automatisations des tâches répétitives ainsi qu'à une meilleure coordination et compréhension du travail ; les erreurs sont plus rares.
- 4) **Communication facilitée** : Grâce aux informations disponibles concernant les tâches à effectuer et l'état d'avancement des processus, la communication et la transparence du travail sont améliorées.
- 5) **Aide à la prise de décision** : Etant informé du déroulement des processus et des activités, il est plus facile de prendre les bonnes décisions.
- 6) **Amélioration du planning** : les informations disponibles concernant l'organisation, son business et ses processus améliorent les facultés de planning.
- 7) **Communication inter-entreprises** : La gestion de processus inter-entreprises augmente considérablement la productivité de la transparence du marché.

## I.8. Les Avantages et les inconvénients de Workflow

### I.8.1. les avantages

Il y a plusieurs avantages [1] comme:

- 1) **Amélioration de la productivité** : gains de productivité réalisés par une application de Workflow sont de 20 à 50% sur la part des tâches qu'elle automatise.
- 2) **Temps de réponse réduit** : la réduction du temps de réponse entre la prise en compte de l'événement déclencheur d'un cas et son traitement complet est une des caractéristiques essentielles des applications de workflow.
- 3) **Sécurité accrue** avec une application de workflow, c'est le système qui affecte les tâches aux participants.
- 4) **Maîtrise de la qualité et des coûts** : une application de workflow enregistre systématiquement le journal de tous les événements qu'elle contrôle, avec la date et l'heure, la procédure et la tâche concernée et le participant actif. Des outils de traitement de ce journal fournissent des rapports de synthèse sur les coûts et délais de traitement des tâches et des procédures.

### **I.8.2. les inconvénients**

Comme aussi nous avons trouvé des inconvénients [1] ont citons :

- 1) Inadapté si l'on n'est pas dans un contexte « procédural ».
- 2) Analyse longue et difficile, car la création d'un Workflow nécessite une analyse du projet. Cette analyse est souvent longue et difficile.
- 3) Contraintes imposées par le logiciel : L'utilisation d'un Workflow nécessite que tous les participants à un projet saisissent leur état d'avancement dans chacune des tâches qu'ils ont à effectuer. Ceci est très lourd et souvent les utilisateurs n'en voient pas la nécessité. Ils ne le font donc pas régulièrement et parfois ne le font pas du tout.
- 4) Lors de l'utilisation de l'application, les utilisateurs peuvent rencontrer différents problèmes. (par exemple, des problèmes d'impression des documents du workflow à partir du navigateur web, ou bien des indisponibilités réseau.)

### **I.9. Conclusion**

Nous présentons dans cette partie une synthèse bibliographique sur les principaux concepts de workflow et ses différents types (production, administratifs, ad hoc ou collaboratifs), et ces avantages et des inconvénients, nous avons aussi cité les phases de cycle de vie de workflow.

Dans le chapitre suivant nous présentons une généralisation sur l'ordonnement.

## Chapitre II : Généralités sur l'ordonnancement

## Chapitre II : Généralités sur l'ordonnancement

### II.1. Introduction

L'ordonnancement est la fonction d'exploitation qui gère les planifications de traitements informatique. Son rôle principal est la soumission détaches, ou de séquence des tâche son peut distinguer plusieurs approches de résolution d'un problème d'ordonnancement. Dans ce chapitre nous avons exposé les notions fondamentales de l'ordonnancement et les différentes techniques que peuvent adopter les problèmes d'ordonnancement des tâches.

### II.2. Définitions

- Un problème d'ordonnancement peut être considéré comme un sous problème de planification dans lequel il s'agit de décider de l'exécution opérationnelle des tâches (jobs) planifiées, et ainsi d'établir leur planning d'exécution et leur allouer des ressources visant à satisfaire un ou plusieurs objectifs sous une ou plusieurs contraintes. En se basant sur les concepts de tâche, ressource, contrainte et objectif, l'ordonnancement peut également être déterminé.
- Ordonnancer un ensemble de tâches c'est programmer leur exécution en leur allouant les ressources requises et en fixant leurs dates de début. La théorie de l'ordonnancement traite de modèles mathématiques mais analyse également des situations réelles fortes complexes, aussi le développement de méthodes utiles ne peut-il être que le fruit de contacts entre la théorie et la pratique [24].
- L'ordonnancement est la programmation dans le temps de l'exécution d'une série des tâches (activités, opérations) sur un e<sup>2</sup>nsemble de ressources physiques (humaines et techniques), en cherchant à optimiser certains critères, financiers ou technologiques, et en respectant les contraintes de fabrication et d'organisation. Le problème d'ordonnancement consiste à organiser dans le temps la réalisation d'un ensemble de tâches, compte tenu de contraintes temporelles (délais contraintes d'enchaînements,...) et de contraintes portant sur l'utilisation et la disponibilité des ressources requises [25].

### II.3. Les domaines concernés de l'ordonnancement

Cette figure (Figure II. 1) représente les domaines d'utilisé l'ordonnancement [24].

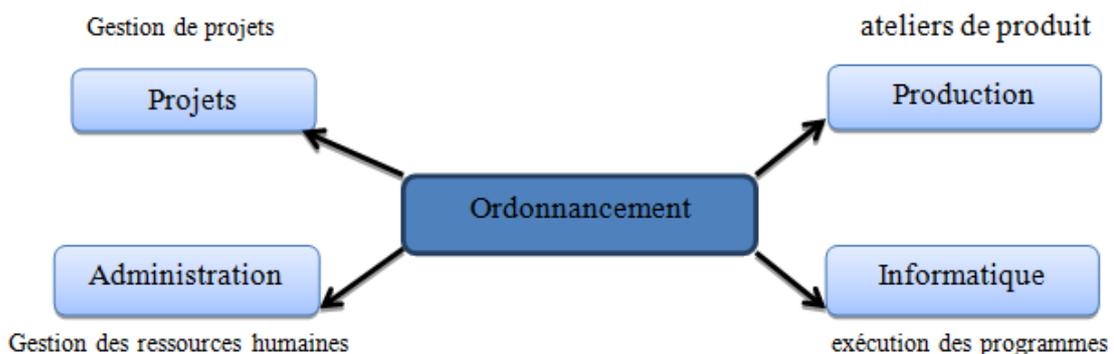


Figure II. 1: les domaines de l'ordonnancement.

## II.4. L'objectif de l'ordonnancement

Les objectifs des entreprises se sont diversifiés et le processus d'ordonnancement est devenu de plus en plus multicritères. D'une manière générale, on distingue plusieurs classes les objectifs concernant un ordonnancement [25].

- 1) **les objectifs liés au temps** : on trouve par exemple la minimisation du temps total d'exécution, du temps moyen d'achèvement, des durées totales de réglage ou des retards par rapport aux dates de livraison.
- 2) **les objectifs liés aux ressources** : maximiser la charge d'une ressource ou minimiser le nombre de ressources nécessaires pour réaliser un ensemble de tâches sont des objectifs de ce type.
- 3) **les objectifs liés au coût** : ces objectifs sont généralement de minimiser les coûts de lancement, de production, de stockage, de transport.

## II.5. Les notions fondamentales de l'ordonnancement

Il y a trois notions pour l'ordonnancement :

### 1) Les tâches

Une tâche ou un job est une entité élémentaire localisée dans le temps, par une date de début et une date de fin, et dont la réalisation nécessite une durée préalablement définie. Elle est constituée d'un ensemble d'opérations qui requiert, pour son exécution, certaines ressources et qu'il est nécessaire de programmer de façon à optimiser un certain objectif [26].

### 2) Les ressources

La ressource est un moyen technique ou humain destiné à être utilisé pour la réalisation d'une tâche et disponible en quantité limitée, sa capacité. Plusieurs types de ressources sont à distinguer. Une ressource est renouvelable si après avoir été allouée à une ou plusieurs tâches, elle est à nouveau disponible en même quantité (les hommes, les machines, l'équipement en général) ; la quantité de ressource utilisable à chaque instant est limitée. Dans le cas contraire, elle est consommable (matières premières, budget) ; la consommation globale (ou cumul) au cours du temps est limitée. Une ressource est doublement contrainte lorsque son utilisation instantanée et sa consommation globale sont toutes deux limitées (l'argent en est un bon exemple). Qu'elle soit renouvelable ou consommable, la disponibilité d'une ressource peut varier au cours du temps. Sa courbe de disponibilité est en général connue a priori, sauf dans les cas où elle dépend du placement de certaines tâches génératrices. On distingue par ailleurs principalement dans le cas de ressources renouvelables, les ressources disjonctives qui ne peuvent exécuter qu'une tâche à la fois (machine-outil, robot manipulateur) et les ressources cumulatives qui peuvent être utilisées par plusieurs tâches simultanément mais en nombre limité (équipe d'ouvriers, poste de travail) [26].

### 3) Les contraintes

Les contraintes expriment des restrictions sur les valeurs que peuvent prendre simultanément les variables de décision. On distingue :

Des contraintes temporelles concernent les contraintes de temps alloué, issues généralement d'impératifs de gestion et relatives aux dates limites des tâches (de lais de livraisons, disponibilité des approvisionnements) ou à la durée totale d'un projet et les

contraintes de cohérence technologique, ou contraintes de gammes, qui décrivent des relations d'ordre relatif entre les différentes tâches.

Des contraintes de ressources décrivent les contraintes d'utilisation de ressources qui expriment la nature et la quantité des moyens utilisés par les tâches, ainsi que les caractéristiques d'utilisation de ces moyens et les contraintes de disponibilité des ressources qui précisent la nature et la quantité des moyens disponibles au cours du temps. Toutes ces contraintes peuvent être formalisées sur la base des distances entre débuts de tâche ou potentiels [26].

## II.6. Les étapes de l'ordonnancement

L'ordonnancement se déroule en trois étapes [27] :

- A) **La planification** : qui vise à déterminer les différentes opérations à réaliser, les dates correspondantes et les moyens matériels et humains à y affecter.
- B) **L'exécution** : qui consiste à la mise en œuvre des différentes opérations définies dans la phase de planification.
- C) **Le contrôle** : qui consiste à effectuer une comparaison entre planification et exécution, soit au niveau des coûts, soit au niveau des dates de réalisation.

## II.7. Méthodes de résolution des problèmes d'ordonnancement

### II.7.1. Modèles de planification de tâches d'un projet

Parmi les principaux modèles de planification de tâches d'un projet [28] on peut citer celles qui seront définies dans les sous sections suivantes.

#### II.7.1.1. La méthode PERT<sup>4</sup>

Consiste à mettre en ordre sans forme d'un graphe orienté afin d'atteindre l'objectif du projet le graphe est constitué des étapes (cercle) et des tâches (flèche).

##### 1) Principe de la méthode

Réduire la durée totale d'un projet par une analyse détaillée des tâches ou activités élémentaires et de leur enchaînement. On étudie les délais sans prendre en compte les charges.

##### 2) Notions de base

La méthode s'appuie en grande partie sur une représentation graphique qui permet de bâtir un «réseau PERT».

Un réseau PERT est constitué par des tâches et des étapes

- a) **Étape**: commencement ou fin d'une tâche. Une étape n'a pas de durée. On symbolise une étape (ou «nœud») sur le réseau par un cercle.
- b) **Tâche**: déroulement dans le temps d'une opération. Contrairement à l'étape la tâche est pénalisante car elle demande toujours une certaine durée, des moyens (ou ressources) et coûte de l'argent. Elle est symbolisée par un vecteur (ou arc

<sup>4</sup>PERT: Programmed Evolution rive nique.

orienté, ou liaison orientée) sur lequel seront indiqués l'action à effectuer et le temps estimé de réalisation de cette tâche.

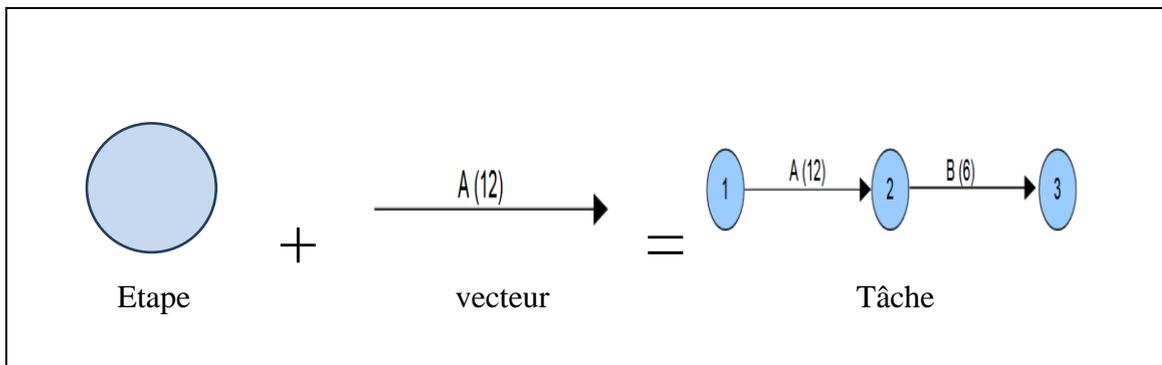


Figure II. 2: les composants d'une tâche.

### Remarques

- La longueur des arcs n'est pas proportionnelle au temps d'exécution.
- Pour alléger la représentation, on ne note pas le nom complet de la tâche mais une lettre ou code le représentant.

### 3) Les types des tâches en PERT

#### a) Tâches successives

Une ne peut pas commencer que la précédente soit terminée (dépendance).

#### b) Tâches simultanées

Plusieurs tâches peuvent être exécutées en même temps, elles patient de la même étape (non dépendance).

#### c) Tâches convergentes

Plusieurs tâches vont vers la même étape.

#### d) Tâche fictive

Une tâche représente une contrainte entre tâches non indépendantes, chaque tâche fictive est représentée par une flèche pointillée sa durée est nulle et n'occupe aucune source.

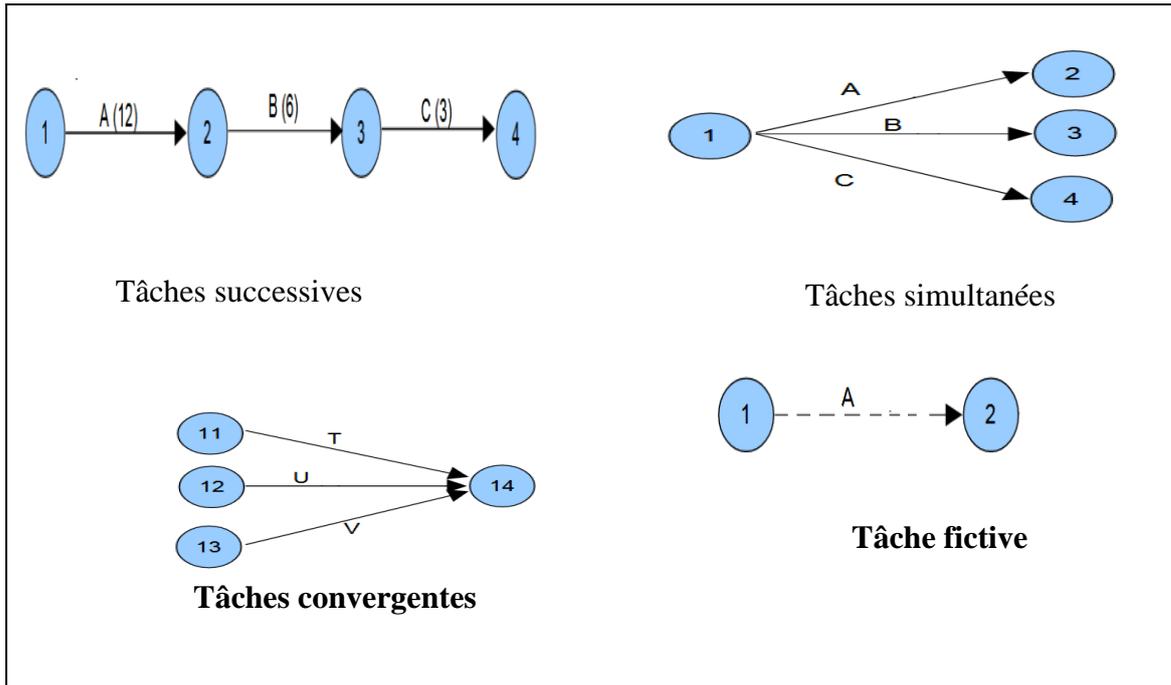


Figure II. 3: Les types des tâches en PERT.

#### 4) Méthodologie de construction d'un réseau PERT<sup>2</sup>

- Établir la liste des tâches (faire le partitionnement des tâches en fonction des ressources).
- Déterminer des antériorités : tâches immédiatement antérieures, et tâches antérieures.
- Déterminer les niveaux d'exécution ou rang des tâches (optionnel).
- Construire le réseau PERT.
- Calculer la durée du projet, les dates début et de fin des tâches. Déterminer le chemin critique. Mettre en évidence les marges.

#### 5) Quelques définitions à retenir

- **Début au plus tôt d'exécution d'une tâche** : C'est le maximum des fins au plus tôt des tâches qui la déclenche (Il peut exceptionnellement y avoir un retard ou chevauchement si le cahier des charges du projet le précise et que la faisabilité est vérifiée).
- **Début au plus tard d'une tâche** : C'est la date de fin au plus tard de la tâche moins la durée de la tâche.
- **Fin au plus tôt** : C'est la date de début au plus tôt plus la durée de la tâche.
- **Fin au plus tard** : C'est le minimum des dates de début au plus tard des tâches qu'elle enclenche.
- **Marge totale** : C'est le retard admissible du début d'une tâche qui n'entraîne aucun recul delà date de fin du projet, mais qui consomme les marges libres des opérations suivantes. C'est la date de début au plus tard moins la date de début au plus tôt.
- **Marge libre** : C'est le retard admissible sur une tâche qui n'entraîne pas de modification des calendriers des tâches suivantes. C'est la date de début au plus

tôt de la tâche suivante moins la durée de la tâche moins la date de début au plus tôt de la tâche.

- **Chemin critique** : C'est l'ensemble des tâches dont la marge totale et la marge libre est nulle. C'est le chemin dont la succession des tâches donne la durée d'exécution la plus longue projet et fournit le délai d'achèvement le plus court. Si l'on prend du retard sur la réalisation de ces tâches, la durée globale du projet est allongée. Pour l'application précédente, on peut dresser un tableau de synthèse comme suit :

### Exemple

Taches	Durée (jours)	Début +tot	Début+ tard	Fin+ tot	Fin+ tard	Marge libre	Marge totale	Chemin critique
A	3	0	0	3	3	0	0	A
B	4	0	1	4	5	1	1	-
C	2	3	3	5	5	0	0	c
D	3	3	6	6	9	3	3	-
E	4	5	5	9	9	0	0	E
...								
...								

Tableau II. 1: les tâches d'un projet dans méthodes PERT.

#### Résultats de l'étude :

La durée globale du projet (délai d'achèvement le plus court) = 9 jours.

Le chemin critique est constitué des tâches : A, C, et E.

Il faut vérifier que ces tâches se déroulent correctement et le cas échéant adopter des actions correctives afin de les fiabiliser [28].

#### II.7.1.2. Le Diagramme de Gantt

Ce type de diagramme a été mis au point par un américain Henry Gantt, Le diagramme de Gantt est un outil permettant de modéliser la planification de tâches nécessaires à la réalisation d'un projet. Le principe de ce type de diagramme est de représenter au sein d'un tableau, en ligne les différentes tâches et en colonnes les unités de temps (exprimées en mois, semaines, jours, etc.). Cette méthode a été développée au début des années 1960 et elle est actuellement intégrée dans tous les logiciels de planification de projets (MS Project, Gantt Project) [25].

##### 1) Réalisation

Les différentes étapes de réalisation d'un diagramme de Gantt sont les suivantes :

**Etape 1** : On détermine les différentes tâches (ou opérations) à réaliser et leur durée.

**Etape 2** : on définit les relations d'antériorité entre tâches.

**Etape 3** : on représente d'abord les tâches n'ayant aucune antériorité, puis les tâches dont les tâches antérieures ont déjà été représentées, et ainsi de suite...

**Etape 4** : on représente par un trait parallèle en pointillé à la tâche planifiée la progression réelle du travail.

## 2) Exemple

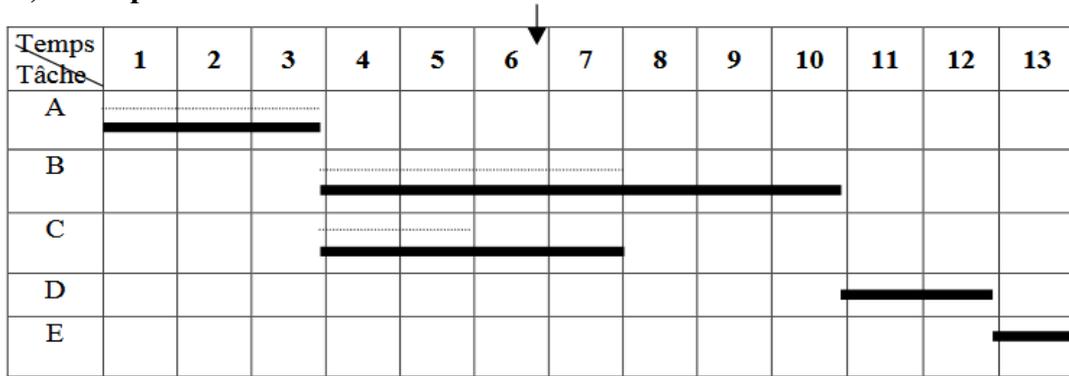


Figure II. 4: le diagramme de GANTT.

## 3) Remarque :

Chaque colonne représente une unité de temps.

Les durées d'exécution prévues des tâches sont représentées par un trait épais.  
(4 unités de temps pour C).

Les contraintes de succession se lisent immédiatement.

-Les tâches B et C succèdent à la tâche A.

-D succède à B.

Le déroulement d'exécution des tâches figure en pointillé, au fur et à mesure des contrôles on est à la fin de la 6<sup>ème</sup> unité de temps, B est en avance d'une unité et, C'est en retard d'une unité.

On peut alors déterminer le chemin critique: qui est formé d'une succession de tâches, sur le chemin le plus long en termes de durées. Il est appelé chemin critique car tout retard pris sur l'une des tâches de ce chemin, entraîne du retard dans l'achèvement du projet (Chemin critique : A, B, D, E) [29].

## 4) Avantages

- Permet de déterminer la date de réalisation d'un projet.
- Permet d'identifier les marges existantes sur certaines tâches (avec une date de début au plus tôt et une date au plus tard).
- La date au plus tard de début d'une tâche, la date à ne pas dépasser sans retarder l'ensemble du projet.

## 5) Inconvénient

- Ne résout pas tous les problèmes, en particulier si l'on doit planifier des fabrications qui viennent en concurrence pour l'utilisation de certaines ressources.

**Remarque :** il y'a d'autre Méthodes de planification comme MPM<sup>5</sup> et PDM<sup>6</sup> et CPM<sup>7</sup>

<sup>5</sup>MPM: Méthode des Potentiels Métra

<sup>6</sup>PDM: Processus Diagram Methode.

<sup>7</sup>CPM: Critical path Methode.

## II.7.2. les approches classiques (Centralisées)

En utilisant des algorithmes de résolution exacts ou bien approchés.

### II.7.2.1. Les méthodes de résolution exactes

Une méthode exacte permet de trouver une solution optimale à un problème donné. Toutefois, ces méthodes peuvent devenir rapidement coûteuses en temps d'exécution, notamment pour les problèmes NP-difficiles. En effet, le temps de traitement et la complexité du problème sont généralement liés (plus c'est complexe, plus le temps d'exécution sera important) [28]. Ci-dessous, quelques méthodes exactes parmi les plus connues :

- Procédure par Séparation et Évaluation (La méthode de "Branch and Bound ")
- Programmation dynamique
- Programmation par contraintes.
- Programmation linéaire.

#### 1) Branch-and-Bound

L'algorithme Branch-and-Bound (exploration par partitionnement et évaluation de bornes) est une autre méthode conçue pour éviter le calcul exhaustif de l'espace de recherche. Le problème initial est séparé en plusieurs sous-problèmes (en général sans recouvrement mutuel) qui à leurs tours sont également partitionnés. Cela permet de construire un arbre de recherche, dont les branches seront successivement éliminées, en démontrant que la solution d'un sous-problème ne peut être meilleure (ou au moins aussi bonne) que la meilleure solution trouvée jusque-là.

Afin de tester la qualité d'une branche, des limites (en anglais « bounds ») sont calculées selon une méthode de relaxation, qui consiste à construire une nouvelle fonction à optimiser Cette fonction englobe toutes les solutions faisables du problème initial (approximation extérieure) et sa fonction objective est inférieure à la fonction objective initiale (sous-estimation). Si le problème relaxé ne peut pas être résolu, le problème initial n'a pas de solution non plus et par conséquent, la branche est élaguée. Si par contre le problème possède une solution, celle-ci peut être considérée comme la nouvelle borne inférieure de la fonction objective. Si cette borne n'est pas meilleure ou aussi bonne que la meilleure solution déjà trouvée (stockée dans une liste des meilleurs résultats) on peut en conclure que le sous-problème n'apporte pas d'amélioration au problème et la branche est également éliminée [30].

#### 2) La programmation dynamique

Introduite par Bellmann en 1955, cette méthode consiste à rechercher en premier temps les Solutions optimales des sous-problèmes et construire par la suite la solution optimale globale Les solutions correspondantes aux sous-problèmes sont calculées une seule fois et stockées Dans des tables. Il existe deux versions algorithmiques: itérative et récursive [31].

### 3) Programmation par contraintes

se basent d'une part sur une relaxation successive du problème de RCPSP<sup>8</sup> à chacune de ses ressources et d'autre part sur la relaxation des contraintes de préséance à des fenêtres temporelles afin d'établir des conditions nécessaires d'existence d'une solution de durée inférieure à H. Ces techniques ont été présentées dans la littérature par Brackeret al. Pour Yedder-finding disjonctif par Dorndorf et al. pouredge-finding disjonctif et le raisonnement énergétique.

Baptiste et Le Pape pour Y edge-finding cumulatif. Baptiste et al. Ils ont mis à jour des dominances entre différentes règles d'ajustement applicables au RCPSP. Ils proposent aussi une étude expérimentale des règles plus complexes de Yedge-finding cumulatif et du raisonnement énergétique. La méthode de programmation par contraintes est présentée de façon détaillée par Baptiste et al. Les auteurs présentent les modèles, les méthodes de résolution et les applications industrielles des divers problèmes de planification et d'ordonnancement [32].

### 4) La programmation linéaire

C'est une méthode très utilisée pour la modélisation des problèmes d'ordonnancement et qui permette minimiser ou bien de maximiser une fonction objectif de variables, soumise à un ensemble de contraintes exprimées sous forme d'équations ou bien d'inéquations linéaires. L'algorithme du simplexe mis au point par Dantzig en 1947 permet de résoudre la plupart des problèmes de programmation linéaire, même si sa complexité théorique demeure exponentielle.

#### II.7.2.2. Les algorithmes approchés

Contrairement aux méthodes exactes qui donnent des solutions optimales, les méthodes approchées permettent d'obtenir des solutions satisfaisantes en un temps de calcul raisonnable [31]. Ces méthodes peuvent être classées en cinq catégories :

- a. Les méthodes par voisinage.
- b. Les méthodes par construction.
- c. Les méthodes par décomposition.
- d. Les méthodes par relaxation de contraintes.
- e. Les méthodes liées à l'intelligence artificielle.

L'objectif d'une méthode approchée est de obtenir une solution exécutable, prenant en considération la fonction objectif mais sans garantie d'optimalité. Son utilisation offre de multiples avantages par rapport à une méthode exacte [33], ont citons:

1. Elles sont plus simples et plus rapides à mettre en œuvre quand la qualité de la solution n'est pas trop importante.
2. Elles sont plus souples dans la résolution des problèmes réels. En pratique, il arrive souvent que l'on doive prendre en considération de nouvelles contraintes qu'on ne pouvait pas formuler dès le départ. Ceci peut être fatal pour une méthode exacte si les nouvelles contraintes changent les propriétés sur lesquelles s'appuyait la méthode.
3. Elles fournissent des solutions et des bornes qui peuvent être utiles dans la conception de méthodes exactes.

<sup>8</sup>RCPSP: Ressource Constrained Project Scheduling Problem.

### 1) Les heuristiques

Les heuristiques sont des méthodes empiriques qui donnent généralement de bons résultats sans pour autant être démontrables. Elles se basent sur des règles simplifiées pour optimiser un ou plusieurs critères. Le principe général de cette catégorie de méthodes est d'intégrer des stratégies de décision pour construire une solution proche de celle optimale tout en cherchant à avoir un temps de calcul raisonnable. Parmi ces stratégies, nous distinguons [24] :

- a) **FIFO**<sup>9</sup> : où la première tâche arrivée est la première à être ordonnancée.
- b) **SPT**<sup>10</sup> : où la tâche ayant le temps opératoire le plus court est traitée en premier.
- c) **LPT**<sup>11</sup> : où la tâche ayant le temps opératoire le plus important est traitée en premier.
- d) **EDD**<sup>12</sup> : où la tâche ayant la date due la plus petite est la plus prioritaire.

### 2) Les méta-heuristiques

Un méta heuristique est un processus de génération itérative qui permet de guider une heuristique subordonnée par la combinaison de divers concepts tels que l'exploitation et l'exploration de l'espace de recherche, ainsi que des stratégies d'apprentissage qui sont utilisées pour structurer efficacement l'information afin de déterminer des solutions d'excellente qualité. Les méta-heuristiques sont en général non-déterministes et ne donnent aucune garantie d'optimalité mais ils peuvent faire usage de l'expérience accumulée durant la recherche de l'optimum, pour mieux guider la suite de la procédure de recherche. On distingue les méta-heuristiques à solution unique parmi lesquelles on retrouve de nombreux algorithmes de recherche locale, et les méta-heuristiques à base de population [32].

Il y a deux types de méta heuristiques qui proposent des solutions

1. méta heuristiques à solution unique.
2. méta heuristiques à population de solutions.

#### ❖ méta heuristiques à solution unique

Les méthodes itératives à solution unique sont toutes basées sur un algorithme de recherche de voisinage qui commence avec une solution initiale, puis l'améliore pas à pas en choisissant une nouvelle solution dans son voisinage [24].

Nous présenterons ici les méthodes les plus utilisées et leur utilisation en extraction de connaissances : les méthodes de descente, le recuit simulé et la recherche tabou.

#### a) Le recuit simulé

Le recuit simulé est une méthode de recherche locale dont l'origine est attribuable aux résultats des travaux de Kirkpatrick et al. en 1983. Elle est inspirée d'un processus thermodynamique de refroidissement utilisé en métallurgie. Le processus consiste à refroidir graduellement un corps en fusion afin d'obtenir un corps solide dans un état d'énergie minimale. Le refroidissement se fait en plusieurs paliers de température. La longueur de ces paliers est choisie de telle manière que le corps atteigne l'équilibre thermique à chaque température. Par analogie avec ce processus physique de refroidissement, le recuit simulé suggère que la fonction objective  $f$  à minimiser soit considérée comme étant la fonction d'énergie, et qu'une solution  $s$  du problème soit

<sup>9</sup>FIFO: First In First Out.

<sup>10</sup>SPT: Shortest Processing Time.

<sup>11</sup>LPT: Longest Processing Time.

<sup>12</sup>EDD: Earliest Due Date.

considérée comme étant un état du système physique. Dans ce cas, la valeur  $f(s)$  exprime l'énergie associée à l'état  $s$  [34].

### b) La recherche Tabou (Tabu Search)

La recherche tabou se comporte comme toute méthode de voisinage et améliore à chaque étape la valeur objectif. La recherche tabou est une méthode de recherche locale combinée avec un ensemble de techniques permettant d'éviter d'être piégé dans un minimum local ou la répétition d'un cycle. Lorsque l'on atteint par contre un minimum local, on se permet de passer au moins mauvais des voisins. Se comporte comme toute méthode de voisinage et améliore à chaque étape la valeur objectif. Le principe de cette méthode d'examiner le voisinage de la solution courante à chaque itération. L'algorithme enregistre la meilleure solution parmi les voisinages. Même si elle moins bonne que la solution courante L'acceptation des solution moins performante que la solution courante permet d'éviter de tomber dans optimum local Pour éviter de tourner dans un cercle enter plusieurs solution l'algorithme interdit le passage par des solution récemment visitées[25].

### c) les méthodes de descente

La méthode descente utilise une recherche locale simple consiste à partir d'une solution  $s$  appartient  $S$  et à choisir une solution  $S'$  dans le voisinage de  $s$  telle que  $f(s') < f(s)$  c'est ta dire amélioré la recherche dans le voisinage.

### ❖ Les méta-heuristiques à population de solutions

Les méthodes d'optimisation à population de solutions améliorent, Avec répétition continue un ensemble de solutions. L'intérêt de ces méthodes est d'utiliser la population comme facteur de diversité [24].

### a) Algorithmes génétique

Les algorithmes génétiques fournissent des solutions aux problèmes n'ayant pas de solutions calculables en temps raisonnable de façon analytique ou algorithmique [12]. Ils ont été adaptés à l'optimisation par John Holland [13], également les travaux de David Goldberg ont largement contribué à les enrichir. Selon cette méthode, des milliers de solutions plus ou moins bonnes sont créés au hasard puis sont soumises à un procédé d'évaluation de la pertinence de la solution imitant l'évolution des espèces : les plus "adaptés", c'est-à-dire les solutions au problème qui sont les plus optimales survivent davantage que celles qui le sont moins et la population évolue par générations successives en croisant les meilleures solutions entre elles et en les faisant muter, puis en relançant ce procédé un certain nombre de fois afin d'essayer de tendre vers la solution optimale.

Les avantages des algorithmes génétiques:

Les algorithmes génétiques ont plusieurs avantages citons:

- a) Ils sont adaptables à plusieurs types de problèmes.
- b) Puissants
- c) Facile à mettre en œuvre.
- d) Hybridation facile.
- e) C'est facile à mettre en parallèle.

Les inconvénients des algorithmes génétiques

Parmi les inconvénients des algorithmes génétiques, nous trouvons :

- a) Il n'y a aucune garantie de rapprochement.
- b) Le temps de calcul est important (si la taille de la population est grande).

#### b) L'optimisation par colonie de fourmi

La méta heuristique d'optimisation par colonies de fourmis a été initialement introduit par Dorigo, Maniezzo et Colorni et a été inspirée par les études sur le comportement des fourmis réelles effectuées par Deneubourg et al. A l'origine, l'optimisation par colonie de fourmis a été conçue pour résoudre le problème du voyageur de commerce en proposant le premier algorithme ACO : 'Ant System' (AS). Par la suite, un nombre considérable d'applications de ACO a été proposé telles que l'affectation quadratique, le routage des véhicules, le problème de satisfaction de contraintes [34].

#### • La Comparaison de la méthode exacte et la méthode approchée

Le tableau suivant présente les différences points entre la méthode exacte et approche [1].

Exploration de S	Systématique (intelligente)	Partielle (astucieuse)
Méthode/solution	Exacte	Approchée
Caractéristique	Optimalité	Approximation
Preuve	Formelle	Empirique (£-appro.)
Aspect	Déterministe	Stochastique (guidée)
Complexité	O (2 <sup>n</sup> ) (souvent)	O (n <sup>k</sup> )
Handicap	Explosion combinatoire	Convergence précoce
Adaptée aux POC	De petites tailles	NP-difficiles de grandes tailles

Tableau II. 2: La différence entre méthode exacte et méthode approchée.

#### II.7.3. Les approches robustes

Comme nous l'avons déjà souligné précédemment, les données du problème d'ordonnancement ne sont en pratique pas toujours connues à l'avance, en plus de l'environnement de production qui est sujet à des perturbations. Ainsi et pour prendre en compte cet aspect très important, plusieurs approches dites robuste sont été développées dans la littérature.

Notons que l'objectif recherché n'est pas l'optimalité de la solution trouvée, mais l'obligation de fournir rapidement une solution sans trop perturber le fonctionnement du système. Les auteurs proposent une classification des approches robustes selon la manière dont l'incertitude est traitée (hors-ligne ou en-ligne). Dans ce qui suit, nous donnons un bref aperçu des trois principales approches existantes [31]:

- Les approches proactives
- Les approches réactives
- Les approches proactives-réactives.

## II.7.4. Ordonnement distribué

Ces dernières années, l'apparition de l'intelligence artificielle distribuée (IAD), des systèmes de production intelligents, des systèmes de production holoniques, en plus du progrès technologique réalisé dans le domaine des réseaux informatiques, ont montré l'intérêt de la Distribution des décisions et de la coopération. Dans le contexte d'un environnement d'application dynamique et incertain, une solution distribuée est tout à fait appropriée pour résoudre un problème d'ordonnement de la production compte tenu de la nature distribuée de ce dernier. Il s'agit en effet d'aborder ce problème selon plusieurs vues contrairement aux approches centralisées [31].

### II.7.4.1 Mécanismes de coopération

Il existe dans la littérature plusieurs définitions du terme coopération du fait qu'il soit largement utilisé dans plusieurs domaines et disciplines (économie, sociologie du travail, etc.).

Dans le cadre des systèmes de production, la coopération vise à mettre en place des mécanismes d'interaction entre les entités ou bien les centres de décision qui composent un atelier de production. La coopération est définie comme étant «le support de la mise en œuvre de la décision entre plusieurs centres de décisions».

En effet, il s'agit d'exploiter les interdépendances des différentes entités afin d'aboutir à une solution globale qui dépend de la forme de coopération choisie coordination ou bien coopération.

La coordination est définie dans comme «une mise en cohérence des actions des acteurs qui entreprennent différentes activités en réduisant au minimum les coûts de division du travail». La gestion et la synchronisation des actions des différentes entités est faite par un superviseur qui assure la cohérence des décisions par rapport au fonctionnement global du groupe.

La coopération quant à elle, implique la participation des acteurs dans l'exécution d'une action pour aboutir à un résultat final mais n'entraîne pas de prise de décisions collective. Lorsque les différents acteurs procèdent à une prise de décision en coopération les uns avec les autres, on parlera alors de codécision. Cette dernière résulte de la négociation (coopération pour la première fois) ou bien la renégociation (remise en cause d'une décision déjà prise) entre les acteurs du système. Différentes phases de négociation, coordination et renégociation sont engagées lors d'un processus de coopération, et cela se fait selon un protocole bien défini. Il existe dans la littérature plusieurs protocoles de négociation, nous en citons quelques-uns :

- Les systèmes de vote: ce genre de système est utilisé pour élire une alternative parmi les différentes alternatives possibles en recueillant les votes pour et les votes contre cette alternative. L'alternative qui satisfait au mieux la population est appelée choix social.
- Ce dernier peut être réalisé en utilisant différentes procédures telles que: le scrutin uninominal majoritaire à un tour, la méthode de Broda, la dictature, etc.
- Les enchères: dans ce cas, la négociation prend la forme d'une vente aux enchères où une ressource propose un objet à vendre et les autres ressources voudront l'acheter selon le type d'enchère adopté.
- Négociation à base du protocole Contract-Net: proposé par Smith en 1980, le Contract-Net est un protocole de haut niveau qui assure la communication entre les entités d'un résolveur de problèmes distribué en facilitant l'exécution de tâches coopératives avec une communication efficace [31]. La négociation est caractérisée par:

- a. Un processus local qui n'implique pas un contrôle centralisé.
- b. Un échange d'informations qui a lieu dans les deux sens.
- c. Chaque partie évalue les informations de sa propre perspective.
- d. L'accord final est réalisé par une sélection mutuelle.

### II.7.4.2 Systèmes d'ordonnement distribués

Une distribution induit naturellement à une décomposition du problème initiale sous-problèmes partiels. Cette décomposition est fortement influencée par la modélisation initiale du problème global, et constitue en elle-même dans certains cas le problème principal. Cela est dû à la variété démodés de distribution qui existence qui complique la tâche de classification des systèmes d'ordonnement distribués. Dans l'auteur a identifié les principales classes les systèmes d'ordonnement distribués: (i) Ordonnement par l'aide à la décision, (ii) par coordination des ressources et (iii) et par coopération des ressources [31].

#### ❖ Ordonnement par l'aide à la décision

Dans ce type de systèmes qui fait partie des Systèmes Interactifs d'Aide à la Décision (SIAD), l'utilisateur humain est intégré au processus de prise de décision afin d'assouplir la rigidité des systèmes d'ordonnement classiques. La résolution du problème dans ce cas est distribuée entre deux acteurs: un acteur logiciel et au moins un acteur humain. Le principal inconvénient de ces systèmes réside dans le non prise en compte de l'impact des décisions de l'utilisateur sur l'atelier de production. Plusieurs travaux dans la littérature abordent ce genre d'ordonnement [31].

#### ❖ Ordonnement par coordination des ressources

Dans ce cas, le problème étant décomposé en sous-problèmes, chaque centre de décision prend en charge une partie du problème global. Ces centres sont interdépendants, et une entité

Unique appelée superviseur assure la cohérence des décisions prises localement par les différents centres de décision afin de maintenir une performance globale satisfaisante. Notons dans le cas où une perturbation apparaît au niveau d'une ressource, le centre de décision correspondant informe le superviseur afin qu'il puisse interagir avec les autres centres de décision concernés afin d'absorber la perturbation en établissant un nouveau ordonnement

Global. Ce type d'organisation hiérarchique limite ainsi l'autonomie des centres dans leur capacité de prise de décision [31].

#### ❖ Ordonnement par coopération des ressources

Contrairement à l'ordonnement basé sur la coordination qui n'autorise qu'une coopération verticale pour la prise de décision, l'ordonnement coopératif s'appuie sur une certaine autonomie des acteurs et sur la capacité de ces derniers à négocier pour trouver la solution globale du problème.

Dans ce mode d'organisation, l'objectif est d'arriver à limiter les conflits résultants des interdépendances entre les différentes ressources tout en respectant les objectifs locaux. Le cas d'un problème d'atelier dans un contexte multi-acteurs pour la l'ordonnement distribué a été étudié dans Le principe de base de l'approche de résolution consistait à assimiler les ressources à des centres de décision, possédant chacune une autonomie décisionnelle (ordonnement robuste à une machine), et prenant chacune en charge l'exécution d'un sous-ensemble d'opérations. L'ordonnement est considéré sous forme d'une fonction distribuée où la solution globale résulte d'une coopération entre les différents acteurs qui se communiquent des intervalles de livraison/consommation des

opérations que chacun gère. L'objectif recherché est la cohérence des décisions prises collectivement [31].

## II.8. Ordonnement à l'aide de systèmes multi-agents

Dans ce type d'approche, des entités logicielles indépendantes, appelées « agents » coopèrent pour trouver une solution à un problème d'ordonnement. Ces agents sont dotés de la capacité à percevoir leur environnement (par exemple l'espace de recherche du problème d'ordonnement), d'une certaine intelligence pour le comprendre et de moyens de communication. Les avantages de l'approche multi-agents pour traiter le problème de l'ordonnement sont [26] :

- a) La robustesse- Les agents peuvent réagir dynamiquement aux perturbations rencontrées.
- b) La possibilité de mise à l'échelle - Les systèmes multi-agents, par leur nature distribuée, se prêtent bien à des problèmes de taille variable.
- c) Le parallélisme- Dans certaines situations, les agents, grâce à leur nombre élevé peuvent traiter un problème vaste ou un problème distribué plus rapidement qu'une unité monolithique.
- d) L'accès aux données distribuées- Les agents peuvent accéder à des informations géographiquement distribuées ou même cachées, qui ne pourraient pas être collectionnées par un ordonnanceur central.
- e) La capacité de confidentialité- Une partie des données d'un problème distribué peut être soumise à des contraintes de confidentialité, ce qui rend les approches centralisées inapplicables. Un agent peut analyser localement une partie confidentielle du problème et ne communiquer que les données « neutres », c'est à dire non-confidentielles ou anonymisées.

## II.9. Conclusion

Dans ce chapitre on a étudié les différentes méthodes de résolution des problèmes d'ordonnement. On distingue des méthodes sont : les Modèles de planification de tâches d'un projet (PERT, GANTT..) et les approches classiques (Les méthodes exactes les méthodes approches). Les approches robustes et Ordonnement distribué.

Dans le chapitre suivant nous présentons l'implémentation de notre application.

## Chapitre III Les Systèmes Multi-Agents

## Chapitre III : Les Systèmes Multi-Agents

### III.1 Introduction

Dans le monde réel, il existe une très grande variante de système, et certains systèmes requièrent d'utiliser plusieurs composants qui subies des changements dans son système. Par conséquent, les systèmes Multi-Agents (SMA) sont très bien adaptés pour ces types de situation. La technologie des agents offre une nouvelle méthode pour analyser concevoir et implémente des applications car ils font partie du domaine IAD.

Les SMA ont été utilisés avec succès dans de nombreux domaines : les bases de données coopératives distribuées et du génie logiciel etc...

Ce chapitre commence par définir ce qu'est un agent, ses différents types et son architecture, puis s'intéresse aux systèmes multi-agent (SMA), leurs propriétés pour aborder par la suite les différentes interactions et la communication entre les agents.

A la fin, un aperçu des différents domaines d'applications des SMA est donné.

### III.2. Positionnement historique

Les systèmes multi-agents se positionnent au carrefour de la programmation (ce sont des logiciels), de l'intelligence artificielle (leur autonomie de décision), et des systèmes répartis (leur décentralisation).

Historiquement, on peut replacer le concept d'agent et de système multi-agents dans l'histoire de l'intelligence artificielle et de manière duale dans l'histoire de la programmation. La notion d'agent est de fait à la base des débuts de l'intelligence artificielle (**IA**). Mais cette discipline s'est focalisée sur la modélisation des capacités intelligentes d'une unique entité pour résoudre des problèmes. Il en a résulté une première génération de programmes informatiques évolués, tels les systèmes experts.

Mais, même restreint à un domaine spécialisé (domaine expert), Un tel système expert était censé résoudre tout seul les problèmes de manière autarcique.

L'accent a donc été mis progressivement à partir de la fin des années 70 sur une résolution distribuée de problèmes complexes, par coordination d'un certain nombre d'agents, ce que l'on a alors commencé à appeler « systèmes multi-agents ». On utilise également le terme quasiment équivalent « **IA distribuée** » (avec son acronyme **IAD**, en anglais **DAI**<sup>13</sup>) pour bien montrer l'opposition à l'**IA** classique autarcique et **IA** centralisée.

Il est important pour la suite de pouvoir définir certains termes qui reviendront très souvent dans ce mémoire à savoir Agent, Système multi-agents, etc. Il est important de notifier que les définitions que nous présentons ici s'inscrivent dans le cadre de notre travail (Approche multi-agents pour l'ordonnancement des tâche de demande de congé) et peuvent donc avoir d'autres connotations dans d'autres domaines.

### III.3. Qu'est-ce qu'un agent ?

Il y a un nombre important d'ouvrages offrant des définitions des agents et des systèmes multi-agents. Les difficultés sont semblables en quelque sorte à celles rencontrées par les scientifiques qui ont essayé de définir la notion d'intelligence artificielle. Pourquoi a-t-il été si difficile de définir l'intelligence artificielle et pourquoi est-il si difficile de définir les systèmes d'agents, quand d'autres concepts de

<sup>13</sup> **DAI**: Distributed Artificial Intelligence.

l'informatique, tels que l'objet et l'orienté objet, le calcul distribué, etc..., n'ont pas rencontré une si grande résistance à être définis.

Une réponse possible est que la notion d'agent, ainsi que la notion d'intelligence artificielle, ont émergé des humains et de la société humaine. Il est évidemment difficile de modéliser ou de simuler le comportement spécifique humain au moyen de programmes informatiques.

Il y'a plus de 30 ans, les scientifiques ont essayé d'écrire des programmes pour mimer le comportement humain intelligent, dont le but était de créer un système artificiel ayant les mêmes capacités qu'une personne intelligente.

Il n'y a pas une définition acceptée en unanimité pour la notion d'agent. Dans ce qui suit, on présente les définitions les plus importantes.

- Un agent est une entité qui perçoit son environnement et agit sur celui-ci [37].
- Un agent est un système informatique, situé dans un environnement, et qui agit d'une façon autonome pour atteindre les objectifs pour lesquels il a été conçu [38].
- Un agent est une entité qui fonctionne continuellement et de manière autonome dans un environnement où d'autres processus se déroulent et d'autres agents existent [39].
- Un agent est une entité autonome, réelle ou abstraite, qui est capable d'agir sur elle-même et sur son environnement, qui dans un univers multi-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 [40].

Nous pouvons identifier deux tendances principales en comparant les définitions données ci-dessus pour les agents et les agences. Quelques chercheurs considèrent que nous pouvons définir un agent en isolation, alors que d'autres considèrent les agents principalement comme entités agissant dans une société d'autres agents, à savoir le paradigme des systèmes multi-agents.

Les deux tendances ont déjà donné des résultats. C'est vrai qu'on pense que c'est le paradigme des SMA<sup>14</sup> qui va s'imposer comme prépondérant car il est plutôt difficile de compter qu'un agent existe seulement comme une entité pour lui seul et ne rencontrera pas d'autres agents (soit artificiels ou humains) dans son environnement.

Les agents personnels, ou les agents d'information, qui ne sont pas censés principalement travailler collectivement pour résoudre des problèmes, auront certainement beaucoup à gagner s'ils agissent par interaction avec d'autres agents et bientôt, avec la diffusion large de la technologie multi-agents, ils ne pourront pas réaliser leurs tâches en isolation. Par conséquent, les chercheurs considèrent la dimension sociale d'un agent comme une de ses caractéristiques essentielles.

---

<sup>14</sup> SMA: Système Multi-agents.

D'autres considèrent la mobilité en tant qu'une des caractéristiques des agents informatiques. On a des réserves sur cette opinion parce que la mobilité est un aspect relié principalement à la mise en place ou à la réalisation des agents, soient-ils logiciels (agents mobiles sur la toile) ou matériels (robots qui se déplacent dans le monde physique) et peut être comprise dans la capacité des agents d'interagir avec l'environnement.

En partant de l'ouvrage de [38], et des définitions citées, on peut identifier les caractéristiques suivantes pour la notion d'agent:

- Situé – l'agent est capable d'agir sur son environnement à partir des entrées sensorielles qu'il reçoit de ce même environnement;
- Autonome – l'agent est capable d'agir sans l'intervention d'un tiers (humain ou agent) et contrôle ses propres actions ainsi que son état interne;
- Proactif – l'agent doit exhiber un comportement proactif et opportuniste, tout en étant capable de prendre l'initiative au bon moment;
- Capable de répondre à temps – l'agent doit être capable de percevoir son environnement et d'élaborer une réponse dans le temps requise;
- Social – l'agent doit être capable d'interagir avec d'autres agents (logiciels ou humains) afin d'accomplir des tâches ou aider ces agents à accomplir les leurs.

Bien que presque toutes les caractéristiques ci-dessus des agents puissent être considérées en tant que partage de quelque chose avec le comportement intelligent, les chercheurs ont essayé de donner une différence nette entre les agents logiciels (software) et les agents logiciels intelligents, glissant dans le monde des agents la différence si recherchée entre les programmes et les programmes intelligents. D'un point de vue, il est clair que, si dans la conception d'un système agent ou multi-agents, nous utilisons des méthodes et des techniques spécifiques à l'intelligence artificielle, alors ces agents peuvent être considérés comme étant intelligents, évidemment dans le sens de l'intelligence artificielle.

L'agent peut également posséder d'autres propriétés (de second ordre) notamment la mobilité, la faculté d'apprentissage, etc. Il est important de mentionner que celles-ci ne sont pas essentielles pour l'agent. En combinant diverses qualités propres aux agents, on peut définir différents types d'agents comme l'indique la figure (Figure III 1) suivant :

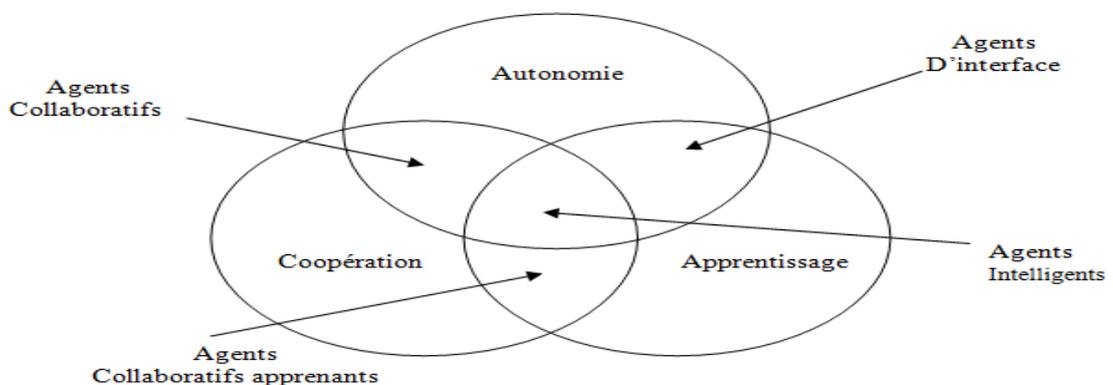


Figure III. 1: Différents types d'agents.

Par exemple, dans la figure (Figure III.1), les propriétés telles que la coopération l'apprentissage et l'autonomie permettent de définir quatre types particuliers d'agents :

- ◆ les agents intelligents.
- ◆ les agents collaboratifs.
- ◆ les agents collaboratifs apprenants.
- ◆ les agents d'interface.

### III.3.1. L'architecture d'agent

La représentation d'agent en tant que boîte noire (black box) est considérée comme une architecture minimale, commune à tous les domaines de recherche d'IA [41,42].

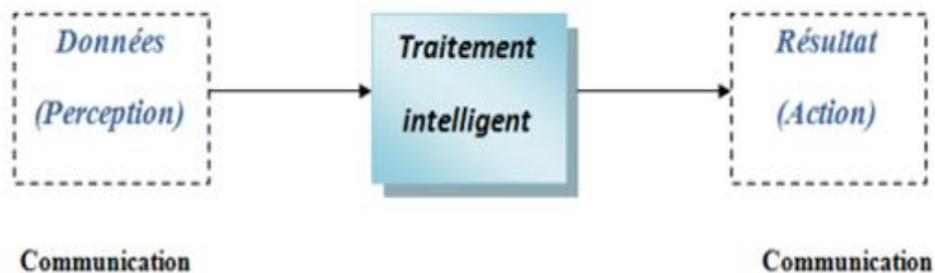


Figure III. 2: L'architectureur minimale d'un agent.

L'architecture minimale (Figure III. 2) est composée de trois modules : le module de perception, le module d'action et le module de traitement intelligent. Ce dernier détermine le comportement d'agent qui peut ainsi analyser les données reçues et construire la réponse sous la forme d'action ou de message.

En général, les architectures existantes sont divisées en trois familles principales : les agents cognitifs (délibératifs), les agents réactifs et les agents hybrides.

### III.3.2. Une comparaison avec les objets

Bien des programmeurs croient encore que les agents ne sont en réalité que de simples objets, au sens attribué à ce terme dans le domaine de la programmation. En fait un objet est une certaine modélisation d'une entité réelle ou abstraite. Cet objet encapsule un état et il est possible de modifier cet état en invoquant différentes méthodes faisant aussi partie de l'objet. Les agents sont aussi des entités informatiques qui possèdent un état interne (privé), qui sont capables d'agir et de communiquer par échanges de messages. Ce qui différencie les objets des agents, c'est que ces derniers reposent sur un concept très fort d'autonomie leur permettant d'agir de manière très flexible. La négociation, processus par lequel différents groupes arrivent à un accord est un concept étranger au monde objet. La démarche en vue d'une négociation requiert un comportement flexible afin de pouvoir faire certaines concessions de part et d'autre.

De plus, les agents contrôlent d'une certaine façon leur état interne. Par exemple si un objet **O1** invoque une méthode **M1** sur l'objet **O2**, cette méthode sera exécutée si aucune erreur ne survient. Cependant, si un agent **A1** effectue une requête **R1** à l'agent **A2**, c'est ce dernier qui décidera si la requête sera effectuée ou non. Toute la différence se situe dans les termes invocation et requête.

Le tableau ci-dessous (Tableau III.1) présente un récapitulatif des éléments de base de l'approche orientée objet versus l'approche orientée agent présenté par **Shoham**<sup>15</sup> [39].

Propriétés	AOO	AOA
Elément de base	Objet	Agent
Paramètre définissant l'élément de base	Sans contraintes	Croyances, engagements, capacités, choix
Type d'exécution	Envoie et réception de messages	Envoie et réception de messages
Type de messages	Sans contraintes	Actes du langage
Contraintes sur les méthodes	Sans contraintes	Honnête, consistant

**Tableau III. 1:** Approche orientée objet (AOO) versus approche orientée agent (AOA).

### III.3.3. Classe d'agent

De manière générale, il est possible de classer un agent selon les trois principaux points suivants :

#### a)- Selon leur nature

- Agents réactifs : Ce sont des agents qui n'ont pas une représentation explicite de leur environnement, ni de mémoire de leur passé, ni de but explicite et leur comportement est de type stimulus réponse.
- Agents délibératifs : Ce sont des agents qui possèdent une représentation explicite de leur environnement. Ils se basent sur la connaissance qu'ils ont de leur environnement et leur habileté à raisonner sur leurs connaissances. Les agents dirigés par leurs buts ou par une fonction d'utilité font partie de ce groupe.
- Agents hybrides : De façon habituelle, un agent n'est pas complètement de type réactif ni de type délibératif. C'est plutôt une combinaison de ces deux approches. Dans certaines circonstances, un agent doit agir très rapidement alors que dans d'autres il aura la possibilité de prendre plus de temps pour avoir un processus délibératif de meilleure qualité. Dans ce cas, les agents sont dits hybrides.

#### b)- Selon l'utilisation

- Agents collaboratifs : Ces agents ont des habiletés de coopération. Un regroupement de ces agents permet, entre autres, de réduire un problème complexe en sous-problèmes moins complexes, c'est le cas dans le domaine médical.
- Agents d'interface : Ces agents collaborent avec l'utilisateur pour effectuer certaines tâches.
- Agents pour la recherche d'informations : Ces agents effectuent, en premier lieu, une recherche d'informations parmi une collection de données et, en second lieu, procèdent à une analyse des informations utiles trouvées afin de découvrir de nouvelles connaissances.

<sup>15</sup> **Yoav Shoham** : Professeur à Université de Stamford.

– Agents pour le commerce électronique : La montée de l'Internet a bien entendu créé de nouvelles nécessités. Les agents issus de cette tendance permettent la promotion, la vente ainsi que l'achat de produits et de services par l'entremise des réseaux informatiques, etc.

### c)- Selon la technologie employée

– Agents stationnaires : Il s'agit du cas où l'agent s'exécute toujours sur la même machine.

– Agents mobiles : Ces agents s'exécutent sur différentes machines en se promenant d'un hôte à l'autre. Typiquement, ils suivent ce que l'on appelle un "itinéraire".

Le plus grand avantage réside dans le fait qu'ils réduisent la charge des réseaux en se rendant directement sur la machine "utile" pour y effectuer les actions de manière locale.

D'un point de vue général, les chercheurs aimeraient que les agents soient des composants logiciels réutilisables qui permettraient d'accéder et/ou contrôler des services et des ressources informatiques. Ces agents seraient les unités de base des applications informatiques et seraient organisés en réseau dans une optique de collaboration.

L'ensemble des entités (Agents) en interaction et en vue de réaliser un but commun forme ce qu'on appelle un système multi-agents, Alors la question qui se pose est :

## III.4. Qu'est-ce qu'un système multi-agents (SMA) ?

### III.4.1 Les systèmes multi-agents

Les systèmes multi-agents font partie de l'Intelligence Artificielle Distribuée (IAD), qui, à la différence d'IA classique, s'intéresse aux comportements intelligents résultant de l'activité coopérative de plusieurs agents [43].

Bien qu'il soit difficile de donner une définition générale pour l'expression « Système Multi-Agents "SMA" », les différentes caractérisations proposées par [44] [45], [46], [47],[48] et autres permettent de dégager des idées fondamentales communes. La définition, mentionnée ci-dessous, est conforme à la plupart de ces idées.

### III.4.2 Définition: Système multi-agents (SMA)

Le système multi-agents est considéré comme un système distribué composé d'un certain nombre d'entités autonomes<sup>16</sup> (les agents), qui travaillent selon les modes complexes d'interaction, pour réaliser leurs propres buts et par-là même atteindre l'objectif global désiré. Les agents peuvent interagir en communiquant directement entre eux ou par l'intermédiaire d'un autre agent ou en agissant sur leur environnement.

En général, dans les systèmes multi-agents, deux types d'agents sont identifiés : les agents IAD<sup>17</sup> et les agents **mobiles**. Les agents IAD sont capables de communiquer et de coopérer à l'aide de protocoles basés sur les réseaux contractuels et les actes de

<sup>16</sup> L'**autonomie** signifie que l'entité (l'agent) est capable de travailler sans l'intervention d'un humain ou des autres agents, et de contrôler ses actions ainsi que son état interne

<sup>17</sup> IAD : Intelligence Artificiel Distribué.

langage. Les agents **mobiles** peuvent se déplacer de site en site dans le réseau pour accomplir des tâches spécifiques.

En général, ils peuvent avoir deux types de mobilité :

- ✓ L'exécution distante:  
L'agent (le programme et les données) est transféré sur le site distant, où il est exécuté.
- ✓ La migration:  
Pendant son exécution, l'agent se déplace vers un autre site afin d'accomplir progressivement sa tâche. Autrement dit, l'agent est capable de suspendre son exécution sur un site, de se déplacer vers un autre en transportant le code, les données et l'état d'exécution, et de reprendre l'exécution depuis le point de suspension.

D'après la définition du système multi-agents, proposée ci-dessus, pour qu'un système soit considéré comme SMA, il est nécessaire de satisfaire certains critères :

- Disposer d'un ensemble des agents autonomes, fonctionnant en parallèle et cherchant à satisfaire un but ;
- Les agents doivent posséder un mécanisme d'interaction de haut niveau indépendant de problème à résoudre. Ils peuvent utiliser les langages de communication d'agents (ACL), par exemple, KQML<sup>18</sup>, FIPA ACL, etc...
- L'agent doit percevoir une partie de son environnement qui peut être le monde physique, d'autres agents, le réseau Internet, etc. Il doit répondre dans un délai acceptable.

On peut déduire, par conséquent, que la distribution modulaire d'une application ne constitue pas forcément un système multi-agents. En effet, les modules n'ont pas de buts à atteindre ou la fonction de satisfaction. Leurs mécanismes d'interactions sont de bas niveau : appels de procédures, etc. En ce qui concerne les agents, ils reçoivent des messages qui peuvent être des demandes d'exécution, mais, aussi, des requêtes d'information sur leurs capacités. Ils tentent de satisfaire les objectifs, ils disposent pour cela d'une autonomie supplémentaire par rapport aux objectifs. L'agent peut refuser une tâche donnée à cause de son manque de compétences ou de sa trop grande occupation.

### III.4.3. Les interactions et les agents

L'interaction est une notion importante dans les systèmes multi-agents. Au dire de **J.Ferber [45]**, l'interaction permet d'avoir une relation dynamique entre deux ou plusieurs agents par le biais d'actions réciproques. Les situations d'interactions sont diverses (l'aide d'un robot à un autre, l'échange de données entre deux serveurs, etc...).

Pour un agent, interagir avec un autre agent constitue la source de sa puissance et l'origine de ses problèmes **[49]**. En effet, seulement en coopérant les uns avec les autres les agents peuvent accomplir leurs tâches collectives. D'autre part, à cause de leurs interactions avec les autres agents, ils doivent coordonner leurs actions et résoudre des conflits.

Traiter le problème d'interaction revient non seulement à décrire les mécanismes permettant aux agents d'interagir, mais aussi à analyser et à concevoir les différentes formes d'interaction, utilisées par les agents pour accomplir leurs tâches et satisfaire

<sup>18</sup> **KQML: Knowledge and Query Manipulation Language.**

leurs buts. Nous présentons le concept d'une de ces formes - la communication - sur laquelle repose toute l'interaction.

### **III.4.3.1. La communication**

La communication est essentielle dans la résolution coopérative des problèmes [50] [46]. Elle permet de synchroniser les actions des agents et de résoudre les conflits des ressources par la négociation. D'après [51], la communication définit l'ensemble des processus physiques et psychologiques par lesquels s'effectue l'opération de mise en relation d'un agent émetteur avec un ou plusieurs agents récepteurs, dans l'intention d'atteindre les objectifs prévus. Les processus physiques décrivent les mécanismes d'exécution des actions, par exemple, l'envoi et la réception de messages. Les processus psychologiques désignent les changements opérés par la communication sur les buts et les croyances des agents.

En général, les actions de communication entre les agents sont considérées comme les actions d'échange d'information.

Il existe plusieurs types de communication, à savoir, la communication par tableau noir, la communication par partage d'informations et la communication par envoi des messages qu'on va détailler dans ce qui suit.

#### **III.4.3.1.1. La communication par envoi des messages**

Ce type de communication permet aux agents d'envoyer leurs messages directement aux destinataires par les mécanismes spécifiés (les canaux ou les ports). Il existe trois types de messages : les questions, les réponses et les informations. Au niveau protocolaire, un envoi de message peut être synchrone (un agent émetteur attend la réponse de son récepteur) et asynchrone (un agent émetteur peut agir immédiatement après avoir placé son message dans une file d'attente).

La communication par l'envoi de messages peut être organisée suivant trois formes différentes :

- La communication point à point.

L'agent émetteur connaît l'agent destinataire et lui transmet directement son message. L'agent destinataire est le seul à recevoir le message envoyé.

- La distribution généralisée.

Il s'agit d'un envoi d'un même message à tous les agents du système multi-agents. L'agent émetteur ne connaît pas forcément les destinataires du message envoyé.

- La distribution restreinte.

Il s'agit d'un envoi d'un même message à un certain groupe d'agents du système multi-agents. L'agent émetteur ne connaît pas forcément tous les destinataires, mais il doit être capable de les atteindre en s'appuyant, soit sur leurs caractéristiques, soit sur la notion de groupe auquel ils appartiennent.

### III.4.4. Les langages de communication

Le concept de communication permet de réaliser les échanges locaux d'information entre des agents. Le problème est alors de combiner ces actions de communication avec les compétences et les connaissances des agents, pour obtenir un comportement collectif entre ces derniers. Dans les systèmes multi-agents, la problématique susmentionnée est étudiée à travers le concept de l'interaction des agents. Cette dernière permet d'avoir une relation dynamique entre deux ou plusieurs agents par le biais d'actions réciproques [46]. La question qui se pose alors, est comment réaliser les interactions des agents ?

D'après [52], pour que les agents puissent interagir de manière efficace, ils doivent posséder un langage de communication commun, leur permettant de se comprendre ainsi que de s'échanger des informations et des connaissances.

Avant d'aborder les langages de communication existants, nous présentons brièvement la théorie des actes de langage, considérée en intelligence artificielle comme un modèle général de communication entre les agents.

#### III.4.4.1. La théorie des actes de langage

La théorie des actes de langage (« **Speech Act Theory** »), constitue un fondement théorique de la communication, basée sur l'idée suivante : « Lorsqu'on parle, on effectue des actions »<sup>19</sup> [14]. Un acte de langage définit un message, qui contient l'affirmation positive ou négative, et provoque les changements de l'environnement (voir, l'annexe C).

Chaque acte de langage comprend trois composants :

- Le composant **locutoire**, qui décrit l'expression d'un message ;
- Le composant **illocutoire**, qui définit les intentions de l'émetteur, associées implicitement au message ;
- Le composant **perlocutoire**, qui décrit les effets d'un acte de langage sur l'environnement.

Dans la théorie des actes de langage, les intentions des émetteurs sont identifiées en utilisant les verbes performatifs. Ces derniers sont classifiés en plusieurs catégories : **les affirmatifs** (informer), **les directifs** (ordonner), **les promissifs** (promettre), **les déclaratifs** (déclarer) et **les expressifs** (exprimer).

D'après cette théorie, chaque acte de langage multi-agents peut être décrit sous la forme d'un message, dont le type est défini par le verbe performatif, tel que « Request » ou « Inform ». Le contenu de ce message est décrit en utilisant les langages de communication (ex: KQML, FIPA ACL).

##### III.4.4.1.1. KQML

KQML (« **K**nowledge and **Q**uery **M**anipulation **L**anguage ») [53] [54] est une approche basée sur les actes de langages, qui permet de réaliser les interactions des agents tenant en compte la diversité des langages de communication.

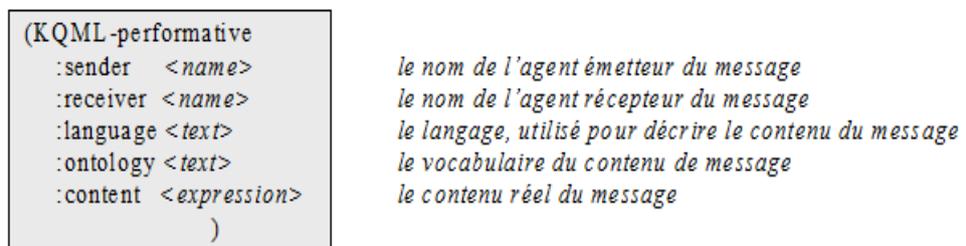
La communication est considérée comme un ensemble d'échanges de messages KQML. Chaque message comprend trois couches [55] :

- La couche de contenu, qui spécifie le contenu réel du message d'agent .

<sup>19</sup> Cette théorie est introduite en 1962, par le philosophe et le linguiste anglais **J.-L. Austin** dans son livre « *How to do things with words* ».

- La couche de communication, qui décrit tous les paramètres de communication de bas niveau, par exemple, l'identificateur de l'agent émetteur et celui de l'agent récepteur l'identificateur de la communication, etc.
- La couche de message, qui est considéré comme le noyau de KQML. Sa fonction principale s'agit d'identifier le protocole de réseau, utilisé pour envoyer le message et de déterminer le performatif, indiquant le type de ce message (**ex**: une affirmation une requête, une commande, etc.).

Le message KQML est représenté sous la forme d'une liste, contenant le performatif, qui correspond au type particulier d'acte de langage (**ex**: tell (transférer l'information aux autres agents), **ask-one** (demander la réponse à l'agent correspondant) etc.), et les arguments associés à ce performatif la figure (Figure III.3).



**Figure III. 3:** La structure d'un message KQML.

**Remarque:** Il est à noter que l'ordre des arguments dans une liste n'est pas important.

Le langage de KQML permet d'utiliser les différents protocoles de communication, notamment, TCP/IP, SMTP (e-mail), HTTP, CORBA, etc. Parmi les travaux réalisés concernant le développement de nouveaux protocoles, basés sur KQML on peut mentionner les langages de coordination **COOL** [56] et **AgentTalk** [57].

Bien qu'il présente un grand intérêt pour les utilisateurs, KQML montre quelques lacunes [58] :

- La signification floue de certains performatifs (**ex**: le cas d'un performatif « dénier ») ;
- Le manque des performatifs promissifs, exprimant l'engagement, auprès d'un tiers d'accomplir une action ;
- L'utilisation de ce langage que pour des communications isolées.

#### III.4.4.1.2. FIPA ACL

Pour résoudre certains problèmes inhérents à KQML, FIPA<sup>20</sup> a proposé un nouveau langage de communication d'agents – FIPA ACL<sup>21</sup>.

Comme KQML, ce langage est fondé sur la théorie des actes de langages. Sa spécification consiste en la définition d'un ensemble des types de messages et en description de leurs pragmatiques, c'est-à-dire, les effets sur les attitudes mentaux des agents émetteurs et des agents récepteurs. La sémantique formelle de FIPA ACL se compose de cinq niveaux [59] :

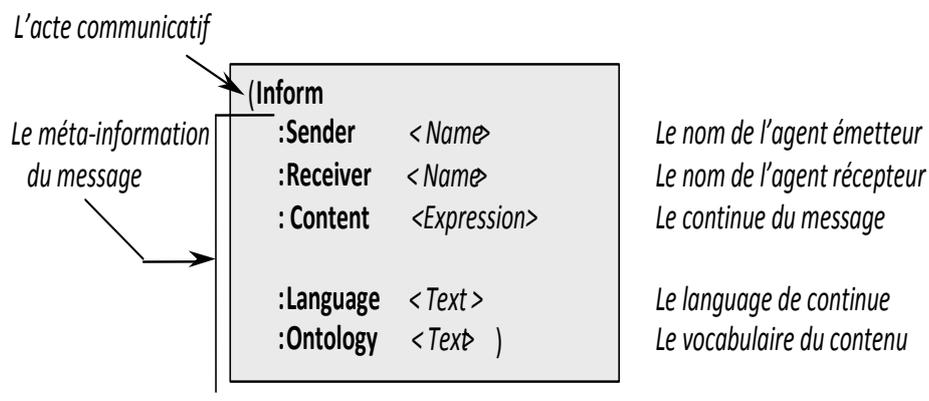
- Le protocole, qui décrit les règles sociales des dialogues entre les agents.

<sup>20</sup> **FIPA:** Foundation for Intelligent Physical Agents.

<sup>21</sup> **FIPA ACL:** FIPA Agent Communication Language.

- Les actes communicatifs (AC), qui définissent le type de communication entre les agents (par exemple, la demande, la confirmation, etc...).
- La méta-information concernant le message (l'identification d'un agent émetteur et d'un agent récepteur, le contexte, etc.).
- Le langage du contenu, qui décrit la grammaire et la sémantique associée, utilisées pour exprimer le contenu d'un message ;
- L'ontologie, qui définit le vocabulaire et les significations des termes et des concepts employés dans le contenu.

La syntaxe de FIPA ACL est identique à celle de KQML, sauf les noms de certains primitifs réservés. Le message FIPA ACL est représenté sous la forme d'une liste, contenant le type de l'acte communicatif (par exemple, INFORM, REQUEST), le nom de l'agent émetteur et celui de l'agent récepteur, le contenu et le contexte du message, l'ontologie à utiliser pour interpréter ce contenu, et le protocole [60] :



**Figure III. 4:** La structure d'un message FIPA ACL.

**FIPA ACL** peut être considéré comme l'extension de KQML possédant deux langages différents. Le langage externe définit la signification intentionnelle du message. Le langage interne (ou le contenu) décrit l'expression à laquelle s'appliquent les croyances, les désirs et les intentions des agents, décrites dans le primitif de communication.

Différemment de KQML, FIPA ACL est basé sur la sémantique logique de la communication. Ceci facilite la description des formats de la communication. Cependant les agents ne possèdent pas toujours les capacités logiques nécessaires. Il est à noter que la sémantique de FIPA ACL est basée en grande partie sur les croyances des agents, qui peuvent être inconnus pour les autres agents.

Une autre grande différence entre FIPA ACL et KQML, concerne les actes communicatifs. FIPA ACL contient un ensemble d'actes communicatifs normatifs, qui peuvent être primitifs ou composés. Les nouveaux actes communicatifs ne peuvent être définis qu'en combinant les actes existants et en utilisant les opérateurs prédéfinis. Ceci permet de maintenir l'intégrité sémantique du langage.

### III.4.5. Sémantique des messages

Pour que les agents puissent comprendre les messages inter-changés, il faut qu'ils partagent un vocabulaire commun. Il s'agit de garantir que les concepts et les entités véhiculées au travers des applications ont la même signification, même si différentes applications utilisent des noms différents les référant.

Ce vocabulaire ce n'est autre qu'une ontologie qui représente une sorte d'un vocabulaire d'un domaine bien précis.

L'intérêt de l'ontologie est essentiellement de décrire le vocabulaire d'un domaine (cardiologie par exemple), elle permet la réutilisation des connaissances, en outre, elle ajoute (enrichit) une sémantique aux messages inter-changés entre les agents.

### III.4.6. La coopération

La coopération est l'une des caractéristiques fondamentales des systèmes multi-agents.

D'après [50], le problème de la coopération peut se ramener à résoudre les différents sous problèmes qui comprennent la collaboration des agents par répartition des tâches, la coordination d'actions et la résolution de conflits. **J.Ferber** présente deux points de vue sur la coopération [46] [57] :

- ✓ Une attitude intentionnelle d'agents qui décident de travailler ensemble. Dans ce cas, les agents coopèrent s'ils effectuent une action commune, après avoir identifié et adopté un but commun. Ce concept présente deux inconvénients principaux : d'une part, on considère que la coopération existe, même si les agents obtiennent meilleurs résultats en travaillant individuellement ; d'autre part d'après l'attitude intentionnelle, la possibilité pour les agents réactifs de pouvoir coopérer est supprimée ;
- ✓ Une qualification d'une activité du groupe d'agents, observée par l'observateur qui interprète les comportements à partir de critères physiques et sociaux. Dans ce cas, plusieurs agents coopèrent si (1) l'ajout d'un nouvel agent permet d'accroître les performances d'un groupe, et (2) l'action d'agent sert à résoudre ou à éviter les conflits. Ces indices de coopération représentent la collaboration entre les agents ainsi que la résolution des conflits.

S'appuyant sur ces points de vue, on peut distinguer deux types de coopération : la coopération intentionnelle (ou celle d'agents cognitifs), où les agents ont l'intention de coopérer, et la coopération réactive.

**J.Ferber**<sup>22</sup> [46] précise plusieurs méthodes de coopération : le regroupement et la multiplication, la communication, la spécialisation, la collaboration par partage de tâches et de ressources, la coordination d'action ainsi que la résolution de conflit par arbitrage et la négociation. Toutes ces méthodes nécessitent d'être structurées au sein de l'organisation. Cette dernière permet de décrire la manière dont les agents sont positionnés dans un groupe ainsi que les techniques de travail coopératif efficace.

<sup>22</sup> **Jacques Ferber** est un professeur de l'informatique à l'université de la Science de Montpellier. Il a mené le groupe de recherche de Myriade à Paris à partir de 1988 jusqu'en 1997. Il mène maintenant un nouveau groupe de recherche qui est également consacré aux systèmes de multi agent.

Dans la section ci-dessus on va présenter la plateforme Jade, par laquelle, on va créer et gérer les agents de notre application.

### III.5. Outils & plateformes de développement des SMA

Le meilleur moyen pour construire un système multi-agents (SMA) est d'utiliser une plate-forme multi-agents. Cette dernière est un ensemble d'outils nécessaire à la construction et à la mise en service d'agents au sein d'un environnement spécifique. Ces outils peuvent servir également à l'analyse et au test du SMA ainsi créé. Ces outils peuvent être sous la forme d'environnement de programmation (API) et d'applications permettant d'aider le développeur.

Actuellement, il existe plusieurs plates-formes développements des agents et des SMA parmi ceux, on cite : **JADE**, **MADKIT**, **AGENT**

On va étudier ci-après la plate-forme **JADE**<sup>23</sup>.

**JADE** 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**<sup>24</sup> [60]. **JADE** comprend deux composantes de base : une plate-forme agents compatible FIPA et un paquet logiciel pour le développement des agents Java.

#### III.5.1. La norme FIPA pour les systèmes multi-agents

Les premiers documents de spécification de la norme FIPA, appelés spécifications [60], établissent les règles normatives qui permettent à une société d'agents d'inter opérer. Tout d'abord, les documents FIPA décrivent le modèle de référence d'une plate-forme multi-agents (Figure III.5) où ils identifient les rôles de quelques agents clés nécessaires pour la gestion de la plate-forme, et spécifient le contenu du langage de gestion des agents et l'ontologie du langage.

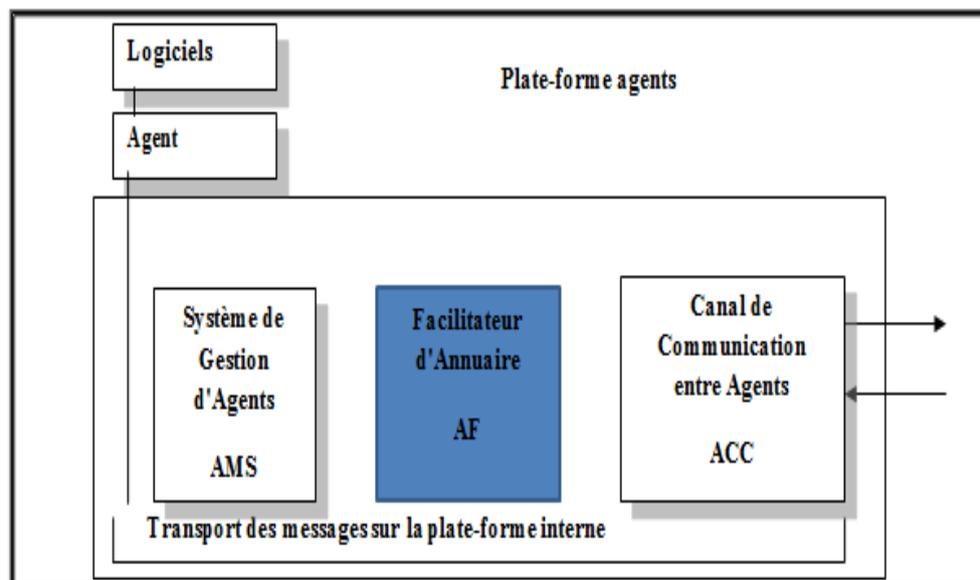


Figure III. 5: Le modèle de référence pour une plate-forme multi-agents FIPA.

<sup>23</sup> **JADE**: Java Agent **DE**velopment Framework.

<sup>24</sup> **FIPA**: Foundation for **I**ntelligent **P**hysical **A**gents.

Dans la figure (Figure III. 5), on voit qu'il existe trois rôles principaux dans une plate-forme multi-agents FIPA :

- Le **Système de Gestion d'Agents** (Agent Management System - **AMS**) est l'agent qui exerce le contrôle de supervision sur l'accès à l'usage de la plate-forme; il est responsable de l'authentification des agents résidents et du contrôle d'enregistrements.
- Le **Canal de Communication entre Agents** (**ACC**<sup>25</sup>) est l'agent qui fournit la route pour les interactions de base entre les agents dans et hors de la plate-forme; c'est la méthode de communication implicite qui offre un service fiable et précis pour le routage des messages; il doit aussi être compatible avec le protocole **IIOP**<sup>26</sup> pour l'interopérabilité entre les différentes plates-formes multi-agents.
- Le **Facilitateur d'Annuaire** (Directory Facilitator - **DF**) est l'agent qui fournit un service de pages jaunes à la plate-forme multi-agents.

Le standard spécifie aussi le **Langage de Communication d'Agents** (Agent Communication Language - **ACL**), la communication des agents est basée sur l'envoi de messages. Le langage **FIPA ACL** est le langage standard des messages et impose le codage, la sémantique et la pragmatique des messages. La norme n'impose pas de mécanisme spécifique pour le transport interne de messages. Plutôt, puisque les agents différents pourraient s'exécuter sur des plates-formes différentes et utiliser technologies différentes d'interconnexion, FIPA spécifie que les messages transportés entre les plates-formes devraient être codés sous forme textuelle. On suppose que l'agent est en mesure de transmettre cette forme textuelle.

### III.6. domaine d'application

Les systèmes multi agents étant issus du domaine de l'intelligence artificielle distribuée, ils permettent de modéliser des applications où une modélisation classique est inadéquate, et où le système est souvent naturellement distribué. Si un problème n'est pas naturellement distribué, on peut choisir de le distribuer pour des raisons de simplification. C'est alors à la charge du concepteur de trouver la meilleure distribution possible afin qu'elle corresponde au problème qu'il souhaite résoudre. Les applications qui sont naturellement distribuées sont, par exemple, celles concernant la simulation d'écosystèmes où l'on associe à chaque individu un agent.

- **Ecosystème et modèle individus centrés**

Les modèles individus centrés, c'est-à-dire orientés individus sont des simulations basées sur les conséquences globales d'interactions locales entre membres d'une application. Ces individus peuvent représenter des plantes et des animaux dans un écosystème, des véhicules dans la circulation, des personnes dans une foule, ou des personnages autonomes dans une animation ou un jeu. Ces modèles consistent typiquement en un environnement ou un cadre globale dans lequel les interactions se passent et un certain nombre d'individus définit par leur comportement (règle comportementale) et des paramètres caractéristiques. Dans un modèle orienté-individus les caractéristiques de chaque individu peuvent être suivies de façon continue. Ceci

<sup>25</sup> **ACC**: Agent Communication Channel.

<sup>26</sup> **IIOP** : Protocole de communication permettant d'intégrer et de mettre en réseau des applications de provenance diverse.

contraste avec les techniques de modélisation où le modèle tente de simuler des changements dans les caractéristiques moyennées d'une population globale. Les modèles orientés individus sont aussi appelés « orientés entités » ou « orientés-agents », et on parle de simulation multi agents. Dans ces systèmes, un agent (individu) correspond toujours à une entité du monde réel (un animal, une personne, un objet, une bactérie, etc.). De nombreuses applications de simulation utilisent des SMA basés sur un modèle individu centré [61].

- **Systèmes complexes**

Les systèmes complexes sont des systèmes où les techniques de modélisation classique sont difficilement utilisables. En effet, dans ces systèmes, les paramètres sont beaucoup trop nombreux ou contradictoires pour pouvoir être pris en compte. Parfois même, il n'est pas possible de connaître l'ensemble des paramètres qui interviennent pour la modélisation. L'approche multi agents permet alors d'avoir recours à une modélisation locale. Cette modélisation permet grâce aux principes d'émergence présents dans les SMA d'obtenir un système ayant les propriétés attendues. Il s'agit notamment de l'ensemble des systèmes permettant la simulation d'écosystèmes. Un bon exemple concerne la simulation d'une fourmilière [62].

En effet, lors de ses travaux de thèses, Alexis Drogoul a modélisé une fourmilière en utilisant des agents pour modéliser les fourmis. Il a alors réussi à démontrer que l'on pouvait obtenir un but global qui était la survie de la fourmilière sans jamais avoir programmé cet élément dans le système mais uniquement à partir de l'interaction des agents par émergence organisationnelle.

- **Systèmes d'aide à la décision et SMA**

Les systèmes d'aide à la décision (**SAD**<sup>27</sup>) sont présents dans de nombreux domaines et ont pour objectif d'aider le décideur dans sa tâche en lui fournissant tous les éléments pertinents pour la prise de décision. Cela consiste très souvent à extraire de l'information depuis de multiples sources et à la traiter. Des traitements de l'information pour une prise de décision multicritères sont alors nécessaires. Les systèmes multi agents apparaissent comme étant bien adaptés pour traiter de l'information qui peut revêtir diverses formes et provenir de diverses sources. En effet, il faut pouvoir faire la corrélation entre l'ensemble des éléments obtenus pour les présenter à l'utilisateur. L'approche à base d'agents permet d'effectuer cette corrélation en utilisant la négociation et la coopération entre agents. Il s'agit donc d'un domaine d'application pour les SMA [63].

- **Le domaine du commerce électronique et les agents du WEB**

Le domaine du commerce électronique est un domaine en plein essor qui permet de favoriser les transactions commerciales, Ce domaine utilise l'outil informatique et plus particulièrement les ressources mises à disposition par Internet pour rapprocher les acteurs commerciaux dans certains domaines. Ce domaine se rapproche de celui de l'aide à la décision et peut même être confondu avec celui-ci dans certaines circonstances car il est caractérisé par la mise en place de moyens permettant d'extraire de l'information sur les produits, les marchés, Les acteurs du marché [64].

L'utilisation de systèmes multi agents est une bonne solution ici aussi car elle permet comme pour le domaine de l'aide à la décision de faire de la fusion d'informations. La

---

<sup>27</sup> **SAD** : Systèmes d'Aide à la Décision

dénomination «agent du WEB» désigne les agents qui circulent sur le réseau Internet pour extraire de l'information. Ces agents qui sont en général mobiles se déplacent au travers des sites WEB. Certains systèmes ont pour objectif de fournir au consommateur une valeur ajoutée en lui fournissant des informations supplémentaires afin qu'il fasse un choix approprié selon les critères qu'il s'est fixé [65]. Il peut s'agir d'informations comparatives entre produits ou encore d'études qualitatives.

Prenons un exemple concret: vous souhaitez acheter une nouvelle voiture mais vous n'avez pas encore décidé du modèle exacte. Jusqu'à maintenant, vous deviez visiter un certain nombre de concessionnaires, étudier des documentations techniques (parfois ardues), acheter des magazines spécialisées qui effectuent eux-mêmes des tests et études comparatives. Tout ce travail est fastidieux et coûteux en temps. Le but des applications dans le domaine du commerce électronique va être d'extraire toute cette information qui est disponible à divers endroits du WEB et de n'en retenir que celle qui vous intéresse. Il va s'agir d'un travail d'extraction, de filtrage et d'analyse de l'information disponible. La multiplicité des sources d'informations disponibles sur le WEB et la complexité des traitements à effectuer rend particulièrement appropriée l'utilisation du paradigme agent pour le développement des applications de commerce électronique [63].

### **III.7. Conclusion**

Les systèmes multi-agents ont apporté une nouvelle vision des choses et une nouvelles façon de concevoir des solutions à des problèmes qui au pare-avant était impossible ou extrêmement difficiles à résoudre.

Dans ce chapitre Nous avons essayé de définir le plus précisément possible, ce qu'un agent et ce qu'un système multi-agent tout en mettant l'accent sur les concepts qui s'y rapportent tel que celui de l'environnement d'interaction et de communication. Dans le chapitre suivant, nous allons vous présenter Un moyen utilisé pour construire un système multi-agent qui est la plate-forme jade.

Chapitre IV

Implémentation

## Chapitre IV : Implémentation

### IV.1. Introduction

Ce dernier chapitre est réservé à la partie implémentation qui consiste à faire une présentation des différents outils utilisés lors du développement de notre application ainsi que la description de son fonctionnement par des images explicatives. Nous avons principalement utilisé le langage JAVA et l'environnement de développement JCREATOR. Pour les agents de notre application, nous avons utilisé la plateforme JADE qui fournit de puissants outils pour le développement de ces genres d'application.

### IV.2. Type d'agents développés

Dans notre application, nous avons développé deux types d'agents, le premier est l'agent Serveur (Directeur de l'entreprise), c'est le responsable de démarrage de la plateforme JADE du système. Cet agent sera lancé au départ puis, on lance les agents de type client ces derniers agents représentent le personnel de cette entreprise, dont on trouve des agents : administrateur principal, Administrateur, Agent de saisie, Agents comptable secrétaire générale (SG), agent de secrétariat, chef de service, chef de bureau...etc. Ces derniers agents seront lancés après avoir lancé l'agent serveur au départ. Donc, voici l'architecture distribuée de notre application, présentée dans (Figure IV.1).

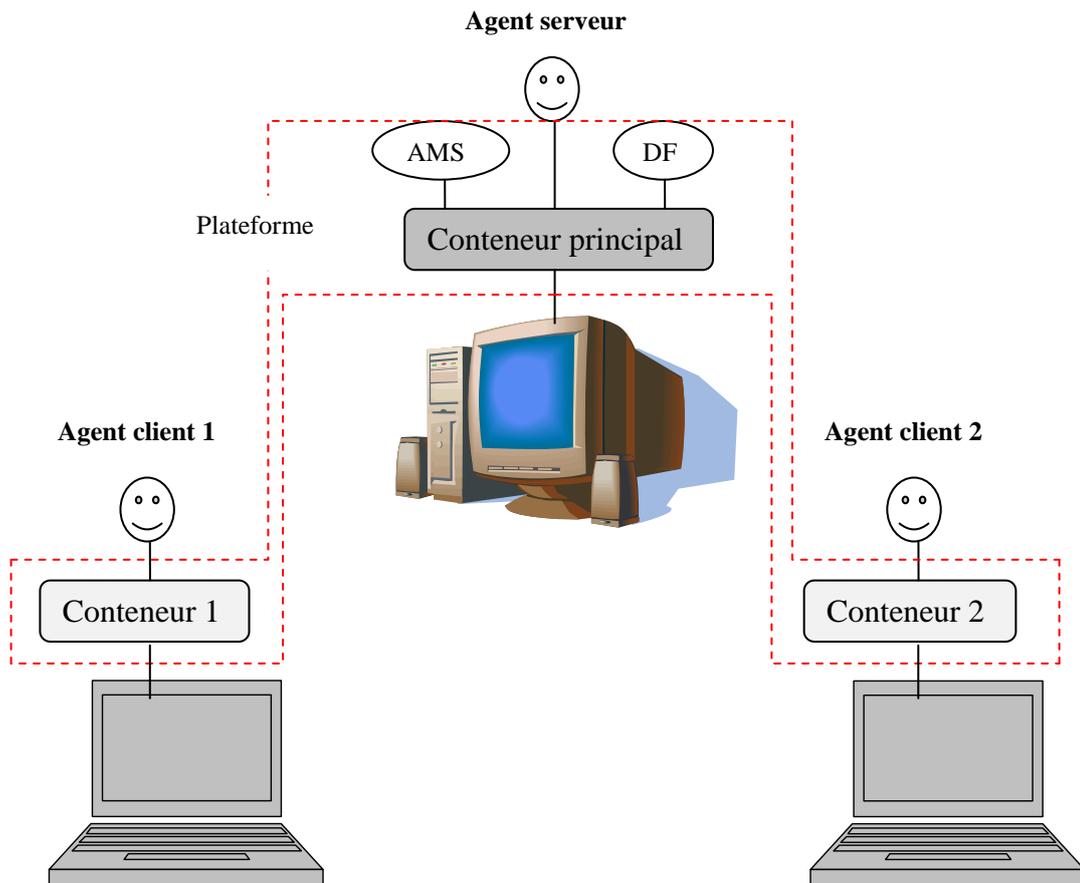


Figure IV. 1: Architecture distribuée de l'application.

Dans la figure précédente, nous avons deux conteneurs 1 et 2 qui sont secondaires et qui sont enregistrés par le conteneur principal de la plateforme Jade, chaque conteneur vit physiquement dans un pc à part, mais logiquement, ils vivent ensemble dans le Framework Jade. Ce dernier fournit des outils tels que l'AMS (Agent Management System) pour la gestion des agents de la plateforme, le DF (Directory Facilitator) qui joue le rôle d'un annuaire, service page jaune.

Jade nous fournit aussi, un langage de communication s'appelant Fipa-Acl, qui est propre aux agents qui vivent dans une plate-forme Jade, ce qu'est un point essentiel pour les agents intelligents.

### IV.3. Pourquoi la plateforme JADE ?

On a développé notre application sous la plate-forme multi agent JADE qui offre les avantages suivants :

- Plate-forme assez facile à mettre en place.
- Disponibilité de packages sur lesquels nous nous sommes appuyés pour développer notre application.
- Documentation claire et complète.

#### IV.3.1. 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** [51].



#### IV.3.2. Une plateforme Jade

La figure suivant présente la structure de plateforme jade [43].

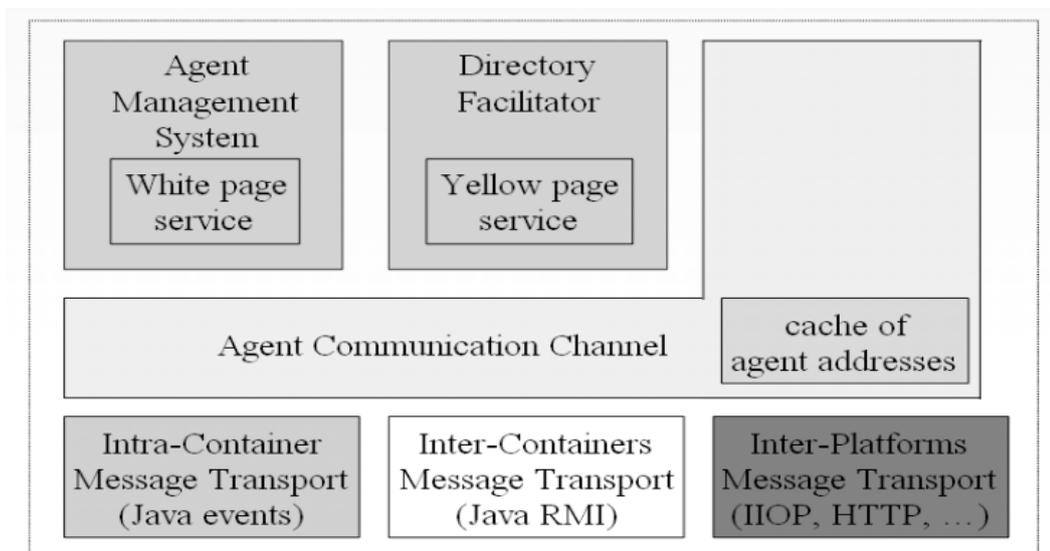


Figure IV. 2: Plateforme globale de jade.

**1. Agent Management System (AMS)**

- Gestion du cycle de vie des agents.
- Maintient une liste de tous les agents qui résident sur la plate-forme (White pages).
- Contrôle l'accès ainsi que l'utilisation du canal de communication des agents (ACC).

**2. Directory Facilitator**

- Enregistre les descriptions des agents ainsi que les services qu'ils offrent.
- Les agents peuvent enregistrer leurs services auprès d'un DF ou demander à DF de découvrir les services offerts par d'autres agents (Yellow Pages).

**3. Agent Communication Channel**

- Gère les communications entre les agents
  - Intra plateformes, Intra containers : Java Events.
  - Intra plateformes, Inter containers : RMI.
  - Inter Plateformes : IIOP Corba.
- Messages ACL (FIPA).

**IV.4. modèle de workflow**

La figure suivante représente le processus de demande de congé, il se déroule de la manière suivante :

1. La préparation de la demande par l'agent client (demandeur) en remplissant les champs (le nom de l'agent remplaçant, date de début, date de fin, raison)
2. L'agent client demandeur envoie sa demande de congé vers l'agent serveur
3. Dans le message reçu, l'agent serveur cherche le nom de l'agent client (remplaçant) puis il lui envoie la confirmation de remplacement.
4. L'agent client (remplaçant) reçoit la demande de confirmation par l'agent serveur après ça, il lui envoie sa réponse.
5. Suivant la réponse reçue, l'agent serveur prend la décision si :
  - oui : un message de « Demande de congé est accordée » est envoyé à l'agent client demandeur du congé.
  - non : un message de « Demande de congé est refusée » est envoyé. L'agent client demandeur du congé.

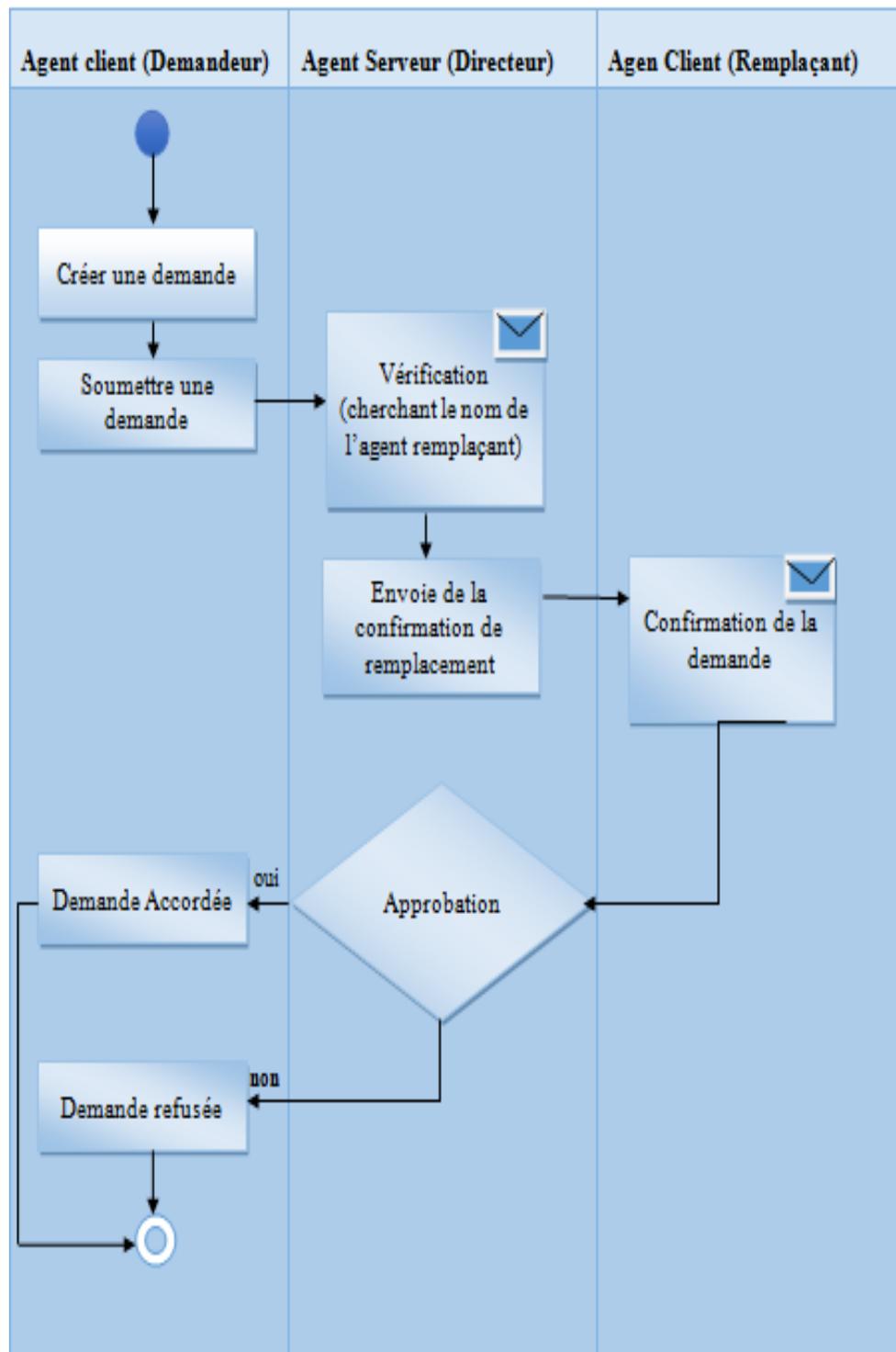
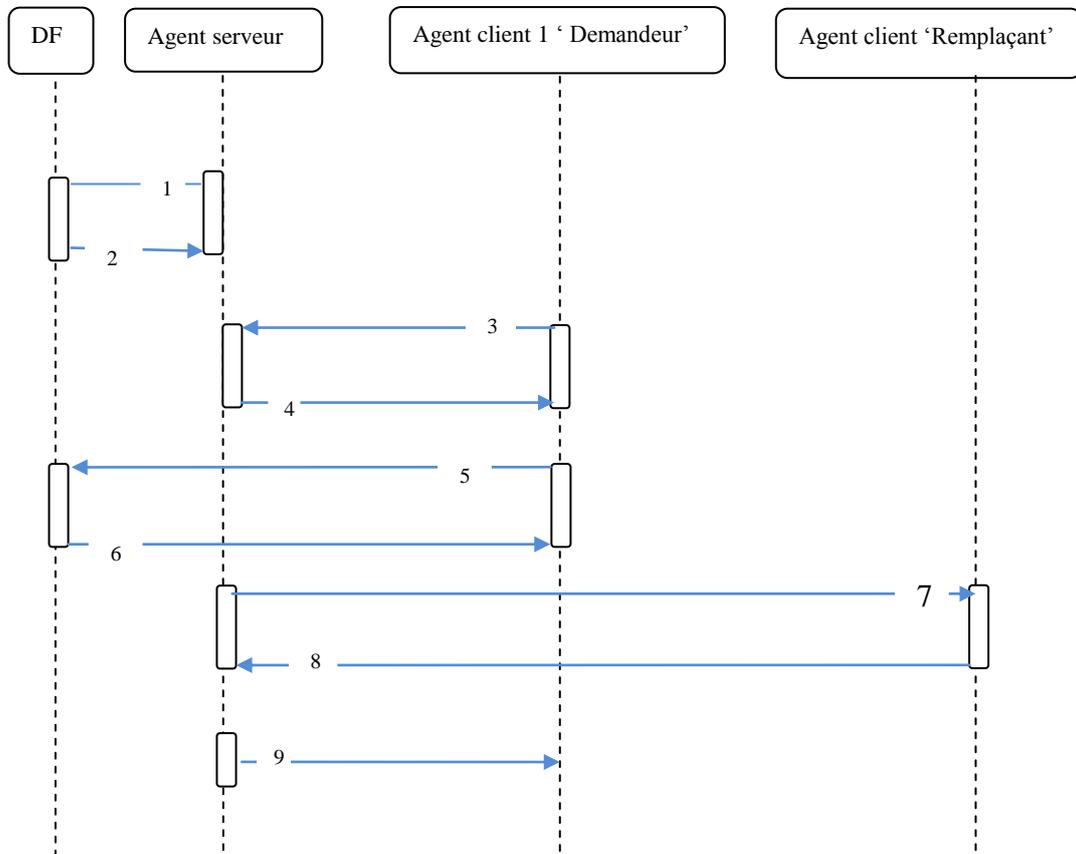


Figure IV. 3: le modèle de workflow.

## IV.5. Énumération des actes de communication des agents

Les actes de communications entre agents suivent un processus qui se déroule suivant un ordre donné. Nous présentons sur la figure suivante un tel processus.



**Figure IV. 4:** Diagramme de séquence entre les agents de système.

Les différentes interactions ou actes de communication se déroulent suivant un ordre donné comme indiqué ci-après :

- 1) Demande d'enregistrement de l'agent Serveur.
- 2) Réponse de l'agent DF, pour la demande d'inscription de l'agent "Agent Serveur". (INFORM)
- 3) Demande de l'authentification de l'agent client1 (respectivement l'agent client2). (REQUEST)
- 4) Validation de l'authentification. (INFORM) si le pseudo et le mot de passe sont valides.
- 5) Demande d'enregistrement de l'agent Client 1 « demandeur » (respectivement "client2" « remplaçant») au DF. (INFORM)
- 6) Réponse de l'agent DF, pour la demande d'inscription de l'agent "Agent Client 1" (respectivement "l'agent client2"). (INFORM)
- 7) L'agent "Agent client 1 "envoie une demande de congé à l'agent "Agent Serveur (REQUEST)
- 8) L'agent « Agent Serveur » envoie la demande de confirmation à l'agent « Agent Client 2». (REQUEST).

- 9) L'agent « Agent Serveur » envoie un message de décision à l'agent « Agent Client1 ». (INFORM)

## IV.6. Présentation de l'application

La première fenêtre après le lancement de la plateforme, cette dernière contient des boutons « lancement du serveur » pour lancer le serveur et « Je me connecte » pour la connexion vers un compte de client existant et «Créer un compte » c'est l'espace pour créer des nouveaux comptes pour les clients. La figure (Figure IV.5) présente la première fenêtre.

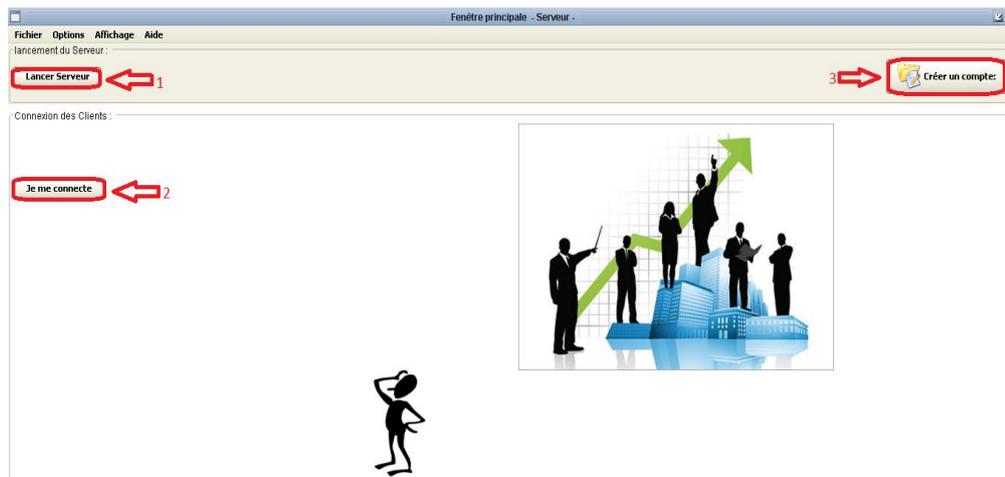


Figure IV. 5: Fenêtre principale de l'agent serveur.

Cette fenêtre contient des boutons.

Pour lancer le serveur cliquez sur (1) et la connexion d'un client vers leur compte cliquez sur(2) et la création d'un nouveau compte par (3).

- Pour Lancer l'agent « Serveur » en cliquant sur le bouton : **lancer Serveur**. comme illustre la figure suivante :

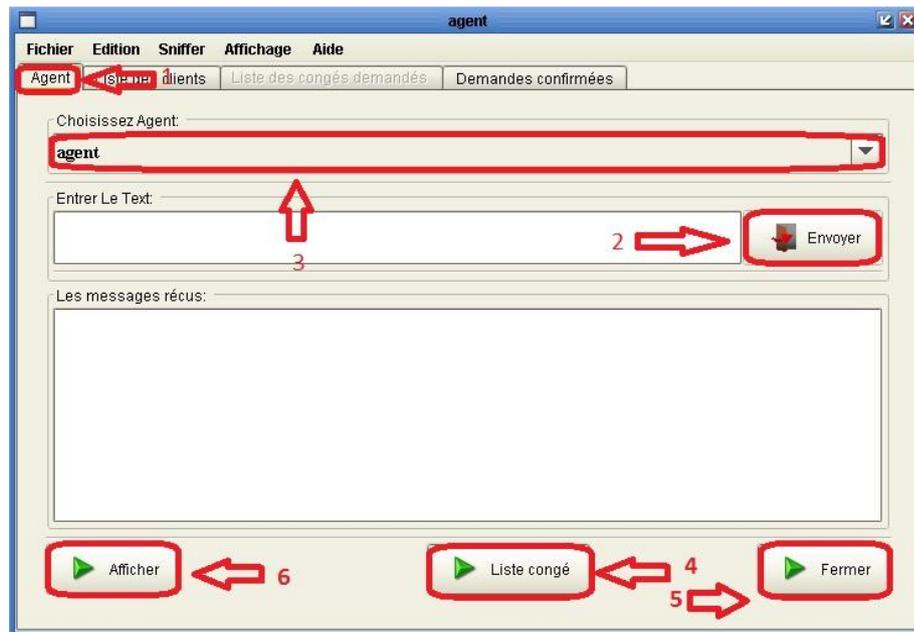


Figure IV. 6: Fenêtre de l'agent serveur.

- D'après (1) dans la figure (Figure IV.6) L'agent permet de écrire des messages et choisir un client parmi les clients qui est en ligne pour discuter à partir (3) en fin envoyer le message par (2).

(4) Que en clique sur bouton « Liste congé » afficher la liste des demandes congé des clients, (5) bouton fermer pour fermer la fenêtre de l'agent.

#### a. Créer un compte

La figure suivant représente les données nécessaires d'inscription :

L'utilisateur doit remplir leur information personnel dans les champs pour créer un compte (Code, Pseudo, Nom, Prénom, Mot de passe, Confirmer, Fonction).

L'inscription des Clients :	
- Code :	<input type="text" value="15"/>
- Pseudo :	<input type="text"/>
- Nom :	<input type="text"/>
- Prénom :	<input type="text"/>
- Mot de passe :	<input type="text"/>
- Confirmer :	<input type="text"/>
- Fonction :	<input type="text" value="Chef de service"/>
<input type="button" value="Afficher BD"/> <input type="button" value="Valid.."/> <input type="button" value="Annuler"/>	

Figure IV. 7: Formulaire pour créer un compte client.

L'agent serveur clique sur bouton « Actualiser » pour voir les nouveaux comptes agents clients dans « Liste des clients »

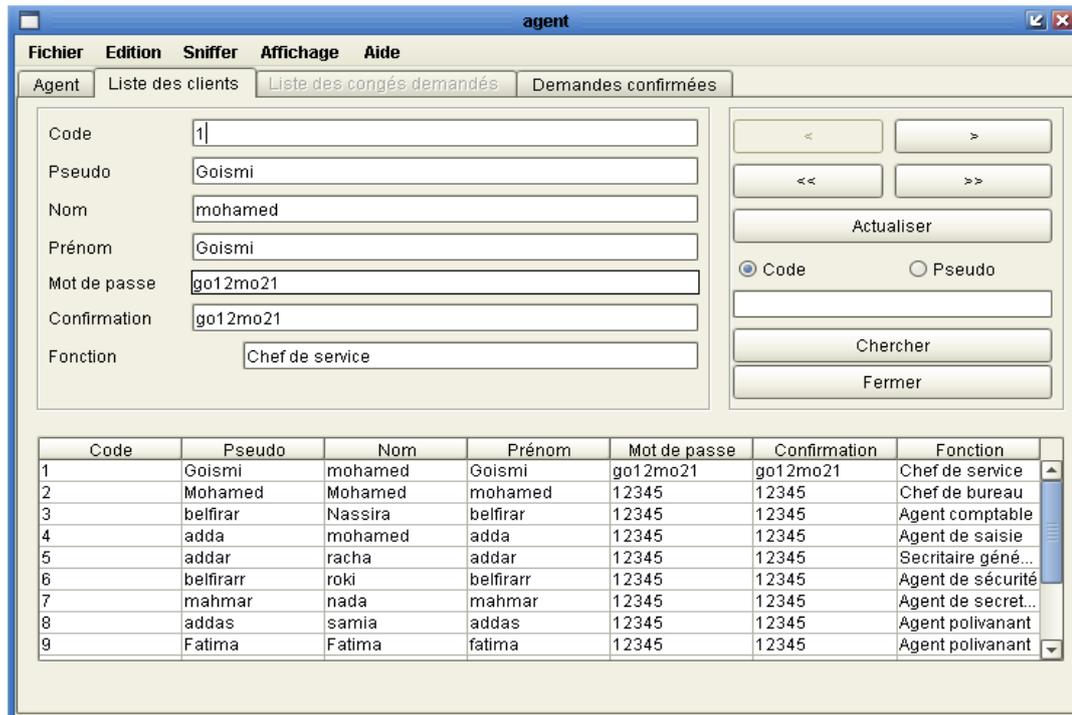


Figure IV. 8: Liste des agents clients existe dans le serveur.



Figure IV. 9:L'agent client « belfirar ».

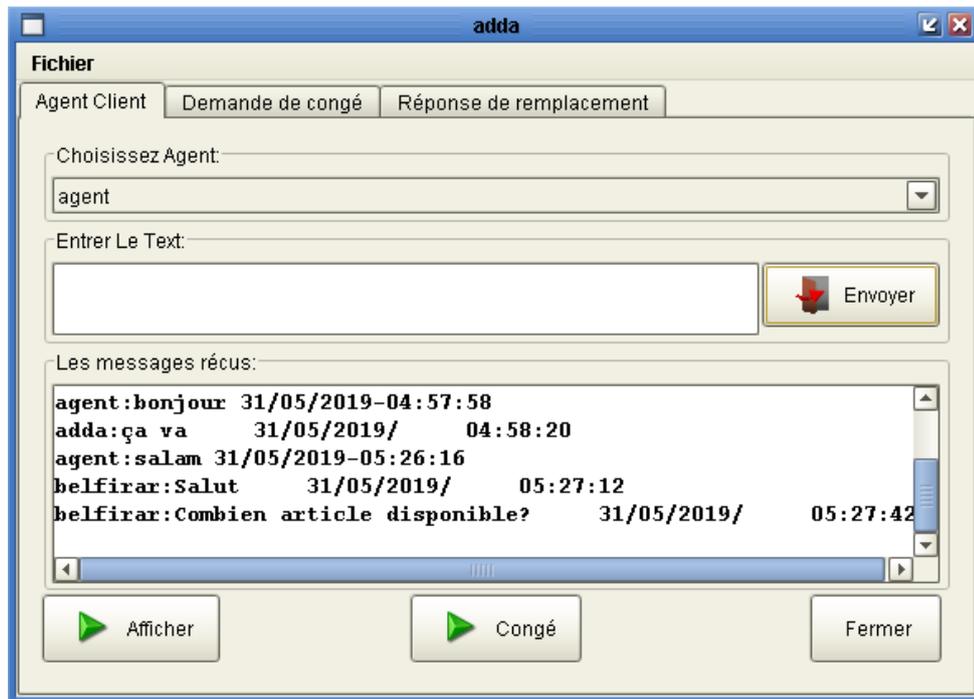


Figure IV. 10: L'agent client « adda ».

L'agent client faire la mise à jour après la connexion vers leur compte pour contacté les agents en ligne.

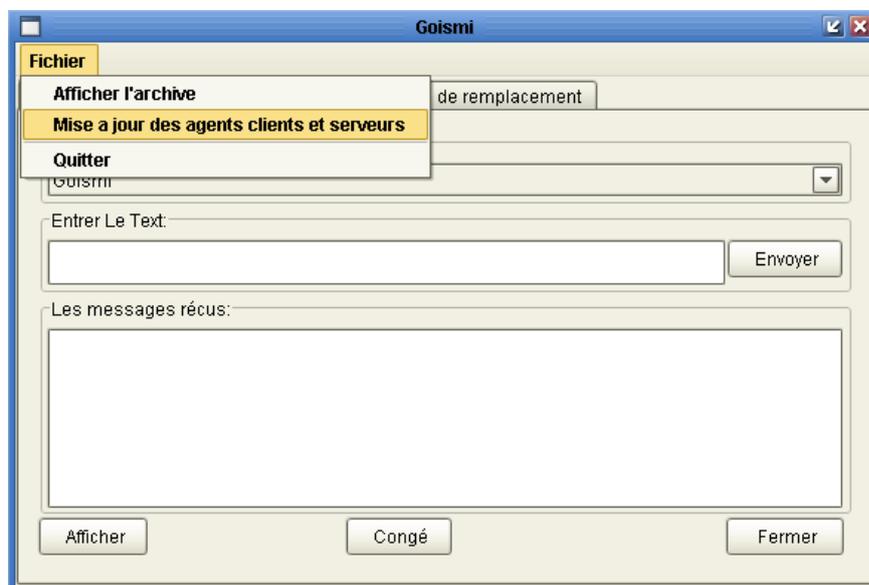


Figure IV. 11: L'agent client « Goismi ».

### b. L'envoi d'un message

La séquence des messages échangés entre les agents du système est illustrée par la figure suivante :

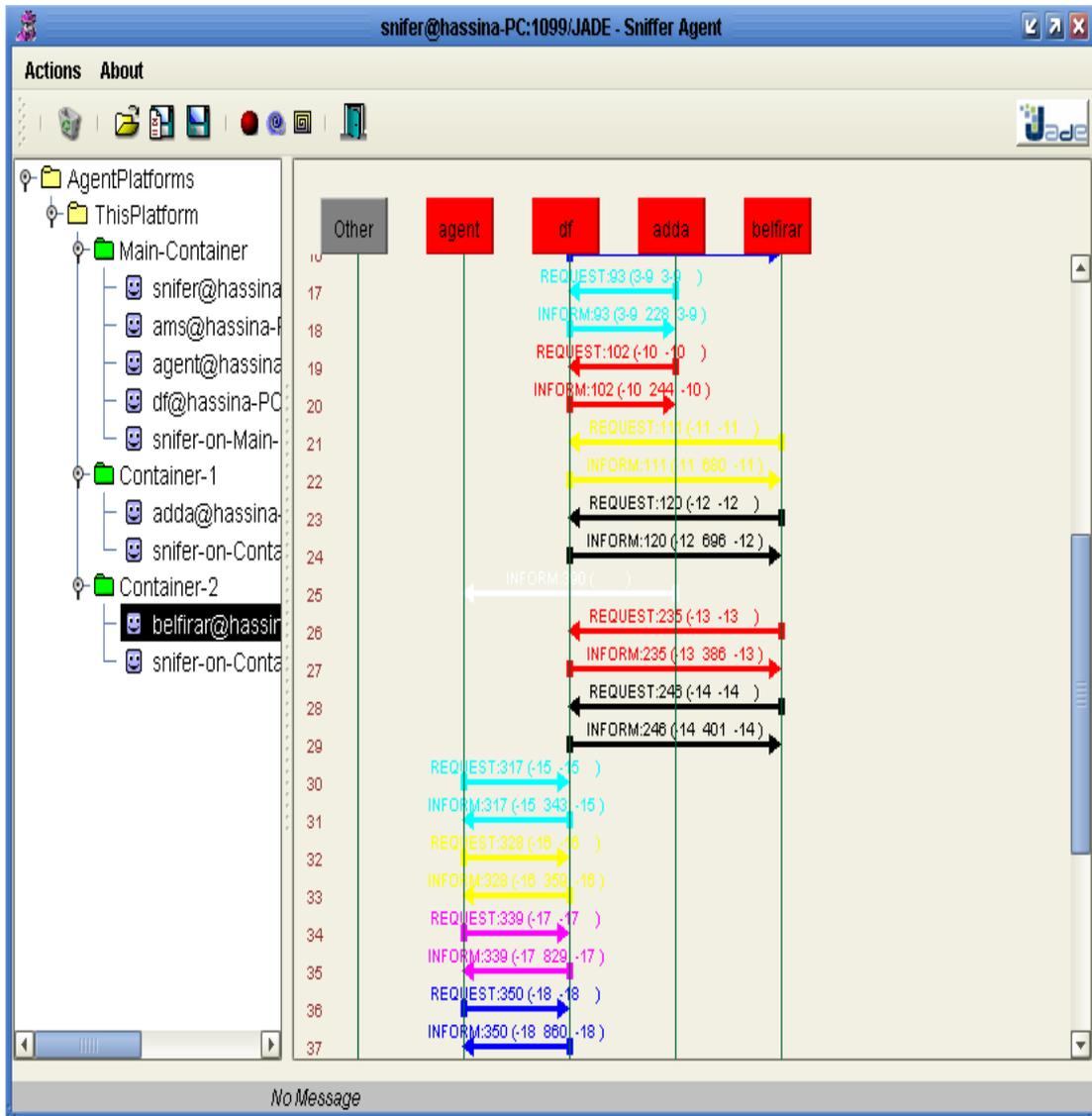


Figure IV. 12: SNIFER, contrôleur des messages échangés entre les agents du système.

Comme la messagerie instantanée est maintenant un mode précieux de communication dans l'environnement de travail, nous avons créé une boîte de conversation instantanée afin de rendre la communication facile entre les différents membres de cette société. Ceci permettant aux membres de l'entreprise de vivre dans un environnement privé confidentiel et bien sécurisé, d'une part.

Voici un exemple pour cette communication entre l'agent serveur et un agent client : Après la connexion de l'agent client et la mise à jour dans les deux fenêtres : celles de l'agent serveur et de l'agent client lancés (La mise à jours nous a donné la liste des agents connectés), l'agent serveur a la possibilité de sélectionner l'agent « Goismi » et lui envoie son message dont son contenu est « *A ce que vous avez faire un demande congé cet année , Goismi* » en cliquant sur le bouton « Envoyer ».

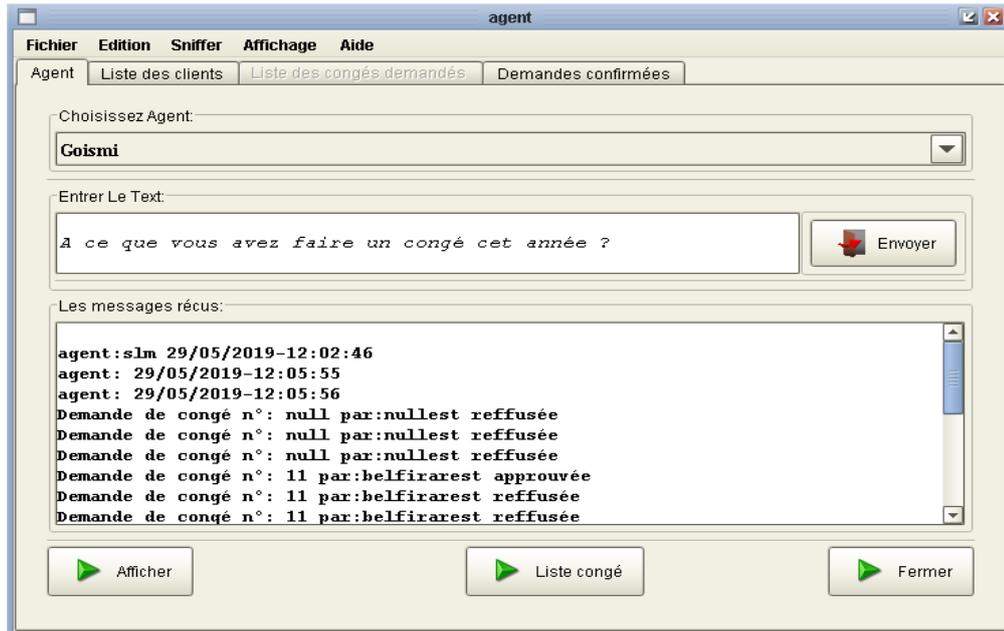


Figure IV. 13: Envoie du message à l'agent client « agent ».

La figure suivant représente

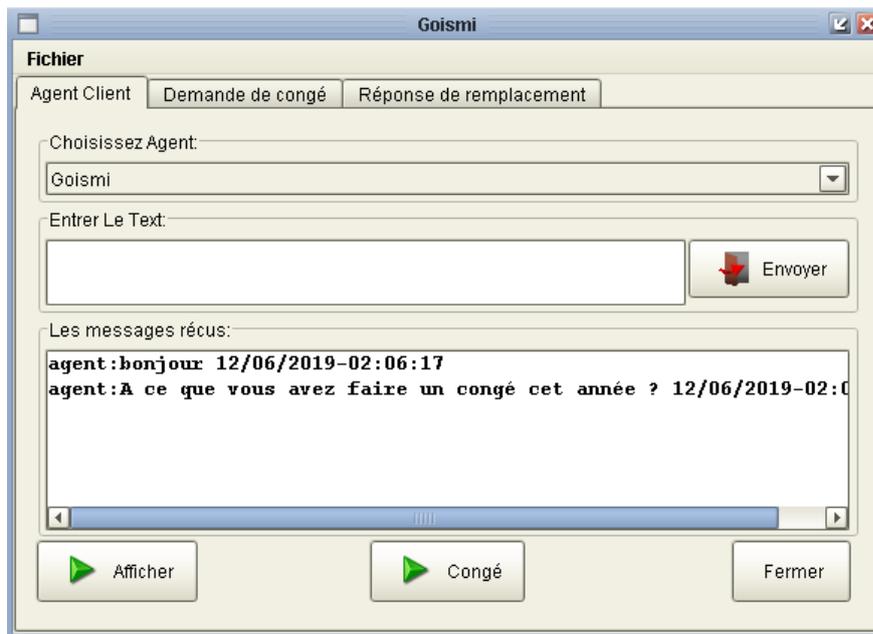


Figure IV. 14: Réception du message de l'agent Serveur.

### c. Demande du congé

- L'agent client demandeur envoie sa demande de congé vers l'agent serveur :
- L'agent choisit un remplaçant et la date de début et la fin de congé et leur raison.
  - 1) demande 1 : 'adda' remplaçant 'Goismi'.
  - 2) demande 2 : 'belfirar' remplaçant 'Goismi'.
- Les deux demandes vers le même remplaçant 'Goismi'.

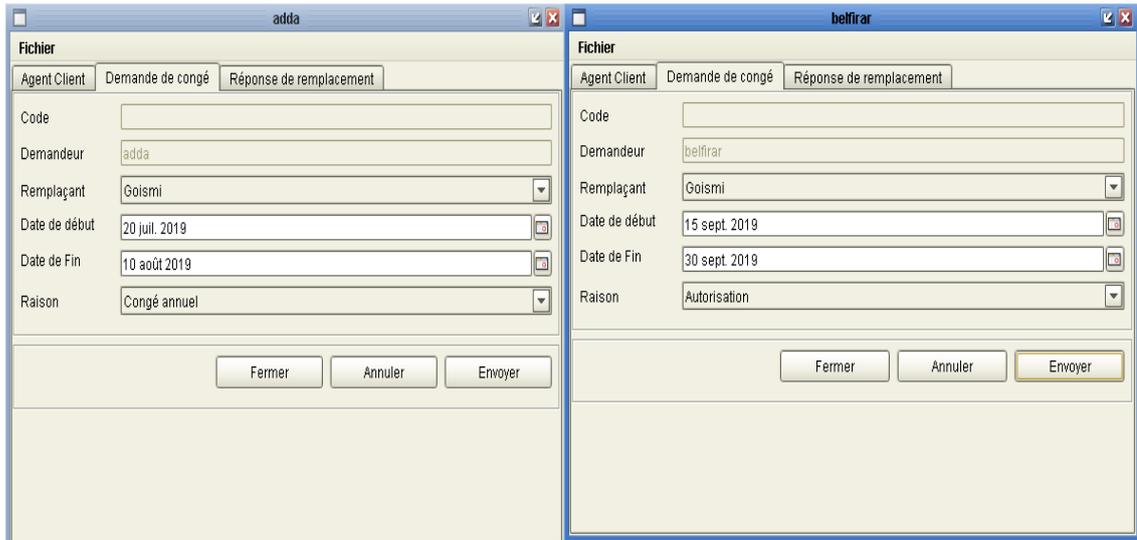


Figure IV. 15 : demande de conge rempli par le demandeur.

- Après la réception de la demande par l’agent serveur, il clique sur le bouton « **Confirmer** » et la demande sera envoyée à l’agent client « Remplaçant » sélectionné.

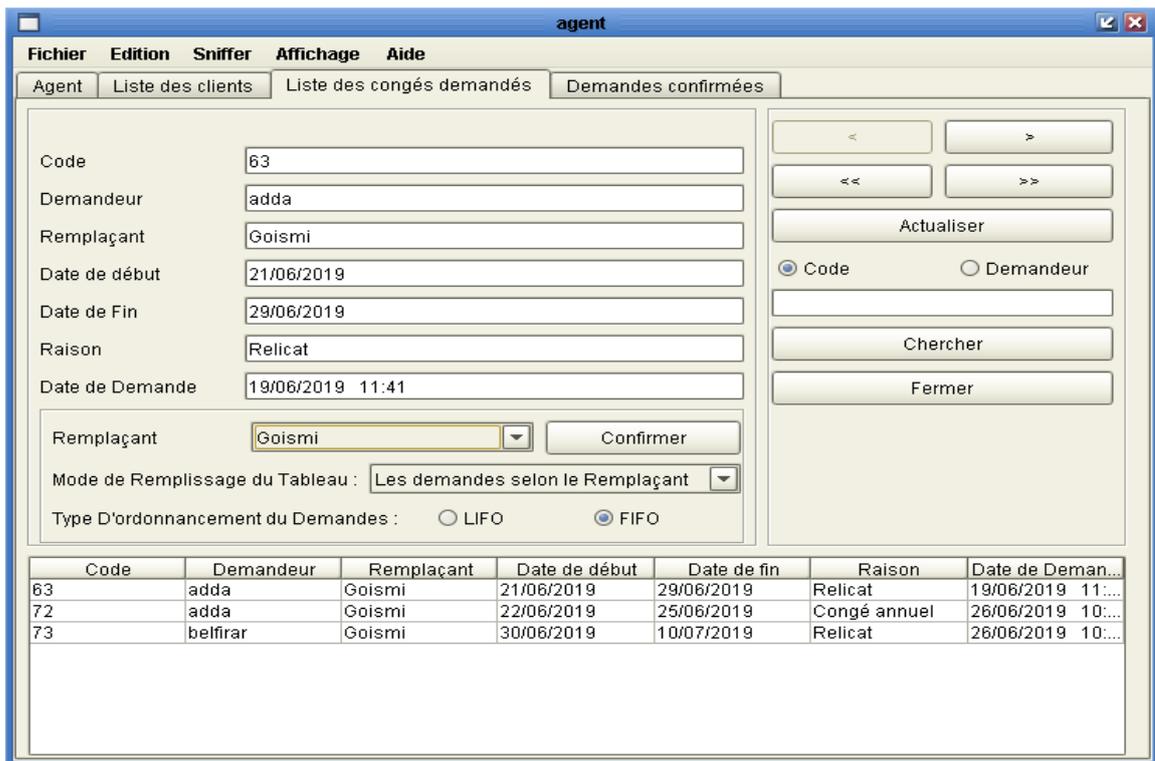


Figure IV. 16: la liste de demande congé.

- L’agent remplaçant valide cette demande par le choix de (**OUI/NON**) pour accepter ou refuser cette dernière et envoie sa réponse au serveur, en cliquant sur le bouton « **Envoyer** ».

La figure suivant résumé cet étape :

The screenshot shows a window titled 'Goismi' with a 'Fichier' menu. It has three tabs: 'Agent Client', 'Demande de congé', and 'Réponse de remplacement'. The 'Réponse de remplacement' tab is active. The form contains the following fields:

- Code: 20
- Demandeur: belfirar
- Remplaçant: Goismi
- Date de début: 15/09/2019
- Date de Fin: 30/09/2019
- Raison: Autorisation
- Réponse: OUI (selected in a dropdown menu)

An 'Envoyer' button is located at the bottom right of the form.

Figure IV. 17: la réponse de remplaçant l'agent client 'Goismi'.

Le remplaçant 'Goismi' envoyer la réponse vers le serveur pour valider la demande de l'autre agent 'adda'.

- Après la décision

❖ un message sera envoyé au serveur contenant soit (OUI) soit (NON).

C'est la date de demande est différent et le remplaçant est le même alors Le serveur étudié la demande par la priorité, clique sur bouton radio FIFO ou bien LIFO pour ordonnancé les demandes.

The screenshot shows a window titled 'agent' with a menu bar: 'Fichier', 'Edition', 'Sniffer', 'Affichage', 'Aide'. It has four tabs: 'Agent', 'Liste des clients', 'Liste des congés demandés', and 'Demandes confirmées'. The 'Agent' tab is active. It contains a 'Choisissez Agent:' dropdown menu with 'agent' selected, and an 'Entrer Le Text:' input field with an 'Envoyer' button. Below is a list of received messages:

```

Les messages reçus:
belfirar:bonjour 31/05/2019/ 04:58:52
adda:wSalam 31/05/2019/ 05:26:51
Demande de congé n°: 15 par:belfirarest approuvée
Demande de congé n°: 18 par:addaest approuvée
Demande de congé n°: 18 par:addaest refusée
Demande de congé n°: 18 par:addaest refusée
agent:Bonjour 31/05/2019-05:48:49
Demande de congé n°: 22 par:hassinaest approuvée
Demande de congé n°: 23 par:addaest approuvée
  
```

At the bottom, there are three buttons: 'Afficher', 'Liste congé', and 'Fermer'.

Figure IV. 18: la réception de message de confirmation de la demande par l'agent serveur.

Un message sera envoyé à l'agent client « demandeur ».

Après la validation, La demande sera enregistrée dans la liste des demandes confirmées.

The screenshot shows a software window titled 'agent' with a menu bar (Fichier, Edition, Sniffer, Affichage, Aide) and a tabbed interface. The 'Demandes confirmées' tab is active. The form contains the following fields:

- Code: 6
- Demandeur: adda
- Remplaçant: Goismi
- Date de début: 20/06/2017
- Date de Fin: 15/06/2017
- Raison: Personnel

Navigation buttons include '<', '>', '<<', '>>', 'Actualiser', 'Chercher', and 'Fermer'. Radio buttons for 'Code' (selected) and 'Demandeur' are also present.

Code	Demandeur	Remplaçant	Date de début	Date de fin	Raison
6	adda	Goismi	20/06/2017	15/06/2017	Personnel
8	belfirar	Mohamed	15/06/2017	16/06/2017	Congé annuel
10	hassina	fatiha	02/06/2017	24/06/2017	Congé annuel
11	belfirari	mahmar	25/05/2015	15/05/2019	Relicat

Figure IV. 19: Liste des demandes confirmée.

## IV.6. Conclusion

Dans ce chapitre, nous avons vu notre application où nous avons essayé de mettre en œuvre l'ensemble des idées qui caractérise l'architecture proposée en se concentrant sur l'implémentation de la mobilité des agents tout cela est fait à l'aide d'une présentation des interfaces et des captures d'écran du système.

### Conclusion générale

Notre travail consiste à réaliser une application qui permet d'améliorer et de faciliter le processus de demande de congé en utilisant les systèmes multi-agents et leur principe de mise en œuvre en essayant de profiter de cette dernière pour avoir une technologie innovante, émergente et distribuée qui possède un fort potentiel.

Le mémoire avait comme objectif la réalisation d'une application traitant l'ordonnancement distribué des tâches dans un workflow de demande de congé dans une entreprise où on a utilisé le concept d'agents et des systèmes multi-agents pour avoir la possibilité de faire distribuer ces tâches entre les différents acteurs de cette entreprise.

Bien que l'application développée est loin d'être employée sur le plan pratique, l'implémentation nous a permis :

- ✚ De faire connaissance avec le langage JAVA, un langage en plein essor et de plus en plus utilisé dans presque tous les domaines.
- ✚ De voir en pratique des concepts de programmation orientée agent.
- ✚ Cette expérience est un bon complément de notre formation de base, elle nous a permis d'enrichir nos connaissances et constitue la base de départ pour des futurs travaux.

### **PERSPECTIVE**

Nous sommes conscients que ce travail présente des limites, il ne prétend pas à donner une solution parfaite à la problématique posée. Des améliorations lui peuvent être apportées, Citons quelques perspectives, pouvant être traitées dans le futur :

- l'intégration de la plateforme **JBPM** pour la création et la gestion des workflow sous JAVA.
- l'étude de traitement de congé, tel qu'il est en réalité respectant tous ces aspects (étude spécifiée de chaque composant) pour une demande de congé parfaite et complète.

## Références bibliographiques

- [1] Seif Ennasar Hazar. Pawor point sur les moteurs de workflow, institut supérieur des études technologiques de kébili, 2013.
- [2] Maker. Workflow Meets Business Objects .In Proceedings of OOPSLA'96 Workshop, Business Objects Design and Implementation II: Business Objects as Distributed ApplicationComponents–the enterprise solution, 1996.
- [3] Mohan, G.Alonso, R.Gunthor ETM.Kamath. Exotica:A research perspective on workflow management systems.Data Engineerin, 1995.
- [4] Pudhota, L. Tierney, A. and Chang, E. Services integration monitor for collaborative workflow management. In Proceedings of the 14th IEEE International Workshops on EnablingTechnologies: Infrastructure for Collaborative Enterprise. (2005).
- [5] The Workflow Mangement Coalition. (199): Workflow Management Coalition Terminology and Glossary, technical report WfMC-TC-1011.
- [6] W.M.P. VAN DER AALST.''PROCESS-ORIENTED ARCHITECTURES FOR ELECTRONIC COMMERCE AND INTERORGANIZATIONAL WORKFLOW'', Eindhoven University of Technology.1999.
- [7] Lotfi BOUZGUENDA.'' Coordination Multi-Agents pour le Workflow Inter organisationnel Lâche''. Doctorat de L'Université Toulouse I, Spécialité INFORMATIQUE.2005.
- [8] Zhao, X. Liu, C. and Yang, Y. An organizational perspective on collaborative business processes. In Proceedings of the 3rd International Conference on Business Process Management, 2005.
- [9] Thomas Vantroys, Yvan Peter. ''Un système de workflows flexible pour la formation ouverte et à distance'', Université des Sciences et Technologies de Lille-Laboratoire TRIGONE –Equipe NOCE, 2002.
- [10] Amirreza Tahamtan, Modeling and Verification of Web Service Composition Based Inter organizational Workflows, Thèse de doctorat, Université de Vienna, 2009.
- [11] Chebbi Issam. Coop Flow : une approche pour la coopération ascendante de workflows dans le cadre des entreprises virtuelles. Thèse de doctorat, Institut National des Télécommunications, France. 2007.
- [12] FIPA Repository, <http://www.fipa.org/repository/aclspeccs.html>. (Consulté 22 /03/2019).
- [13] Lambrinouidakis, C. Kokolakis, S. Kar yda, M. Tsoumas, V. Gritzalis, D. and Katsikas, S Electronic voting systems: security implications of the administrative workflow. In Proceedings of the14th International Workshop on Database and Expert Systems Applications, 2003.
- [14] Budinska, I. Oravec, V. Gatial, E. Laclavik, M. Seleng, M. Balogh, Z. Frankovic, B. Forgac,R. Mokris, I. and Hluchy, L. A Knowledge Support System For Administrative Workflow Processes, In Proceedings of the Seventh International Conference on Application of Concurrency to System Design. 2007.

## Références bibliographiques

---

- [15] Muehlberger, R. Orłowska, M.E. and Kiepuszewski, B. Backward step: The right direction for production workflow systems. In Proceedings of the Australian Database Conference, 1999.
- [16] Leymann, F. Production workflow: concepts and techniques, Prentice Hall PTR Upper Saddle River, NJ, USA, 1999.
- [17] SAADOUN Mélissa. Technologies de l'information et management. Paris, 252p, Hermès Sciences, 2000.
- [18] G.T.S. Ho, H.C.W. Lau, C.K.M. Lee, A.W.H. Ip, and K.F. Pun. An intelligent production workflow mining system for continual quality enhancement, The International Journal of Advanced Manufacturing Technology, 28(7-8), pp.792-809, 2006.
- [19] Jiang, P. Mair, Q. and Newman, Using uml to design distributed collaborative workflows: from uml to xpd. In Proceedings of the Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003
- [20] Khetawat, A. Lavana, H. and Brglez, F. Collaborative workflows: A paradigm for distributed benchmarking and design on the internet. Technical report, North Carolina State University, 1997.
- [21] Huth, C. Erdmann, I. and Nastansky, L. Groupprocess: using process knowledge From the participative design and practical operation of ad hoc processes for the design of Structured workflows, In Proceedings of the 34th Annual Hawaii International Conference on System Sciences., 2001.
- [22] Levan, S.K, le projet workflow, 2ème édition, Eyrolles. [Manolis & Dimitris 2009]: Business Process Modeling and design: a formal model and methodology, tirage 2000.
- [23] Mlle MEBKHOUT Djohar et Mlle TERCHOUNE Nouzha, approche multi-agent pour la conception d'un workflow coopératif et administratif, Université IBN KHALDOUN-Tiaret, 2017.
- [24] Adjiri Hadjer, Les problèmes d'ordonnancement d'atelier: M-Machine identique en parallèle, 2018.
- [25] DERADRA Nesrine et KEDJAR Nawal, Ordonnancement du projet de rénovation de la base de vie de la station de pompage SP3-OB1 (W. M' Sila), Université Abderahmane Mira de Béjaia Faculté des Sciences Exactes, thèse de master, 2017.
- [26] Sebastian KANZOW, Optimisation d'ordonnancement publiquement, 2004.
- [27] Brahim S, mémoire Master, Conception et mise en œuvre d'un outil de vérification workflow, 2011.
- [28] c.jossin  
TRAVAIL\MAINT\Planification\_Ordonnancement\\_PLANIFICATION\_et\_Ordonnancement.doc
- [29] G.BAVIER Techniques d'ordonnancement.

## Références bibliographiques

---

- [30] BOUMEDIENE MEROUANE Hocine, Approche pour l'ordonnancement distribué de workflows dans le contexte D'entreprises virtuelles,
- [31] Mohamed Lamine Berrandjia, Approche multi-agents pour l'ordonnancement par coopération multi-ressources avec délais de production, thèse Magister, Sciences et Technologies de l'Information et de la Communication 2015.
- [32] Eunice Adjarath LEMAMOU, Ordonnancement de projet sous contraintes de ressources à l'aide d'un algorithme génétique à croisement hybride de type OER, 2009.
- [33] LES PROBLEMES D' ORDONNANCEMENT A MACHINES PARALLELES, DE TACHES DEPENDANTES d'allocation de ressource dans les cloud computing, 2016.
- [34] Résolution des Problèmes Multi Objectifs à Base de Colonies de FourmSAHA ADEL.
- [35] Eunice Adjarath LEMAMOU, Ordonnancement de projet sous contraintes de ressources à l'aide d'un algorithme génétique à croisement hybride de type OER, 2009.
- [36] RA V ARY JÉRÔME, ORDONNANCEMENT SOUS INCERTITUD, 2008.
- [37] Rationality and intelligence. Artificial Intelligence, Russell, Vol. 94. 1997.
- [38] Michael Wooldridge, Nicholas R. Jennings, « Intelligent Agents: Theory and Practice » The Knowledge Engineering Review, v.10, n°2, 115-152 p, 1995.
- [39] Yoav Shoham, « Agent-oriented programming », *Artificial intelligence*, v.60, n°1, 51-92 p, 1993.
- [40] Ferber Jacques, *Les systèmes multi-agents : vers une intelligence collective*, 1995.
- [41] Jörg P. Müller, *The design of intelligent agents. A layer approach*, Lecture Notes of Computer Science v.1177, Springer-Verlag, 1996.
- [42] Brenner Walter, Zarnekow Rüdiger, Wittig Harmut, *Intelligent Software Agents: Foundations and Applications*, Springer-Verlag, Berlin, 1998.
- [43] Loïc Péliissier, Régis Lhoste, Jade Java Agent DEvelopmentFramework.
- [44] Bonabeau Eric, Theraulaz Guy, Intelligence collective, Hermès, Paris, 1994.
- [45] Ferre L., "Selection of components in principal analysis: A comparaison of methods", Computational Statistics and Data Analysis pp. 669-682, 1995.

## Références bibliographiques

---

- [46] G.M.P. O'Hare, N.R. Jennings (Eds.), *Foundations of Distributed Artificial Intelligence*, Wiley & Sons, New York, 1996. OHA, 96],
- [47] Stan Franklin, Art Graesser, « Is it an agent or just a program ? : a taxonomy for autonomous agents », Springer-Verlag, 1996
- [48] Michael Wooldridge, Nicholas R. Jennings, « Pitfalls of Agent-Oriented Development », Minneapolis, USA, 385-391 p, 1998.
- [49] Ferber Jacques, « Technologie multi-agent », *Mémento : Les interfaces intelligentes et images de communication et l'images*, France Télécom Recherche, n°8, Octobre 1996.
- [50] S. Labidi, W. Lejouad, « De l'Intelligence Artificielle Distribuée aux Systèmes multi-agents », *Rapport de recherche n°2004*, INRIA, Août 1993.
- [51] Bouron T., Structure de communication et d'organisation pour la coopération dans un univers multi-agents, Thèse 3ème cycle, Université Paris VI, 1992.
- [52] Finin Tim, Labrou Yannis, Mayfield James, « KQML as an Agent Communication Language », Bradshaw Jeffrey (Eds.), 1997.
- [53] Labrou Yannis, Finin Tim, « History, State of the Art and Challenges for Agent Communication Languages », 17-24 p, 2000.
- [61] Downing K, Zvirinsky, The simulated evolution of biochemical guilds: Reconciling Gaia theory and natural selection, 1999.
- [62] Drogoul A, De la simulation multi agents à la résolution collective de problèmes: une étude de l'émergence de structure d'organisation dans les systèmes multi agents, thèse de doctorat, Université de Paris VI, 1993.
- [63] Duvallet C, Des systèmes d'aide à la décision temps réel et distribué : modélisation par agents. thèse de doctorat, Université du Havre, 2001.
- [64] Ghidini C, Serafini L'Information integration for electronic commerce, Minneapolis USA, 1998.
- [65] Guttman R, Maes P. Agent mediated interrogation negotiation for retail electronic commerce, p. 70-90, Minneapolis, USA, 1998.
- [66] Philippe Caillou. Master IAC 2013-2014
- [67] Olivier Boissier, DÉVELOPPEMENT DE SMA, MASTER WEB INTELLIGENCE 2010.
- [68] Giovanni Caire, JADE PROGRAMMING FOR BEGINNERS, thèse LICENSE, 2009.
- [69] Gauthier Picard et Laurent Vercouter, mémoire Initiation à la programmation orientée-objet avec le langage Java. Pôle Informatique 2014.