

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE  
UNIVERSITE MOHAMED BOUDIAF - M'SILA

FACULTE DES MATHÉMATIQUES ET  
DE L'INFORMATIQUE

DEPARTEMENT D'INFORMATIQUE

N° :.....



DOMAINE : MATHÉMATIQUES ET  
INFORMATIQUE

FILIERE : INFORMATIQUE

OPTION : RESEAUX

**Mémoire présenté pour l'obtention  
Du diplôme de Master Académique**

**Par: Bouchenafa Elkhansa**

**Intitulé**

**La recherche sémantique d'information :  
approche basée sur les agents mobiles**

**Soutenu devant le jury composé de :**

	Université de M'sila	Président
Meliouh Amel	Université de M'sila	Rapporteur
	Université de M'sila	Examineur

**Année universitaire : 2016 /2017**

# TABLE DES MATIERES

Introduction générale.....	6
<b>Chapitre 1 : les agents mobiles</b>	
1. Introduction.....	8
2. Définition d'un agent.....	8
3. Les caractéristiques d'un agent.....	8
3.1 Autonomie.....	8
3.2 Connaissance.....	8
3.3 Social.....	8
4. Les types des agents.....	8
4.1 Agent réactif.....	9
4.2 Agent cognitif.....	9
4.3 Agent hybride.....	9
5. Domaine d'application des agents.....	9
6. Système multi agent.....	9
7. Agent mobile.....	9
7.1 Définition.....	10
7.2 La migration des agents.....	10
7.2.1 La migration forte.....	10
7.2.2 La migration faible.....	10
7.3 Les caractéristiques d'un agent mobile.....	10
7.4 Avantage des agents mobiles.....	10
7.4.1 Flexibilité.....	11
7.4.2 Performance et efficacité.....	11
7.4.3 Fiabilité.....	11
7.4.4 Portabilité.....	11
7.4.5 Tolérance aux pannes.....	11
7.5 Domaine d'application des agents mobiles.....	11
7.5.1 Maintenance répartie.....	12
7.5.2 Grille de calcul.....	12
7.6 Environnement d'exécution des agents mobiles.....	12
7.7 Les différents types d'agent mobile.....	12

7.7.1	Les agents légers.....	13
7.7.2	Les agents lourds.....	13
7.8	Services requis pour l'exécution d'agents mobiles.....	13
7.8.1	Création d'agents mobiles.....	13
7.8.2	Service de nommage.....	13
7.8.3	Plate-forme et standardisation.....	13
7.8.4	Communication.....	15
7.8.5	Exécution d'un agent.....	18
7.8.6	Sécurité.....	18
7.9	Cycle de vie et contrôle de l'agent mobile.....	18
7.9.1	Création et initialisation.....	19
7.9.2	Migration.....	19
7.9.3	Activation et désactivation.....	19
7.9.4	Terminaison.....	19
8.	Conclusion.....	19

## **Chapitre 2 : la recherche sémantique d'information**

1.	Introduction.....	20
2.	Les ontologies.....	20
a.	Origine.....	20
b.	Définition.....	20
c.	Composants d'une ontologie.....	21
i.	Les concepts.....	21
ii.	Les relations.....	21
iii.	Les fonctions.....	21
iv.	Les axiomes.....	21
v.	Les instances.....	21
d.	Type d'ontologie.....	21
i.	Ontologie de représentation des connaissances.....	22
ii.	Ontologie de haut niveau / supérieure.....	22
iii.	Ontologie générique.....	22
iv.	Ontologie du domaine.....	22
v.	Ontologie de tache.....	22
vi.	Ontologie d'application.....	22

e. Domaine d'application d'ontologies.....	22
f. Formalismes et représentation des ontologies.....	23
i. Les frames.....	23
ii. Graphes conceptuels.....	23
3. La recherche d'information.....	23
a. Définition.....	23
b. Concepts de base de la RI.....	23
i. Collection de documents.....	23
ii. Documents.....	23
iii. Requête.....	23
iv. Pertinence.....	23
v. Besoin d'information.....	24
c. Les modèles de recherches d'information.....	24
i. Modèle booléen.....	24
ii. Modèle vectoriel.....	24
iii. Modèle probabiliste.....	25
d. Système de recherche d'information.....	25
e. Architecture d'un système de recherche d'information.....	25
f. Les phases du processus de recherche d'information.....	25
i. Indexation.....	26
ii. L'appariement requête-document.....	26
4. La recherche sémantique d'information.....	27
4.1 Définition.....	27
4.2 La recherche d'information guidée par les ontologies.....	27
4.2.1 Architecture d'un système de recherche guidée par les ontologies.....	27
4.2.2 Contraintes liées à la recherche sémantique.....	28
4.2.2.1 Présélection des ontologies.....	28
4.2.2.2 Intégration des ontologies.....	28
4.2.2.3 Annotation du corpus.....	28
4.3 Indexation conceptuelle.....	28
4.3.1 L'indexation sémantique.....	29
4.3.1.1 La méthode de voorhees.....	29
4.3.1.2 La méthode de krovetz & croft.....	29

4.4 Similarités entre concepts dans une ontologie.....	29
4.4.1 Extension des requêtes via d'une ontologie.....	29
4.4.1.1 Identification des concepts candidats.....	30
4.4.2 Mesure reposant sur la distance.....	30
4.4.2.1 Mesure de RADA.....	30
4.4.2.2 Mesure de wu-palmer.....	30
4.4.3 Mesure reposant sur le contenu en informations des concepts.....	31
4.4.3.1 Mesure de resnik.....	31
5. Conclusion.....	31
<b>Chapitre 3 : contribution</b>	
1. Introduction.....	32
2. Objectif et motivations de la démarche.....	32
3. Description générale de la démarche adoptée.....	32
3.1 Description de système.....	34
3.1.1 La recherche sémantique d'un terme.....	34
3.1.2 Création de conteneurs.....	35
3.1.3 Création d'un agent.....	35
3.1.4 La migration d'un agent.....	36
3.1.5 Accéder à l'ontologie par un agent.....	36
3.1.6 Traitement de la requête.....	36
3.1.7 Obtenir et affichage de résultat.....	36
4. Les algorithmes principaux.....	36
a. La recherche sémantique d'information.....	36
b. La migration.....	36
5. Mise en œuvre du système.....	36
5.1 Environnement du développement.....	36
5.2 Choix de plate-forme.....	39
6. Quelques fenêtres de notre application.....	41
6.1 Interface de monitoring rma et sniffeur avant la migration d'un agent.....	41
6.2 Interface de monitoring rma et sniffeur après la migration d'un agent.....	42
6.3 Interface de l'agent de recherche.....	43
6.4 Interface de l'ontologie.....	43
6.5 Interface de recherche.....	44

6.6 Interface de résultat.....	44
7. Conclusion.....	45
Conclusion générale.....	46
Bibliographie.....	48
Résumé	

## **INTRODUCTION GENERALE**

La recherche d'information(RI) est un domaine très important qui permet de faire des études sur tous ce qu'il concerne la structure,l'analyse,le stockage et la découverte de l'information dont le but est de faciliter l'accès à l'information pour réaliser les besoins en informations d'un utilisateur, alors la RI est une collection des procédures et des méthodes qui permettent de trouver à partir d'un ensemble de documents ceux qui correspondent aux exigences de l'utilisateur.

La recherche sémantique d'information(RSI) est une branche principale de la RI qui vise à étudier les significations des termes dans les documents pour comprendre le contenu de document et découvrir le sens de l'information.

La RSI est devenue avec le temps une tâche très complexe puisqu'elle nécessite la manipulation sémantique de l'information c'est-à-dire chercher la signification et le sens des termes à partir d'un ensemble de méthodes et d'algorithmes qui ne sont pas faciles à trouver pour la compréhension et la découverte du sens de cette information.

Pour trouver la sémantique d'un terme nous utilisons des algorithmes et des méthodes parmi ces méthodes l'indexation sémantique qui s'appuie sur le sens des termes, elle est basée sur des algorithmes de désambiguïsation de mots (wsd) qui permettent de ramener tous les mots qui possèdent le même sens, Dans le but est d'indexer les documents et les requêtes avec le sens des mots (mots-sens), ensuite la méthode de Voorhees qui est une partie de la méthode d'indexation sémantique, elle repose sur le terme WordNet, ce terme est un réseau utilisé dans le domaine sémantique et concerne le concept synset. Un synset consiste à rassembler des termes que ce soit simples ou composés ayant le même sens dans un contexte donné. Et ainsi les méthodes de Krovetz & Croft qui permettent de connaître la relation de correspondance du sens des mots dans les requêtes et les documents. Leur but était de trouver l'existence d'une relation entre la correspondance/non correspondance des sens.

Dans le cadre de ce travail nous proposons une approche basée sur la technologie d' agents mobiles permettant d'aider à la RSI .Ces agents sont des entités logicielles qui ont la capacité de se déplacer d'une machine à une autre de manière autonome sans perdre leurs code, leurs état d'exécution, et leurs données propres, alors ces entités s'exécutent dans un environnement qui offre des services permettant la création, la migration, la communication et l'échange des messages entre ces agents. Nous utilisons aussi une ontologie qui est un ensemble structuré de connaissances dans un domaine particulier qui permet de chercher à en

représenter le sens des termes, elle exprime les relations entre ces connaissances, dont le but est la découverte du sens de l'information c'est-à-dire sa sémantique.

Les ontologies sont essentielles pour manipuler la sémantique de l'information, elles fournissent des représentations qui permettent la compréhension du sens de ce qu'elles manipulent dans le but est que la machine comprenne le sens de l'information qu'elle véhicule. Alors dans un Système de recherche d'informations guidée par les ontologies les documents et les requêtes sont représentés par un intervalle, cet intervalle contient des mots précis et simples qu'on trouve dans les formes classiques, alors tout d'abord, il faut découvrir les termes qui existes dans la requête, ensuite étendre ces termes par les termes qui indiquent d'autres concepts qui sont proches de la requête, ces termes sont identifiés grâce aux liens sémantiques entre concepts qu'offre l'ontologie. On dit que La similarité est idéale dans le cas où les deux concepts possèdent le même sens donc, elles sont identiques.

En conséquence pour trouver la sémantique d'une information, Un agent mobile déplace d'une machine à une autre de manière autonome avec ses données propres, il accède ensuite à l'ontologie, pour chercher le sens d'une information et rendre les résultats obtenus pour une recherche sémantique d'une information.

Ce mémoire est organisé en trois chapitres :

Dans le premier chapitre, nous avons introduit les agents mobiles, leurs caractéristiques, leurs avantages, et le cycle de vie de ces entités.

Dans le deuxième chapitre nous nous sommes intéressés à la recherche sémantique d'information ou nous avons parlé sur les ontologies comme composante dans le processus de recherche, avec quelques notions de base de la recherche sémantique d'information et certaines méthodes existantes pour réaliser cette recherche.

Dans le dernier chapitre nous avons présenté notre contribution dans ce travail. Ainsi que les résultats obtenus pour une recherche sémantique d'une information.



# CHAPITRE 1

## LES AGENTS MOBILES

### 1. Introduction

De nos jours, la plupart des applications de la recherche d'informations nécessitent l'interaction et la communication entre différentes entités afin d'échanger des données entre eux. Ces applications sont basées sur la technologie d'agent mobile. Ces agents sont des entités logicielles qui ont la capacité de se déplacer d'une machine à une autre de manière autonome sans perdre leurs codes, ni leurs états d'exécution et s'exécutent dans un environnement qui offre des services permettant la création, la migration, la communication et l'échange des messages entre ces agents. Dans ce chapitre, nous allons présenter les notions de base de la technologie agent mobile.

### 2. Définition d'un agent

Un agent est une entité logicielle ou matérielle qui fonctionne de manière autonome et possède des ressources propres, capable d'agir dans un environnement utilisé pour satisfaire certains besoins de clients.

### 3. Les caractéristiques d'un agent

#### 3.1 Autonomie :

Un agent est une entité autonome disposant son propre environnement, c'est-à-dire un agent fonctionne de manière indépendamment du client, il décide lui-même où il va se déplacer et ce qu'il doit y faire, en fonction des informations lui provenant de son environnement.

#### 3.2 Connaissance :

Un agent dispose d'une connaissance de son environnement courant, ce qui permet de prendre des décisions adéquates.

#### 3.3 Social :

Un agent doit être capable d'interagir avec d'autres agents afin d'accomplir des tâches.

### 4. Les types des agents

Il existe plusieurs types d'agent à savoir :

#### 4.1 Agent réactif :

Il se caractérise par des comportements simples, sa structure purement tend à la simplicité, de petite taille, ces agents qui se résument à des réactions aux stimuli de l'environnement, la communication entre ces agents est accomplie par des traces ou des signaux.

#### 4.2 Agent cognitif :

Ces agents disposent d'une mémoire, ils possèdent une connaissance propre, ils ont des capacités de cognition leur permettant de raisonner sur les autres agents.

#### 4.3 Agent hybride :

La combinaison entre agent réactif et agent cognitif. Il prend les propriétés des agents réactifs et des agents cognitifs pour composer une entité qualifiée de se comporter encore différemment.

### 5. Domaine d'application des agents

Les agents sont utilisés dans plusieurs domaines, on peut citer :

- Les Systèmes ouverts (exemple internet).
- Les Systèmes distribués par nature :(géographiques, humaine).
- Les Systèmes complexes : société d'entités coopérantes, robotique collective, [8].

### 6. Système multi agent

Est un ensemble organisé d'agents, qui interagissent entre eux dans un environnement commun.

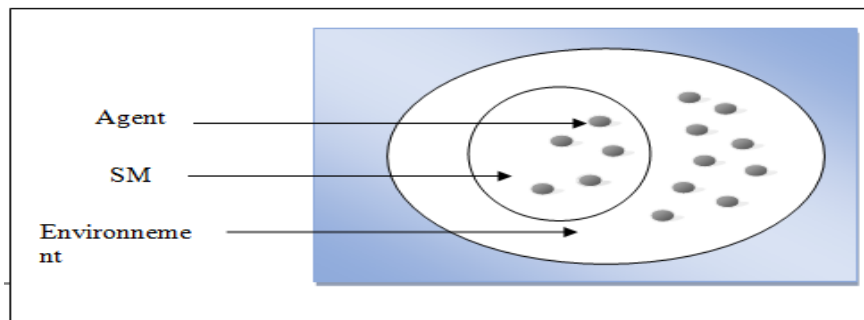


Figure 1.1 Les agents dans SMA

Un SMA est caractérisé par :

- Chaque agent possède des informations et des capacités qui lui permettent de résoudre des problèmes.
- Il n'y a aucun contrôle global du SMA.
- Les données sont décentralisées.

### 7. Agent mobile

#### 7.1 Définition

Il s'agit d'une entité logicielle qui se déplace de manière autonome entre les machines, avec son code, ses données propre et son état d'exécution afin d'accomplir les tâches de clients. Un agent mobile consiste en un code et un état, contient son propre contrôle. Il est capable de déplacer son code et son état entre les machines, ce déplacement est appelé migration.

## 7.2 La migration des agents

Une migration est le déplacement d'un agent d'une machine à une autre sans perdre ni son code, ni son état d'exécution, afin d'accomplir leur tâche.

Deux types de migrations ont été proposés:

7.2.1 La migration forte permet à un agent de se déplacer quelque soit l'état d'exécution dans lequel il se trouve. Dans ce type de migration l'agent se déplace avec son code, son contexte d'exécution et ses données. Dans ce cas, l'agent reprend son exécution après la migration exactement là où elle était avant son déplacement. La migration forte nécessite un mécanisme de capture instantanée de l'état d'exécution de l'agent. Elle peut être proactive ou réactive. Dans la migration proactive, la destination de l'agent est déterminée par l'agent lui-même. Dans la migration réactive, la migration de l'agent est dictée par une partie ayant une relation avec l'agent mobile.

7.2.2 La migration faible ne fait que transférer avec l'agent son code et ses données. Sur le site de destination, l'agent redémarre son exécution depuis le début en appelant la méthode qui représente le point d'entrée de l'exécution de l'agent, et le contexte d'exécution de l'agent est réinitialisé. Pour que l'agent se branche sur une instruction particulière de son code après sa migration, le programmeur doit inclure dans l'état de l'agent des moments privilégiés (explicites) dans le code de l'agent (point d'arrêt) pour pouvoir le relancer [22].

## 7.3 Les caractéristiques d'un agent mobile

-Un agent mobile peut se déplacer entre les machines avec son code, ses données propres et son état d'exécution.

-La décision de migration revient seulement aux agents.

-Un agent mobile est capable de migrer plusieurs fois vers différentes destinations afin d'accomplir leur tâches.

-Un agent mobile est capable de suspendre son exécution à un point donné, de migrer vers un autre site, puis de relancer son exécution depuis le même point.

## 7.4 Avantages des agents mobiles

Parmi les avantages des agents mobiles, on peut citer :

#### 7.4.1 Flexibilité :

Un agent mobile peut se déplacer entre machines de différents systèmes tout en restant opérationnel.

#### 7.4.2. Performance et Efficacité :

L'utilisation des agents mobiles augmente les performances des applications. Elle fournit une meilleure utilisation des ressources, permet une haute qualité de services, réduit la latence et la consommation de la bande passante.

#### 7.4.3. Fiabilité :

Un système multi agent est tolérant aux fautes. Si un agent est hors-service il reste d'autres agents qui peuvent accomplir leurs fonctions ou même régénérer l'agent, si le système donne la possibilité.

#### 7.4.4 Portabilité :

Les agents supportent plus facilement les systèmes distribués et hétérogènes, plusieurs environnements de développement des agents mobiles supportent plusieurs langages de programmation en même temps ce qui permet une compatibilité avec plusieurs systèmes [20].

#### 7.4.5. Tolérance aux pannes :

Le modèle d'exécution de l'agent mobile implique son interaction avec plusieurs sites ce qui expose l'agent à une éventualité de disparition à cause de la défaillance ou de la déconnexion soudaine et imprévue d'un site sur lequel il s'exécute. La disparition d'un agent entraîne un dysfonctionnement de l'application basée sur ce dernier. Une application distribuée sûre doit pouvoir continuer de fonctionner en cas de défaillance d'une partie du système. Pour certains types d'applications, il est essentiel que les environnements d'exécution d'agents mobiles offrent des mécanismes de tolérance aux fautes. Pour faire face à la défaillance, les environnements d'agents mobiles offrent un mécanisme de point de reprise. Les points de reprise sont conservés sur disque, support supposé fiable. Ils peuvent ainsi être utilisés ultérieurement pour restaurer l'agent en cas de défaillance. Cependant, lors de la restauration de l'agent il faut veiller à ne pas avoir deux agents actifs en même temps (problème rencontré en cas de défaillance lié à la perte de la connexion avec le site d'accueil de l'agent par exemple).

### **7.5 Domaines d'application des agents mobiles**

Dans cette partie, nous présentons quelques domaines d'applications envisageables pour les agents mobiles [21].

### 7.5.1 Maintenance répartie

La maintenance répartie c'est la mise à jour des ressources (services ou matérielles) réparties au sein d'un réseau. Ces mises à jour consistent à remplacer, ou à ajouter, des fonctionnalités à un ensemble d'éléments. L'utilisation des agents mobiles permet de décrire simplement l'exploration d'un réseau et permet d'envisager une mise à jour totalement découplée d'un service central en déléguant une partie de l'administration aux agents mobiles. Dans ce cas, un agent possède la mise à jour à appliquer et va se déplacer d'un site à un autre et va actualiser tous les éléments trouvés. En se déplaçant, les agents peuvent plus facilement accéder aux éléments appartenant aux infrastructures décentralisées et surtout peuvent appliquer la mise à jour sans craindre d'être interrompus, cela grâce à leur traitement local [10].

### 7.5.2 Grille de calcul

La grille de calcul consiste à regrouper dynamiquement au sein d'un même réseau virtuel tout un ensemble de machines, de ressources et d'utilisateurs. L'objectif des grilles de calcul est de permettre à des organisations dispersées (entreprises, universités, etc.) de partager des applications (services), des données et des ressources (processus, disque). Le but est de mettre en commun les capacités de chaque entité. Ainsi, on peut très bien imaginer qu'une entreprise, spécialisée dans les calculs à haute performance, loue du temps de calcul sur ses serveurs à une entreprise souhaitant mettre en place une expertise précise (service). Nous sommes donc face à un environnement hétérogène, décentralisé et dynamique. Avec les grilles de calcul, les agents mobiles trouvent un champ d'application naturel. En effet, les caractéristiques de déplacement, d'adaptation, de coopération vont permettre de tirer partie de ces environnements [2].

## 7.6 Environnement d'exécution des agents mobiles

Les environnements d'exécution d'agents mobiles fournissent des services permettant l'exécution d'un agent mobile sur une machine. Ces services sont les suivants : création, migration, exécution d'un agent mobile, communication et échange de messages entre les agents mobiles, accès aux ressources locales par les agents mobiles.

## 7.7 Les différents types d'agents mobiles

Les agents mobiles sont classifiés en deux grands types d'agents mobiles : les légers et les lourds.

### 7.7.1 Les agents légers :

Ils sont nommés « légers », parce qu'il s'agit d'agents de courte taille dans le sens où leur code exécutable est diminué le plus possible et les informations qu'ils véhiculent sont peu volumineuses.

#### 7.7.2 Les agents lourds :

Ces agents sont dits « lourds », car la taille du code exécutable ainsi que celle des données transportées seront beaucoup plus volumineuses que celles des agents légers.

### 7.8 Services requis pour l'exécution d'agents mobiles

Nous allons présenter les services suivants:

- Création d'un agent.
- Transfert d'un agent.
- Désignation d'un agent, c'est-à-dire offrir un nom globalement unique à un agent.
- Plate-forme et Standardisation.
- Localisation d'un agent.
- Communication entre agents.
- Exécution d'un agent.
- Protection d'un agent.

#### 7.8.1 Création d'agents mobiles

Dans la mesure où ces agents peuvent se déplacer, on peut se poser la question, lors de leur création, de l'endroit où l'on souhaite qu'ils démarrent leurs activités. Une fois créé, l'agent peut être actif, c'est-à-dire que l'exécution de son code est déclenchée [22].

#### 7.8.2 Service de nommage

Dans un système à agents mobiles, les agents ont besoin de communiquer entre eux, ce qui nécessite de les nommer. Le nom donné à l'agent doit être globalement unique. Dans la plupart des systèmes à agents mobiles ce nom est construit à partir du nom de la machine (ou adresse IP) où l'agent s'exécute, d'un numéro de port et d'un identificateur localement unique [22].

#### 7.8.3 Plate-forme et Standardisation

La plupart des plates-formes d'agents mobiles ont introduit une collection de termes identiques qui ont servi d'apport dans la standardisation des intergiciels agents.

Maintenant, certaines plateformes de développement d'agents mobiles ont et commercialisées. L'ensemble des services fournis varie d'une plateforme à une autre. Cette variété est due à plusieurs facteurs, notamment le domaine d'application et le modèle d'agent adopté (réactif ou proactif).

On différencie trois approches pour concevoir et implémenter une plateforme d'agents mobiles. La première consiste à recourir à un langage de programmation qui comprend des instructions pour les agents mobiles.

La deuxième approche consiste à implanter le système d'agents mobiles comme des extensions du système d'exploitation. Enfin, la dernière approche construit la plate-forme comme une application spécialisée qui tourne au-dessus d'un système d'exploitation.

Plusieurs systèmes, commerciaux et autres, supportant la mobilité sont actuellement disponibles. Ils sont basés généralement sur la dernière approche et implémenté comme une collection des bibliothèques java.

➤ Aglets

Aglet est une norme basée sur java proposée pour développer des agents mobiles. Elle a été développée par une équipe de chercheurs du laboratoire de recherche d'IBM à Tokyo au début 1995, son but est de fournir une plateforme uniforme pour les agents mobiles dans un environnement hétérogène.

Les aglets sont des objets java mobiles qui peuvent se déplacer d'une machine à une autre. Ainsi, un aglet qui s'exécute sur un hôte peut stopper son exécution, se déporter vers un hôte distant et continuer cette exécution dans son nouvel environnement.

La technologie Aglets et son environnement de développement ASDK (Aglets Software Développement Kit) sont actuellement maintenus au travers d'un site communautaire, selon un modèle libre et open source. L'ASDK permet de programmer rapidement et simplement des agents logiciels mobiles en Java. Le modèle d'agent est simple, associant à un objet java un thread et des méthodes d'activation spécifiques. L'exécution est supportée par un serveur d'Aglets nommé Tahiti qui permet de gérer les communications, la sécurité et la mobilité des agents.

➤ Jade

Jade est une plateforme de développement de système multi agent qui est implémentée en java, jade possède trois modules principaux :

DF (Directory Facilitator) : fournit un service de pages jaunes à la plateforme.

ACC (Agent Communication Channel) : gère la communication entre les agents.

AMS (Agent Management System) : supervise l'enregistrement des agents, leur authentification, leur accès et l'utilisation du système.

Ces trois modules sont activés à chaque démarrage de la plateforme, cette plateforme héberge un ensemble d'agents identifiés de manière unique pouvant communiquer de manière bidirectionnelle avec les autres agents.

➤ **PLANGENT**

La plateforme PLANGENT est basée sur le langage Java. Elle s'intéresse à l'adaptation des agents durant leurs déplacements au sein des environnements dynamiques. Le but de la plateforme PLANGENT est de proposer aux agents des mécanismes pour modifier les objectifs intermédiaires nécessaires à la réalisation de leur tâche finale. Cette plateforme est très ancienne.

➤ **Grasshopper**

La plateforme Grasshopper permet la création d'applications multi-agents en fournissant un environnement d'exécution approprié pour eux. Cette plateforme est mise en œuvre en Java et supporte une mobilité faible ainsi que les invocations distantes de manière transparente pour le programmeur.

#### 7.8.4 Communication

Ensemble des processus physiques et psychologiques par lesquels s'effectue l'opération de mise en relation d'un émetteur avec un ou plusieurs récepteurs, dans l'intention d'atteindre certains objectifs.

La communication est un moyen ou une méthode de coopération ou d'interaction entre agents.

➤ **Pourquoi communiquer ?**

Les agents communiquent et interagissent :

- pour synchroniser leurs actions et pour résoudre des conflits (de ressources, de buts ou d'intérêts).
- pour s'aider mutuellement ou pour suppléer aux limites de leurs champs de perception.

➤ **Quand et avec qui communiquer ?**

Les agents communiquent :

- lorsqu'ils sont face à un problème qu'ils ne savent pas résoudre (manque de compétences ou de ressources).
- lorsqu'il est nécessaire de coordonner leurs actions.
- lorsqu'il y a un conflit entre plusieurs agents.

➤ **Comment communiquer ?**

Deux procédures de communication pour véhiculer les messages (porteurs d'informations ou d'actions) entre agents :

1. communication par partage d'informations.
2. communication par envoi de messages.

### ➤ **Les Interactions entre agents**

Comme les systèmes multi-agents sont constitués d'une société d'agent, l'interaction entre ces agents est essentielle pour la mise en œuvre de l'organisation. " Une interaction est la mise en relation dynamique de deux ou plusieurs agents par le biais d'un ensemble d'actions réciproques.

### **Coopération**

La coopération est la forme générale d'interaction la plus étudiée dans les S.M.A.

**1-Coopération par partage de tâches et des résultats** : avec la possibilité de prendre en compte localement les plans des autres.

**2-Commande** : Un agent supérieur « A » décompose le problème en sous-problèmes qu'il répartit entre les autres agents « Xi ». Ceux-ci résolvent et renvoient les solutions partielles à « A ».

**3-Appel d'offre** : « A » décompose le problème en des sous-problèmes dont il diffuse la liste. Chaque agent « Xi » qui souhaite résoudre l'un des sous-problèmes envoie une offre ; « A » choisit parmi celles-ci et distribue les sous-problèmes. Le système travaille ensuite en mode commande.

**4-Compétition** : Dans le mode compétition, « A » décompose et diffuse la liste des sous-problèmes comme dans le mode appel d'offre. Chaque agent « Xi » résout un ou plusieurs sous-problèmes et envoie les résultats correspondants à « A » qui à son tour fait le tri.

### **Coordination**

Une coordination peut être nécessaire pour améliorer le fonctionnement global du système. La notion de coordination entre agents peut être définie comme la description des différentes étapes respectées par les intervenants lors d'une coopération afin d'en garantir le déroulement intégral et interrompu. Les agents travaillent sur des problèmes dont les solutions sont utiles pour les autres agents. Leur travail doit donc être coordonné dans le temps. La coordination permet aux agents de considérer toutes les tâches et de ne pas dupliquer le travail. La coordination des actions est liée à la planification et à la résolution des conflits, car c'est à ce niveau qu'on tient compte des actions (plans) des autres agents.

On peut distinguer deux types de coordination :

**1. la coordination due à la gêne :**(problème de navigation : les agents doivent coordonner leurs plans de navigation pour s'éviter mutuellement).

**2. la coordination due à l'aide :** (manutention : dans un environnement multi-robots, les agents doivent synchroniser leurs actions pour pouvoir agir efficacement et transporter un objet.

### **Négociation**

Les activités des agents dans un système distribué sont souvent interdépendantes et entraînent des conflits. Pour les résoudre, il faut considérer les points de vue des agents, les négocier, et utiliser des mécanismes de décision concernant les buts sur lesquels le système doit se focaliser.

La négociation est caractérisée par :

- Un faible nombre d'agents impliqués dans le processus.
- Un protocole minimal d'actions : proposer, évaluer, modifier et accepter ou refuser une solution.

Le problème de la négociation ne consiste pas forcément à trouver un compromis mais peut s'étendre à la modification des croyances d'autres agents pour faire prévaloir un point de vue.

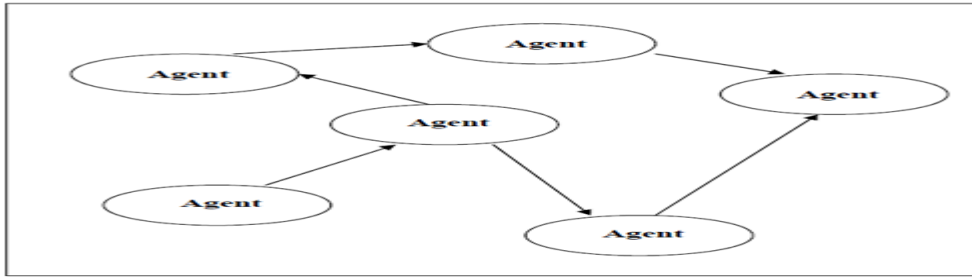
#### ➤ **Modèles de communication pour les agents mobiles**

En général, les techniques de communication des agents mobiles sont classées en deux types de modèle de communication:

- Passage des messages

Cette technique de communication permet aux agents d'envoyer des messages entre eux. C'est l'une des formes de communication directe ou l'agent qui envoie un message doit connaître le nom et l'emplacement de l'agent qui doit recevoir ce message, cela nécessite que soit l'agent connu à l'avance les autres agents, soit par la mise en œuvre d'un service qui permet aux agents d'obtenir des informations sur les autres agents. D'une autre part l'agent récepteur doit comprendre le message reçu et décider quelle action à faire.

La forme la plus simple de passage de message est la communication point à point dont un seul agent émetteur envoie un message à un seul récepteur. La deuxième forme de passage de message est la communication multi point (multicast ou broadcast), ce type est utilisé lorsque plusieurs agents travaillent ensemble, diffusent plusieurs messages aux autres agents pour résoudre un problème.



**Figure 1.2** Communication par passage de message

- Espace d'information

Ce modèle de communication fournit à tous les agents un seul espace pour l'échange d'information. C'est une forme indirecte de communication. Tout simplement un agent peut écrire une donnée dans un espace commun et les autres agents peuvent le lire, d'après [1].

#### 7.8.5 Exécution d'un agent

Un système d'agents mobiles doit offrir la possibilité d'exécuter un agent sur les machines qui vont accueillir ce dernier, ainsi l'agent utilise les ressources disponibles sur la machine afin d'accomplir la tâche demandée par le client. Sur la machine d'accueil, les agents mobiles peuvent être amenés à effectuer des entrées/sorties, accéder à l'interface utilisateur, au système de gestion de fichiers, aux applications externes et à l'interface réseau. Les machines qui forment le système d'exécution n'auront pas toutes le même type de système d'exploitation. Par conséquent, maintenir l'indépendance vis à vis du système d'exploitation d'un environnement d'agents mobiles tout en autorisant en même temps l'accès aux ressources est un problème important à surmonter dans les environnements d'exécution d'agents mobiles. C'est pour cette raison que la plupart des environnements d'agents mobiles utilisent le langage Java comme langage d'implémentation du code mobile [6].

#### 7.8.6 Sécurité

Garantir la sécurité dans les environnements d'agents mobiles est important du fait de la nature même du modèle d'exécution des agents et du fait des interactions des agents mobiles avec plusieurs systèmes et ressources. Un système à agents mobiles doit offrir des mécanismes permettant une exécution sécurisée des agents au sein du système [22].

### 7.9 Cycle de vie et contrôle de l'agent mobile

Un environnement d'exécution pour les agents mobiles doit offrir à l'utilisateur la possibilité de contrôler les activités de l'agent. Depuis sa création jusqu'à sa terminaison, un agent peut passer par plusieurs étapes (cycle de vie d'un agent). Un langage de programmation d'agents mobiles doit offrir au programmeur des points

d'entrée qui vont lui permettre de contrôler l'activité de son agent. Un agent passe par une partie ou la totalité des étapes suivantes:

#### 7.9.1 Création et initialisation

Lors de sa création, l'agent peut être initialisé avec des informations nécessaires à son exécution telles qu'un itinéraire, des préférences de son utilisateur, etc. Par ailleurs, un agent est initialisé par le système avec des informations nécessaires à son interaction avec son environnement, comme par exemple l'identité de son utilisateur.

#### 7.9.2 Migration

L'agent se déplace entre les machines du système. Le but de cette migration est souvent motivé par une coopération en local avec des agents fixes ou d'autres agents mobiles s'exécutant sur le même site. Avant de reprendre ses activités, l'agent aura besoin des informations sur le nouveau site.

#### 7.9.3 Activation et désactivation

Dans une application basée sur les agents mobiles, un agent se désactive en suspendant son exécution afin d'être sauvegardé sur un support non volatile. L'agent mobile est donc gelé. L'activation est l'opération inverse par laquelle l'agent est restauré afin de continuer son exécution.

#### 7.9.4 Terminaison

Une fois que sa tâche est réalisée, l'agent se termine et son processus d'exécution est tué. Avant sa terminaison l'agent a besoin de livrer un bilan à son utilisateur, [22].

## 8. Conclusion

Dans ce chapitre nous avons introduit les agents mobiles, leurs caractéristiques, leurs avantages, et le cycle de vie de ces entités. Les agents mobiles ont un bon rendement dans les applications de la recherche d'informations.

Dans un prochain chapitre, nous allons présenter des approches à base des agents mobiles pour la recherche sémantique d'information.

## CHAPITRE 2

### LA RECHERCHE SEMANTIQUE D'INFORMATION

#### 1. Introduction

La recherche d'information (RI) est un domaine qui s'intéresse à la structure, à l'organisation, à l'analyse, aux stockages et à la découverte de l'information dont le but est de faciliter l'accès à l'information pour un utilisateur ayant un besoin en information donc la RI est un ensemble des techniques qui permettent de sélectionner à partir d'un ensemble de documents ceux qui sont susceptibles de répondre aux besoins de l'utilisateur.

La recherche sémantique d'information est une branche de la RI qui vise à étudier les significations des termes et s'intéresse à la découverte du sens de l'information.

Le but de ce chapitre est de présenter les notions de base de la recherche sémantique d'information.

#### 2. Les ontologies

##### a. Origine

Ontologie est un terme qui a été défini comme une branche fondamentale de la métaphysique en philosophie, elle est apparue au début des années 1990, qui permet de traiter la nature des êtres, en tant qu'être, c'est-à-dire étudier la notion d'existence et étudier aussi les critères principaux de l'existence, ce terme est construit à partir des racines grecques **onto** qui veut dire ce qui existe et **logos** qui veut dire l'étude donc l'étude de l'existence.

##### b. Définition

Dans le cadre de l'intelligence artificielle, **Neeches** et ses collègues furent les premiers à proposer une définition à savoir : «une ontologie définit les termes et les relations de base du vocabulaire d'un domaine ainsi que les règles qui indiquent comment combiner les termes et les relations de façon à pouvoir étendre le vocabulaire» [13].

- ❖ En 1993, **Gruber** propose la définition suivante : «spécification explicite d'une conceptualisation» qui est jusqu'à présent la définition la plus citée dans la littérature en intelligence artificielle» [6]. Cette définition a été modifiée légèrement par **Borst** comme «spécification formelle d'une conceptualisation partagée» [18].

Ces deux définitions sont regroupées dans celle de **Studer** comme «spécification formelle et explicite d'une conceptualisation partagée» [14].

- ❖ Formelle : l'ontologie doit être lisible par une machine, ce qui exclut le langage naturel. Explicite : la définition explicite des concepts utilisés et des contraintes de leur utilisation.
- ❖ Conceptualisation : le modèle abstrait d'un phénomène du monde réel par identification des concepts clefs de ce phénomène.
- ❖ Partagée : l'ontologie n'est pas la propriété d'un individu, mais elle représente un consensus accepté par une communauté d'utilisateurs.

Pour **Guarino & Giaretta** «Une ontologie est une spécification rendant partiellement compte d'une conceptualisation» [7].

**Swartout** et ses collègues la définissent comme suit : « une ontologie est un ensemble de termes structurés de façon hiérarchique, conçue afin de décrire un domaine et qui peut servir de charpente à une base de connaissances» [15].

La même notion est également développée par **Gomez** comme: «une ontologie fournit les moyens de décrire de façon explicite la conceptualisation des connaissances représentées dans une base de connaissances. » [4].

Donc une Ontologie est un ensemble structuré de connaissances dans un domaine particulier qui permet de chercher à en représenter le sens des termes, elle exprime les relations entre ces connaissances.

### c. Composants d'une ontologie

Les composants d'une ontologie sont :

- i. Les concepts : appelés aussi termes ou classe de l'ontologie, est un élément de la pensée sémantiquement estimable et communicable.
- ii. Les relations : appelés aussi propriétés sont l'ensemble des interactions et associations existant entre les concepts qui permettant de construire des représentations de la connaissance du domaine à étudier.
- iii. Les fonctions : constituent des cas particuliers des relations.
- iv. Les axiomes : appelés aussi règles, sont des expressions qui sont toujours justes, l'objectif de ces règles est la description des concepts et des relations qui permettent de représenter les concepts dans la phase sémantique.
- v. Les instances : appelés aussi individus c'est la représentation des éléments dans un domaine à étudier.

### d. Types d'ontologie

On distingue six types :

i. Ontologie de représentation des connaissances

Ce type d'ontologie permet de formaliser les concepts dans la formalisation des connaissances.

ii. Ontologie de haut niveau / supérieure

C'est une ontologie générale qui décrit des concepts tels que : les entités, les relations, les propriétés c'est-à-dire étudie les critères des choses qui existe.

iii. Ontologie générique

C'est une Méta-ontologie utilisable dans différents domaines, les concepts qui y sont définis sont génériques, ces concepts permet de résoudre des problèmes génériques.

iv. Ontologie du domaine

Cette ontologie régit un ensemble de vocabulaires et de concepts qui décrivent un domaine d'application ou monde cible. Elle permet de créer des modèles d'objets du monde cible. L'ontologie du domaine est une méta-description d'une représentation des connaissances, c'est-à-dire une sorte de méta-modèle de connaissance dont les concepts et propriétés sont de type déclaratif. La plupart des ontologies existantes sont des ontologies du domaine. Selon Mizoguchi, l'ontologie du domaine caractérise la connaissance du domaine où la tâche est réalisée. Dans le contexte de la formation à distance, un domaine serait par exemple : le téléapprentissage [11].

Donc l'ontologie de domaine permet de décrire des entités du domaine, et leurs propriétés, elles sont réutilisables dans différentes applications dans le même domaine.

v. Ontologie de tâches

Ce type d'ontologies est utilisé pour conceptualiser des tâches spécifiques dans les systèmes, telles que les tâches de diagnostic, de planification, de conception, de configuration, de tutorat, soit tout ce qui concerne la résolution de problèmes. Elle régit un ensemble de vocabulaires et de concepts qui décrit une structure de résolution des problèmes inhérente aux tâches et indépendante du domaine. [11].

vi. Ontologie d'application

Cette ontologie est la plus spécifique. Les concepts dans l'ontologie d'application correspondent souvent aux rôles joués par les entités du domaine tout en exécutant une certaine activité [9]. C'est-à-dire elles comportent un grand ensemble d'informations essentielles qui permet de modéliser les connaissances pour une application spécifique.

**e. Domaine d'application d'ontologies**

Les ontologies sont utilisées dans plusieurs domaines, les plus répandus sont :

- La communication
- L'interopérabilité entre les systèmes
- Ingénierie des systèmes.

#### **f. Formalismes de représentation des ontologies**

Il existe de nombreux formalismes pour représenter les ontologies, parmi ces formalismes :

##### **i. Les frames**

Introduit dans les années 70 par minsky, c'est un modèle de base qui permet de représenter une connaissance ou un certain concept dans un domaine.

##### **ii. Graphes conceptuels**

Sont conçus pour l'analyse et la compréhension applicables à la représentation des connaissances, ils sont décomposés en deux niveau :

- le niveau terminologique ou sont écrits les concepts, les relations, les instances de concepts.

- le niveau assertionnel ou sont représentés les règles et les contraintes.

### **3. La recherche d'information**

#### **a. Définition**

La recherche d'information est un ensemble des techniques et des procédures permettent de sélectionner à partir d'un ensemble de documents ceux qui sont susceptibles de répondre aux exigences du client

#### **b. Concepts de base de la RI**

Il existe plusieurs concepts de base de la recherche d'information, on peut citer :

- i. Collection de documents** : est un ensemble des informations qui sont utilisées et exploitées par un utilisateur pour réaliser certains objectifs, elle est constitué d'un ensemble de documents.
- ii. Documents** : c'est une information élémentaire d'une collection de documents qui représente tout ou une portion d'un document. C'est-à-dire toute unité qui peut composer une réponse à une exigence informationnel de l'utilisateur.
- iii. Requête** : est l'expression du besoin informationnelle de l'utilisateur.
- iv. Pertinence** : c'est un concept principal dans le domaine de la recherche d'information, elle est définie comme la correspondance entre un document et une requête.

v. **Besoin d'information** : la notion de besoin en information en recherche d'informations est souvent appropriée au besoin de l'utilisateur. Trois types de besoin utilisateur ont été définis :

- ✓ **Besoin vérificatif** : l'utilisateur cherche à vérifier le texte avec les données connues qu'il possède déjà. Il recherche donc une donnée particulière, et sait même souvent comment y accéder. La recherche d'un article sur Internet à partir d'une adresse connue serait un exemple d'un tel besoin. Un autre exemple serait de chercher la date de publication d'un ouvrage dont la référence est connue. Un besoin de type vérificatif est dit stable, c'est-à-dire qu'il ne change pas au cours de la recherche.
- ✓ **Besoin thématique connu** : l'utilisateur cherche à clarifier, à revoir ou à trouver de nouvelles informations dans un sujet et un domaine connus. Un besoin de ce type peut être stable ou variable : il est très possible en effet que le besoin de l'utilisateur s'affine au cours de la recherche. Le besoin peut aussi s'exprimer de façon incomplète, c'est-à-dire que l'utilisateur n'énonce pas nécessairement tout ce qu'il sait dans sa requête mais seulement un sous-ensemble. C'est ce qu'on appelle dans la littérature le label.
- ✓ **Besoin thématique inconnu** : cette fois, l'utilisateur cherche de nouveaux concepts ou de nouvelles relations en dehors des sujets ou des domaines qui lui sont familiers. Le besoin est intrinsèquement variable et est toujours exprimé de façon incomplète.

### c. **Les modèles de recherche d'information**

Dans cette partie, nous allons discuter les trois approches principales de modélisation de la recherche d'information, à savoir le modèle booléen, le modèle vectoriel, le modèle probabiliste.

#### i. **Modèle booléen**

Le modèle booléen est appuyé sur la théorie des ensembles. Dans ce modèle, les documents et les requêtes sont représentés par des ensembles de mots clés. Chaque document est représenté par une réunion logique des termes non pondérés qui constitue l'index du document.

#### ii. **Modèle vectoriel**

Il a été introduit en 1975 et développé par la suite par d'autres chercheurs. Il représente les documents et les requêtes comme des vecteurs de poids des concepts. Le poids dans un vecteur peut être le nombre d'occurrences d'un terme correspondant dans ce document ou dans la requête. Ces vecteurs prennent leur signification dans un espace

vectorel qui est défini par l'ensemble des concepts que le système a rencontré durant l'indexation.

iii. **Modèle probabiliste**

Dans la RI, le modèle probabiliste classe les documents dans un ordre décroissant selon leur probabilité de pertinence pour une requête exprimant le besoin d'information d'un usager. Les concepteurs de ce modèle ont bâti leur approche sur la théorie de probabilités ainsi que celle des statistiques pour évaluer les probabilités de pertinence.

**d. Système de recherche d'information**

C'est un système informatique qui permet de retrouver à partir d'un ensemble de documents, ceux dont le contenu correspond le mieux à leur exigence d'information de l'utilisateur.

**e. Architecture d'un Système de recherche d'information**

Architecture d'un Système de recherche d'information intègre les fonctionnalités suivantes :

- La représentation de documents et de requêtes (indexation).
- La recherche.
- La reformulation de requêtes.

❖ **La représentation :**

Pour réaliser la correspondance entre un document et une requête, il faut comparer les concepts qui présentent dans ces deux éléments, bien que, les collections comportent un nombre élevé de documents de taille hautes donc la comparaison devient coûteuse, afin de diminuer cette complexité il faut transformer les documents de manière plus faciles à manipuler.

❖ **La recherche :**

Dans ce processus, le système décide quels sont les documents qui correspondent à la requête de l'utilisateur, cette décision est reposée sur un ensemble de fonctions mathématiques afin de pouvoir évaluer certaines relations particulièrement la relation d'appariement entre la requête et le document.

❖ **La reformulation de requêtes :**

Le but de cette étape est de modifier la requête de l'utilisateur en ajoutant des termes qui peuvent provenir de différentes sources.

**f. Les phases du processus de recherche d'information**

L'objectif principal d'un processus de RI est de sélectionner les documents du besoin en information de l'utilisateur décrit par une requête. Donc on a deux phases importantes dans le déroulement du processus : indexation et appariement requête/documents.

i. Indexation

L'indexation est une étape primordiale dans le processus de RI. Elle consiste à extraire les termes qui représentent le contenu d'un document ou d'une requête, qui couvrent au mieux son contenu sémantique de documents. Donc chaque document associer une liste de mots clés appelée descripteur qui associer à des poids qui représentent leurs degrés de représentativité de l'unité textuelle.

**Les Mode d'indexation :** l'indexation peut être manuelle, automatique ou semi-automatique :

-Indexation manuelle : chaque document est analysé par un spécialiste du domaine correspondant ou par un documentaliste.

-Indexation automatique : chaque document est analysé à l'aide d'un processus automatisé. il se base sur des procédures algorithmiques et elle passe par un ensemble d'étape pour crée de façon automatique l'index, ces étapes sont : l'analyse lexicale, l'élimination de mots vides, la normalisation, la sélection des descripteurs, la taille de chaque document et enfin création de l'index.

-Indexation semi-automatique : l'indexation est réalisée par un programme informatique et un spécialiste de domaine

Le but de l'indexation est l'extraction des descripteurs qui peuvent être :

-Des mots simples du texte du document.

-Les racines des mots extraire.

-Des concepts qui sont des expressions contenant un ou plusieurs mots.....

**Fonction de pondération :**

La pondération est une fonction principale dans la recherche d'information, elle permet d'affecter à chaque terme une valeur qui indique son importance de ce terme dans ce document, la majorité des méthodes de pondération se basent sur des aspects statistiques, elles tirent leur origine de la loi de Zipf et de la conjecture de luhn.

ii. L'appariement requête-document

Les SRI intègrent un processus de recherche qui permet de sélectionner l'information pertinente pour l'utilisateur. A cet effet, une mesure de similitude entre la requête indexée et les descripteurs des documents de la collection est calculée. Seuls les documents dont la similitude dépasse un seuil prédéfini sont sélectionnés par le SRI.

Il existe deux types d'appariement :

✓ Appariement exact

Le résultat est une liste de documents respectant exactement la requête spécifiée avec des critères clairs. Les documents retournés ne sont pas triés.

✓ Appariement approché

Le résultat est une liste de documents censés être pertinents pour la requête. Les documents retournés sont triés selon un ordre de mesure.

## 4. La recherche sémantique d'information

### 4.1 Définition

La recherche sémantique d'information est une branche de la RI qui vise à étudier les significations des termes dans les documents pour permettre une meilleure compréhension du contenu de document et du sens de l'information afin de faciliter l'accès à l'information pour un utilisateur ayant un besoin en information.

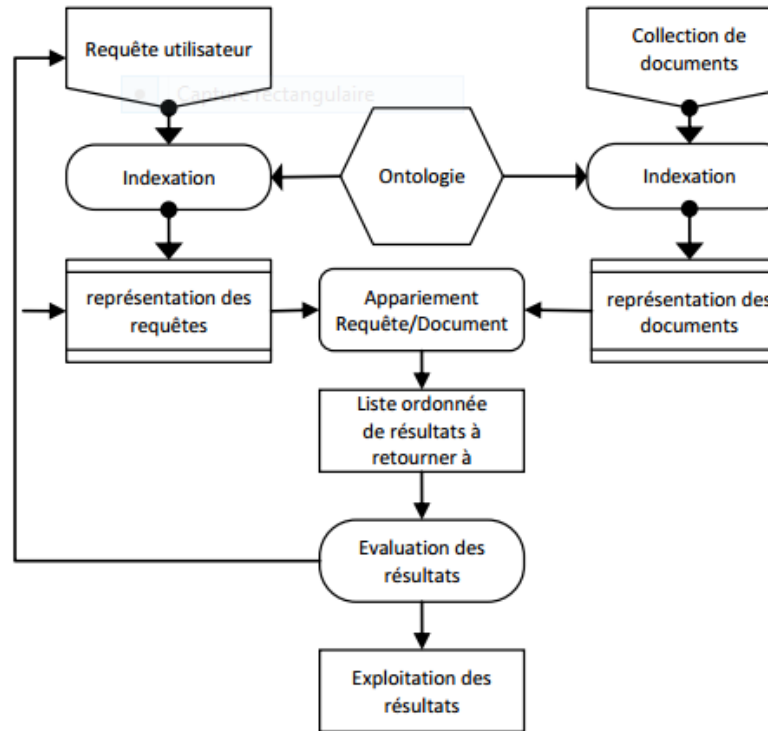
### 4.2 La recherche d'information guidée par les ontologies

Dans le domaine de sciences de l'information Les ontologies sont essentielles pour manipuler la sémantique de l'information, elles fournissent des représentations qui permettent la compréhension du sens de ce qu'elle manipule dans le but est que la machine comprenne le sens de l'information qu'elle véhicule.

#### 4.2.1 Architecture d'un système de recherche guidée par les ontologies

Dans un Système de recherche d'informations guidée par les ontologies les documents et les requêtes sont représentés par un intervalle, cet intervalle contient des mots précis et simples qu'on trouve dans les formes classiques, ce système est construit en trois couches, **la couche application** qui représente l'interface du système, elle permet à l'utilisateur de sélectionner des documents et de formuler des requêtes, **La couche logicielle** offre les implémentations pour la recherche sémantique d'informations. **La couche pérennisation** des données.

Le processus d'une recherche sémantique peut être illustré comme suit :



**Figure 2.1** processus de la recherche sémantique [26].

#### 4.2.2 Contraintes liées à la recherche sémantique

L'opération de la recherche sémantique d'informations implique le respect des contraintes liées aux ontologies existantes, on distingue généralement trois types de contraintes :

##### 4.2.2.1 Présélection des ontologies

Cette contrainte consiste à choisir les ontologies du domaine. Elle est dite présélection des ontologies, quand elle est réussite.

##### 4.2.2.2 Intégration des ontologies

Dans cette contrainte Lortal, (2002) et Koli, (2005) dressent un bilan complet de l'intégration des ontologies et distinguent entre l'intégration par assemblage et l'intégration par fusion : *merging/matching*.

##### 4.2.2.3 Annotation du corpus

Cette contrainte consiste à décrire le contenu des documents qui appelé Annotation du corpus, La construction de l'index sémantique.

Une annotation est une information liée à un document qui peut prendre plusieurs formes : des images, des icônes, des sons etc.

### 4.3 Indexation conceptuelle

L'indexation conceptuelle est un cas particulier d'indexation en recherche d'informations (RI). Elle consiste à passer par tous le texte, ainsi que trouver toutes les informations associées et stocker le résultat dans le but de pouvoir retrouver celles-ci facilement.

#### 4.3.1 L'indexation sémantique

C'est une partie d'indexation qui s'appuie sur le sens des termes, elle est basée sur des algorithmes de désambiguïsation de mots (*wsd*) qui permettent de ramener tous les mots qui possède le même sens, Dans le but est d'indexer les documents et les requêtes avec le sens des mots (*mots-sens*).

##### 4.3.1.1 La méthode de Voorhees

En 1993, **Voorhees** a créé un outil particulier à la désambiguïsation qui repose sur le terme *WordNet*. Ce terme est un réseau utilisé dans le domaine sémantique et structuré concernant le concept *synset*, Un *synset* consiste à rassembler des termes que ce soit simples ou composés ayant le même sens dans un contexte donné.

Pour supprimer l'ambiguïté d'un mot, les *synsets* de ce mot calculent la valeur entre le contexte de ce mot et un voisinage contenant les mots du *synsets* dans la hiérarchie de *WordNet*.

Après l'application de cette approche sur l'ensemble de tests de désambiguïtés effectué par Voorhees, ils ont trouvé que l'outil de désambiguïsation était incapable de déterminer de manière exacte le sens attendu des mots.

##### 4.3.1.2 La méthode de Krovetz&Croft

Dans les années 1992, Krovetz et Croft faire une grande étude qui permet de connaître la relation de correspondance du sens des mots dans les requêtes et les documents. Leur but était de trouver l'existence d'une relation entre la **correspondance/non correspondance** des sens.

### 4.4 Similarités entre concepts dans une ontologie

La similarité sémantique est une évaluation qui permet d'évaluer le lien sémantique entre deux concepts pour apprécier le degré par lequel les concepts sont proches dans leur sens. La similarité entre deux concepts est liée aux certaines caractéristiques. On dit que La similarité est idéale dans le cas où les deux concepts possèdent le même sens donc, elles sont identiques.

#### 4.4.1 Extension des requêtes via une ontologie

Dans un système de recherche d'informations, on utilise les ontologies pour faire des modifications sur les requêtes, tout d'abord, il faut découvrir les termes qui existes dans la requête, ensuite étendre ces termes par les termes qui indiquent d'autres concepts qui sont

proches de la requête, ces termes sont identifiés grâce aux liens sémantiques entre concepts qu'offre l'ontologie.

#### 4.4.1.1 Identification des concepts candidats

Pour accomplir les modifications à la requête, il doit identifier les termes de la requête qui représentent des concepts dans l'ontologie, pour ce faire il nécessite découvrir les termes formés par des termes uniques ou un ensemble des mots, La méthode qui est utilisée consiste à concaténer les mots voisins (qui sont trouvés dans le texte) et voir s'ils correspondent à une (des) entrée(s) dans l'ontologie.

Il existe deux cas:

- **Projeter le document sur l'ontologie** : dans ce cas, Pour chaque concept candidat formé, il ordonne les mots voisins qui existent dans le texte, ensuite on utilise ces mots pour vérifier la correspondance entre ces mots et les entrées qui sont trouvées dans l'ontologie, s'ils ne correspondent pas alors on utilise leurs formes de base qui permet de résoudre les problèmes attachés aux variations de la morphologie des mots.
- **Projeter l'ontologie sur le document** : dans ce cas, il identifie les termes qui apparaissent dans le document. Cette approche permet la construction d'une ressource réutilisable.

#### 4.4.2 Mesures reposant sur la distance

Les mesures qui s'appuient sur la distance observent que la similarité entre deux concepts est calculée par le nombre de liens qui séparent les deux concepts.

##### 4.4.2.1 Mesure de RADA

C'est une méthode basée sur la mesure du « Edge Counting » introduite par Rada (1989) [24], c'est une méthode qui consiste à mesurer la distance sémantique par le nombre de liens qui séparent les concepts, on utilise le plus court chemin. Donc la similarité sémantique entre deux concepts correspond à l'inverse de la distance entre deux concepts. Plus deux concepts sont distants, moins ils sont similaires.

##### 4.4.2.2 Mesure de Wu-Palmer

C'est une méthode qui a été introduite par Wu et Palmer (1994), consistant sur la profondeur des concepts.

Pour calculer la similarité entre deux concepts  $c_1$  et  $c_2$ , la formule suivante est utilisée :

$$Sim_{WP}(c_1, c_2) = \frac{2 * depth(c)}{depth(c_1) + depth(c_2)} \quad (1)$$

Où depth ( $c_i$ ) correspond le niveau de profondeur du concept  $c_i$  et  $c$  représente le concept le plus spécifique qui généralise  $c_1$  et  $c_2$ . La valeur de la similarité est comprise entre 0 et 1 (1 signifiant que les concepts sont totalement similaires).

#### 4.4.3 Mesures reposant sur le contenu en informations des concepts

Il s'agit de considérer que le contenu en information des concepts de l'ontologie, Le contenu informationnel d'un concept traduit la pertinence d'un concept dans le corpus en tenant compte de sa spécificité ou de sa généralité. Le principe, plus un ensemble de concepts partagent de l'information, plus ils sont similaires [Var, 2005] [Jia, 1998].

Les différentes mesures proposées pour apprécier la similarité entre concepts basé sur ce calcul.

##### 4.4.3.1 Mesure de Resnik

Dans cette méthode, on calcule la similarité entre deux concepts par le contenu en informations de ces concepts qui été proposée dans Resnik (1995). Elle prend des valeurs entre 0 et 1.

4.4.3.2 Une autre approche, [Sec, 2004] qui calcule le contenu informatif d'un concept à partir de WordNet, elle repose sur l'idée que plus un concept « C » a des descendants, moins il est informatif.

## 5. Conclusion

Dans ce chapitre nous nous sommes intéressés à la recherche d'information, ou nous avons parlé sur les ontologies comme composante dans le processus de recherche, et nous avons présenté quelques notions de base de la recherche sémantique d'information et certaines méthodes existantes. Cette recherche fait l'objectif de notre travail, ou nous allons utiliser la technologie agent mobile pour faciliter cette recherche sémantique.

# CHAPITRE 3

## CONTRIBUTION

### 1. Introduction

Dans les dernières années, nous remarquons un accroissement rapide dans la masse d'informations, ce qui pose le problème d'accès à ces informations : La question qui se pose c'est comment peut-on accéder aux informations cherchées sans avoir d'autres similaires syntaxiquement ?

Pour trouver ces informations, nous faisons appel aux différents outils renvoyant un bon nombre de réponses, qui permet de répondre aux besoins de l'utilisateur.

Dans ce chapitre nous déterminons le cadre conceptuel et méthodologique de notre système de recherche sémantique d'informations, utilisant la technologie agent mobiles. Nous présentons dans ce cas, la démarche suivie, l'architecture du système ainsi que les résultats obtenus.

### 2. Objectif et motivations de la démarche

Le but de notre travail est de proposer une approche pour la recherche sémantique d'information en utilisant des agents mobiles pour réaliser une meilleure recherche.

La proposition rassemble la motivation suivante :

- ✓ Approche basée sur les agents mobiles : Le système proposé est composé d'un ensemble d'agents mobiles. L'importance de ces agents est due à plusieurs raisons :

-Flexibilité :

Un agent mobile peut se déplacer entre différents conteneurs tout en restant opérationnel.

-Performance et Efficacité :

L'utilisation des agents mobiles augmente les performances des applications.

### 3. Description générale de la démarche adoptée

La démarche de notre travail se définit comme une suite de recherches qui doivent être effectuées pour obtenir les informations cherchées par un utilisateur.

Notre travail vise à proposer un système de recherche sémantique d'information capable de délivrer des informations pertinentes relatives au sens d'un terme. Nous voulons développer un système basé sur les agents mobiles pour la recherche sémantique

d'information. L'organigramme suivant montre l'ensemble des composants du système et leurs fonctionnalités.

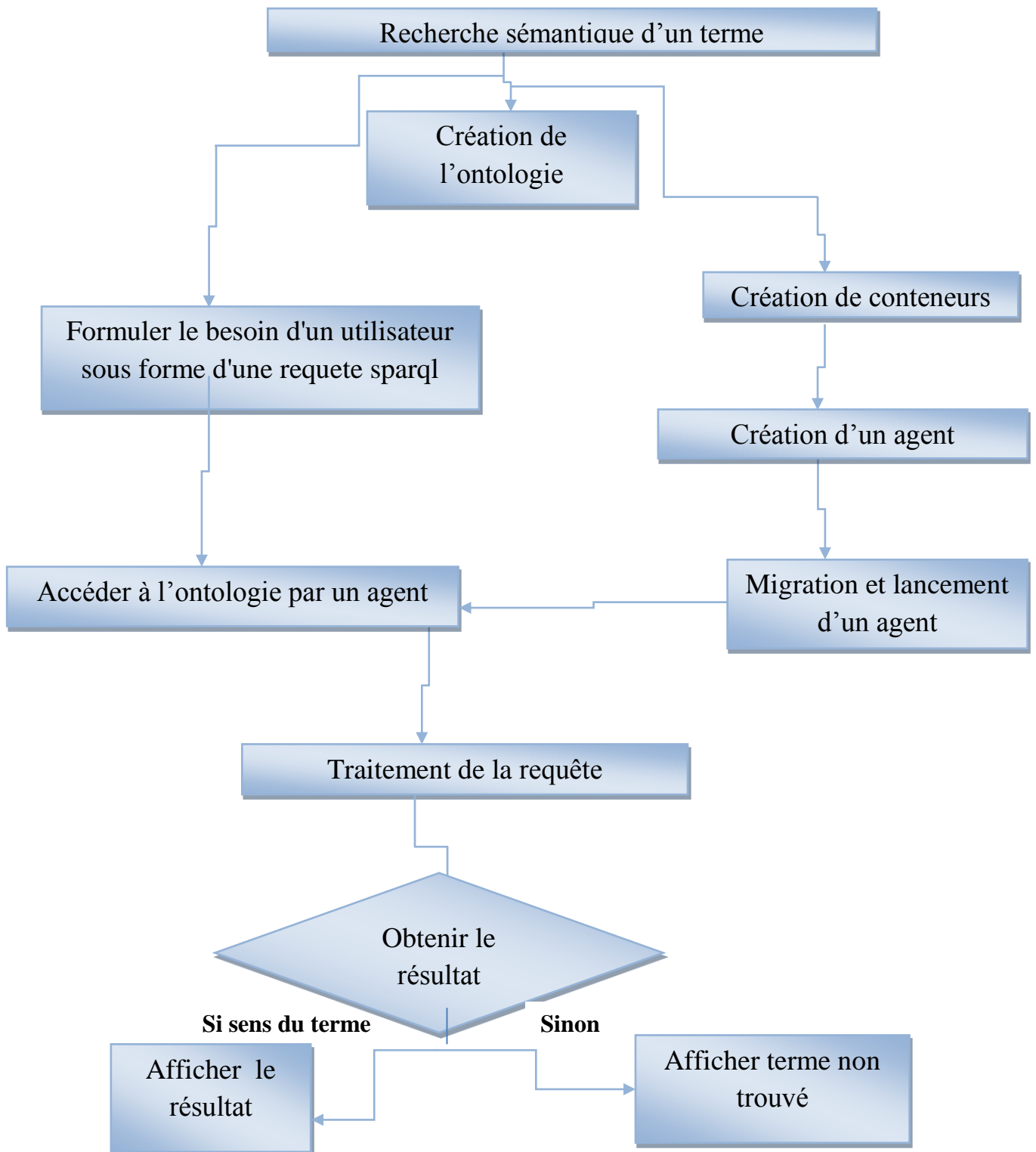


Figure 3.1 Organigramme générale du système

### 3.1 Description de système :

Le fonctionnement du système est composé en différents étapes, à savoir : la recherche du sens d'un terme, et la migration des agents mobiles. Par ailleurs le système est doté des interfaces qui expriment le besoin de l'utilisateur à travers une requête.

#### 3.1.1 La recherche sémantique d'un terme

Dans la recherche sémantique d'un terme nous étudions les sens des termes pour permettre une meilleure compréhension du contenu de l'information afin de faciliter l'accès à l'information pour un utilisateur ayant un besoin en information. Cette recherche contient les étapes suivantes :

##### 3.1.1.1 La création à l'ontologie

Nous avons créé une ontologie Famille par :

##### a- La création des classes et sous-classes de l'ontologie Famille:

- La première classe est la classe personne c'est la classe mère de tous les autres classes.
- Deuxième classe est la classe enfant qui contient de sous classe : la classe fille et la classe fils.
- La classe femme qui est composée des sous classes suivantes : la classe grand-mère, mère, sœur.
- La classe fraterie possède les sous classes suivantes : frère, et sœur.
- La classe grand parent comprend la classe grand-père et grand-mère.
- La classe homme inclut la classe frère, grand-père, et père.
- Et enfin la classe parent englobe la classe mère et père.

##### b- La création des propriétés pour les classes:

-Une personne possède un nom, un age, une nationalité.

- Création de la propriété (type datatype) nom avec le domaine personne et le range xsd:string
- Création de la propriété (type datatype) age avec le domaine personne et le range xsd:int
- Création de la propriété (type datatype) nationalité avec le domaine personne et le range xsd:string

- Création de la propriété (type object) estFilsDe qui est la sous propriété estEnfantDe avec le domaine homme et le range personne
- Création de la propriété (type object) estFilleDe qui est la sous propriété estEnfantDe avec le domaine femme et le range personne
- Création de la propriété (type object) estFrèreDe qui est la sous propriété estEnRelationDeFraterieAvec avec le domaine homme et le range personne
- Création de la propriété (type object) estSoeurDe qui est la sous propriété estEnRelationDeFraterieAvec avec le domaine femme et le range personne
- Création de la propriété (type object) estPèreDe qui est la sous propriété estParentDe avec le domaine homme et le range personne
- Création de la propriété (type object) estMèreDe qui est la sous propriété estParentDe avec le domaine femme et le range personne

c- Assigner les types pour les propriétés:

- la propriété se\_marier\_avec et estEnRelationDeFraterieAvec sont symétriques
- La propriété estEnRelationDeFraterieAvec est transitive
- La propriété estEnfantDe est la propriété inverse de la propriété estParentDe
- La propriété nom, age, nationalité sont fonctionnelle

d- Assigner des instances:

-Création des instances pour la classe homme:

a- Mohamed, 70, de nationalité algérienne.

b- Omar, 40, de nationalité algérienne.

-Création des instances pour la classe femme:

a- Aycha, 69, de nationalité algérienne.

-Réglage des instances pour les deux classes homme et femme:

- Mohamed se\_marier\_avec Aycha
- Omar estFilsDe Mohamed

### 3.1.1.2 Formuler le besoin d'un utilisateur sous forme d'une requete sparql

Après la création de l'ontologie, nous formulons une requete sparql qui permet d'exprimer le besoin d'un utilisateur pour réaliser leur exigence.

### 3.1.2 Création de conteneurs

Pour créer un agent, il faut tout d'abord créer des conteneurs qui permettent la création des agents.

### 3.1.3 Création d'un agent

Afin de réaliser la migration d'un agent, tout d'abord nous créons l'agent qui va migrer entre les conteneurs pour accomplir leur tâche.

#### 3.1.4 La migration d'un agent

Dans la migration d'un agent, nous avons utilisé la migration inter-plateforme qui permet à un agent de migrer entre les différents conteneurs qui existent dans la même plateforme.

#### 3.1.5 Accéder à l'ontologie par un agent

Après la migration d'un agent entre les conteneurs, l'agent va accéder à l'ontologie pour faire la recherche et réaliser le besoin d'un utilisateur.

#### 3.1.6 Traitement de la requete

Lorsque l'agent accède à l'ontologie, il faire la recherche par une requete sparql, selon le contenu de cette requête.

#### 3.1.7 Obtenir et l'affichage de résultat

Après les étapes précédentes, l'agent obtenu le résultat, si il trouve la réponse de requete, il va afficher, sinon il n'affiche rien.

### 4. Les algorithmes principaux

#### a. La recherche sémantique d'information

**Début:**

1. Formuler le besoin d'un utilisateur sous forme d'une requete sparql
2. Migration et lancement d'un agent mobile
3. Accéder à l'ontologie par l'agent
4. Traiter la requete
5. Obtenir le résultat
6. Affichage de résultat par l'agent

**Fin**

#### b. La migration

**Début:**

1. Création de conteneurs
2. Création d'un agent
3. Migration d'un agent entre les conteneurs

**Fin**

### 5. Mise en œuvre du système

## 5.1 Environnement de développement

Avant de commencer la phase de l'implémentation, nous présentons les outils utilisés durant cette phase



Langage de programmation (JAVA)

Pour la préférence de programmation de notre système, nous avons choisis le langage JAVA pour les arguments suivantes :

-Java est complètement orienté objet, cela vous permet de créer des programmes modulaires et code réutilisable, vous permet et vous pousse même à développer vos applications d'une façon orientée objet et vous permet d'avoir une application bien structurée, modulable, maintenable beaucoup plus facilement et efficace. Cela augmente une fois de plus votre productivité.

-Java est facile à apprendre, c'est-à-dire il a été conçu pour être facile à utiliser et il est donc facile à écrire, compiler, déboguer et apprendre que les autres langages de programmation.

-Java est indépendant de la plateforme :

L'un des avantages les plus importants de Java est sa capacité de se déplacer facilement d'un système informatique à un autre. La capacité à exécuter le même programme sur de nombreux systèmes différents est essentielle pour le logiciel World Wide Web, et Java succède à cela en étant indépendant de la plateforme à la fois la source et le niveau binaire.

-Java est distribuée :

Java est conçue pour rendre l'informatique répartie facile avec la fonctionnalité de réseau qui est intrinsèquement intégrées. Écriture de programmes de réseau en Java, c'est comme envoyer et recevoir des données vers et à partir d'un fichier.

-Java est sécurisée :

Java considère la sécurité dans le cadre de son design. Le langage Java, le compilateur, interprète, et l'environnement d'exécution ont chacun été développés avec la sécurité à l'esprit.

-Java est robuste :

Consistant: la fiabilité. Java met beaucoup d'accent sur la vérification rapide pour d'éventuelles erreurs, que des compilateurs Java sont en mesure de détecter de nombreux problèmes qui seraient d'abord montrer au cours de délai d'exécution dans d'autres langues.

-Java est multi-thread :

Multithread est la possibilité pour un programme à exécuter plusieurs tâches simultanément dans un programme. En Java, la programmation multithread a été intégrée

harmonieusement, tandis que dans d'autres langues, système d'exploitation des procédures spécifiques doivent être remises en vue de permettre le multithreading.



Pour la formalisation des ontologies, nous avons choisi pour l'éditeur d'ontologies Protégé. Voici un aperçu sur cet éditeur :

**Protégé** est un système auteur pour la création d'ontologies. Il a été créé à l'université Stanford et est très populaire dans le domaine du Web sémantique et au niveau de la recherche en informatique.

Protégé est développé en Java. Il est gratuit et son code source est publié sous une licence libre (la Mozilla Public License).

Protégé peut lire et sauvegarder des ontologies dans la plupart des formats d'ontologies : RDF, RDFS, OWL, etc.

Il possède plusieurs concurrents tels que Hozo, OntoEdit et Swoop. Il est reconnu pour sa capacité à travailler sur des ontologies de grandes dimensions.

Protégé est une interface modulaire, il permet l'édition, la visualisation et le contrôle d'ontologies, Protégé permet la définition de méta-classes, dont les instances sont des classes, ce qui permet de créer son propre modèle de connaissances avant de bâtir une ontologie. Son interface est conçue de manière excellente, et son architecture logicielle permettant l'insertion de plugins pouvant apporter de nouvelles fonctionnalités, ces pourquoi motivent notre choix de cet éditeur.

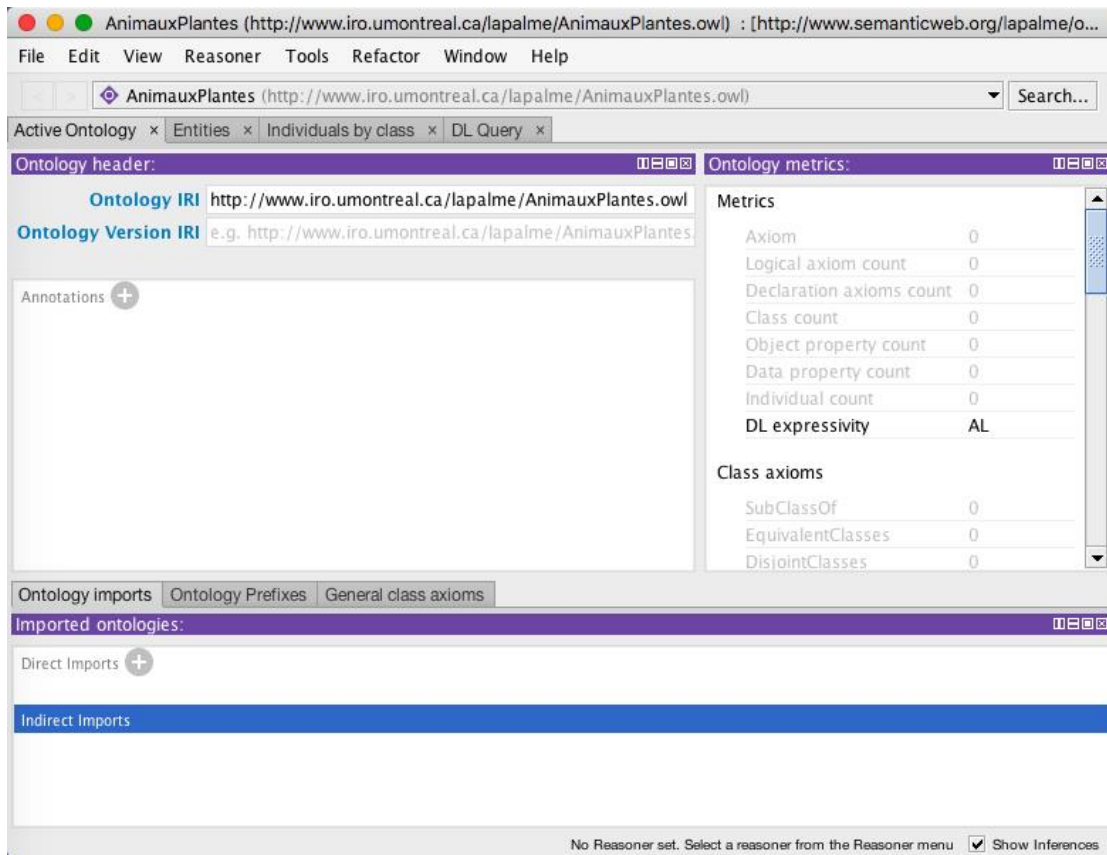


Figure 3.2 Interface principale de protégé

## SPARQL



**SPARQL** (prononcé sparkle en anglais : « étincelle ») est un langage de requête et un protocole qui permet de rechercher, d'ajouter, de modifier ou de supprimer des données RDF disponibles à travers Internet. Son nom est un acronyme récursif qui signifie *SPARQL Protocol and RDF Query*

*Language*.

SPARQL est l'équivalent de SQL car comme en SQL, on accède aux données d'une base de données *via* ce langage de requête alors qu'avec SPARQL, on accède aux données du Web des données. Cela signifie qu'en théorie, on pourrait accéder à toutes les données du Web avec ce standard. L'ambition du W3C est d'offrir une interopérabilité non pas seulement aux niveaux des services, comme avec les services Web, mais aussi aux niveaux des données structurées ou non qui sont disponibles à travers l'Internet.

Ce standard a été créé par le groupe de travail DAWG (*RDF Data Access Working Group*) du W3C (*Consortium World Wide Web*). SPARQL est considéré comme l'une des technologies clés du Web sémantique et le 15 janvier 2008, la version 1.0 est devenue une recommandation officielle du W3C. La version 1.1 permettra d'enregistrer des données et de

fusionner des données de sources différentes. La version 1.1 est devenue depuis le 21 mars 2013 une recommandation.

Les implémentations de SPARQL au sein de triplestores se multiplient. « SPARQL fera une énorme différence », selon Tim Berners-Lee dès mai 2006.

## 5.2 Choix de plateforme

Notre choix s'est porté sur la plate-forme JADE. Notons que nous avons utilisé JADE de la version 4.4.0

### 5.2.1 Plateforme JADE

JADE (Java Agent Development Framework) est une plateforme multi agents répartie (multi-hôtes) développé par F.Bellifemine & A. Poggio, G. Rimassa, P. Turci par la société Telecom Italia Lab « Tilab, anciennement CSELT » en 1999.

JADE développée en Java, fonctionne sous tous les systèmes d'exploitation, inclut tous les composants obligatoires qui contrôlent un SMA, et possède une architecture très précise permettant la construction dite « normalisée » d'agents. Pour tout cela la plateforme JADE contient :

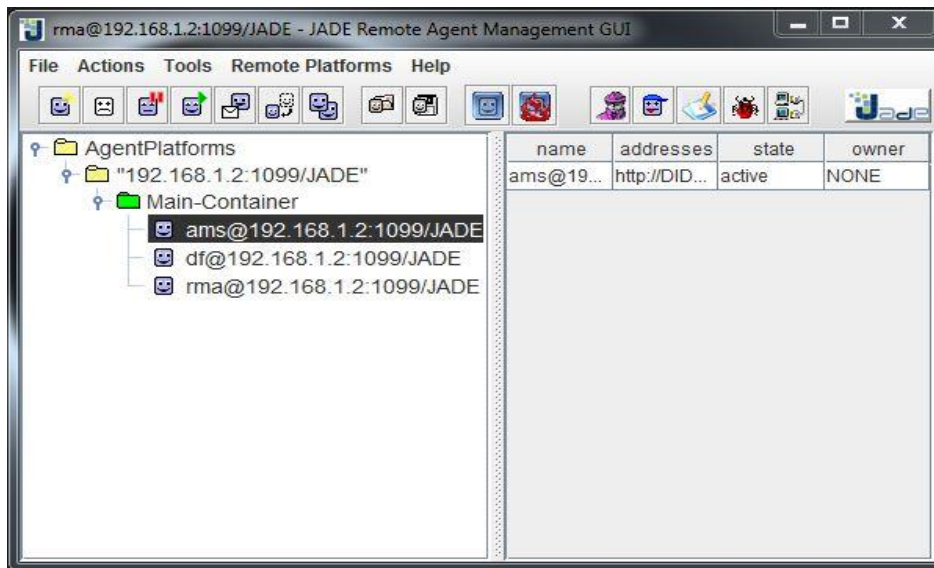
- 1. Un runtime Environment** : l'environnement où les agents peuvent vivre. Il doit être activé pour pouvoir lancer les agents.
- 2. Une librairie de classes** : que les développeurs utilisent pour écrire leurs agents.
- 3. Une suite d'outils graphiques**: qui facilitent le débogage, la gestion et la supervision de la plateforme des agents.

JADE fournit aussi des classes qui implémentent JESS pour la définition du comportement des agents: JESS (outil de raisonnement à base de règles) est le moteur qui exécute tout le raisonnement nécessaire. On a donc la possibilité de créer des agents JADE intelligents en déléguant le raisonnement à d'autres outils.

Chaque instance du JADE est appelée conteneur « container en anglais », qui peut contenir plusieurs agents. Ainsi qu'un ensemble de conteneurs constituent une plateforme et chaque plateforme doit contenir un conteneur spécial appelé *main-container* et tous les autres conteneurs s'enregistrent auprès de celui-là dès leur lancement.

Un *main-container* se distingue des autres conteneurs car il contient toujours deux agents spéciaux appelés AMS et DF qui se lancent automatiquement au lancement du *main-container*. Ainsi on a un conteneur par machine et lorsque l'on lance un conteneur :

- soit il est seul, il devient alors le conteneur principal (main container)
- soit on lui indique l'adresse du conteneur principal.



**Figure 3.3** Interface graphique de la plate-forme Jade.

L'interface graphique de JADE (GUI) a été implémentée comme un agent appelé **RMA** « Remote Monitoring Agent ».

Nous avons choisis la plateforme JADE pour les raisons suivantes :

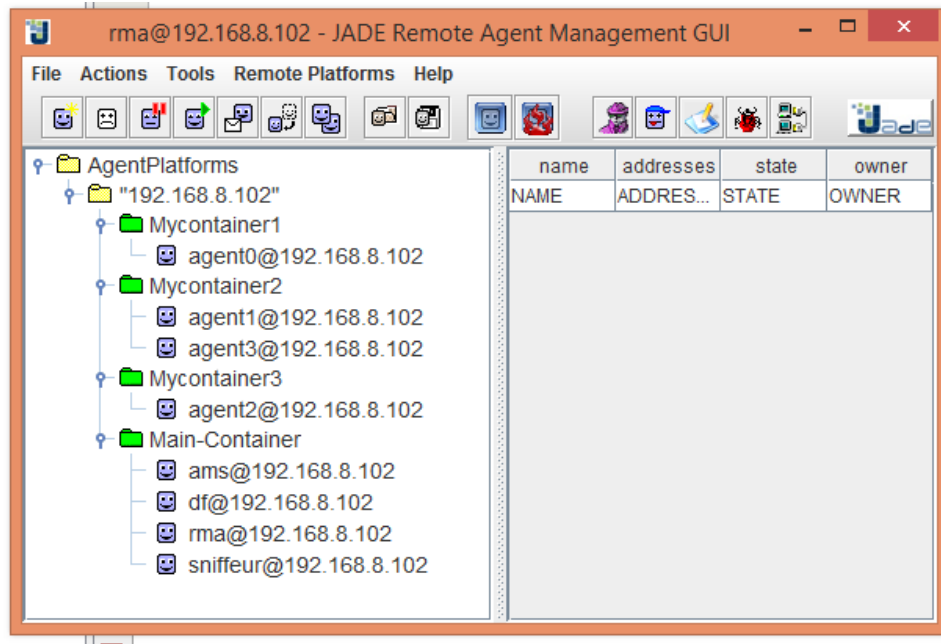
- Facile à utiliser
- Documentation nombreuse et variée
- Evolutif : on peut commencer par quelques agents simples puis faire évoluer l'architecture avec l'application
- Léger (900 ko)
- Mature : première livraison le 13/09/2000(v 1.25)

## 6. Quelques fenêtres de notre application

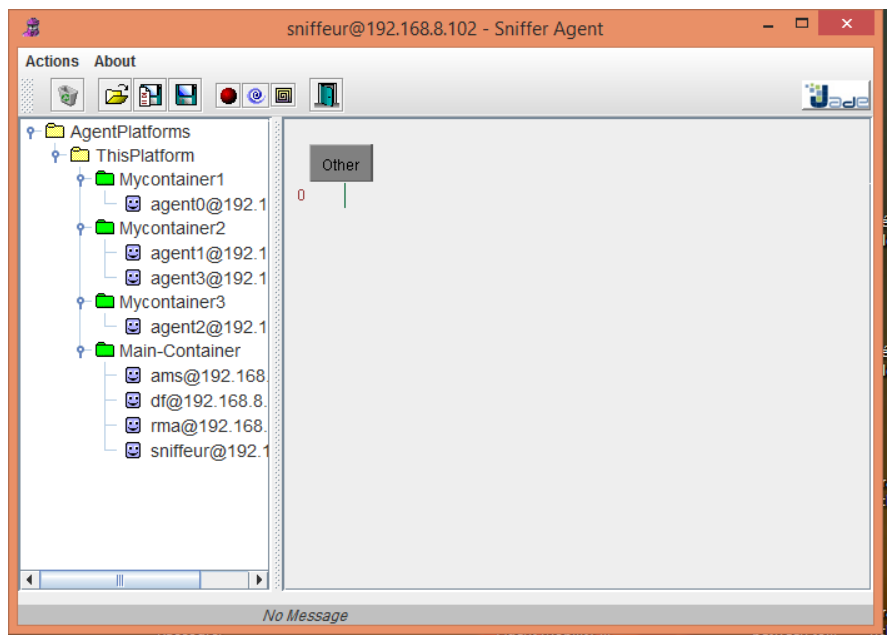
Dans cette partie nous présentons les résultats obtenus dans notre application

### 6.1 Interface de monitoring rma et sniffeur avant la migration d'un agent

Les deux interfaces rma et sniffeur sont des interfaces de surveillance de la plateforme et la communication, voici les deux interfaces avant la réalisation de la migration d'un agent.



**Figure 3.4** Interface rma (remote monitoring agent)



**Figure 3.5** Interface sniffeur

## 6.2 Interface de monitoring rma et sniffeur après la migration d'un agent

Les deux interfaces rma et sniffeur après la réalisation de la migration d'un agent entre les différents conteneurs.

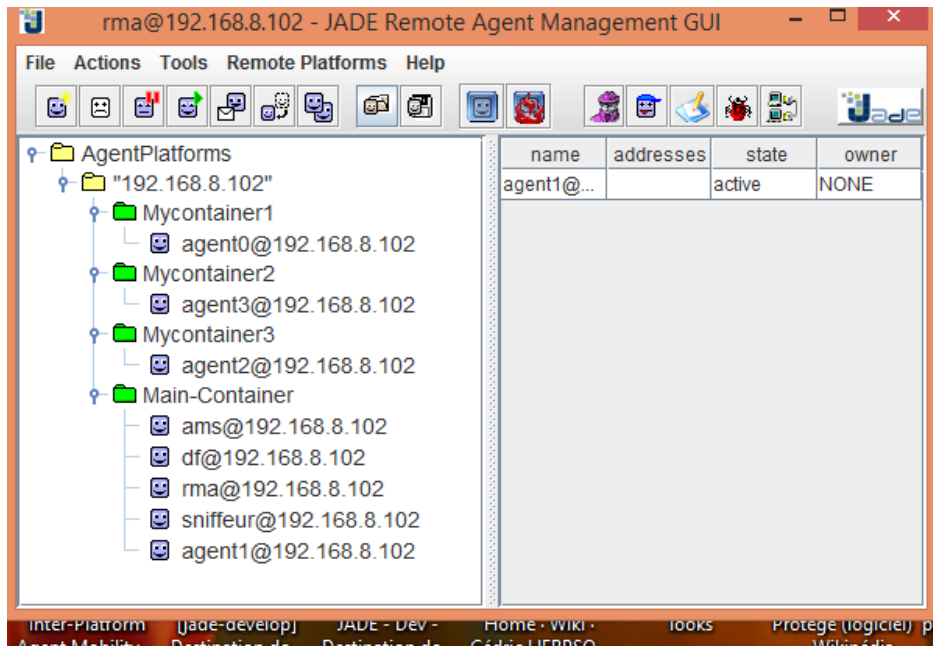


Figure 3.6 Interface rma (remote monitoring agent)

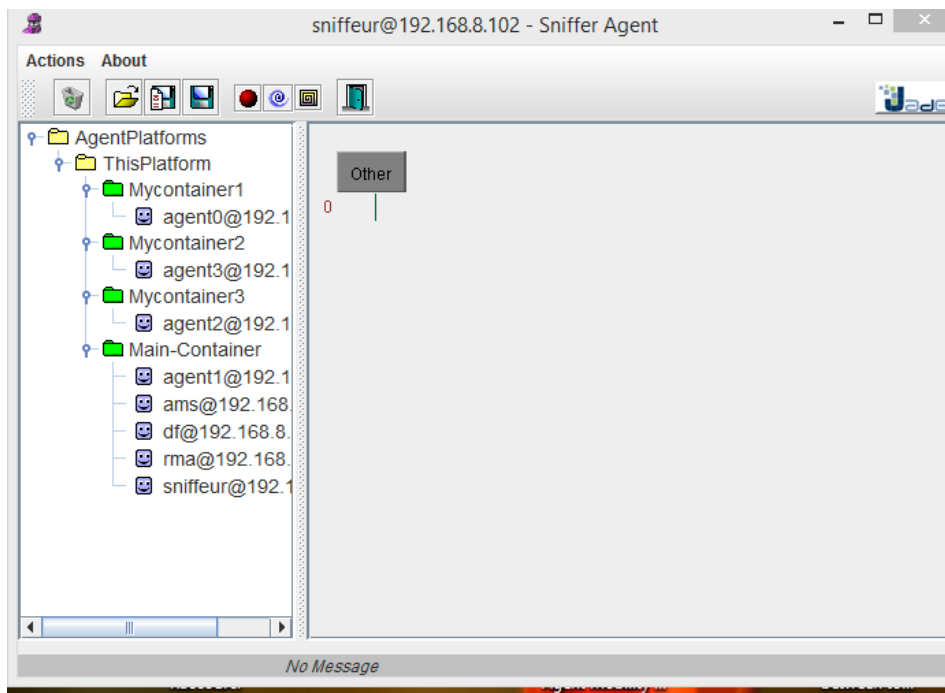


Figure 3.7 Interface sniffeur

### 6.3 Interface de l'agent de recherche

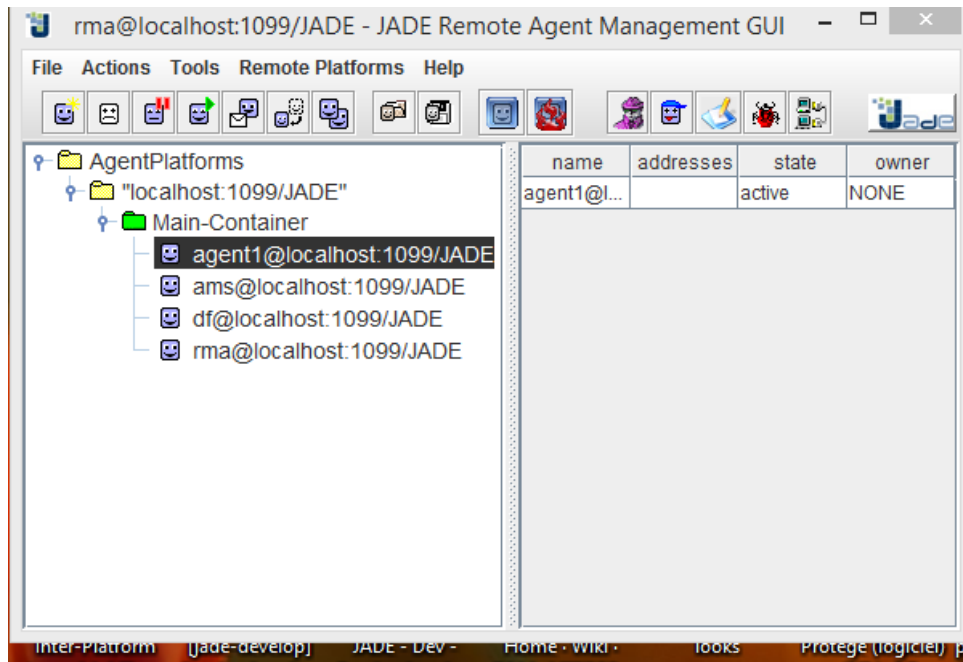


Figure 3.8 Interface de l'agent de recherche

## 6.4 Interface de l'ontologie

C'est l'interface qui utiliser pour faire la recherche

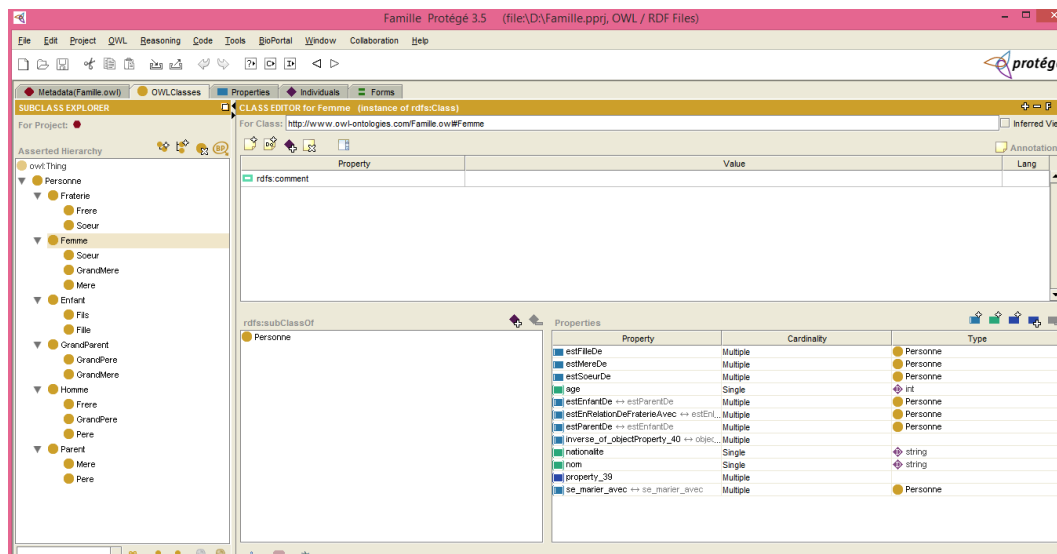


Figure 3.9 Interface de l'ontologie

## 6.5 Interface de recherche

L'interface principale de recherche qui permet:

- Le Lancement de recherche (demande d'informations) selon le choix de l'utilisateur.

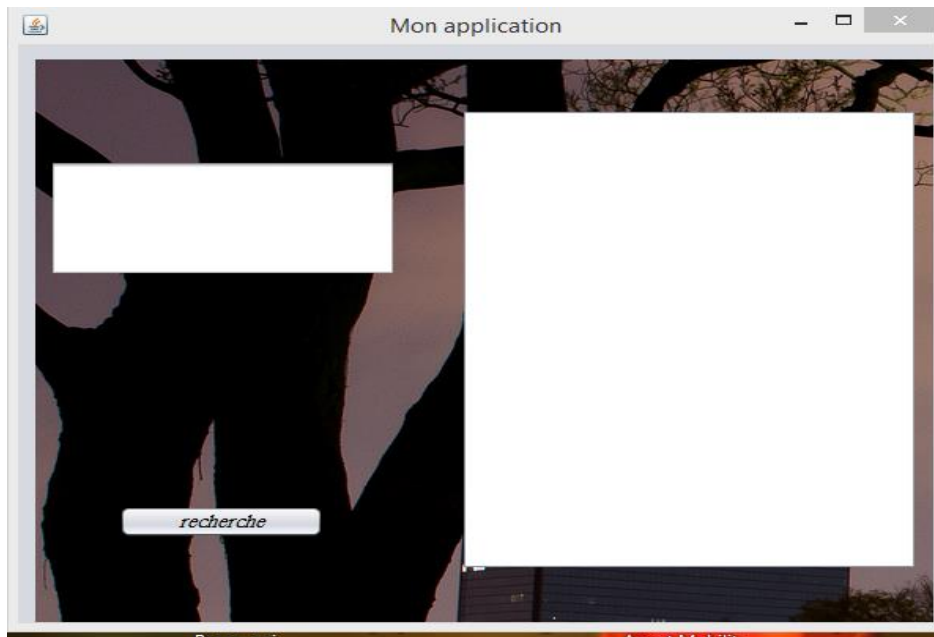


Figure 3.10 Interface de recherche

### 6.6 Interface de résultat

C'est l'interface qui permet d'afficher le résultat de recherche.

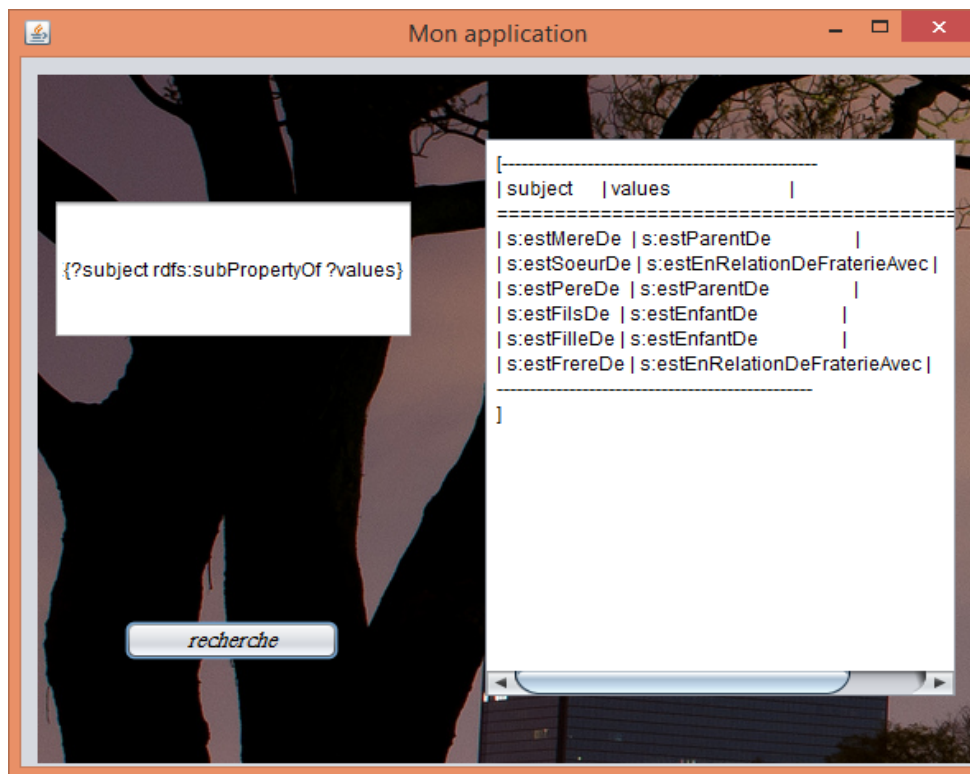


Figure 3.11 Interface de résultat

## **7. Conclusion**

Dans ce chapitre, nous avons présenté la démarche adoptée pour la conception d'un système de recherche sémantique d'informations à base d'agent mobile en décrivant l'architecture de cette démarche. Nous avons montré le rôle de l'agent dans notre système et comment les tâches sont réalisées par cet agent. Les résultats obtenus sont montrés ainsi que les outils d'implémentation utilisés.

## CONCLUSION GENERALE

La mobilité est une caractéristique inhérente de l'informatique moderne, lorsqu'il s'agit de La recherche sémantique d'information.

Dans ce mémoire, nous nous sommes intéressés par l'utilisation des agents mobiles pour la recherche sémantique d'information.

Tout au long de ce mémoire, nous avons présenté les technologies nécessaires pour la réalisation de notre travail. Dans un premier temps, nous avons présenté le terme de la mobilité d'un agent, c'est-à-dire comment réaliser la mobilité d'un agent, ensuite comment accomplir la recherche sémantique d'information par un agent mobile en accédant à une ontologie qui joue un rôle très important pour faire cette recherche.

Dans ce travail nous avons utilisé une ontologie famille comme une entité nécessaire pour la réalisation d'une recherche sémantique. Une ontologie est une base de connaissances qui permet de faire une meilleure recherche, et aide l'utilisateur à satisfaire ses exigences. Nous avons implémenté aussi la mobilité qui permet à un agent de se déplacer entre les conteneurs dans une plateforme JADE afin de réaliser la recherche du sens d'un terme saisi par l'utilisateur de notre système. En fin si le terme est trouvé une explication détaillée sera affichée à l'utilisateur.

Dans ce travail, nous avons rencontré deux problématiques :

- 1- La migration d'un agent entre les machines avec la plateforme JADE, c'est-à-dire la migration inter-plateforme qui permet à un agent de migrer entre les différents plateformes c'est-à-dire entre les différents machines, alors c'était impossible de la réaliser malgré les efforts que nous avons fais car cette plateforme ne supporte pas ce type de migration.
- 2- La recherche sémantique d'information est une tâche difficile à réaliser, alors nous avons réalisé une recherche simple parce que le temps n'était pas suffisant pour arriver à notre recherche sémantique.

Ce mémoire constitue une base de travail à partir du quel, des nouvelles activités de recherche peuvent être lancées afin d'améliorer le travail présenté. Les perspectives que nous proposons peuvent donc s'orienter vers les directions suivantes :

- La réalisation de la migration inter-plateforme avec la plateforme JADE, c'est-à-dire la migration entre les différentes plateformes (les différentes machines).

- La réalisation de la recherche sémantique d'information, c'est-à-dire appliquer un bon algorithme de recherche dans une ontologie.

## BIBIOGRAPHIE

- [1] Peter Braun and Wilhelm Rossak, “ mobile agents basic concepts, mobility models, and the tracy toolkit, ” livre, Morgan Kaufmann publishers is an imprint of Elsevier, 500 Sansome Street, Suite 400, San Francisco, CA 94111, édition 2005.
- [2] M.Baker, Rajkumar Buyya, and Domenico Laforenza. «Grids and grid technologies for wide-area distributed computing». *International Journal of Software : Practice and Experience (SPE)*, 32(15) :1437– 1466, December 2002.
- [3] Bettahar Aoued, les aglets d’IBM, université de Montréal.
- [4] Gomez P.A., 1999. Développements récents en matière de conception, de maintenance et d’utilisation d’ontologies. 3èmes rencontres Terminologie et intelligence artificielle TIA. 19 :9-20.
- [5] J. Gomoluch and M. Schroeder, *Information agents on the move: A survey on loadbalancing with mobile agents*, 2001.
- [6] Gruber T., 1993. A translation approach to portable ontology specifications. *Knowledge Acquisition Journal*. 5:199 -220.
- [7] Guarino N., et Giaretta P., 1995. Ontologies and Knowledge Bases: Towards a Terminological Clarification. In *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*, Mars N. J. I., Amsterdam: IOS Press. pp. 25–32.
- [8] C.Hanachi, C.Sibertin-blanc, *introduction aux systèmes multi agents*, université Toulouse i & irit.
- [9] Maedche A., 2002. *Ontology Learning for the Semantic Web*. Boston: Kluwer Academic Publishers.
- [10] P Marques, Paulo Simões, Luís M. Silva, Fernando Boavida, and João G. Silva. «Providing applications with mobile agent technology». In *OpenArch’01 - Fourth IEEE International Conference on Open Architectures and Network Programming*, Anchorage, Alaska, April 2001.
- [11] Mizoguchi R., Kozaki K., Sano T. et Kitamura Y. 2000. Construction and Deployment of a Plant Ontology. *The 12th International Conference, EKAW2000*, Juan-les-Pins, France, pp. 113-128.
- [12] Yassine Mrabet 1,2 — Nacéra Bennacer 2 — Nathalie Pernelle 1 — Mouhamadou Thiam 1,21 LRI, *Une approche pour la recherche sémantique de l’information dans les documents semi-structurés hétérogènes*, Université Paris-Sud 11, INRIA Saclay F-91893

Orsay Cedex, France 2 SUPELEC Sciences des Systèmes (E3S) Département Informatique 91192 Gif-sur-Yvette Cedex, France.

[13] Neeches R., Fikes R. E., Finin T., Gruber T. R., Senator T. et Swartout W. R., 1991. Enabling technology for knowledge sharing. *AI Magazine*. 12: 36- 56.

[14] Studer R., Benjamins V. R. et Fensel D., 1998. Knowledge engineering: Principles and Methods. *Data Knowledge Engineering*, 25: 161-197.

[15] Swartout B., Patil R., Knight K. et Russ T., 1997. Towards Distributed Use of Large Scale Ontologies. *Spring Symposium Series on Ontological Engineering*, Stanford University, CA. pp. 138-138.

[16] Haïfa Zargayouna, Catherine, Roussey, Jean-Pierre Chevallet, Recherche d'information sémantique: état des lieux, LIPN (CNRS UMR 7030 - Université Paris 13 Sorbonne Paris Cité), Villetaneuse, France haifa.zargayouna@lipn.univ-paris13.fr Irstea UR TSCF, 9 avenue Blaise Pascal, CS 20085, Aubière, France catherine.roussey@irstea.fr, LIG (CNRS UMR 5217 - Université de Grenoble Alpes), Grenoble, France.

[17] M Baziz, (2005). Indexation conceptuelle guidée par les ontologies pour la recherche d'information. Thèse de doctorat, Université Paul Sabatier, Toulouse.

[18] Borst W. N., 1997. Construction of Engineering Ontologies. Thèse de doctorat, Center for Telematica and Information Technology, Université de Tweent, Enschede.

[19] Mr. Abdelkrim bouramoul, recherche d'information contextuelle et sémantique sur le web, thèse Pour obtenir le grade de Docteur en Sciences Spécialité: informatique Présentée et soutenue publiquement par Le: 25/ 09/ 2011, Université mentouri de Constantine.

[20] Mr bouzaher Abdelaziz, Approche agent mobile pour l'adaptation des réseaux mobiles ad hoc, Mémoire Présenté en vue de l'obtention du diplôme de Magister en Informatique. Université Mohamed Khider Biskra.

[21] Christophe Cubat Dit Cros « Agents Mobiles Coopérants pour les Environnements Dynamiques », thèse de doctorat, Institut National Polytechnique de Toulouse ,2 décembre 2005.

[22] Salah el falou, Programmation répartie, optimisation par agent mobile, thèse doctorat de l'université de caen 07 (août 2006).

[23] Hamida Souraya, Une approche basée agent mobile pour le m-service web sémantique, Thèse Présentée en vue de l'obtention du diplôme de Docteur en 3ième cycle LMD en Informatique, Université Mohamed Khider – Biskra, juin 2014.

- [24] Hernandez, N. (2006). Ontologies de domaine pour la modélisation du contexte en recherche d'informations. Thèse de Doctorat, Université Paul Sabatier, Toulouse.
- [25] Kolli Manel, Intégration d'ontologies dans le cadre du web sémantique : une détection des relations sémantiques basée sur le RÀPC, thèse Présentée pour obtenir le grade de doctorat en sciences informatiques, Université Mentouri Constantine.
- [26] Mr Ali Sriti, la recherche d'information sémantique d'informations dans le cadre du web sémantique, Mémoire Présenté en vue de l'obtention du diplôme de Magistère en informatique, Université Mohamed Khider – Biskra, 25 Juin 2012.

**ملخص:** البحث الدلالي لمعلومة عبارة عن فرع اساسي في البحث المعلوماتي الذي يهدف الى دراسة معاني المصطلحات في الوثائق من اجل فهم محتوى الوثيقة و اكتشاف معنى المعلومة بغرض تسهيل الوصول للمعلومة و تحقيق احتياجات المستعمل. في اطار هذا العمل نستعمل مقاربة تتركز على تكنولوجيا عون متنقل تسمح بالمساعدة على البحث الدلالي للمعلومة.

**الكلمات المفتاحية:** البحث الدلالي لمعلومة, عون متنقل

**Abstract :** The semantic information research is main branch of information research which aims to study the meaning of terms in the documents to understand document content and discover the meaning of information the purpose of which is to facilitate access to information to realize the needs of a user. Within the framework of this work we use an approach based on mobile agents technology to assist in the semantic research of information.

**Key words :** Semantic information research, mobile agents, Ontology.

**Résumé :** La recherche sémantique d'information (RSI) est une branche principal de la recherche d'information (RI) qui vise à étudier les significations des termes dans les documents pour comprendre le contenu de document et découvrir le sens de l'information dont le but est de faciliter l'accès à l'information pour réaliser les besoins d'un utilisateur. Dans le cadre de ce travail nous utilisons une approche basée sur la technologie d'agents mobiles permettant d'aider à la RSI.

**Mots clés :** la recherche sémantique d'information(RSI), les agents mobiles, Ontologie.