

1. Introduction

Dans ce chapitre nous allons présenter, dans un premier temps, l'environnement de développement JADE ainsi que l'environnement matériel et logiciel utilisé dans toutes les phases de réalisation. Un use case sera présenté également à la fin de ce chapitre.

2. Environnement de développement

Dans la réalisation de ce travail, nous avons employé beaucoup des techniques (langages, outils, environnements, APIs¹, etc.) à savoir : java, JADE, MYSQL, etc... Dans les sections suivantes, nous allons présenter ces différentes techniques, en les organisant selon leurs catégories.

2.1. La plate-forme JADE

JADE (Java Agent Development Framework) est une plate-forme multi-agents développée en Java par CSELT (Groupe de recherche de Gruppo Telecom, Italie), il fournit un environnement de développement et d'exécution des systèmes multi-agents compatibles avec les standards FIPA [JADE, 2002]. JADE comprend deux composantes de base : la plate-forme d'agents compatible FIPA et un package logiciel pour le développement des agents dans Java. [13]

JADE fournit les facilités suivantes : Une plate-forme agent distribuée : la plateforme peut être distribuée (partagée) entre plusieurs hôtes connectées via RMI de Java, de telle façon qu'une seule application Java, par conséquent une seule "Machine Virtuelle Java" est exécutée sur chaque hôte.

- ✓ Une interface utilisateur graphique (GUI): l'interface GUI assure un traitement plus commode de la plate-forme, elle permet à l'utilisateur d'exécuter plusieurs ordres tel que créer un nouvel agent dans la même plate-forme, cloner l'agent, le déplacer, le suspendre, le tuer, etc....
- ✓ Un support d'exécution : pour les activités multiples, parallèles et concurrentes des agents via le modèle du comportement (behaviour).
- ✓ Un transport efficace des messages ACL : à l'intérieur de la même plate-forme.
- ✓ Une bibliothèque de protocoles : compatibles aux standards FIPA et prêts à être employés pour gérer l'interaction inter-agent.

2.2. FIPA (Foundation for Intelligent Physical Agents)

FIPA est une association internationale sans but lucratif, travaillant pour produire des spécifications pour des technologies d'agent générique.

Le premier document de la FIPA, et le FIPA97, établissant les règles normatives qui permettent à une société d'agents d'inter opérer. Les documents FIPA décrivent le modèle de référence d'une plate-forme multi-agents ou ils identifient les rôles de quelques agents clés nécessaires pour la gestion de la plateforme, et spécifient le contenu du langage de gestion des agents et l'ontologie du langage. Trois rôles (agent) principaux sont identifiés dans une plate-forme d'agent : [19]

2.2.1. Le Système de gestion d'Agent (AMS- Agent Management System)

Agent qui exerce le contrôle de supervision sur l'accès et l'usage de la plateforme, il est responsable d'authentifier les agents résidents et de contrôler là les enregistrements.

2.2.2. Le Canal De communication (ACC- Agent Communication Canal)

Agent qui fournit le chemin pour les interactions de base entre les agents dans et en d'hors de la plateforme, c'est la méthode de communication implicite qui offre un service fiable et précis pour le routage des messages, il (l'agent) doit aussi être compatible avec le protocole IIOP pour assurer l'interopérabilité entre les différentes plateformes.

2.2.3. Le facilitateur d'Annuaire (DF- Directory Facilitator) [19]

Agent qui fournit un service de pages jaunes à la plateforme. (Voir **Figure 4.1**)

FIPA spécifie aussi le Langage de Communication d'agents (ACL- Agent Communication Language). La communication entre agents ne se fait que par envoi de messages seulement. 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 pourraient s'exécuter sur des plateformes différentes et utilisent des technologies différentes interconnexion, FIPA spécifie que les messages transportés entre la différente plateforme devraient être codé sous forme textuelle.

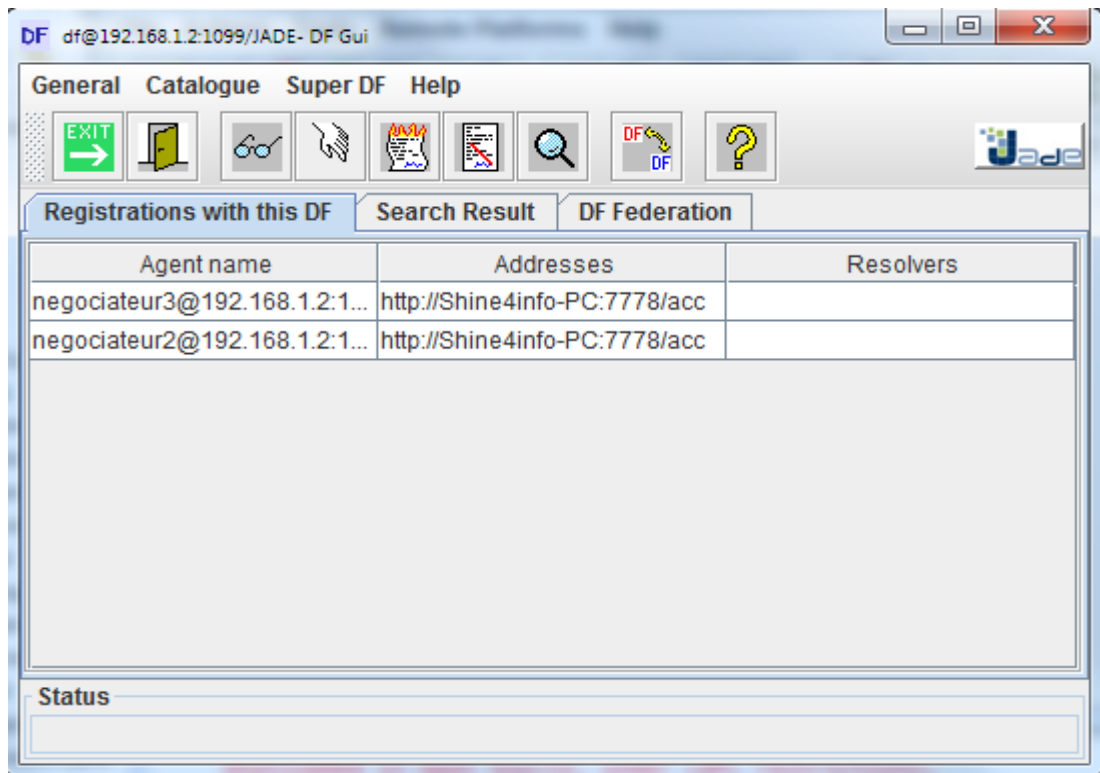


Figure 4.1 :Le facilitateur d'Annuaire (DF)

2.3. Eclipse

Eclipse est un IDE (Integrated Development Environment) c'est-à-dire un logiciel qui simplifie la programmation en proposant un certain nombre de raccourcis et d'aide à la programmation. Il est développé par IBM, est gratuit et disponible pour la plupart des systèmes d'exploitation. Il est d'abord conçu pour le langage Java mais ses nombreux greffons en font un environnement de développement pour de nombreux autres langages de programmation (C/C++, Python, PHP...etc).

2.4. Serveur de bases de données MySQL

MySQL est un système de gestion de bases de données (SGBD). Selon le type d'application, sa licence est libre ou propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels

MySQL est un serveur de bases de données relationnelles SQL développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le

service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées. Il est multi-thread et multi-utilisateur.

3. Les technologies adoptées

3.1. JAVA

Java est un langage de programmation orienté objet développé par *Sun Microsystems*. Les premières versions datent de 1995, il a réussi à intéresser et intriguer beaucoup de développeurs à travers le monde. [18] La syntaxe générale du langage java est très proche de celle du langage C, et parmi ces avantages :

- Java assure une totale indépendance des applications vis-à-vis de l'environnement d'exécution : c'est à dire que toute machine supportant Java est en mesure d'exécuter un programme sans aucune adaptation (ni recompilation, ni paramétrage de variables d'environnement).
- Java est un langage orienté objet, c'est à dire que nous n'allons pas manipuler des fonctions et des procédures mais des objets qui vont s'échanger des messages. Le principal avantage est que l'on peut réaliser une programmation modulaire : tous les objets peuvent être mis au point séparément.
- Java nous permet un accès simplifié aux bases de données, soit à travers la passerelle JDBC-ODBC ou à travers un pilote JDBC spécifique au SGBD.
- Il est particulièrement adapté au développement d'application communiquant à l'intermédiaire d'un réseau. L'API java est très riche : différents packages permettant d'accéder aux réseaux, aux entrées/sorties et aux différents composants graphiques.

Quelques notions de base JAVA :

- **JDK**

Le Java Development Kit (JDK) désigne un ensemble de bibliothèques logicielles de base du langage de programmation Java, ainsi que les outils avec lesquels le code Java peut être compilé, transformé en byte code destiné à la machine virtuelle Java

Il existe plusieurs éditions de JDK, selon la plate-forme Java considérée (et bien évidemment la version de Java ciblée) :

- JSE pour la Java 2 Standard Edition également désignée J2SE.
- JEE, sigle de Java Enterprise Edition également désignée J2EE.

- JME 'Micro Edition', destinée au marché mobiles.
- etc.

À chacune de ces plateformes correspond une base commune de Development Kits, plus des bibliothèques additionnelles spécifiques selon la plate-forme Java que le JDK cible, mais le terme de JDK est appliqué indistinctement à n'importe laquelle de ces plates-formes.

Le Kit de développement comprend plusieurs outils, parmi lesquels :

- *javac*: le compilateur Java
- *java*: un interpréteur d'applications (machine virtuelle)
- *applet viewer*: un interpréteur d'applets
- *jdb*: un débogueur
- *javap*: un décompilateur, pour revenir du *bytecode* au code source
- *javadoc*: un générateur de documentation
- *jar* : un compresseur de classes Java

▪ API

Cela signifie « Application Programming Interface ». Une API contient un ensemble de fonctions qui facilitent la programmation.

L'API de java contient paquetages standard : `java.awt`, `java.io`, `java.lang`, `java.net` et `java.util`.

▪ JVM

La machine virtuelle Java ou JVM (Java Virtual Machine) est un environnement d'exécution pour applications Java. C'est un des éléments les plus importants de la plate-forme Java. Elle assure l'indépendance du matériel et du système d'exploitation lors de l'exécution des applications Java. Une application Java ne s'exécute pas directement dans le système d'exploitation mais dans une machine virtuelle qui s'exécute dans le système d'exploitation et propose une couche d'abstraction entre l'application Java et ce système [20].

3.2. XML

Extensible MarkupLanguage (langage extensible de balisage) est un langage informatique de balisage générique. Il sert essentiellement à stocker/transférer des données de type texte Unicode structurées en champs arborescents. Ce langage est qualifié d'extensible car il permet à l'utilisateur de définir les balises des éléments. L'utilisateur peut multiplier les espaces de nommage des balises et emprunter les définitions d'autres utilisateurs. Le World Wide Web Consortium (W3C), promoteur de standards favorisant l'échange d'informations sur Internet, recommande la syntaxe XML pour exprimer des langages de balisages spécifiques.

4. Etude de cas : Entreprise virtuelle de tourisme

Dans Cette section nous allons simuler la création d'une entreprise virtuelle dans le secteur de tourisme. Pour ce faire, notre application se comporte comme suit :

Lancement de l'environnement graphique de JADE (Voir **Figure 4.2**)

Cette interface comporte initialement, un container (Main-Container) qui inclus trois agents spécifiques (conforme aux spécifications FIPA) : Agent Management System (AMS), Directory Facilitator (DF), Agent Communication Channel (ACC) et les différents agents négociateurs, chacun dans un container.

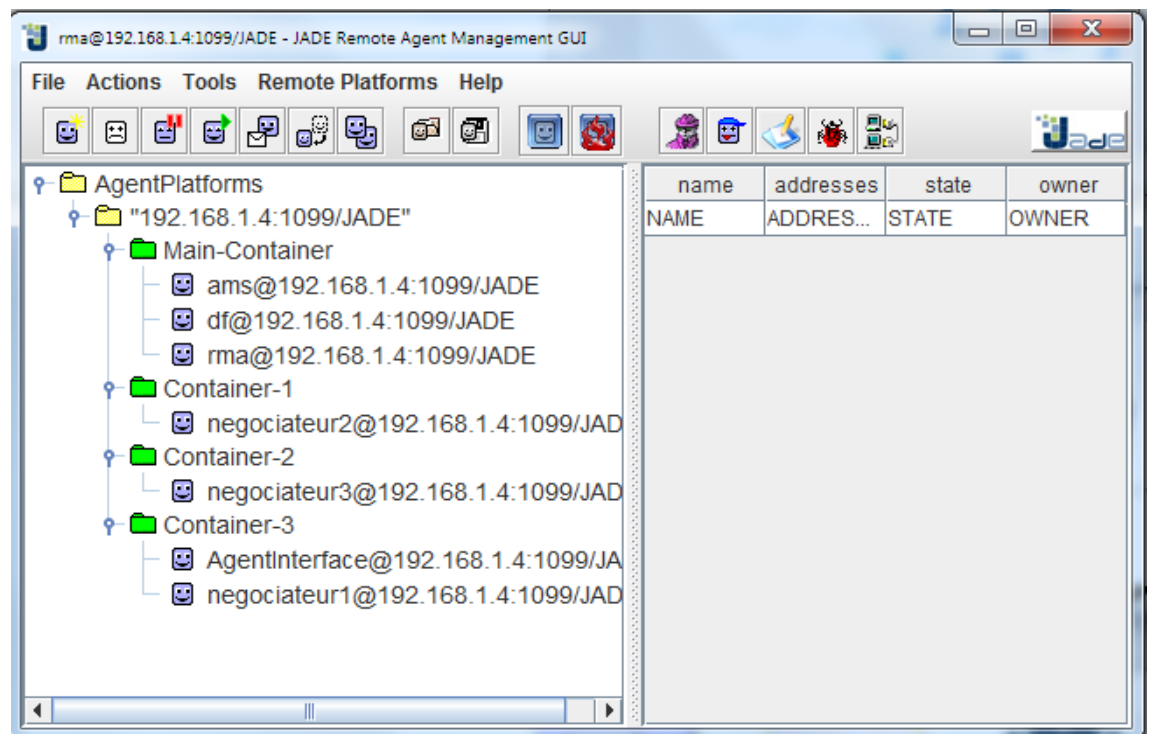


Figure 4.2 : Environnement simulé pour une entreprise virtuelle

JADE fournit des agents pour l'affichage, des messages transmises entre agents. On va utiliser cet agent (*sniffer*) pour représenter les messages échangés entre les agents négociateurs des différentes d'entreprises. (Voir **Figure 4.3**)

Les messages ACL échangés sont représentés sous la forme suivante :

L'agent émetteur et récepteur de message

Contenu de message envoyé

Protocole utilisé entre agents

Figure 4.3 :Message ACL dans JADE

Description de l'environnement de l'Entreprise virtuelle

L'Entreprise virtuelle est composée de plusieurs entreprises liées par des engagements afin de satisfaire les besoins de leurs clients. Nul de ces entreprises ne possède seul toutes les compétences nécessaires pour répondre aux services demandés par le client. Dans le cadre de cette étude, les entreprises individuelles sont regroupées pour créer une entreprise virtuelle consacrée aux services relatifs au tourisme.

Chaque entreprise est enregistrée dans un répertoire de services dans le Cloud, pour représenter ses services, Elle est représentée par un agent négociateur d'entreprise. L'exemple suivant représente une entreprise avec ses services fournis :

```

<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<repertoire>
  <!-- Entreprise1 -->
  <Entreprise ID="negociateur2">
    <Service>
      <nomService>HOTELS</nomService>
      <paye>ITALIE</paye>
      <qualite>HAUT</qualite>
      <activiteSup>PARK</activiteSup>
    </Service>
    <Service>
      <nomService>RESTAURENTS</nomService>
      <paye>TURQUE</paye>
      <qualite>MOYEN</qualite>
      <activiteSup>MUSEE</activiteSup>
    </Service>
    <Service>
      <nomService>RESTAURENTS</nomService>
      <paye>FRANCE</paye>
      <qualite>HAUT</qualite>
      <activiteSup>CONCERT</activiteSup>
    </Service>
  </Entreprise>

```

Figure 4.3: Services fournis par une entreprise

Après la détermination des besoins et des objectifs, l'entreprise lance une offre à l'aide de son agent Interface qui lui permet de créer le profil de l'offre (Voir **Figure 4.4**)

The screenshot shows a window titled 'AgentInterface'. The main area has a decorative header with silhouettes of various landmarks. Below this, there is a form with five fields: 'Type Service' (dropdown with 'HOTELS'), 'Qualité de Servi...' (dropdown with 'HAUT'), 'Pays' (dropdown with 'ITALIE'), 'Prix Max' (empty text field), and 'Activités supplémentaires' (dropdown with 'PARK'). At the bottom of the form, there are two buttons: 'Nouveau Contrat' and 'Ancien Contrat'.

Figure 4.4 : Agent Interface de système

Dans le cas de nouveau Contrat l'agent négociateur consulte le répertoire des services des entreprises dans le Cloud et focalise les entreprises qui fournissent le service demandé. Dans ce cas l'Agent négociateur lance un appel vers les autres négociateurs, une phase de négociation se lance (Voir **Figure 4.5**)



Figure 4.5 : Phase de négociation (Nouveau Contrat)

Dans le cas d'ancien Contrat l'agent négociateur consulte son répertoire des Contrats et focalise les entreprises prêtes à satisfaire son besoin afin de renouveler le contrat (Voir **Figure 4.6**)

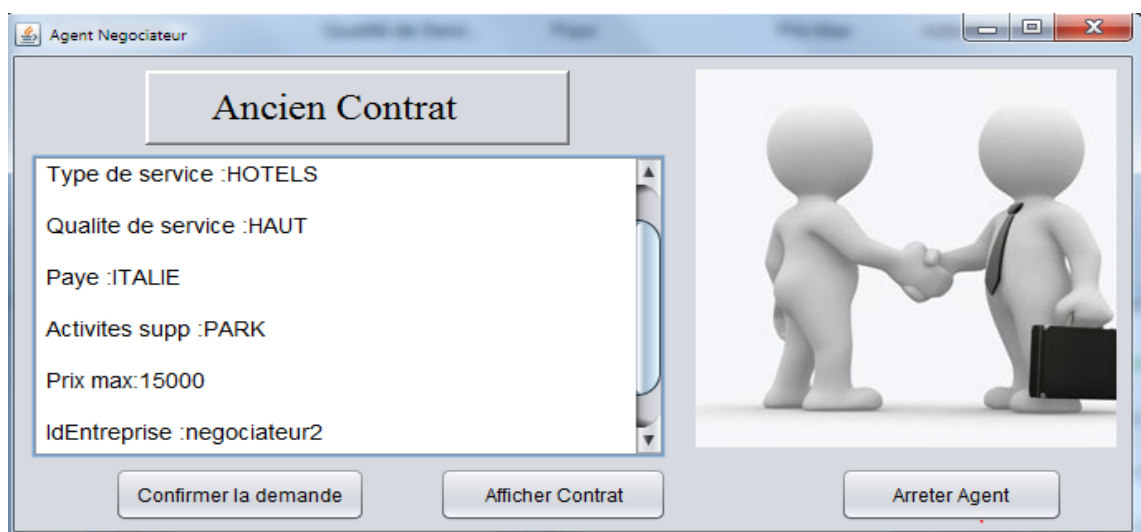


Figure 4.6 : Négociation (Ancien Contrat)

Une fois que les négociateurs se mettent en d'accord, un contrat de travail est instauré entre ces derniers en fixant les contraintes de services demandés (Voir **Figure4.7**).



The image shows a software window titled 'CONTRAT'. The top half of the window features a background image of a fountain pen writing on a document. Below this image, the word 'CONTRAT' is centered in a box. Underneath, there is a table with contract details. At the bottom, there are two buttons: 'Confirmer' and 'Annuler'.

| | |
|--------------------------|--------------|
| Type de Service | HOTELS |
| Qualite de Service | HAUT |
| Paye | ITALIE |
| Activite Supplémentaires | PARK |
| IdEntreprise | negociateur2 |
| Prix | 15000 |

Figure 4.7 : Contrat de travail

Une fois le contrat est établi l'entreprise virtuelle commence à fonctionner en respectant les cahiers de charges.

5. Conclusion

Dans ce chapitre, nous avons proposé une étude de cas comme application de notre architecture. Nous avons essayé de mettre en œuvre l'ensemble des idées qui caractérisent l'architecture proposée en se concentrant sur l'implémentation de l'interaction des agents.