

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université IBN KHALDOUN– Tiaret
Département d'Informatique

Mémoire
Pour l'obtention du diplôme du MASTER
Spécialité : Génie Informatique
Option : Systèmes d'Information et Technologie de Web

**Approche Multi-agent pour la conception d'un
Workflow coopératif et administratif**

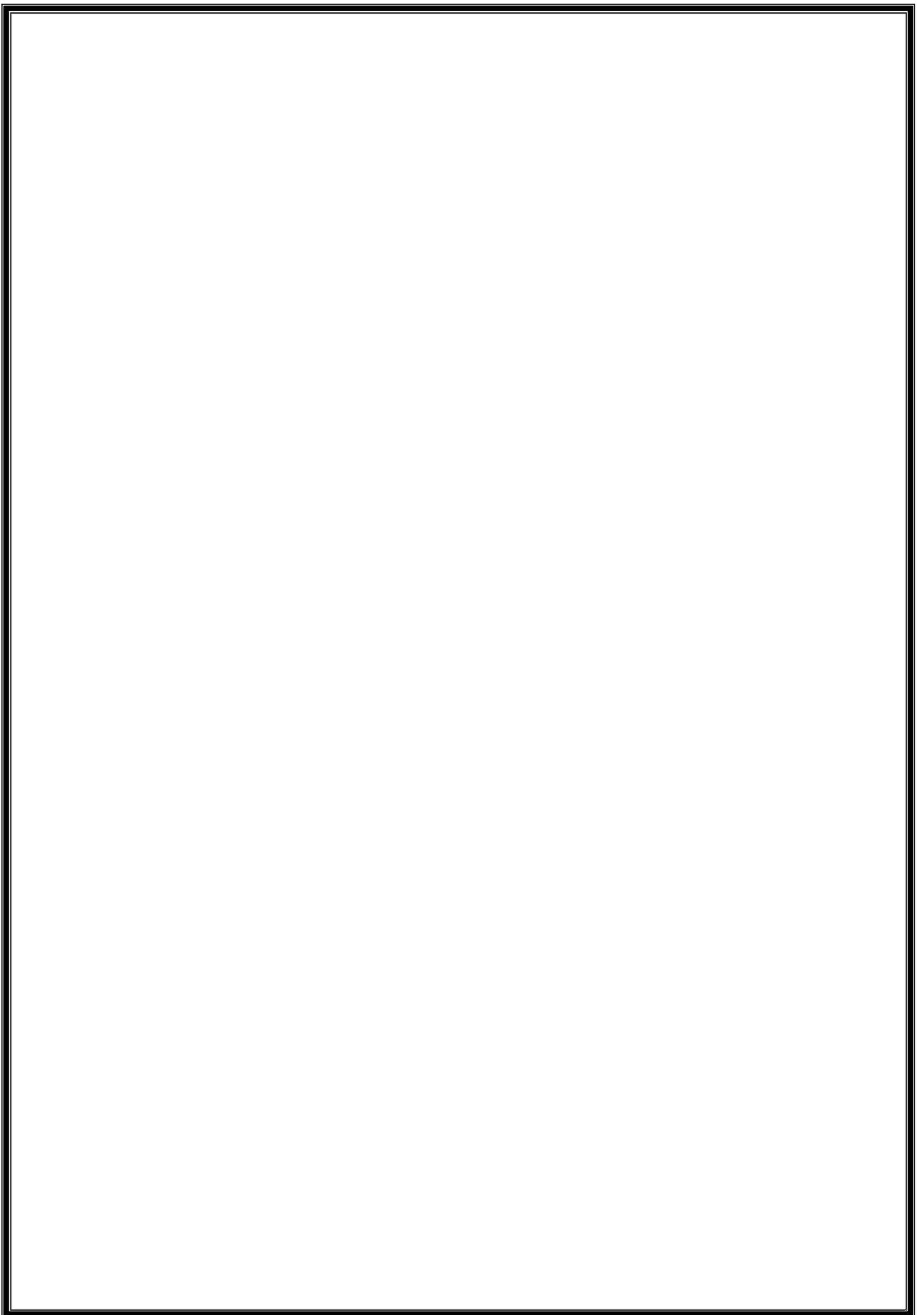
Réalisé par :

1) Mlle MEBKHOUT Djohar 2) Mlle TERCHOUNE Nouzha

Devant le jury composé de :

Mr MEBAREK BENDAOUED	MCA	Président
Mr MAASKRI MUSTAPHA	MAA	Examineur
Mr GOISMI MOHAMED	MAA	Encadreur

Année universitaire : 2016/2017



Remerciement

Au terme de cinq ans d'études, il est une obligation de dire Merci à tous ceux qui ont contribué à la réalisation de ce travail. Nous tenons tout d'abord à remercier ALLAH qui nous a donné la santé et le courage d'accomplir ce travail.

Nous ne pouvons pas oublier de présenter notre gratitude à nos parents pour leur patience et les efforts inlassables qu'ils ne cessent de déployer pour nous.

Nos vifs remerciements vont à Monsieur GOISMI Mohamed, notre promoteur pour tous ses conseils très précieux, ses encouragements, sa patience et ses orientations qui nous ont été très bénéfiques tout au long de ce travail.

Nous exprimons nos reconnaissances à 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, que tous ceux qui ont contribués de près ou de loin à l'aboutissement de ce travail, moralement ou matériellement trouvent ici l'expression de notre sincère gratitude et nos remerciements les plus sincères.

Dédicaces

Avec les sentiments de gratitude les plus profonds. Je dédie ce modeste travail :

A ma mère qui a gouverné mes premiers pas et qui n'a pas cessé de me soutenir dès le début de mes études jusqu'à l'heure où je suis aujourd'hui, pour toutes les peines endurées, toutes les privations et sacrifices consentis « Que la réussite de ce projet soit la récompense de ses efforts » ;

A mon défunt père (الله يرحمه) qui ouvre pour moi le chemin du savoir et que je regrette qu'il n'ait pas eu l'occasion de récolter les fruits de ses semences

A mes tendres frères « Khaled et Ahmed » et mes sœurs « Asma, Zahira, Kheira, Khadidja, Hadjer, Racha », pour tant de confiance, d'amour, de patience et d'abnégation. et mes nièces.

A ma famille de près ou de loin,

Il serait ingrat de ne pas songer à ma chère Binôme TERCHOUNE Nozha et toute sa famille.

Je ne me permettrais surtout pas d'oublier ma très chère amie Nedjma, Houda et Khaldia.

A tous mes collègues que j'ai eu l'occasion de connaître au cours de mes études. Que tous ceux qui se trouveraient omis ne me tiennent pas rigueur mais qu'ils trouvent ici l'expression de ma profonde reconnaissance.

A tous les gens qui m'ont aidé et soutenu tout au long de ce projet. Et à tous ceux que j'aime . .

MEBKHOUT Djohar

Dédicaces

Mon soutien moral et source de joie et de bonheur, celui qui s'est toujours sacrifié pour me voir réussir à toi mon père.

A la lumière de mes jours, la source de mes efforts ma vie et mon bonheur maman que j'adore.

A tous mes frères et mes sœurs, ((Amel, Ahlem, Mouhamed. Fatima)))

Mes nièces "Rimas", "Riham "

A mes cousines ((MERIEM, FATIMA, AICHA))

Aux personnes qui m'ont toujours aidé et encouragé, qui étaient toujours à mes côtés, et qui m'ont accompagnaient durant mon chemin d'études.

A mon binôme "DJOUHAR" mes aimables amies "IMANE, AMINA.S, ZAHRA, MERIEM, SIHAM A mes collègues d'étude.

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, aussi bien au sein de l'entreprise que dans le cadre d'une coopération inter-organisationnelle. Le workflow est un concept permettant de modéliser et de gérer des procédures industrielles ou administratives, impliquant plusieurs acteurs, documents et tâches. 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. L'objectif de ce mémoire est de développer une application workflow coopératif et administratif à l'aide d'un système multi-agent où nous avons mis en place un prototype permettant de concrétiser le processus de demande de congé à base d'agents intelligents.

Mots clés : *workflow coopératif et administratif, Systèmes multi-agents, AUML, JADE.*

Abstract

In recent years, companies have entered the era of computerized collaboration and, in order to remain competitive, they must constantly improve their operational efficiency. Accessibility to a growing volume of information and the integration of varied software solutions create new requirements for collaboration management tools, both within the company and through inter-organizational. Workflow is a concept for modeling and managing industrial or administrative procedures, involving several actors, documents and tasks. It consists of working models to coordinate the activities of each participant and to ensure their perfect interconnection by building on the existing information systems. The objective of this thesis is to develop a cooperative and administrative workflow application using a multi-agent system where we have put in place a prototype to realize the process of requesting leave based on intelligent agents.

Key words: *cooperative and administrative workflow, Multi-agent-systems, AUML, JADE.*

Introduction générale.....	1
----------------------------	---

Chapitre I : le workflow

1. Introduction.....	3
2. Concepts clés.....	3
2.1 Processus métier.....	3
2.2 Modèle de processus.....	3
2.3 Tâche.....	3
2.4 Nœud.....	4
2.4.1 type de nœuds.....	4
2.5 Transitions.....	4
2.6 Evènement.....	5
3. Définitions de workflow.....	5
3.1 Cycle de vie d'un workflow.....	5
3.2 Système de gestion de workflows.....	6
3.3 Le modèle de référence pour le workflow.....	7
3.4 Types de workflow.....	8
3.4.1 Workflow d'administration.....	8
3.4.2 Workflow de production.....	9
3.4.3 Workflow de collaboration ou coopératif.....	10
3.4.4 Workflow ad-hoc.....	10
3.5 Concepts de base de WF [22].....	11
3.6 Les typologies des applications de Workflow.....	12
3.6.1 Une typologie technique.....	12
3.7 Regroupement des workflows.....	14
3.8 Modélisation des workflow.....	14
3.8.1 L'aspect fonctionnel.....	15
3.8.2 L'aspect comportemental.....	15

3.8.3 L'aspect informationnel (donnée)	15
3.8.4 L'aspect organisationnel	15
3.9 Domaines d'application de workflow	15
3.10 Impacts du workflow	16
4. Conclusion	17

Chapitre II : Les Systèmes Multi-Agents

1. Introduction	18
2. Positionnement historique	18
3. L'agent.....	19
3.1 Caractéristiques de l'agent.....	19
3.2 L'architecture d'agent.....	21
3.3 Une comparaison avec les objets	21
3.4 classes d'agents.....	22
3.4.1 Nature	22
3.4.2 Utilisation	25
3.4.3 Technologie employée	26
4. Les systèmes multi-agents	26
4.1 Définition de système multi-agents (SMA).....	27
4.2 Problématique des SMA	28
4.2.1 Agent	28
4.2.2 Environnement	28
4.2.3 Interaction.....	28
4.2.4 Organisation	29
5.1 La communication	30
5.1.1 La communication par envoi des messages	30
5.2 Les langages de communication	31
5.3 La théorie des actes de langage.....	31

5.3.1 FIPA ACL	32
6. la coopération	33
7. Outils & plateformes de développement des SMA	34
8. Conclusion.....	35

Chapitre III : La plate-forme Jade

1. Introduction	36
2. Représentation de la plate-forme JADE	36
2.1 La norme FIPA pour les systèmes multi-agents	37
3. L'environnement JADE.....	38
3.1 Le langage de communication entre agents FIPA ACL	39
4. L'architecture de la plate-forme multi-agents	40
5. les agents techniques dans JADE « outils de débogage»	41
6. bibliothèque du JADE	45
7. Communication entre Agents JADE	45
7.1 Les Actes de communication	46
8. jade et l'ontologie	50
9. Conclusion.....	50

Chapitre IV : implémentation

1. Introduction	55
2. Type d'agents développés	55
3. modèle de workflow	57
4. Enumération des actes de communication des agents	58
5. Scénario d'exécution	59
6. Conclusion.....	69
Conclusion générale.....	71
Perspectives.....	72
Références bibliographiques	73

Figure 1.1: Cycle de vie d'un workflow [07]	6
Figure 1.2: Le modèle de référence du workflow.	7
Figure 1.3: Typologie fonctionnelle des applications de workflow	14
Figure 2.1: Différents types d'agents	20
Figure 2.2: Architecture minimale d'un agent	21
Figure 2.3: Architecture interne de l'agent réactif	23
Figure 2.4: Architecture interne de l'agent cognitif	24
Figure 2.5: Architecture interne de l'agent hybride	25
Figure 2.6: Représentation d'un SMA [25]	27
Figure 2.7: Structure d'un message FIPA ACL	33
Figure 3.1: Modèle de référence pour une plate-forme multi-agents FIPA	37
Figure 3.2: Architecture logicielle de la plate-forme multi-agents JADE	41
Figure 3.3: Agent RMA	42
Figure 3.4: Agent DF	43
Figure 3.5 : Agent Dummy	43
Figure 3.6 : Agent Sniffer	44
Figure 3.7: Agent Introspector	44
Figure 4.1 : Architecture distribuée de l'application	56
Figure 4.2 : Modèle de workflow	57
Figure 4.3 : Diagramme de séquence entre les agents de système	58
Figure 4.4: Fenêtre principal de l'agent serveur	60
Figure 4.5: fenêtre principale des agents clients	60
Figure 4.6: Fenêtre de l'agent serveur	61
Figure 4.7 : Formulaire pour créer un compte client	61
Figure 4.8: Liste des agents clients	62
Figure 4.9: Lancement de l'agent client	62
Figure 4.10: Fenêtre de l'agent client « Terchoune »	63
Figure 4.11: Fenêtres des deux agents clients "demandeur" et "remplaçant	63
Figure 4.12: Mise à jour des agents	64
Figure 4.13 : demande de conge rempli par le demandeur	64
Figure 4.14: Réception de la demande de conge par le Serveur	65
Figure 4.15: confirmation de l'agent client "remplaçant"	65

Figure 4.16: la réception de message de confirmation de la demande par l'agent serveur	66
Figure 4.17 : la réception de message de confirmation de la demande par l'agent client « demandeur »	67
Figure 4.18 : Liste des demandes confirmées	67
Figure 4.19 : SNIFER, contrôleur des messagers échangés entre les agents du système	68
Figure 4.20 : Envoie du message à l'agent client « Mebkhout »	69
Figure 4.21 : Réception du message de l'agent Serveur	69

Tableau 2.1: approche orientée objet (AOO) versus approche orientée agent (AOA) [25]	22
Tableau 3.1: Actes de communication de FIPA-ACL.....	49

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. Ces outils sont soit de type groupware qui est plus orienté vers le travail de groupe, soit de type workflow qui est un concept permettant de modéliser et de gérer des procédures industrielles ou administratives, impliquant plusieurs acteurs, documents et tâches. 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 est purement statiques, ce que lui donne une rigidité l'empêchant de s'adapter aux changements, cette limite a créé un problème de besoin d'adaptation, de flexibilité et de coopération entre les acteurs exécutant des 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 (IA) et plus particulièrement dans celui de l'Intelligence Artificielle Distribuée (IAD) s'intéressant particulièrement à la modélisation du comportement d'entités intelligentes distribuées dans un environnement. Les organisations multi-agent proposées dans ce domaine ne contraignent pas leurs agents à être présents dans un même lieu. Se communiquant des informations sur des sites physiquement distribués, les organisations multi-agents peuvent servir de support à la réflexion pour la conception de systèmes d'informations distribués [01].

L'objectif de ce mémoire est l'implémentation d'une application workflow d'un processus de demande de congé en utilisant la technologie des systèmes multi-agents en essayant de profiter de cette dernière, leurs atouts en termes d'autonomie des agents, coordination et coopération entre eux.

Le présent mémoire s'articule autour de quatre chapitres faisant un tour sur le domaine du workflow et les agents. Ainsi, nous décrivons un outil de développement des Système Multi Agent et nous présentons les résultats du travail réalisé et son implémentation sur un exemple.

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 a pour objectif de donner une brève introduction sur l'élément de base de notre architecture, il s'agit du paradigme agent donc nous allons donner une brève introduction sur les agents et les systèmes multi-agents, présentons des définitions, des types et des langages de communication

Le troisième chapitre, un bref rappel sur une des plateformes de développements et d'implémentations des systèmes multi- agents *Jade*.

Le quatrième chapitre, qui a pour rôle de donner et décrire le travail réalisé.

Nous clôturons ce mémoire par un bilan du travail effectué en donnant des conclusions sur les objectifs atteints, et les perspectives ouvertes pour le développement de ce travail.

Chapitre I : Le workflow

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 WfMC¹, 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.

2. Concepts clés

2.1 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).

2.2 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 .

2.3 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

¹ WfMC : Workflow Management Coalition

humaines et/ou machines. Quand une ressources et requise, la réalisation de l'activité est attribuée à un participant.

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

2.4 Nœud

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

2.4.1 type de nœuds

- Un (unique) *startNode* : étape à l'origine de la création de l'instance du workflow.
- *TaskNode* : étape en attente d'une interaction avec l'utilisateur.
- **State** : étape en attente d'une action extérieure (attente de réponse d'un composant externe a l'application).
- Un ou plusieurs *endNode* : Archivage de l'instance du workflow et libération des ressources.
- Les nœuds *task* et *State* ont un état *wait*.
- *Fork* : Séparation du Workflow en N branches devant se réunir via un nœud de type *join*.
- *DecisionNode* : condition sur une variable de l'instance de workflow (utilisation d'un langage simple ou délégation a une classe java).

2.5 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).

2.6 Evènement

Chaque nœud possède trois évènements déclencheurs :

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

3. Définitions de workflow

Selon [05],

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

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 workflow 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.

3.1 Cycle de vie d'un workflow

Le cycle de vie d'un workflow (Figure 1.1) est composé essentiellement de deux étapes [06].

- 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.

- 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.

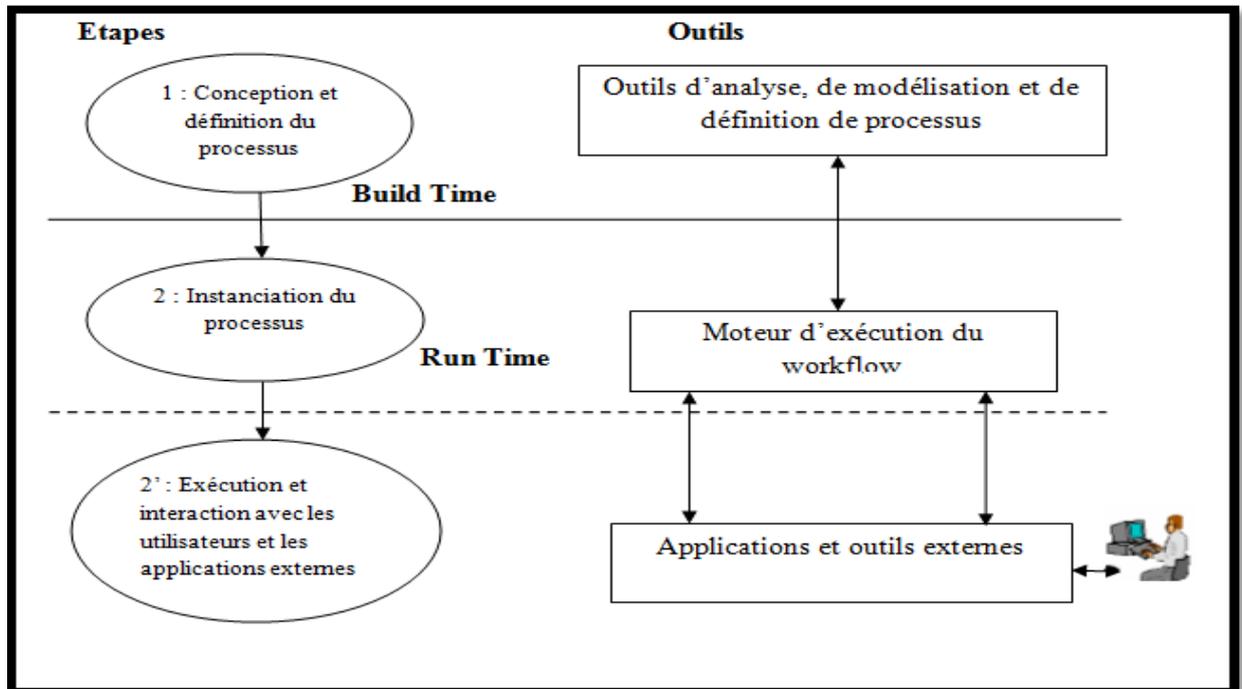


Figure 1.1: Cycle de vie d'un workflow [07]

Pour que les WFs² 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.

3.2 Système de gestion de workflow

Un système de gestion de Workflow (SGWf)³ 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

² WFs : Workflow

³ SGWF : Système de Gestion de Workflow

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.

3.3 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 1.2 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.

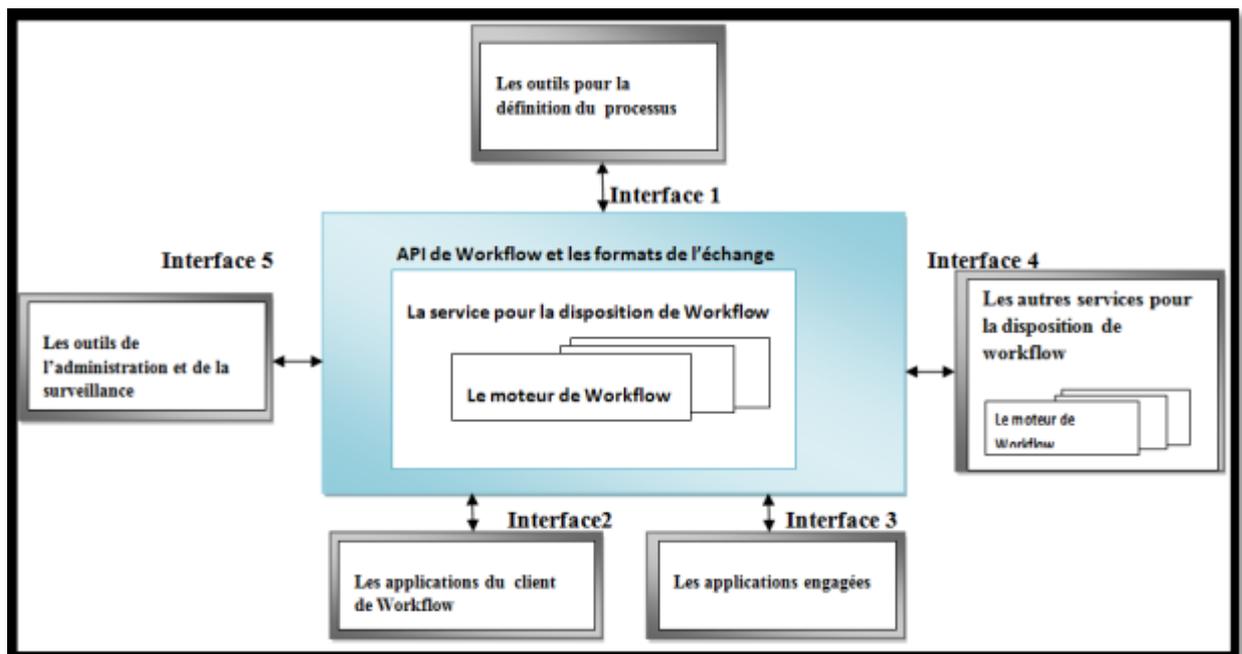


Figure 1.2: Le modèle de référence du workflow.

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

3.4 Types de workflow

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

3.4.1 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.

- **Processus concernés** : les processus de soutien de l'entreprise.
- **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...

3.4.2 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.).

- **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.
- **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...

3.4.3 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.

- **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...
- **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]

3.4.4 Workflow ad-hoc

Un workflow ad-hoc [21] représente une classe de workflows 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.

- **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.
- **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...

3.5 Concepts de base de WF [22]

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

a) 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.

b) 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.

c) Les rôles accomplissent des activités

Enfin, 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.

3.6 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].

3.6.1 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.

3.6.1.1 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 :

- 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>>.

- 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]

3.6.1.2 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 :

- 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.
- Le document reste accessible aux autres personnes pendant le déroulement du processus.
- 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]

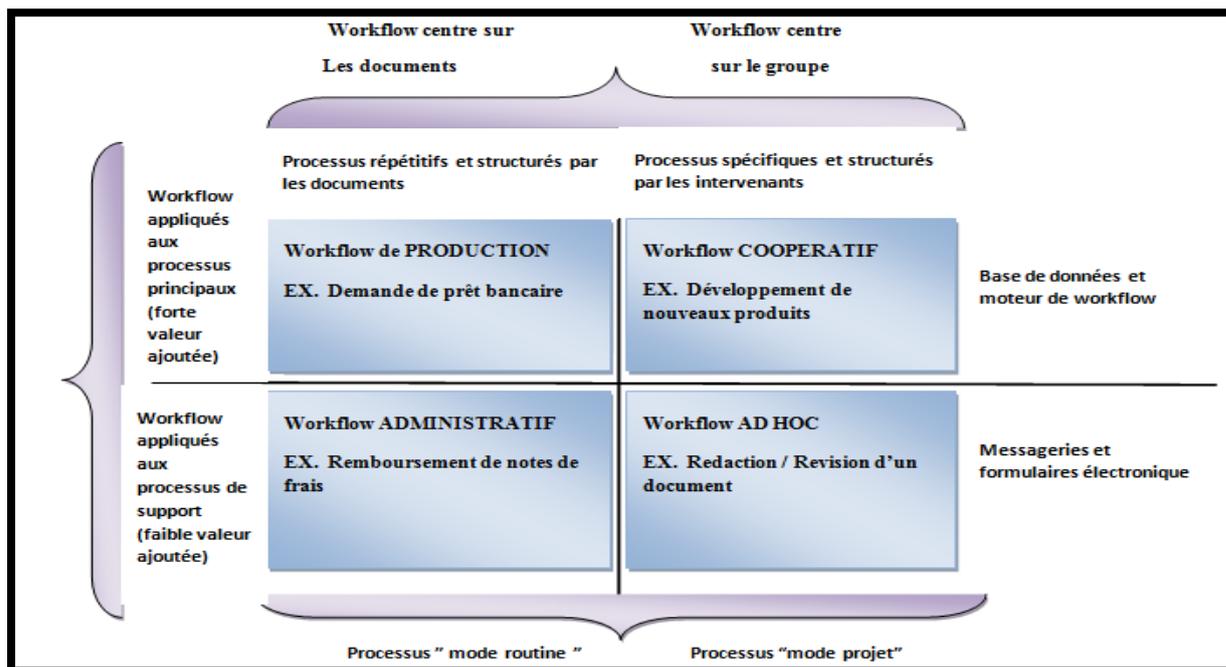


Figure 1.3: Typologie fonctionnelle des applications de workflow

3.7 Regroupement des workflows

Une première manière de regrouper les workflows est la suivante :

- les workflows documentaires (rédaction, validation, traduction de documents),
- 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.

3.8 Modélisation des workflow

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

3.8.1 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*).

3.8.2 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.

3.8.3 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.

3.8.4 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.

3.9 Domaines d'application de workflow

Les WFs ont de multiples applications dans le monde d'aujourd'hui. L'évolution des processus organisationnels de l'entreprise conduit à utiliser cet outil. Il répond à un besoin d'optimisation des processus de travail en termes d'utilisation des ressources et de temps effectif.

Le Workflow est amené à jouer un rôle important dans les entreprises du monde financier. On peut l'étendre à tout processus de travail cyclique dans le monde de l'entreprise.

On s'intéresse aussi à ses applications dans le monde informatique, comme le processus de développement de logiciel ; en intégrant l'aspect de travail coopératif au sein du workflow, on peut lier l'intégration progressive des éléments d'un logiciel avec l'organisation prévue. Le chef de projet dispose ainsi d'un outil de contrôle sur l'avancement du projet et la cohérence du système en termes de délais.

Les workflows peuvent également être utilisés dans des organisations autres que l'entreprise, comme dans le monde médical : suivi du dossier médical d'un patient (on peut le mettre à jour automatiquement selon les traitements médicaux effectués).

On peut imaginer des applications de workflow dans l'éducation par exemple la mise en place de processus de contrôle continu de l'apprentissage via le web.

3.10 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.

- **Gains tangibles**

Du côté des gains tangibles nous retrouvons les éléments suivants :

- ✓ **Réduction des coûts opérationnels** : les organisations utilisant des systèmes de workflow constatent une diminution des coûts de transition.
- ✓ **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.
- ✓ **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.

- **Gains intangibles** : ces gains 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.
 - ✓ **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.
 - ✓ **Facilitation du changement** : Les entreprises peuvent constamment, grâce aux systèmes de workflow, redéfinir et automatiser leur processus.
 - ✓ **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.
 - ✓ **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.
 - ✓ **Aide à la prise de décision** : Etant informé du déroulement des processus et des activités, il est plus facile de prendre les bonnes décisions.
 - ✓ **Amélioration du planning** : les informations disponibles concernant l'organisation, son business et ses processus améliorent les facultés de planning.
 - ✓ **Communication inter-entreprise** : La gestion de processus inter-entreprise augmente considérablement la productivité de la transparence du marché.

4. Conclusion

Dans ce chapitre, nous avons introduit le workflow, ces concepts essentiels et ces avantages, mais l'exécution de ces tâches est purement statiques, ce que lui donne une rigidité l'empêchant de s'adapter aux changements, cette limite a créé un problème de besoin d'adaptation et de flexibilité, de coopération entre les acteurs exécutant des 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) que nous allons la détailler dans le chapitre suivant.

Chapitre II : Les systèmes Multi-Agents

1. Introduction

Le domaine des systèmes multi-agents (SMA)⁴ n'est pas récent, il est actuellement un champ de recherche très actif. Cette discipline est à la connexion de plusieurs domaines en particulier de l'intelligence artificielle, des systèmes informatiques distribués et du génie logiciel. C'est une discipline qui s'intéresse aux comportements collectifs produits par les interactions de plusieurs entités autonomes et flexibles appelées agents, que ces interactions tournent autour de la coopération, de la concurrence ou de la coexistence entre ces agents.

Ce chapitre présente les concepts principaux de la technologie d'agent. Nous n'avons pas l'intention d'effectuer une étude détaillée, mais la définition des concepts nécessaires pour la lisibilité de ce travail.

2. Positionnement historique

Les systèmes multi-agents se positionnent au carrefour de la programmation de l'intelligence artificielle et des systèmes repartis.

Historiquement, on peut remplacer 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

⁴ SMA: Système Multi-agents.

appeler « systèmes multi-agents ». On utilise également le terme quasiment équivalent « IA⁵ distribuée » (avec son acronyme IAD, en anglais DAI⁶) pour bien montrer l'opposition à l'IA classique autracique 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 et peuvent donc avoir d'autres connotations dans d'autres domaines.

3. L'agent

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. Il y a un nombre important d'ouvrages offrant des définitions des agents et des systèmes multi-agents. 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 [23].
- 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 [24].
- 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 [25].
- 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 [26].

3.1 Caractéristiques de l'agent

En partant de l'ouvrage de [24], 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.

⁵ IA : Intelligence Artificielle

⁶ DAI: Distributed Artificial Intelligence.

- *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 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.

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 2.1.

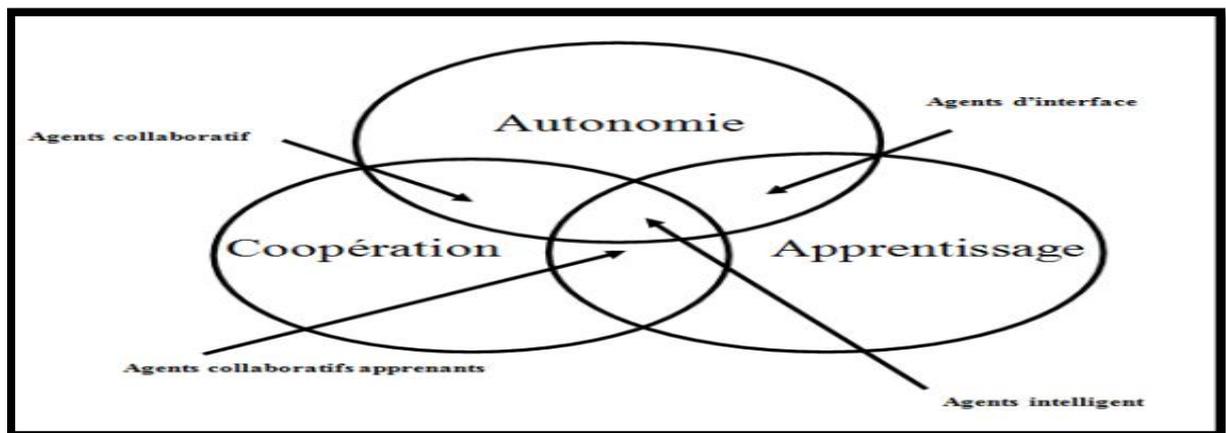


Figure 2.1: Différents types d'agents

Par exemple, dans la Figure 2.1, 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.

3.2 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 [27], [28].

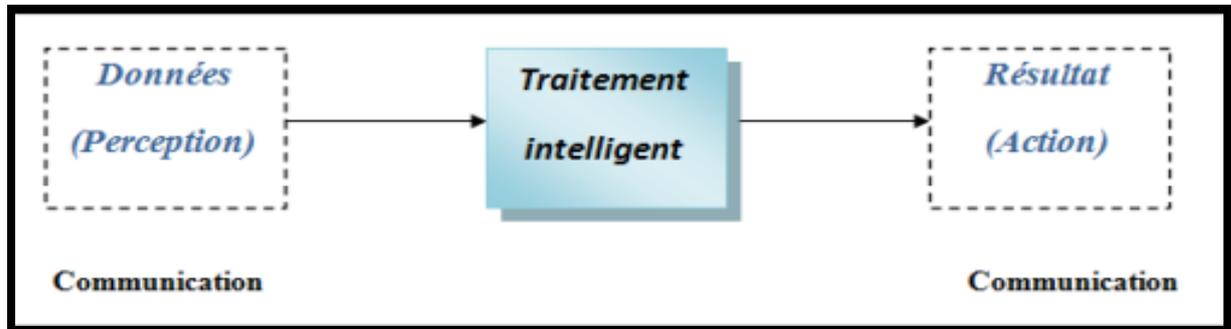


Figure 2.2: Architecture minimale d'un agent

L'architecture minimale (Figure 2.2) 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.

3.3 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.

Le tableau ci-dessous (Tableau 2.1) présente un récapitulatif des éléments de base de l'approche orientée objet versus l'approche orientée agent présenté par *Shoham*⁷ [25].

⁷Yoav Shoham : Professeur à Université de Stamford.

Propriété	AOO ⁸	AOA ⁹
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 message	Sans contraintes	Actes du langage
Contraintes sur les méthodes	Sans contraintes	Honnête, consistant

Tableau 2.1: approche orientée objet (AOO) versus approche orientée agent (AOA) [25]

3.4 classes d'agents

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

3.4.1 Nature

On trouve trois types d'agents différents :

3.4.1.1 Agents réactifs

Ce sont des agents qui :

- fonctionnent selon un mode stimuli/réponse (basé sur la règle perception/ action).
- ne possède pas une représentation complète de son environnement et n'est pas capable de tenir compte de ses actions passées.
- De ce fait, il ne peut avoir des capacités d'apprentissage.
- L'architecture interne de cet agent est représentée par la figure suivante :

⁸AOO:ApprocheOrientéeObjet.

⁹AOA:ApprocheOrientéeAgent

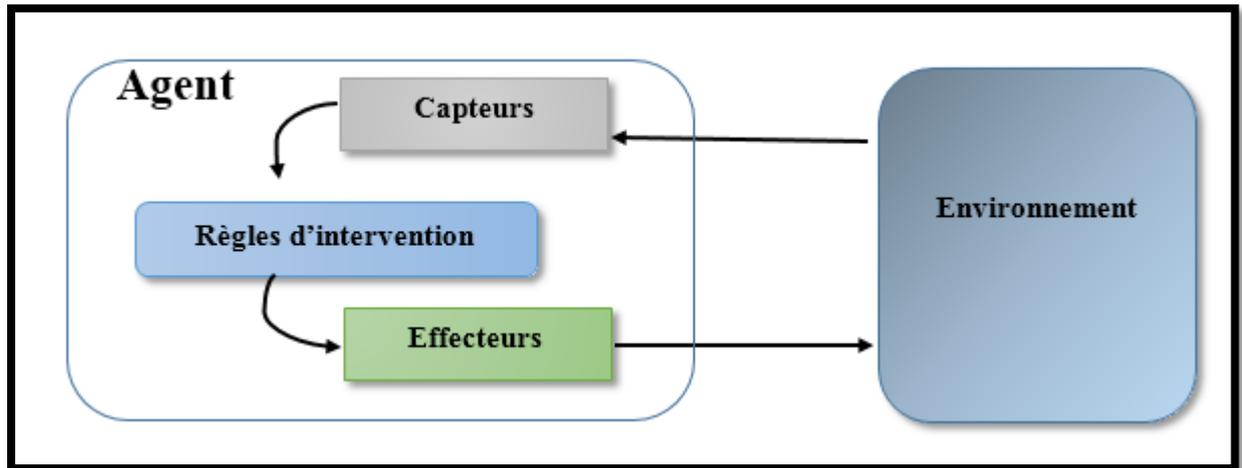


Figure 2.3: Architecture interne de l'agent réactif

- ✓ les capteurs : qui servent à détecter un évènement survenu dans l'environnement.
- ✓ les tâches : qui décrivent une partie du comportement de l'agent.
- ✓ les effecteurs : qui permettent de réaliser les tâches.

3.4.1.2 Agents délibératifs

Ils se basent sur un modèle qui provient d'une métaphore du modèle humain. Ces agents possèdent une représentation explicite de leur environnement, des autres agents et d'eux-mêmes. Ils sont aussi dotés de capacités de raisonnement, planification et de communication. Ce modèle est appelé aussi modèle BDI (Believe, Desire, Intention). Il est fondé sur trois attitudes qui définissent la rationalité d'un agent intelligent : B = Beliefs = Croyances : Il s'agit de l'ensemble d'informations que possède l'agent sur son environnement et sur les autres agents agissant dans le même environnement. Ces informations constituent les connaissances supposées être vraies pour l'agent. D = Desires = Desirs : Ils constituent les différents états de l'environnement et parfois de l'agent lui-même, on peut dire que ce sont les objectifs que se fixe un agent. I = Intentions = Intentions : Ce sont les actions à faire par l'agent pour accomplir ses desirs. Ils se composent de l'ensemble des plans exécutés pour atteindre ses objectifs.

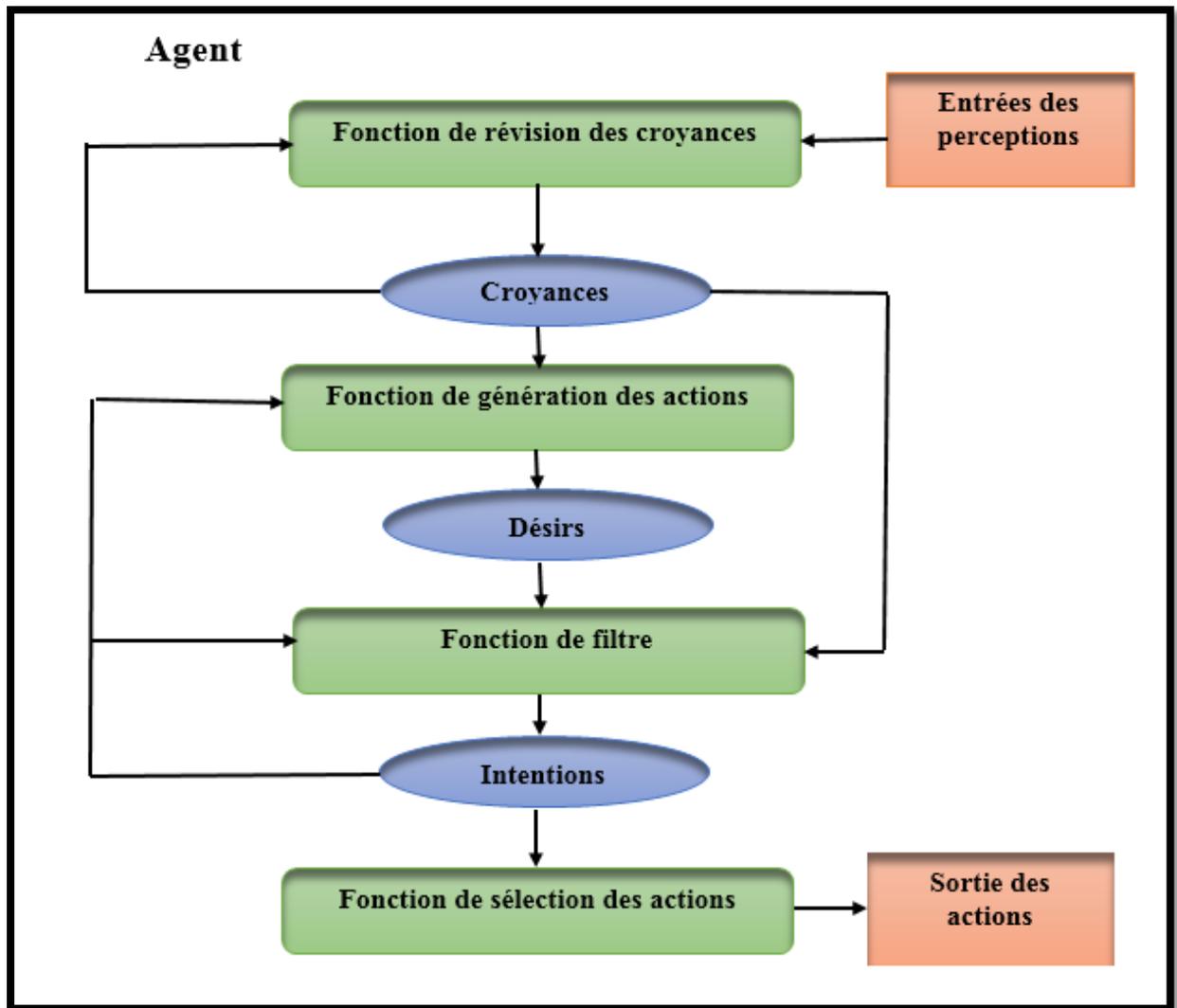


Figure 2.4: Architecture interne de l'agent cognitif

3.3.1.3 Agents hybrides

Ce sont des agents ayant à la fois des capacités cognitives et réactives. Ils conjuguent la rapidité de réponse des agents réactifs ainsi que les capacités de raisonnement des agents cognitifs. Dans ce modèle, les agents sont conçus comme étant composés de niveaux hiérarchiques qui interagissent entre eux. Chaque niveau gère un aspect du comportement de l'agent [29] :

- Bas niveau : couche réactive qui est en relation directe avec l'environnement et qui raisonne suivant les informations brutes de ce dernier.
- Niveau intermédiaire : couche mentale qui fait abstraction des données brutes et qui se préoccupe d'un aspect de l'environnement plus évolué.

- Niveau supérieur : couche sociale qui tient compte de l'interaction avec les autres agents.

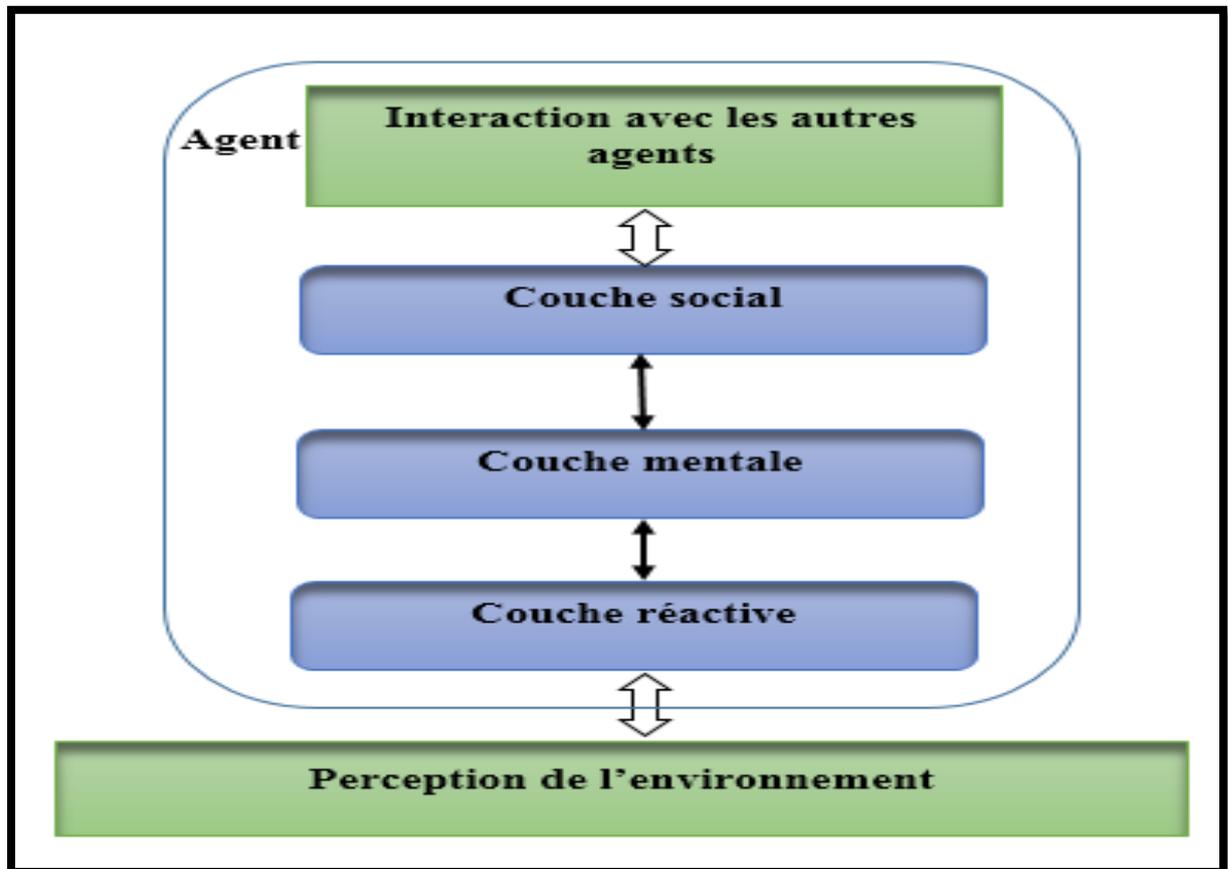


Figure 2.5 : Architecture interne de l'agent hybride

3.4.2 Utilisation

Regroupe les agents aux quatre catégories suivantes :

3.3.2.1 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.

3.4.2.2 Agents d'interface

Ces agents collaborent avec l'utilisateur pour effectuer certaines tâches.

3.4.2.3 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.

3.4.2.4 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.

3.4.3 Technologie employée

Cette dernière contient les deux types :

3.4.3.1 Agents stationnaires

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

3.4.3.2 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 : Qu'est-ce qu'un système multi-agents ?

4. Les systèmes multi-agents

Lorsque plusieurs agents se retrouvent dans un même environnement et que ces agents ont besoin d'interagir entre eux, on parle alors de système multi-agents. Après avoir défini le concept d'agent, son architecture et son fonctionnement, nous nous intéressons dans cette section aux SMA.

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 [30]. Bien qu'il soit difficile de donner une définition

générale pour l'expression « Système Multi-Agents "SMA" » les différentes caractérisations proposées par [26], [31], [32], [33], [34] et autres permettent de dégager des idées fondamentales communes. la définition, mentionnée ci-dessous, est conforme à la plupart de ces idées.

4.1 Définition de système multi-agents (SMA)

De nombreuses définitions ont été proposées pour les SMA présentant chacune une vision propre de l'auteur. *Ferber* [26] définit un SMA comme étant un système composé d'un environnement, un ensemble d'agents qui représentent des entités actives, un ensemble d'objets passifs qui peuvent être perçus, créés, détruits et modifiés par les agents et un ensemble de relations qui unissent les objets ou les agents entre eux (figure 2.6). *Demazeau* met l'accent sur deux autres axes caractérisant les SMA : les interactions et l'organisation [35]. Les agents d'un SMA interagissant, le plus souvent, selon des modes de coopération, de concurrence ou de coexistence [36]. Dans un SMA, il n'y a aucun contrôle global et les données sont décentralisées [36].

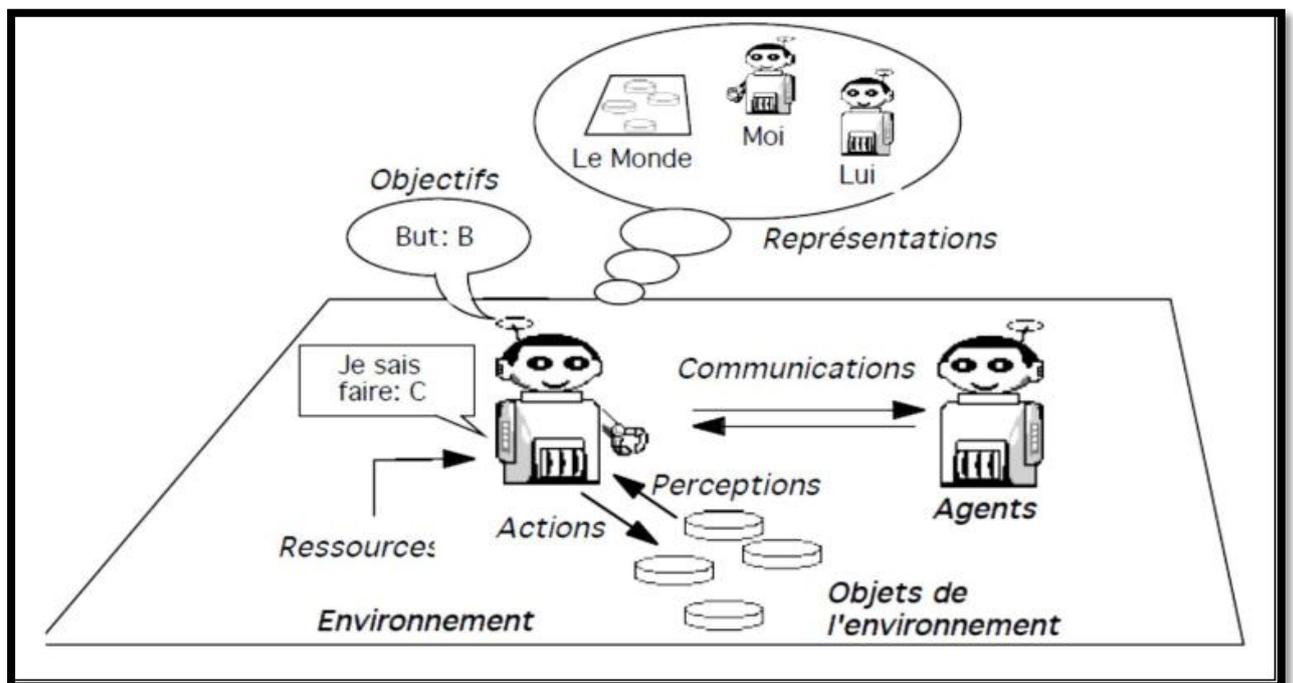


Figure 2.6: Représentation d'un SMA [25]

4.2 Problématique des SMA

L'étude d'un système multi-agent est faite sur un ensemble d'axes, nous introduisons la vision donnée par *Yves Demazeau* appelée la décomposition Voyelles AEIO (Agent, Environnement, Interactions, Organisation).

4.2.1 Agent

Le concept d'agent a été défini plus haut.

4.2.2 Environnement

L'environnement est le milieu dans lequel les agents sont introduits. Son rôle est de permettre la communication entre agents, il supporte les actions des agents en définissant les règles et en renforçant ces actions, il est observable par les agents, et prend en charge l'activité propre des objets et des ressources présentes. Lorsque les agents sont réactifs, l'environnement détient une importance capitale car il est le médiateur de leurs interactions. En effet, comme ces agents ne peuvent communiquer directement entre eux, ils s'influencent mutuellement soit par leurs positions s'ils sont situés, soit par l'intermédiaire d'objets qu'ils perçoivent et modifient.

D'après *Russel* [38] et *Woodbridge* et al [39] Un environnement peut être :

- Accessible si un agent peut, à l'aide des primitives de perception, déterminer l'état de l'environnement et ainsi procéder, par exemple, à une action. Si l'environnement est inaccessible alors il faut que l'agent soit doté de moyens de mémorisation afin d'enregistrer les modifications qui sont intervenues.
- Déterministe, ou non, selon que l'état futur de l'environnement ne soit, ou non, fixé que par son état courant et les actions de l'agent.
- Episodique si le prochain état de l'environnement ne dépend pas des actions réalisées par les agents.
- Statique si l'état de l'environnement est stable (ne change pas) pendant que l'agent prend des décisions. Dans le cas contraire, il sera qualifié de dynamique.
- Discret si le nombre des actions faisables et des états de l'environnement est fini.

4.2.3 Interaction

L'interaction est une notion très complexe et variée dans le domaine des SMA, dans ce qui suit nous donnons quelques aspects concernant cette notion. Une interaction est une mise

en relation dynamique de deux ou plusieurs agents par le biais d'un ensemble d'actions réciproques [27]. Les interactions s'expriment ainsi à partir d'une série d'actions dont les conséquences exercent en retour une influence sur le comportement futur des agents. Les agents interagissent le long d'une suite d'événements pendant lesquels les agents sont d'une certaine manière en contact les uns avec les autres, que ce contact soit direct ou qu'il s'effectue par l'intermédiaire d'un autre agent ou de l'environnement.

L'interaction est caractérisée par : les langages de communication et les protocoles d'interaction.

On distingue différents types d'interaction que les agents peuvent adopter tels que : la coopération, la négociation et la coordination. Ces types sont à la base :

- Des buts des agents qui peuvent être compatible ou incompatible.
- Des ressources extérieures pour la réalisation des tâches des agents.
- Des capacités des agents par rapport aux tâches à accomplir.
- De la situation d'interaction qui est la combinaison des trois facteurs précédents (but, ressource, compétence).

4.2.4 Organisation

L'organisation désigne un ensemble d'agents travaillant ensemble pour la réalisation d'une ou plusieurs tâches. Le concept d'organisation peut être exprimé à partir des concepts plus élémentaires d'agent et de tâches. Par rapport au concept de tâche, l'organisation désigne les processus qui permettent la décomposition des tâches en sous-tâches, l'allocation des tâches aux agents et l'accomplissement des tâches dépendantes de façon cohérente. Par rapport au concept d'agent l'organisation détermine les statuts et les comportements sociaux des agents (les rôles) et les relations qui permettent d'unir les agents au sein d'un groupe, que ce soit vis-à-vis de la décision (les liens d'autorité) ou vis-à-vis de la coordination (les liens d'engagement) [40].

La structure d'une organisation peut être simplement définie comme la somme totale des moyens employés pour diviser le travail en tâches distinctes et pour assurer la coordination nécessaire entre les tâches. [41].

5. les interactions et les agents

L'interaction est une notion importante dans les systèmes multi-agents. Au dire de *J.Ferber* [26], 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 [42]. 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 leurs buts. Nous présentons le concept d'une de ces formes - la communication - sur laquelle repose toute l'interaction.

5.1 La communication

La communication est essentielle dans la résolution coopérative des problèmes [26] [30]. Elle permet de synchroniser les actions des agents et de résoudre les conflits des ressources par la négociation. D'après [39], 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.

5.1.1 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.

5.2 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 [26]. La question qui se pose alors, est comment réaliser les interactions des agents ? D'après [42], 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.

5.3 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 » ¹⁰[31]. Un acte de langage définit un message, qui contient l'affirmation positive ou négative, et provoque les changements de l'environnement. 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, FIPAACL*).

5.3.1 FIPA ACL

FIPA¹¹ a proposé un langage de communication d'agent FIPA ACL¹², Ce langage est fondé sur la théorie des actes de langage. 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 mentales des agents émetteurs et des agents récepteurs. La sémantique formelle de *FIPA ACL* se compose de cinq niveaux [44] :

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

¹⁰ 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* ».

¹¹ FIPA : Foundation for Intelligent Physical Agents.

¹² FIPA ACL: FIPA Agent Communication Language.

- 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, employées dans le contenu.

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

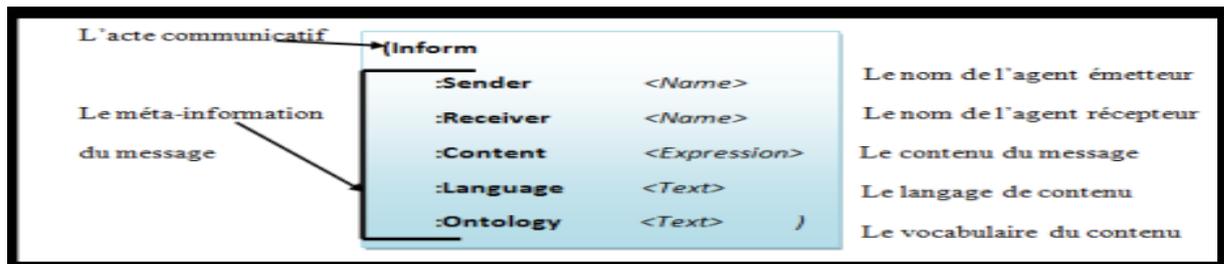


Figure 2.7: Structure d'un message FIPA ACL

FIPA ACL possède 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. *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. En plus, *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.

6. la coopération

La coopération est l'une des caractéristiques fondamentales des systèmes multi-agents. D'après [44], 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. *Ferbert* présente deux points de vue sur la coopération [26], [45] :

- 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**¹³ [26] 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.

7. 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

¹³ **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.

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

8. Conclusion

Dans ce chapitre, nous avons présenté les systèmes multi-agents qui ont des possibilités d'interaction entre les différentes entités (agents) ainsi que des possibilités de raisonnement et d'apprentissage des agents .Cette technologie est utilisée dans les processus complexes qui ne peuvent être prédéfinis où les problèmes à résoudre sont de nature distribuée et nécessitant une certaine autonomie dans la décision et la coopération. Dans le chapitre suivant, nous allons vous présenter Un moyen utilisé pour construire un système multi-agent qui est la plate-forme jade.

Chapitre III : la plate-forme JADE

1. Introduction

Pour construire un SMA il est préférable d'utiliser une plate-forme multi-agent qui offre un ensemble d'outils utilisé pour la construction et la mise en service d'agents au sein d'un environnement spécifique. Les outils peuvent être utilisés pour analyser les SMA, créer les SMA ou bien tester les SMA. Ces outils peuvent être sous la forme d'environnement de programmation (API) et d'applications permettant d'aider le développeur à la programmation d'un SMA ainsi que son débogage. Il existe de nombreuses plates-formes pour le développement de SMA : JACK, Jadex, Madkit, JADE..etc.

Nous avons opté dans le cadre de notre travail pour la plateforme JADE (Java Agent DEvelopment Framework) pour les avantages qu'elle procure et qui nous allons la présenter en détail dans le reste de ce chapitre.

2. Représentation de la plate-forme 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** [41]. JADE comprend deux composantes de base : une plate-forme agents compatible FIPA et un paquet logiciel pour le développement des agents Java.

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

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

2.1 La norme FIPA pour les systèmes multi-agents

Les premiers documents de spécification de la norme FIPA, appelés spécifications FIPA97, 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 11**) 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.

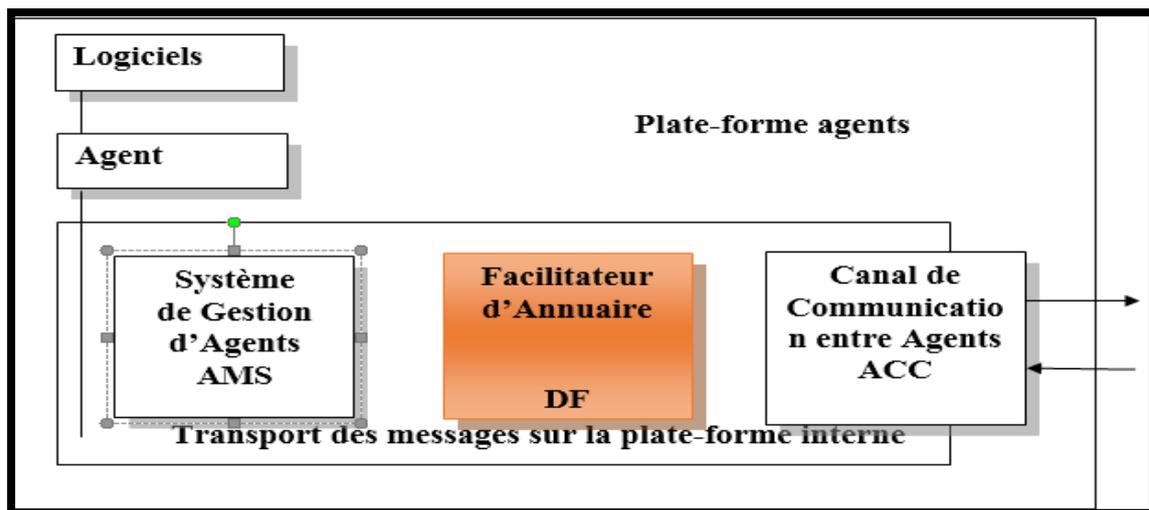


Figure 3.1: Modèle de référence pour une plate-forme multi-agents FIPA

Dans la **Figure 3.1**, 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**¹⁴) 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**¹⁵ pour l'interopérabilité entre les différentes plates-formes multi-agents.

¹⁴ **ACC**: Agent Communication Channel.

¹⁵ **IIOP** : Protocole de communication permettant d'intégrer et de mettre en réseau des applications de provenance diverse

- 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.

3. L'environnement JADE

Le but de *JADE* est de simplifier le développement des systèmes multi-agents en conformité avec la norme *FIPA* pour réaliser des systèmes multi-agents interopérables. Pour atteindre ce but, *JADE* offre la liste suivante de **caractéristiques** au programmeur d'agents :

- **La plate-forme multi-agents compatible FIPA**, qui inclut le Système de Gestion d'Agents (**AMS**), le Facilitateur d'Annuaire (**DF**), et le Canal de Communication entre Agents (**ACC**). Ces trois agents sont automatiquement créés et activés quand la plate-forme est activée.

La plate-forme d'agents peut être distribuée sur plusieurs hôtes, à condition qu'il n'y ait pas de pare-feu entre ces hôtes. Une seule application Java, et donc une seule Machine Virtuelle Java, est exécutée sur chaque hôte. Les agents sont implémentés comme des threads d'exécution Java et les événements Java sont utilisés pour la communication efficace et légère entre agents sur un même hôte. Un agent peut exécuter des tâches parallèles et *JADE* planifie ces tâches d'une manière plus efficace (et même plus simple pour le programmeur) que la planification faite par la Machine Virtuelle Java pour les threads d'exécution.

- Un certain nombre de **DF** (Facilitateurs d'Annuaire) compatibles *FIPA* qui peuvent être activés quand on lance la plate-forme pour exécuter les applications multi_domaines.
- Une interface de programmation pour simplifier l'enregistrement de services d'agents avec un ou plusieurs domaines de type **DF**.

- Le mécanisme de transport et l'interface pour l'envoi et la réception des messages.
- Le protocole **IIOP** pour connecter des plates-formes multi-agents différentes.
- Le transport léger de messages **ACL** sur la même plate-forme d'agents. Dans le but de simplifier la transmission, les messages internes (sur la même plate-forme) sont transférés et codés comme des objets Java et non comme des chaînes de caractères. Quand l'expéditeur ou le récepteur n'appartient pas à la même plate-forme, le message est automatiquement converti en chaîne de caractères. De cette façon, la conversion est cachée au programmeur d'agents qui a seulement besoin de traiter la classe d'objets Java.
- L'enregistrement automatique d'agents dans le Système de Gestion d'Agents (**AMS**).
- Un service d'attribution de noms compatible *FIPA* ; quand on lance la plate-forme, un agent obtient un identificateur unique (GUID¹⁶).
- Une interface graphique utilisateur pour gérer plusieurs agents et plates-formes multi-agents en partant d'un agent unique. L'activité de chaque plate-forme peut être supervisée et enregistrée.

3.1 Le langage de communication entre agents FIPA ACL

La communication entre les agents permet de lier un ensemble d'agents et permet d'augmenter les capacités perceptives des agents en leur permettant de bénéficier des informations et du savoir-faire des autres agents. Sans communication, un agent n'est qu'un individu isolé, sourd et muet qui ne fait qu'agir sur lui-même. La communication donc s'appuie sur des langages de communication standard.

FIPA ACL «FIPA Agent Communication Language » est l'un des langages de communication entre les agents créé par FIPA dont la spécification consiste en un ensemble de types de message («actes de communication ») et en un ensemble de protocoles d'interaction de haut niveau. Cependant, FIPA ACL est indépendant de la langue et le contenu est conçu pour fonctionner avec n'importe quel contenu de la langue et avec toutes les approches de spécification de l'ontologie.

- Le but de la FIPA ACL est d'interagir entre les agents quel que soit le protocole qu'ils utilisent. Pour que les agents puissent supporter/utiliser un nouveau protocole, il suffit

¹⁶ **GUID**: Globally Unique Identifier.

d'implémenter une nouvelle interface. Cependant, Dans FIPA ACL, la communication des agents est basée sur l'envoi de messages.

4. L'architecture de la plate-forme multi-agents

La plate-forme offre une interface graphique utilisateur (**GUI**¹⁷) pour la gestion à distance qui permet de contrôler et superviser les états des agents, par exemple arrêter et remettre en marche un agent. L'interface graphique permet aussi de créer et de commencer l'exécution d'un agent sur un hôte éloigné, à condition qu'un réceptacle d'agents s'exécute déjà sur cet hôte. L'interface elle-même a été implémentée comme un agent, appelé **RMA (Remote Monitoring Agent)**.

La propriété importante d'un agent est son autonomie : un agent ne doit pas se limiter à réagir aux événements externes, mais il doit être aussi capable de prendre l'initiative de nouveaux actes communicatifs d'une façon autonome. Ceci exige que chaque agent ait un thread interne de contrôle ; cependant, un agent peut engager des conversations simultanées multiples, tout en poursuivant d'autres activités qui n'impliquent pas d'échanges de messages.

JADE utilise l'abstraction *Comportement* pour modéliser les tâches qu'un agent peut exécuter et les agents instancient leurs comportements selon leurs besoins et leurs capacités.

De point de vue de la programmation concurrente, un agent est un **objet actif**, ayant un thread de contrôle. JADE utilise un modèle de programmation concurrente "un thread par agent" au lieu d'un modèle "un thread par comportement" pour éviter une augmentation du nombre de threads d'exécution exigés sur la plate-forme d'agents. Ceci signifie que, pendant que les agents différents s'exécutent dans un environnement multithreads de préemption, deux comportements d'un même agent sont planifiés coopérative ment.

En dehors de la préemption, les comportements travaillent tous comme des threads d'exécution coopératifs, mais il n'y a pas de pile qui ait besoin d'être sauvée. Un planificateur (Scheduler), exécuté par la classe de base *Agent* et caché au programmeur, exécute une politique de "round robin" de non préemption entre tous les comportements disponibles dans la file des processus prêts. Ainsi, il permet l'exécution d'une classe dérivée de la classe *Comportement* jusqu'à ce qu'elle abandonne le contrôle d'exécution par elle-même. Si la tâche qui a le contrôle n'est pas encore finie, elle sera re-planifiée pendant le prochain tour du round-robin à moins

¹⁷ **GUI**: **G**raphical **U**ser **I**nterface.

qu'elle ne soit pas bloquée ; en fait, un comportement peut se bloquer lui-même, par exemple pendant qu'il attend des messages, pour éviter le gaspillage de temps de CPU, réalisant ainsi un comportement d'attente occupée. Donc, le développeur d'agents doit étendre la classe Agent et implémenter les tâches spécifiques de l'agent par une ou plusieurs classes Comportement, les instancier et les ajouter à l'agent. La classe Agent représente une super-classe commune pour tous les agents définis par l'utilisateur. Du point de vue du programmeur, la conséquence est qu'un agent JADE est simplement une classe Java qui étend la classe de base Agent. Cela permet à l'agent d'hériter un comportement fondamental caché (qui traite toutes les tâches liées à la plate-forme, telles que l'enregistrement, la configuration, la gestion à distance, etc.), et un ensemble de méthodes qui peuvent être appelées pour implémenter les tâches spécifiques à l'agent, par exemple envoi des messages, utilisation des protocoles d'interaction standard, enregistrement sur plusieurs domaines, etc. De plus, il y a encore deux méthodes qui sont héritées pour gérer la file de comportements d'agents : *addBehaviour (Behaviour)* et *remove Behaviour (Behaviour)*. JADE inclut aussi quelques comportements prêts à être utilisés pour les tâches les plus communes dans la programmation des agents, tels que l'envoi et la réception des messages et la décomposition des tâches complexes en des agrégations de tâches plus simples.

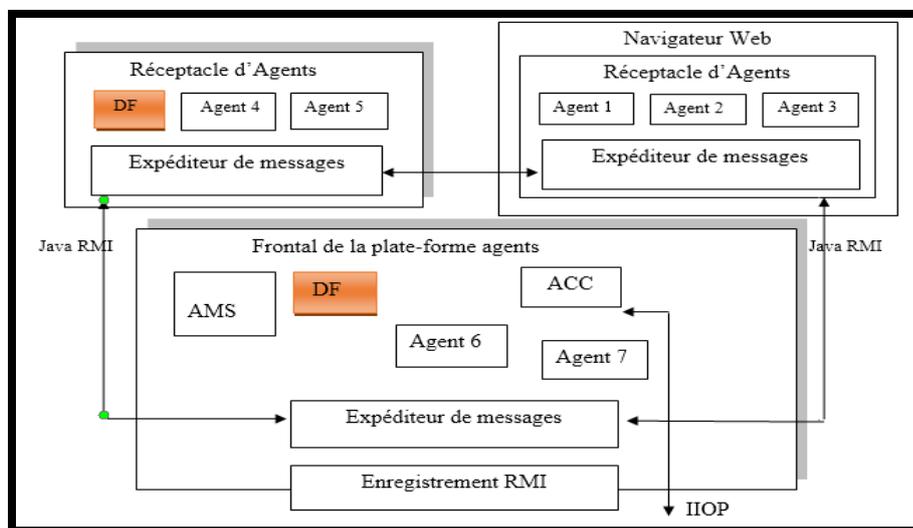


Figure 3.2: Architecture logicielle de la plate-forme multi-agents JADE

5. les agents techniques dans JADE « outils de débogage »

Pour supporter la tâche difficile du débogage des applications multi-agents, la plateforme possède l'ensemble d'agents prédéfinis suivant :

Remote Management Agent « RMA » : le RMA permet de contrôler le cycle de vie de la plateforme et tous les agents la composant. Avec son interface, il est possible de créer, supprimer et de migrer des agents. L'agent RMA permet également la création et la suppression des conteneurs et la fermeture de la plateforme. Il peut être lancé à partir de la ligne de commande **Java jade.Boot -gui**.

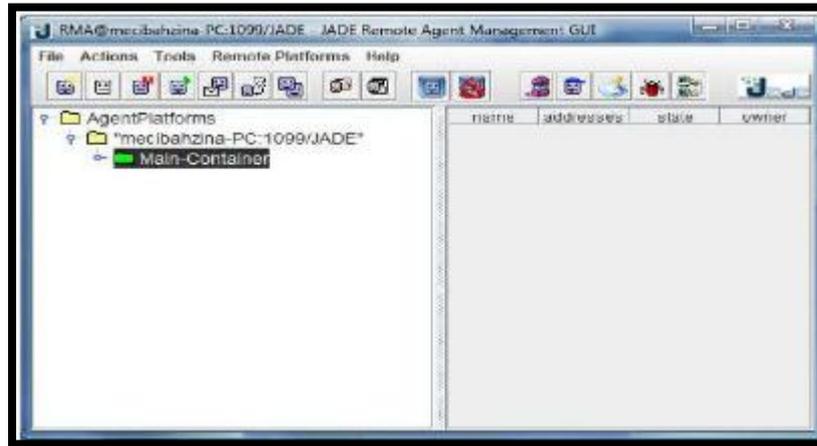


Figure 3.3: Agent RMA

Agent Management System « AMS » : est un agent chargé de la supervision et le contrôle de l'accès et l'usage de la plateforme. Chaque agent doit s'inscrire auprès de l'AMS pour obtenir un AID valide.

Agent Communication Canal « ACC » : c'est le composant logiciel qui contrôle tous les échanges de messages dans la plateforme, incluant également les messages de/vers des plateformes distantes.

Directory Facilitator « DF » : enregistre les descriptions et les services des agents et maintient des pages jaunes de services. Grâce à ce service, un agent peut connaître les agents capables de lui fournir les services qu'il requiert pour atteindre son but. L'interface du DF peut être lancée à partir du menu du RMA.

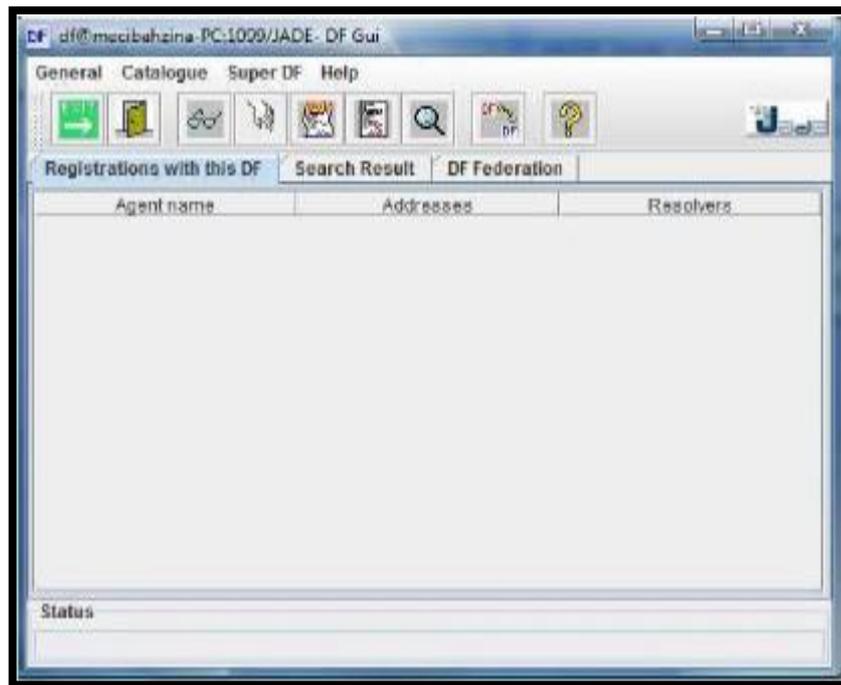


Figure 3.4: Agent DF

Dummy Agent « DA » : l'outil *DummyAgent* permet aux utilisateurs d'interagir avec les agents JADE d'une façon particulière. L'interface permet la composition et l'envoi de messages ACL et maintient une liste de messages ACL envoyés et reçus. Cette liste peut être examinée par l'utilisateur et chaque message peut être vu en détail ou même édité. Plus encore, le message peut être sauvegardé sur le disque et renvoyé plus tard.

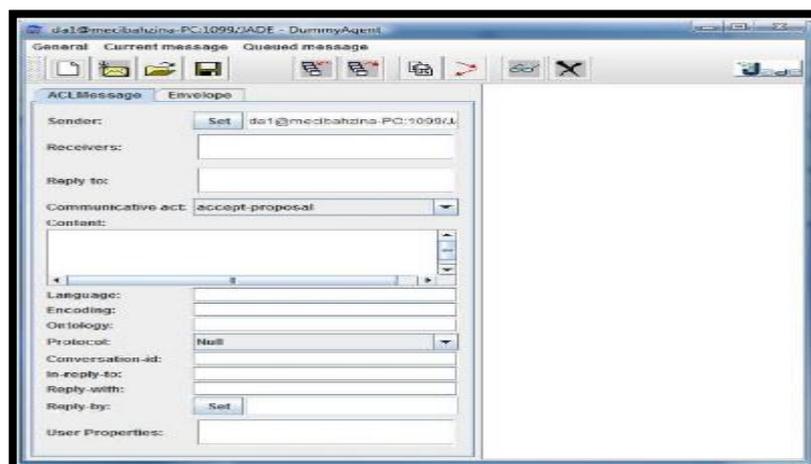


Figure 3.5 : Agent Dummy

Sniffer Agent « SA » : quand un utilisateur décide d'épier un agent ou un groupe d'agents, il utilise un agent Sniffer. Chaque message partant ou allant vers ce groupe est capté et affiché sur l'interface du sniffer. L'utilisateur peut voir et enregistrer tous les messages, pour éventuellement les analyser plus tard. L'agent peut être lancé du menu du RMA ou de la ligne de commande suivant **Java jade.Boot sniffer : jade.tools.sniffer.sniffer.**

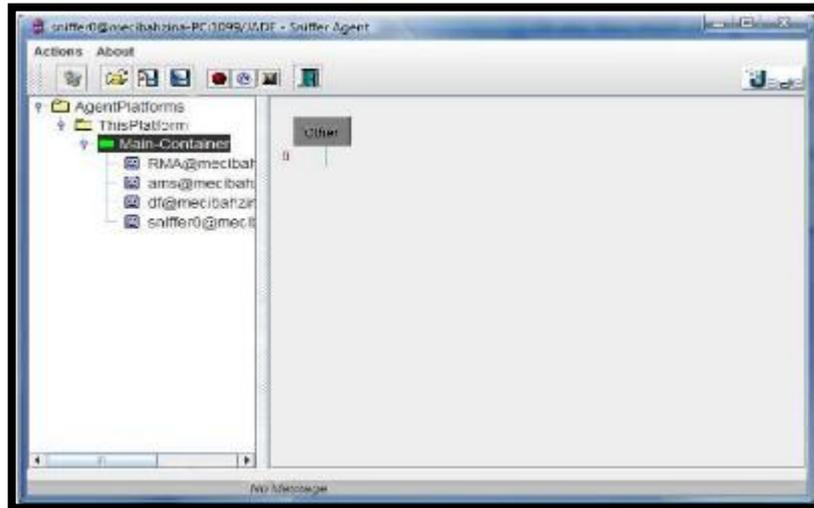


Figure 3.6 : Agent Sniffer

Introspector Agent : Cet agent permet de gérer et de contrôler le cycle de vie d'un agent s'exécutant et de la file de ses messages envoyés et reçus.

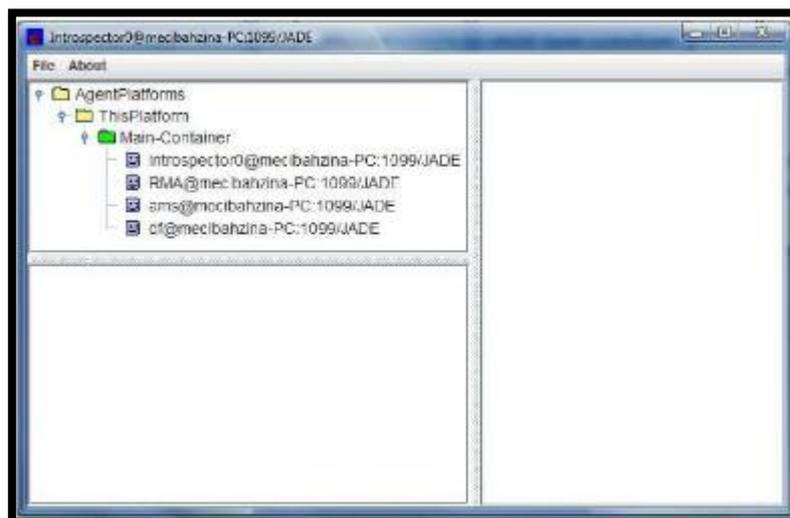


Figure 3.7: Agent Introspector

6. bibliothèque du JADE

JADE est composé des principaux packages suivants :

Jade.core : implante le noyau du système. Il possède la classe `<< Agent >>` qui doit être étendue par les applications des programmeurs et une classe `<< behaviour >>`.

Jade.lang : contient un sous-package pour chaque langage de communication utilisé par JADE et en particulier **jade.lang.acl**.

Jade.content : contient un ensemble de classes qui permettent de définir des ontologies.

Jade.domain : contient toutes les classes Java qui représentent les entités Agent Management définies par FIPA en particulier AMS et DF.

Jade.gui : contient un ensemble générique de classes utiles pour la création de GUIs pour l'affichage, l'édition des messages ACL, et de la description des agents.

Jade.proto : contient des classes qui modélisent les protocoles standards d'interaction. Les classes permettent aussi aux programmeurs d'ajouter d'autres protocoles.

Jade.tools : contient certains outils qui facilitent l'administration de la plate-forme et le développement d'applications.

7. Communication entre Agents JADE

Pour que plusieurs agents JADE arrivent à collaborer, ils doivent s'échanger des messages. Chaque agent JADE possède une sorte de boîte aux lettres qui contient les messages qui lui sont envoyés par les autres agents. Ces boîtes aux lettres sont sous forme d'une liste qui contient les messages selon l'ordre chronologique de leur arrivée.

Les agents JADE utilisent des messages conformes aux spécifications de la FIPA (FIPAACL). Les messages JADE sont des instances de la classe `ACLMessage` du package `jade.lang.acl`. Ces messages sont composés en général de :

- L'émetteur du message : un champ rempli automatiquement lors de l'envoi d'un message.

- L'ensemble des récepteurs du message : un message peut être envoyé à plusieurs agents simultanément.
- L'acte de communication : qui représente le but de l'envoi du message en cours (informer l'agent récepteur, appel d'offre, réponse à une requête, ...).
- Le contenu du message.
- Un ensemble de champs facultatifs, comme la langue utilisée, l'ontologie, le timeout, l'adresse de réponse, ...

Un agent JADE peut ainsi envoyer, recevoir, attendre un message d'un autre agent ou encore choisir un message dans sa boîte aux lettres.

7.1 Les Actes de communication

Les actes de communications proposés par JADE sont : *agree*, *cancel*, *confirm*, *cfp*, *disconfirm*, *failure*, *inform*, *inform_if*, *query_if*, *query_ref*, *reject_proposal*, *request*, *request_when*, *request_whenever*, *subscribe*, *null*, *propose*, *not_understood*.

L'acte de communication traduit la nature du message envoyé entre agents. Les actes se classent dans des catégories, certaines permettent d'amorcer des protocoles alors que d'autres sont utilisés pour répondre. Ainsi **request**, **cfp**, **request_when**, **request_whenever**, **query_if**, **query_ref**, **subscribe** sont utilisés pour amorcer un questionnement ou une demande de proposition et **agree**, **inform**, **failure**, **propose**, **not_understood**, sont des réponses possibles.

Le plus usité est le **not_understood**, dès qu'un agent ne comprend pas le message reçu, il répond par un message **not_understood** pour avoir plus de clarté.

Et voici, le Tableau 3.1 qui explique tous les performatifs :

Actions	Syntaxe	Définition - Sens
Accept Proposal	accept-proposal	Communication de l'accord de l'expéditeur d'effectuer une action qui lui a été préalablement soumise.

Agree	Agree	Communication de l'accord de l'expéditeur pour effectuer une action, sans doute dans le futur.
Cancel	Cancel	Communication de l'annulation de l'accord donnée préalablement par l'expéditeur pour effectuer une action.
Call for Proposal	Cfp	Communication par l'expéditeur d'une demande d'effectuer une certaine action.
Confirm	Confirm	Communication par l'expéditeur de la confirmation de la validité (selon les règles de l'agent) de la proposition préalablement reçue.
Disconfirm	Disconfirm	Communication par l'expéditeur de la confirmation de la non validité (selon les règles de l'agent) de la proposition préalablement reçue.
Failure	Failure	Communication par l'expéditeur de l'échec d'une action essayée.
Inform	Inform	Communication par l'expéditeur d'une proposition, pensée vrai par celui-ci.
Inform If	inform-if	Communication par l'expéditeur d'une proposition (pensée vrai par celui-ci), et demande au receveur une confirmation ou une non-confirmation.

Inform Ref	inform-ref	Communication par l'envoyeur d'une demande de l'objet qui correspond à une description envoyée.
Not Understood	not-understood	Communication par l'envoyeur d'une non compréhension d'une action effectuée par le destinataire.
Propagate	Propagate	Communication par l'envoyeur d'un message à propager à des agents dont la description est fournie. Le destinataire du message traite le sous message à propager comme s'il lui était directement destiné et envoie le message "propagate" au agent qu'il a identifié.
Propose	Propose	Communication par l'envoyeur d'une proposition d'action conditionnée à certains prés conditions données.
Proxy	Proxy	Communication par l'expéditeur d'une demande d'une transmission d'un message à des agents dont la description est donnée.
Query Ref	query-ref	Communication par l'envoyeur d'une demande par l'expéditeur de l'objet référencé par une expression.

Refuse	Refuse	Communication par l'expéditeur de son refus d'effectuer une action donnée, et en donne les raisons.
Reject Proposal	reject-proposal	Communication, pendant une négociation, par l'expéditeur de son refus d'effectuer des actions.
Request	Request	Communication par l'expéditeur d'une demande au destinataire d'effectuer une action.
Request When	request-when	Communication par l'expéditeur d'une demande, au destinataire, d'effectuer une action quand une proposition donnée devient vraie.
Request Whenever	request-whenever	Communication par l'expéditeur d'une demande, au destinataire, d'effectuer une action dès qu'une proposition donnée devient vraie, et à chaque fois que celle-ci redevient vrai.
Subscribe	Subscribe	Communication par l'expéditeur d'une demande d'un objet donné par une référence envoyé par l'expéditeur, et de notifier l'agent ayant souscrit dès que l'objet en question change.

Tableau 3.1: Actes de communication de FIPA-ACL

Même si la plate-forme multi-agents JADE est un projet récent, difficile à vendre, difficile de trouver des personnes le connaissant et a été conçu selon des critères d'avantage

académiques qu'industriels, et même s'il n'a été distribué en Open Source que très récemment, il a été utilisé par de nombreux projets internationaux et dans des domaines différents tel que: le soutien de travail collaboratif, e-learning, e-tourisme, gestion de réseau de divertissement, la gestion des connaissances...etc.[24]

8. jade et l'ontologie

Schémas définissant la structure des types de prédicat, d'actions d'agent et de concept concernant le domaine adressé ont été ajoutés. Ces schémas sont des instances des classes *PredicateSchema*, *AgentActionSchema* et *ConceptSchema* incluses dans le package *jade.content.schema*. Ces classes ont des méthodes au moyen desquelles il est possible de déclarer les champs qui définissent la structure de chaque type de prédicat, d'action d'agent et de concept. Comme une ontologie est fondamentalement une collection de schémas qui typiquement n'évoluent pas pendant la durée de vie d'un agent, il est plus pratique de déclarer l'ontologie comme un objet singleton et de définir une classe qui étend le package *jade.content.onto.Ontology* avec une méthode d'accès statique à cet objet singleton. Ceci permet de partager le même objet ontologie (et tous les schémas inclus) entre différents agents d'une même JVM.

9. Conclusion

Dans ce chapitre, nous avons présenté l'une des plates-formes multi agents : JADE « Java Agent Développement Framework ». Elle est compatible avec la norme FIPA, donc JADE est considérée comme standard. En utilisant JADE, il est facile d'implémenter un agent simple « agent cognitif ou bien réactif ». JADE utilise des techniques avancées (Système Expert JESS) et permet de commencer par quelques agents simples puis faire évoluer l'architecture avec l'application. Par ailleurs, JADE offre aux développeurs des systèmes multi agents la possibilité de planifier les tâches d'un agent d'une manière efficace et assez transparente.

Chapitre IV : implémentation

1. Introduction

Ce chapitre a pour objectif d'expliquer l'application réalisée pour donner une solution de la problématique posée dont on a concrétisé un scénario d'exécution entre les agents représentant le personnel d'une entreprise. Pour cela, nous allons présenter le modèle de workflow illustrant le processus de ce travail et le diagramme de séquence en AUML¹⁸ montrant l'ensemble des communications faites entre les agents du SMA sous la plate-forme Jade.

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 4.1).

¹⁸ AUML : Agent Unified Modeling Language

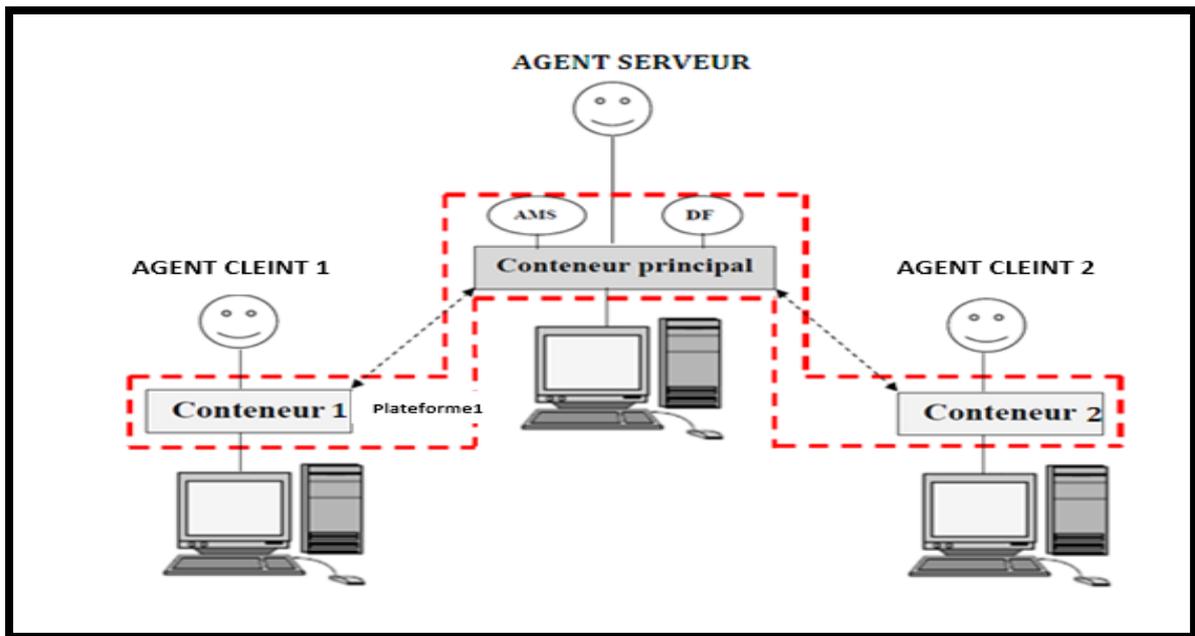


Figure 4.1 : Architecture distribuée de l'application

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.

3. modèle de workflow

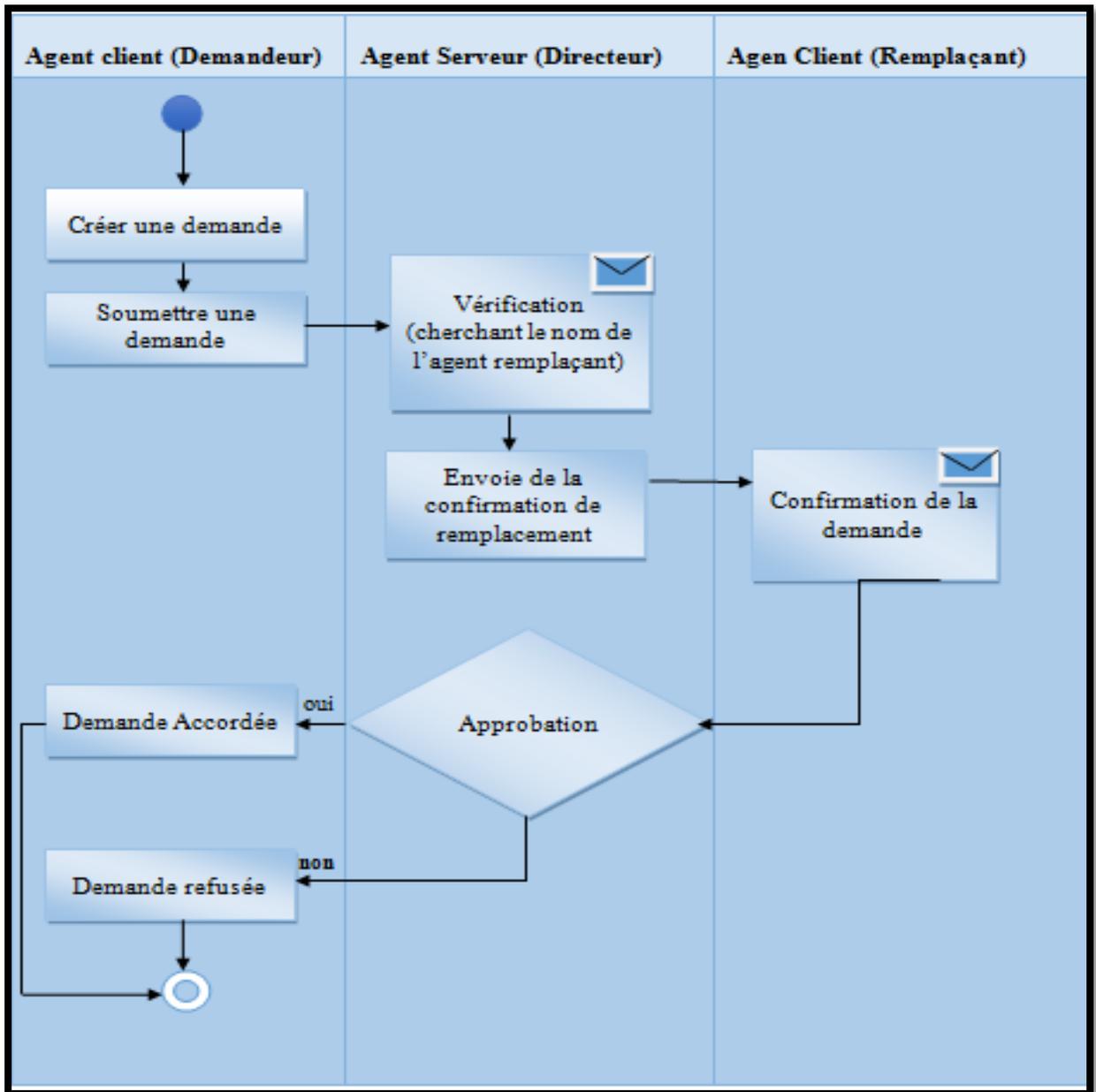


Figure 4.2 : Modèle de workflow

La figure précédente 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 à l'agent client demandeur du congé.

4. Enumération 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.

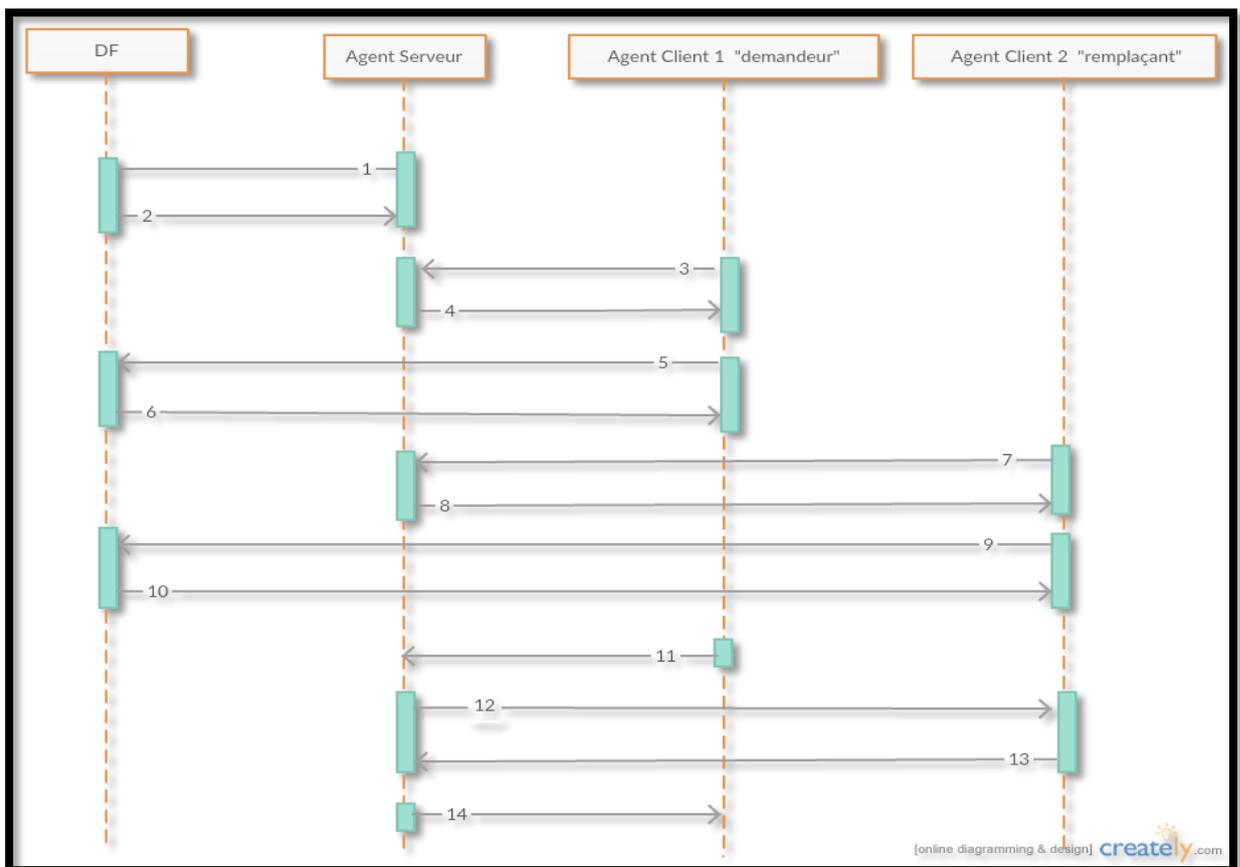


Figure 4.3 : Diagramme de séquence entre les agents de système

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)

3,7 : Demande de l'authentification de l'agent client1 (respectivement l'agent client2).
(REQUEST)

4,8 : Validation de l'authentification. (INFORM) si le pseudo et le mot de passe sont valides.

5,9 : Demande d'enregistrement de l'agent Client 1 « demandeur » (respectivement "client2" « remplaçant») au DF. (INFORM)

6,10 : Réponse de l'agent DF, pour la demande d'inscription de l'agent "Agent Client 1" (respectivement "l'agent client2"). (INFORM)

11 :L'agent "Agent client 1 "envoie une demande de congé à l'agent "Agent Serveur "
(REQUEST)

12 : L'agent « Agent Serveur » envoie la demande de confirmation à l'agent « Agent Client 2».
(REQUEST).

13 : L'agent « Agent Client2 » envoie un message de confirmation à l'agent « Agent Serveur ».
(INFORM)

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

5. Scénario d'exécution

Ce scénario d'exécution illustre un exemple d'utilisation de notre application, nous avons jugé utile donner un exemple complet d'une demande de congé dans un système multi-agent composé d'un agent serveur et deux agents clients (demandeur et remplaçant). Le processus s'effectue comme suit :

- ✓ Lancement de l'application sur des postes situés sur un réseau (deux postes dans notre exemple). Pour se faire, nous commençons par lancer le serveur et éventuellement le client. Ce qui nous donne respectivement les fenêtres ci-dessous :



Figure 4.4: Fenêtre principal de l'agent serveur



Figure 4.5: Fenêtre principale des agents clients

A partir de la figure 4.4 nous pouvons :

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

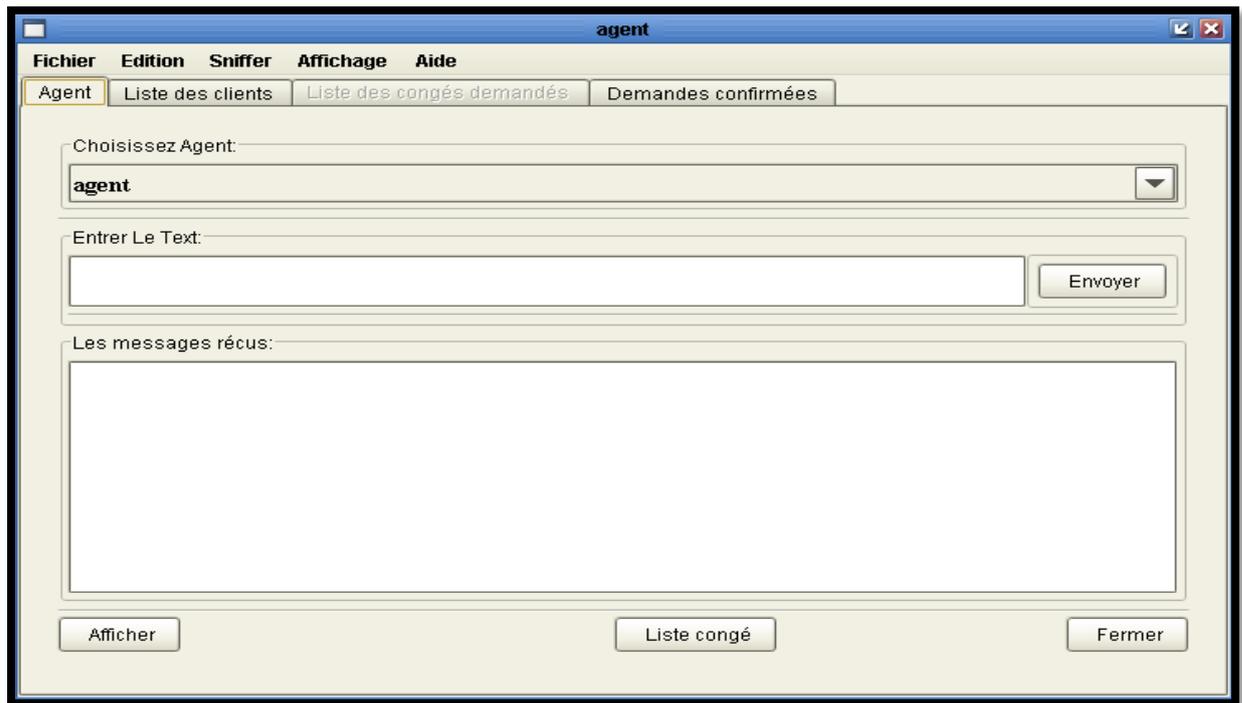


Figure 4.6: Fenêtre de l'agent serveur

2. Créer des comptes pour les agents clients dans le poste du serveur, en remplissant les champs indiqués dans figure suivante :

Figure 4.7 : Formulaire pour créer un compte client

2. Cliquer pour sur :
 - a. le bouton « Afficher » pour afficher la liste des comptes des clients.
 - b. Le bouton « Valider » pour la validation de création d'un compte de client.
 - c. Le bouton « Annuler » pour annuler l'opération de création d'un compte de client.

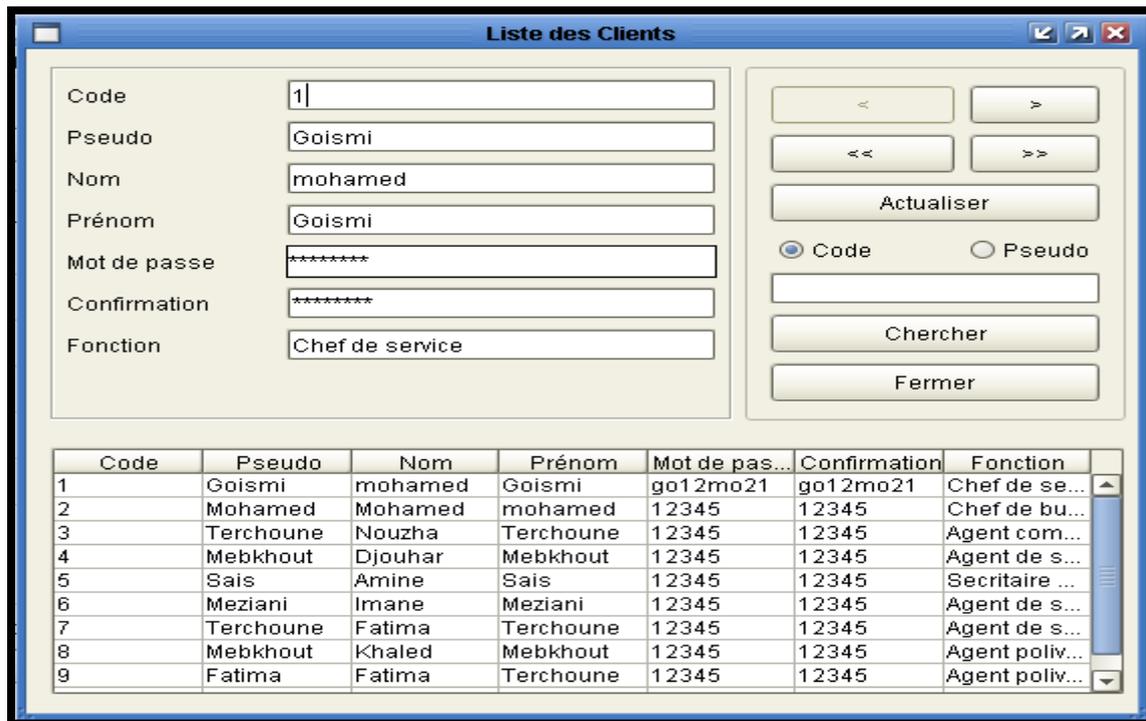


Figure 4. 8: Liste des agents clients

3. Lancer un agent client depuis la fenêtre principale. En cliquant sur le bouton « **je me connecte** ». En remplissant les champs (pseudo et mot de passe) puis cliquer sur le bouton « connexion ».

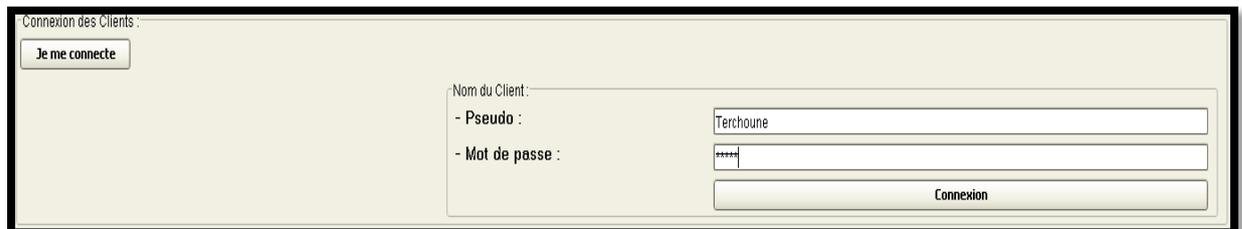


Figure 4.9: Lancement de l'agent client

4. Lancement de l'agent client « Terchoune ».

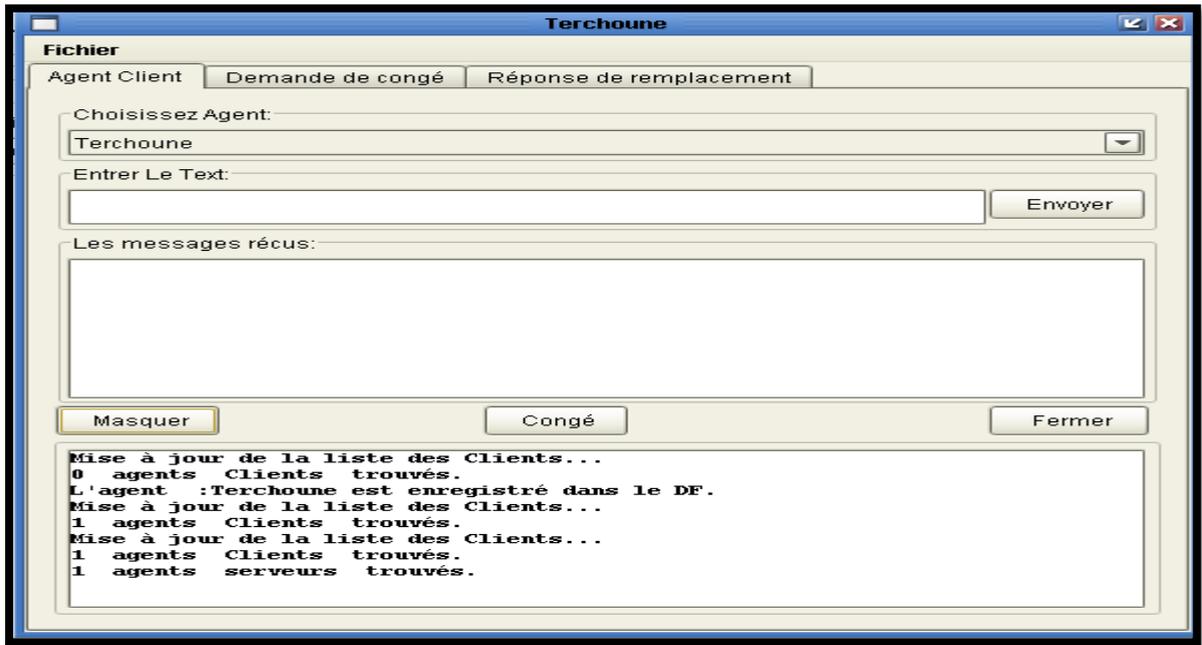


Figure 4.10: Fenêtre de l'agent client « Terchoune »

➤ **Demande du congé**

Pour que la demande de congé est établie :

5. lancer les agents deux du côté client (demandeur et remplaçant) et un autre du côté Serveur).

La figure suivante présente les interfaces graphiques de nos agents « Mebkhout » et « Goismi ».

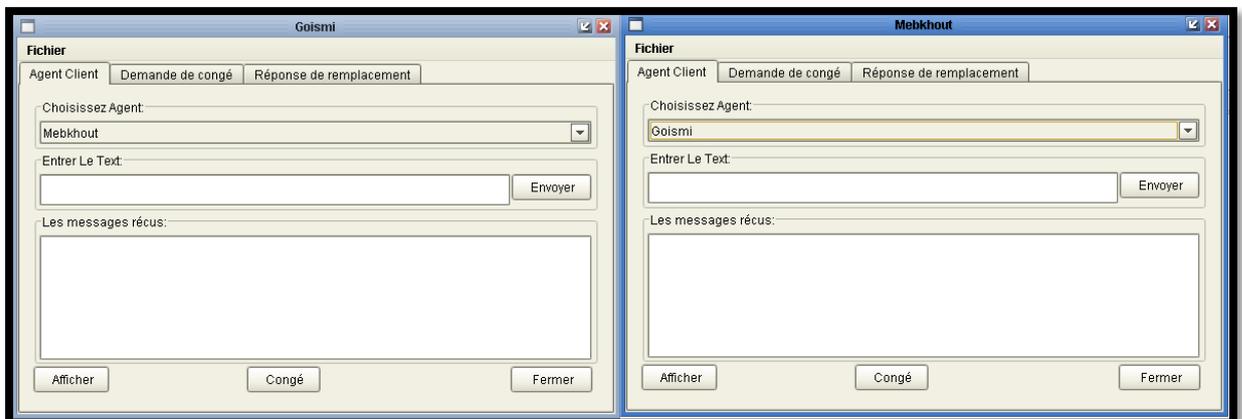


Figure 4.11: Fenêtres des deux agents clients "demandeur" et "remplaçant"

- Avant que la demande soit établie, il faut faire une mise à jour de tous les agents lancés.

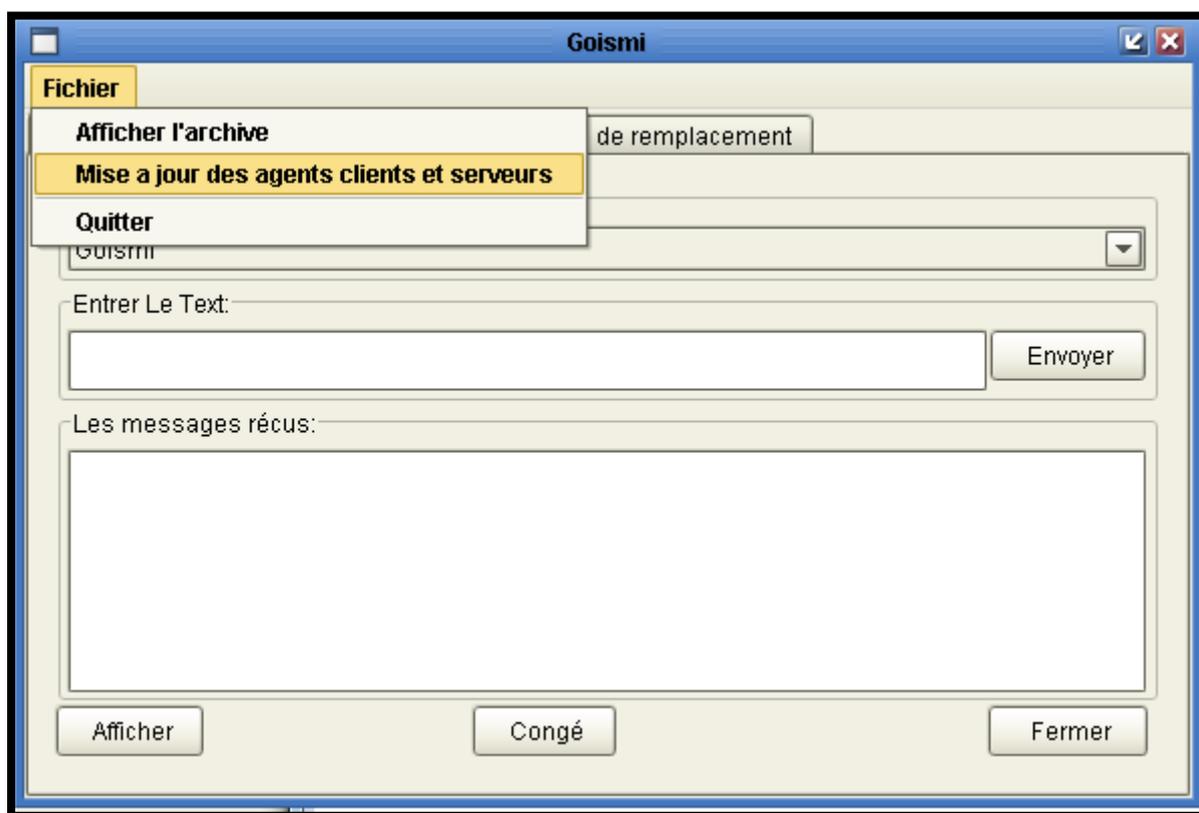


Figure 4.12: Mise à jour des agents

1. Demander un congé en cliquant sur le bouton « **demande conge** » et remplissant les champs indiqués dans figure suivante :



Figure 4.13: demande de conge rempli par le demandeur

2. Dès que le demandeur clique sur le bouton « Envoyer » la demande sera envoyée au serveur.

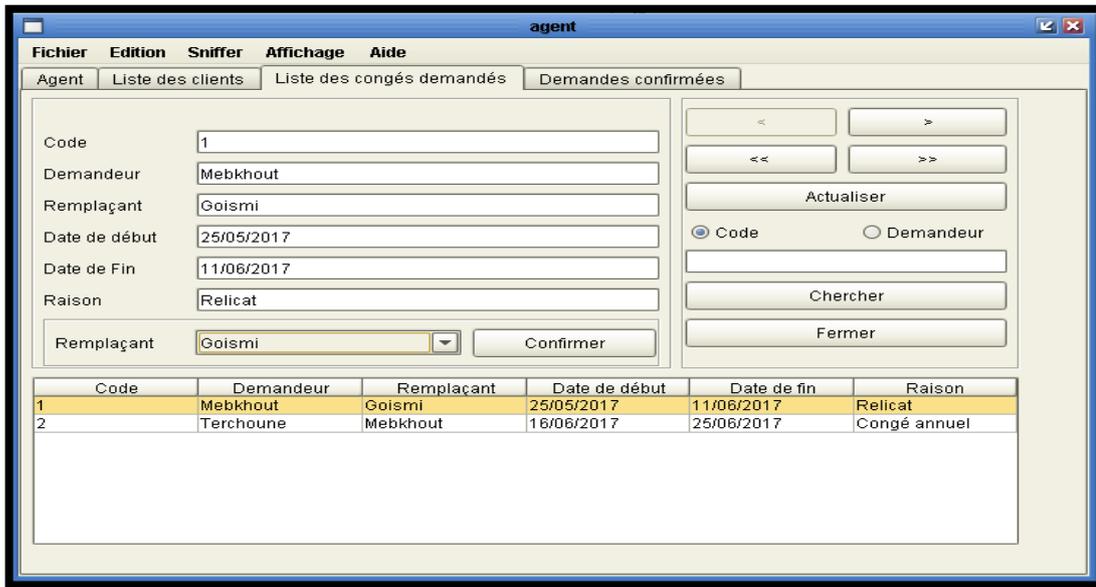


Figure 4.14: Réception de la demande de conge par le Serveur

3. 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.
4. 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** ».

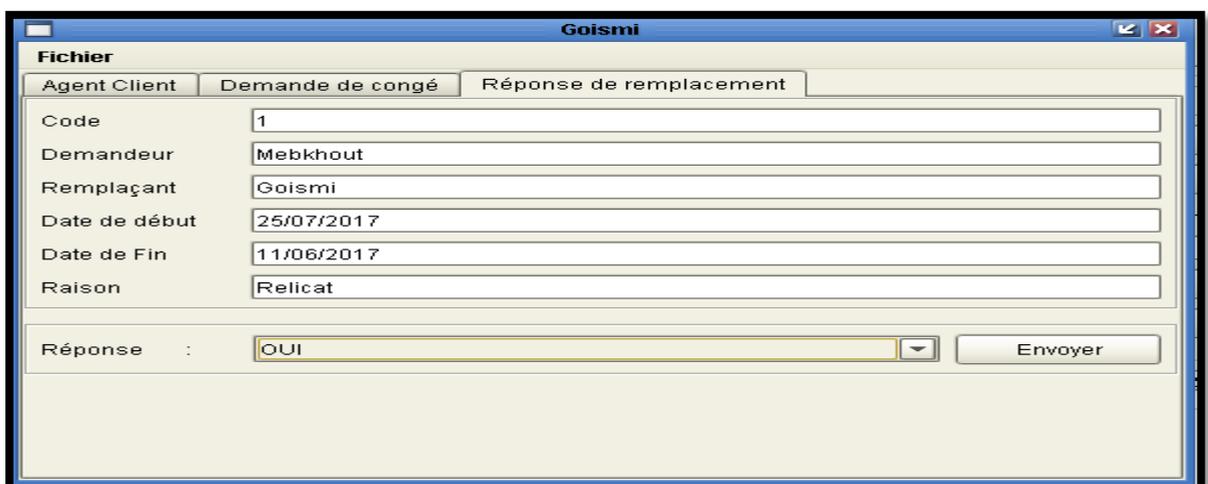


Figure 4.15: confirmation de l'agent client "remplaçant"

5. Après la décision

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

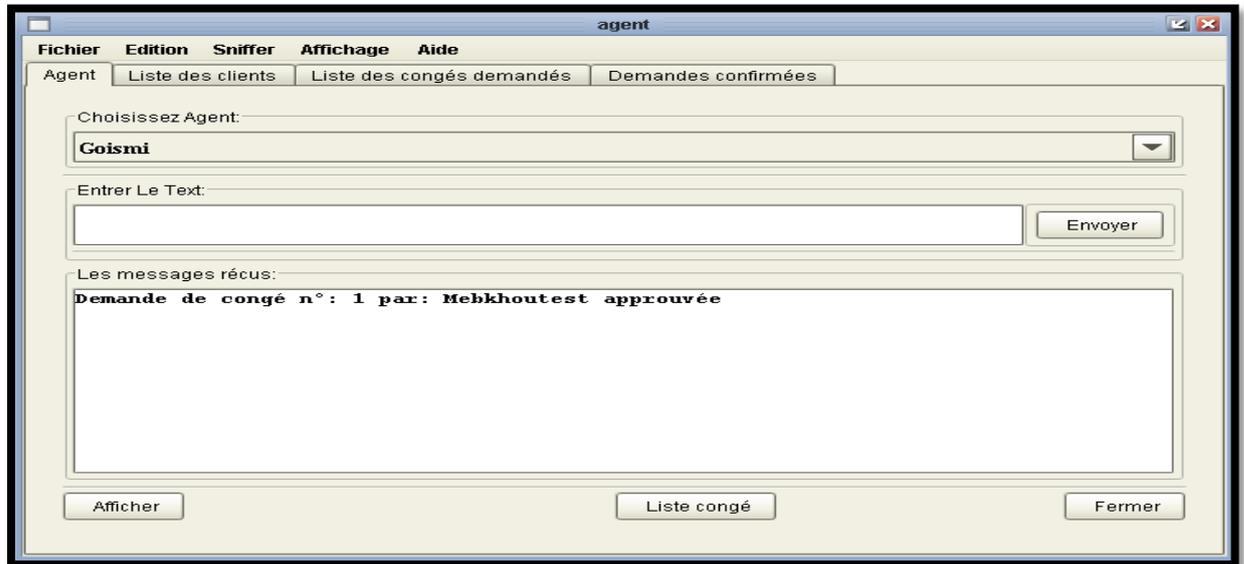


Figure 4.16: la réception de message de confirmation de la demande par l'agent serveur

- un message sera envoyé à l'agent client « demandeur »

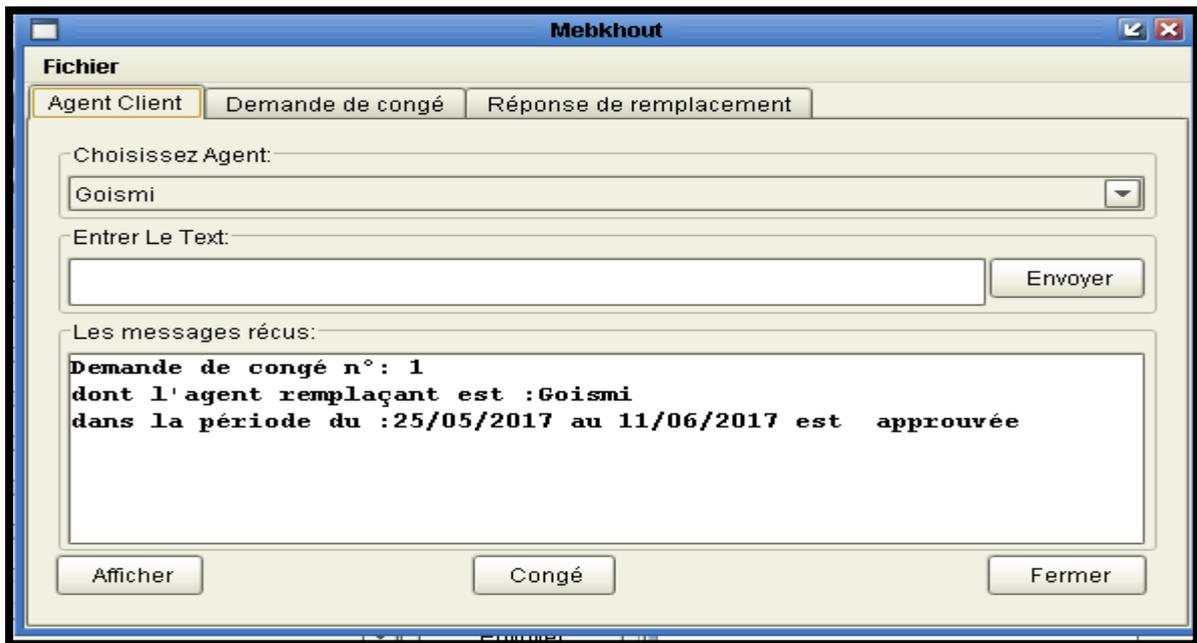


Figure 4.17 : la réception de message de confirmation de la demande par l'agent client « demandeur »

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

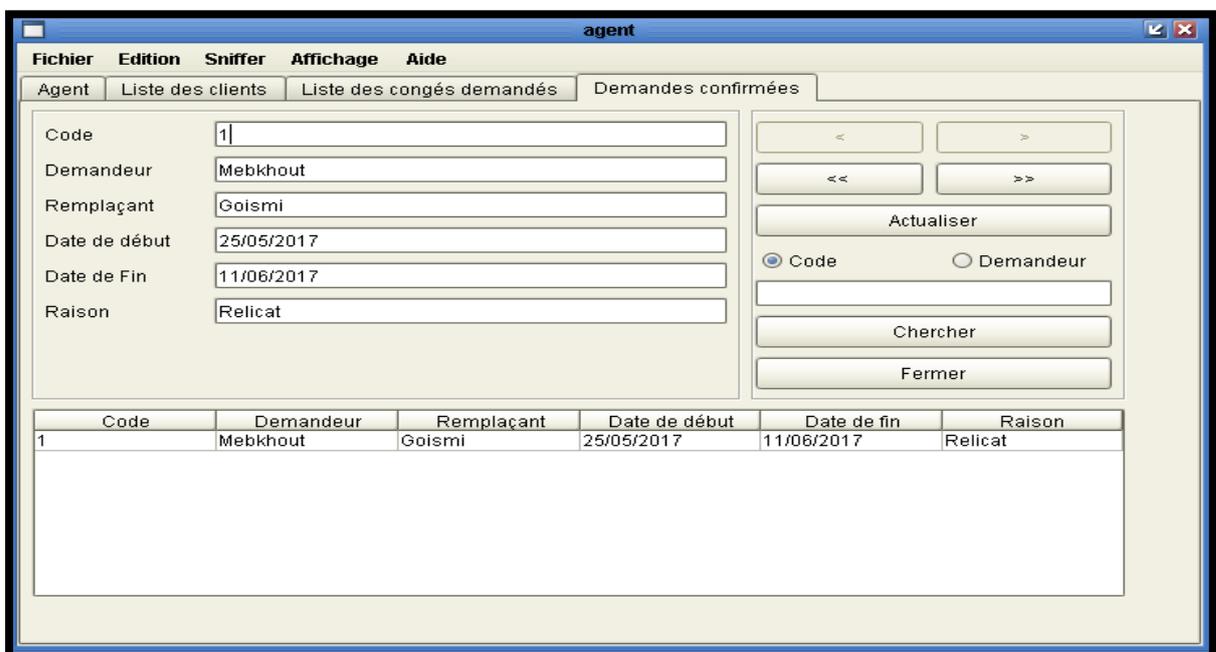


Figure 4.18 : Liste des demandes confirmées

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

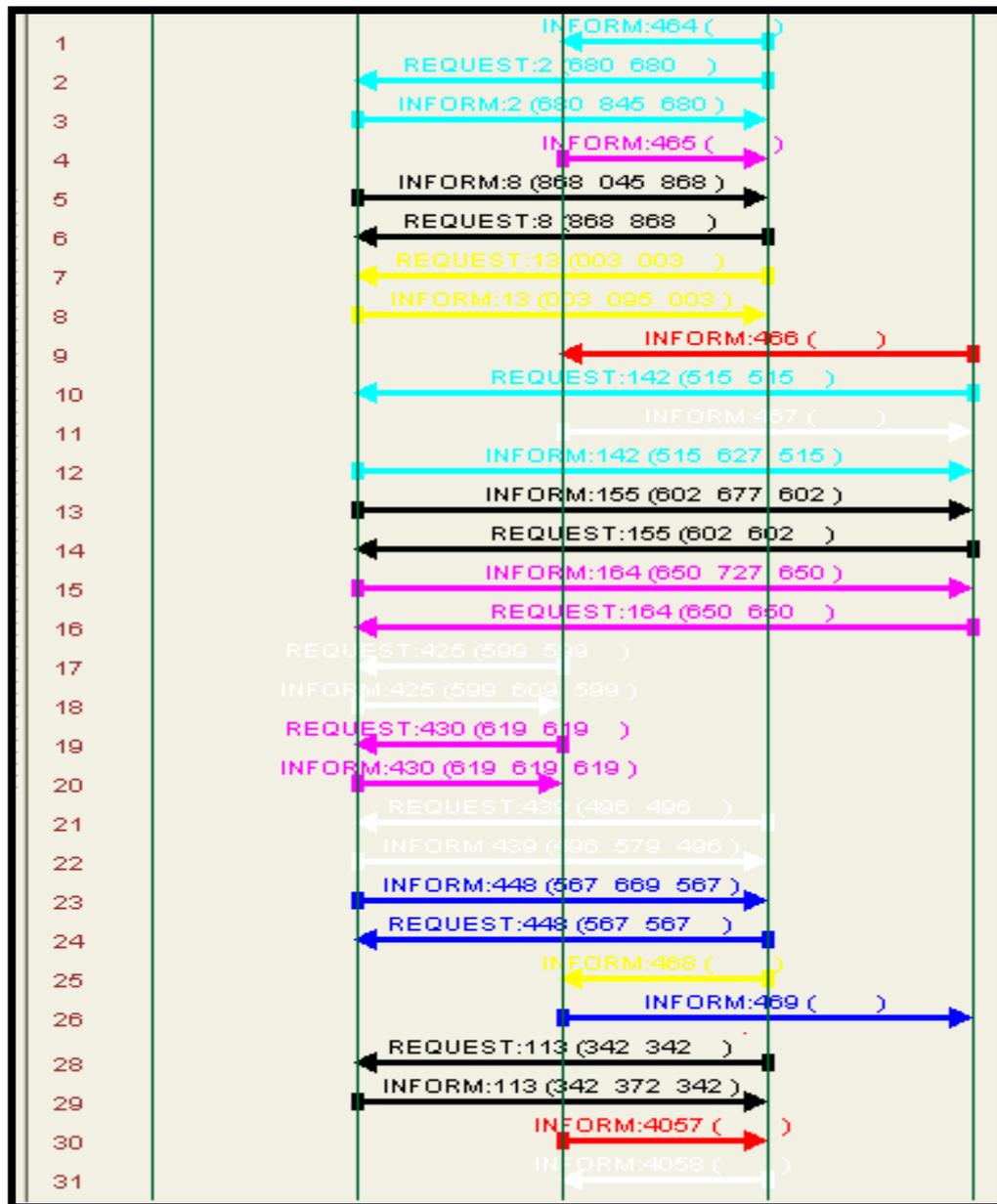


Figure 4.19 : SNIFER, contrôleur des messages échangés entre les agents du système

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 jour nous a donné la liste des agents connectés), l'agent serveur a la possibilité de sélectionner l'agent « Mebkhout » et lui envoie son message dont son contenu est « *bonjour, Mebkhout* » en cliquant sur le bouton « Envoyer ».



Figure 4.20 : Envoi du message à l'agent client « Mebkhout »

L'agent Mebkhout reçoit le message envoyé par le Serveur



Figure 4.21 : Réception du message de l'agent Serveur

6. Conclusion

La meilleure manière pour valider une conception d'un système est de l'implémenter en utilisant les outils disponibles et adéquats au problème posé.

Dans ce chapitre, nous avons expliqué l'approche utilisée pour la réalisation d'une application de demande d'un congé et de messagerie instantanée entre le personnel de

l'entreprise où nous avons utilisé les systèmes multi-agents pour faciliter la communication dans cette dernière, en plus pour simplifier le développement du processus workflow choisi (Demande de congé), l'enrichir et aussi de le rendre flexible.

Conclusion générale

Dans l'objectif de proposer une solution satisfaisante nous avons opté pour les systèmes multi-agents comme une solution aux limites de Workflow.

Notre travail consiste à réaliser une application qui permettra 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 formant une technologie innovante et émergente qui possède un fort potentiel, en termes d'autonomie des agents, coordination et coopération entre eux.

Le but principal de ce mémoire est de fournir un outil garantissant une communication sûre entre les employés appartenant à la même entreprise et détermine un service (demande de congé).

Nous avons pu acquérir une expérience en développant une application. Cette dernière se compose d'un ensemble d'agents au sein d'une entreprise pour effectuer une tâche à l'occurrence, la demande de congé par l'approche multi-agents qui offre la possibilité de faire distribuer les tâches des différents acteurs de l'entreprise, ainsi que nous a permis de faciliter la tâche de la programmation de ce processus.

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.

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 future : l'intégration de la plateforme **JBPM** pour la création et la gestion des workflow sous JAVA. En plus 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

- [1] Yu L., A Conceptual Framework For Agent Oriented and Role Based Workflow Modeling. Proceeding of the Conference on Advanced Information Systems Engineering, (14-15 June 1999; Heidelberg, Germany), CaiSE'99.
- [02] Maker. Workflow Meets Business Objects .In Proceedings of OOPSLA'96 Workshop, Business Objects Design and Implementation II: Business Objects as Distributed Application Components –the enterprise solution, 1996.
- [03] Mohan, G.Alonso, R.Gunthor ET M.Kamath. Exotica: A research perspective on workflow management systems.Data Engineering, 18(1), Mars 1995.
- [04] Pudhota, L. Tierney, A. and Chang, E. (2005): Services integration monitor for collaborative workflow management. In Proceedings of the 14th IEEE International Workshops on EnablingTechnologies: Infrastructure for Collaborative Enterprise.
- [05] The Workflow Mangement Coalition. (1999): Workflow Management Coalition Terminology and Glossary, technical report WfMC-TC-1011.
- [06] W.M.P. VAN DER AALST. "PROCESS-ORIENTED ARCHITECTURES FOR ELECTRONIC COMMERCE AND INTERORGANIZATIONAL WORKFLOW". Department of Mathematics and Computing Science, Eindhoven University of Technology.1999.
- [07] Lotfi BOUZGUENDA."Coordination Multi-Agents pour le Workflow Interorganisationnel Lâche". Doctorat de L'Université Toulouse I, Spécialité INFORMATIQUE.2005.
- [08] Zhao, X. Liu, C. and Yang, Y. (2005): An organizational perspective on collaborative business processes. In Proceedings of the 3rd International Conference on Business Process Management.
- [09] 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. 2009: Modeling and Verification of Web Service Composition Based Interorganizational Workflows. Thèse de doctorat. Université de Vienna.
- [11]Chebbi Issam. (2007) : Coop Flow : une approche pour la coopération ascendante de

workflows dans le cadre des entreprises virtuelles. Thèse de doctorat, Institut National des Télécommunications, France.

[12] FIPA Repository. <http://www.fipa.org/repository/aclspecs.html>, accessed on March 22, 2012

[13] Lambrinouidakis, C. Kokolakis, S. Karyda, M. Tsoumas, V. Gritzalis, D. and Katsikas, S. (2003): Electronic voting systems: security implications of the administrative workflow. In Proceedings of the 14th International Workshop on Database and Expert Systems Applications.

[14] Budinska, I. Oravec, V. Gatial, E. Laclavik, M. Seleng, M. Balogh, Z. Frankovic, B. Forgac, R. Mokris, I. and Hluchy, L. (2007) : A Knowledge Support System For Administrative Workflow Processes. In Proceedings of the Seventh International Conference on Application of Concurrency to System Design.

[15] Muehlberger, R. Orłowska, M.E. and Kiepuszewski, B. (1999) : Backward step: The right direction for production workflow systems. In Proceedings of the Australian Database Conference.

[16] Leymann, F. (1999): Production workflow: concepts and techniques. Prentice Hall PTR Upper Saddle River, NJ, USA.

[17] SAADOUN Mélissa. Technologies de l'information et management. Paris: Hermès Sciences, 2000, 252p

[18] G.T.S. Ho, H.C.W. Lau, C.K.M. Lee, A.W.H. Ip, and K.F. Pun. (2006): An intelligent production workflow mining system for continual quality enhancement. *The International Journal of Advanced Manufacturing Technology*, 28(7-8), pp.792-809.

[19] Jiang, P. Mair, Q. and Newman, J. (2003): 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.

[20] Khetawat, A. Lavana, H. and Brglez, F. (1997): Collaborative workflows: A paradigm for distributed benchmarking and design on the internet. Technical report, North Carolina State University.

[21] Huth, C. Erdmann, I. and Nastansky, L. (2001): 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.

[22] Levan, S.K <le projet workflow> ,2ème édition ,tirage 2000,Eyrolles.[Manolis & Dimitris 2009] : Manolis, K.Dimitris ,P., < Business Process Modeling and design : a formal model and methodology > manolis@di.uoa.gr,dp@csd.usf.edu.gr

[23] Rationality and intelligence. Artificial Intelligence, Russell, Vol. 94. 1997.

[24] Michael Wooldridge, Nicholas R. Jennings, « Intelligent Agents: Theory and Practice », The Knowledge Engineering Review, v.10, n°2, 1995, 115-152 p.

[25] Yoav Shoham, « Agent-oriented programming », *Artificial intelligence*, v.60, n°1, 1993, 51-92 p.

[26] Ferber Jacques, *Les systèmes multi-agents : vers une intelligence collective*, 1995.

[27] S. Labidi, W. Lejouad, « De l'Intelligence Artificielle Distribuée aux Systèmes multi-agents », *Rapport de recherche n°2004*, INRIA, Août 1993

[28] Jörg P. Müller, *The design of intelligent agents. A layer approach*, Lecture Notes of Computer Science v.1177, Springer-Verlag, 1996.

[29] Fischer, K. Müller, J. P. and Pischel, M. (1996) : A pragmatic BDI architecture. Intelligent Agents II (LNAI Volume 1037), pages 203-218. Springer-Verlag: Berlin, Germany.

[30] Brenner Walter, Zarnekow Rüdiger, Wittig Harmut, *Intelligent Software Agents: Foundations and Applications*, Springer-Verlag, Berlin, 1998.

[31] Bonabeau Eric, Theraulaz Guy, Intelligence collective, Hermès, Paris, 1994.

[32] G.M.P. O'Hare, N.R. Jennings (Eds.), *Foundations of Distributed Artificial Intelligence*, Wiley & Sons, New York, 1996.

[33] Stan Franklin, Art Graesser, « Is it an agent or just a program ? :a taxonomy for autonomous agents », In : *Intelligent Agents III : Agents Theories, Architectures and Languages, ECAI'96 Workshop (ATAL)*, Springer-Verlag, 1996.

[34] Michael Wooldridge, Nicholas R. Jennings, « Pitfalls of Agent-Oriented Development », In: *Proceedings of the 2nd Conference on Autonomous Agents (Agents-98)*, Minneapolis, USA, 1998, 385-391 p.

- [35] LabrouYannis, Finin Tim, « History, State of the Art and Challenges for Agent Communication Languages », *Informatik – Informatique*, n°1, Février 2000, 17-24 p.
- [36] Labrou Yannis, Finin Tim, Peng Yun, « Agent Communication Languages: the CurrentLandscape », *IEEE Intelligent Systems&Their Applications*, v.14, n°2, 1999, 45-52 p.
- [37] Mayfield James, LabrouYannis, Finin Tim, « Evaluating KQML as an Agent Communication Language », In : Wooldridge Michael, Müller Jörg P., TambeMilind (Eds.), *Intelligent Agents II : Agent Theories, Architectures, Languages – IJCAI'95 Workshop (ATAL)*, Lecture Notes Artificial Intelligence v.1037, Springer-Verlag, 1996, 347-360 p.
- [38] Rationality and intelligence. *Artificial Intelligence*, Russell, Vol. 94. 1997.
- [39] Bouron T., Structure de communication et d'organisation pour la coopération dans un univers multi-agents, Université Paris VI, (Thèse 3ème cycle, Informatique), Novembre 1992.
- [41] Bouron T., Structure de communication et d'organisation pour la coopération dans un univers multi-agents, Université Paris VI, (Thèse 3ème cycle, Informatique), Novembre 1992
- [42] Finin Tim, Labrou Yannis, Mayfield James, « KQML as an Agent Communication Language », *Software Agents*, Bradshaw Jeffrey (Eds.), AAAI/MIT Press, 1997.
- [43] Foundation for Intelligent Physical Agents, « FIPA Abstract Architecture Specification », August 2000, <http://www.fipa.org>.
- [44] Ferber Jacques, « Technologie multi-agent », *Mémento : Les interfaces intelligentes et images de communication et l'image*, France Télécom Recherche, n°8, Octobre 1996.
- [45] Brassac Christian, Pesty Sylvie, « Coopération dans les systèmes multi-agents : comportement ou conduite ? », *Decentralized Intelligent Multi-Agent Systems'95*, Krakow, Pologne, 22-24 novembre, 1995.
- [46] http://www.memoireonline.com/08/12/6084/m_La-generation-des-diagrammes-UML--partir-de-programme-jade13.html

