

VIETNAM NATIONAL UNIVERSITY (VNU)
INSTITUT FRANCOPHONE INTERNATIONAL (IFI)



ĐẠI HỌC QUỐC GIA HÀ NỘI
VNU
Since 1906



INSTITUT
FRANCOPHONE
INTERNATIONAL

Option : Systèmes Intelligents et Multimédia (SIM)

Promotion : XXI

Systèmes Multi-Agent (SMA)

**« MODELISATION ET IMPLEMENTATION
D'UN SYSTEME DE COMPARAISON DES
PRIX : CAS DE LA LOCATION DES
CHAMBRES D'HOTEL »**

Auteur :

MEDOU Daniel Magloire

Encadrant :

Dr. NGUYEN Manh HUNG (PTIT)

année académique 2017-2018

TABLE DES MATIÈRES

1	Introduction	3
1.1	Objectifs	3
1.2	Mots clés	3
1.3	Plan de travail	3
2	Spécification des exigences	3
2.0.1	Diagramme des cas d'utilisations	4
2.0.2	Description des cas d'utilisations	5
3	Analyse du sujet	6
4	Conception	6
4.1	Conception architecturale	6
4.1.1	Modèle des agents	6
4.1.2	Modèle des services	7
4.1.3	Modèle d'acointances	8
4.2	Conception détaillée	9
4.2.1	Les modèles des classes	9
4.2.2	Les modèles de protocoles	11
4.2.3	Diagramme d'activité du système	13
5	Implémentation	14
5.1	Présentation de quelques plateforme SMA	14
5.2	Structure des données	15
5.3	Choix et justification des outils utilisés	15
5.4	Environnement de développement	15
5.5	Démarrage de l'application	16
5.6	Flux d'interaction des agents	16
5.6.1	Explications et interprétations des messages obtenus	17
6	Présentation de l'application	18
6.1	Agent Main	18
6.2	Agent Propriétaire	18
6.3	Agent Client	19
6.4	Agent Négociateur	20
7	Difficultés rencontrés	21
8	Conclusion	22
	Références	23

LISTE DES FIGURES

1	Diagramme des cas d'utilisations	5
2	Modèle des agents de notre application	7
3	Diagramme de classe client	10
4	Diagramme de classe propriétaire	10
5	Diagramme de classe système	10
6	Modèle de protocole du négociateur	11
7	Initiation du modèle négociateur	11
8	Réponse négociateur	12
9	Modèle de protocole du transaction	12
10	Initiation du modèle transaction	13
11	Transaction réponse	13
12	Diagramme d'activité du système	14
13	Flux de conversassions avec Sniffer Agent	17
14	Présentation de l'agent main	18
15	Présentation de l'agent propriétaire	19
16	Présentation de l'agent client	20
17	Présentation de l'agent négociateur	21

LISTE DES TABLEAUX

1	Description des cas utilisations	6
2	Liste des service sur le client	7
3	Liste des service sur le négociateur	8
4	Liste des service sur le propriétaire	8
5	Modèle d'accointances	8

1 Introduction

Parvenu au terme de l'unité d'enseignement intitulée Systèmes Multi Agents, il a été soumis à notre étude un sujet nous permettant de lier la partie théorique vue en cours à celle pratique lorsque nous réalisons ce projet intitulé : « Système de comparaison des prix : cas de la location des chambres d'hôtels ». Il nous a été demandé de façon générale de mettre en œuvre ce projet et de façon quantitative de faire une analyse du sujet, une conception et implémenter. Ce logiciel orienté agents nous donnera la possibilité de collecter les prix des chambres d'hôtels, les trier selon l'intervalle défini par le client et enfin de donner une décision. Le présent rapport de manière plus détaillée sera constituée des principales parties que sont les spécifications, une analyse, la conception, implémentation, les difficultés et en fin une conclusion.

1.1 Objectifs

L'objectif général de notre projet est de modéliser et d'implémenter un système de comparaison des prix de chambre d'hôtels. Ce travail de façon spécifique devra être subdivisé en quelques parties qui nous permettront de mieux comprendre les tenants et les aboutissants de la programmation orientée agent. Alors ces parties ne sont que les différents points développés dans ce rapport à savoir : les exigences, l'analyse, la conception et en fin l'implémentation.

1.2 Mots clés

- ❖ **Un agent** : est une entité informatique, placée dans un environnement, qui est capable d'exécuter des actions d'une façon flexible et autonome afin d'atteindre les objectifs établis lors de sa conception.
- ❖ **Un agent** : est une entité physique ou virtuelle capable d'agir (ou communiquer) sur son environnement de manière autonome et ce en fonction de perceptions et connaissances partielles.
- ❖ **DF « Director Facilitator »** : fournit un service de « pages jaunes » à la plate-forme ;
- ❖ **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.

1.3 Plan de travail

2 Spécification des exigences

C'est une étape primordiale du projet et le point de départ de tous projets informatiques. En effet, elle permet de déterminer les exigences fonctionnelles ou non fonctionnelles (norme ISO 9126) de l'application. En d'autres termes, c'est un compte rendu entre le Client et les Utilisateurs finaux avec l'équipe de développement.

Par définition, les Exigences Fonctionnelles décrivent ce que le système doit pouvoir faire en termes d'actions et d'attentes. Dans le cadre de notre application, elles sont décrites selon les différents agents ou acteurs.

En effet, construire un diagramme des cas d'utilisation est la meilleur manière d'exprimer les exigences d'une application.

2.0.1 Diagramme des cas d'utilisations

Le diagramme des cas d'utilisation permet de déterminer les exigences des utilisateurs pour la nouvelle application. En d'autres termes, les cas d'utilisation énumèrent toutes les fonctionnalités future de l'application. L'ensemble des cas d'utilisation spécifie la fonctionnalité complète du système. Notons cependant que le diagramme de cas d'utilisation montre un système uniquement au point de vue des acteurs [LEDANG 2001 ; AUDIBERT 2009]. Notre Diagramme des cas d'utilisation est représentée par la (FIGURE 1).

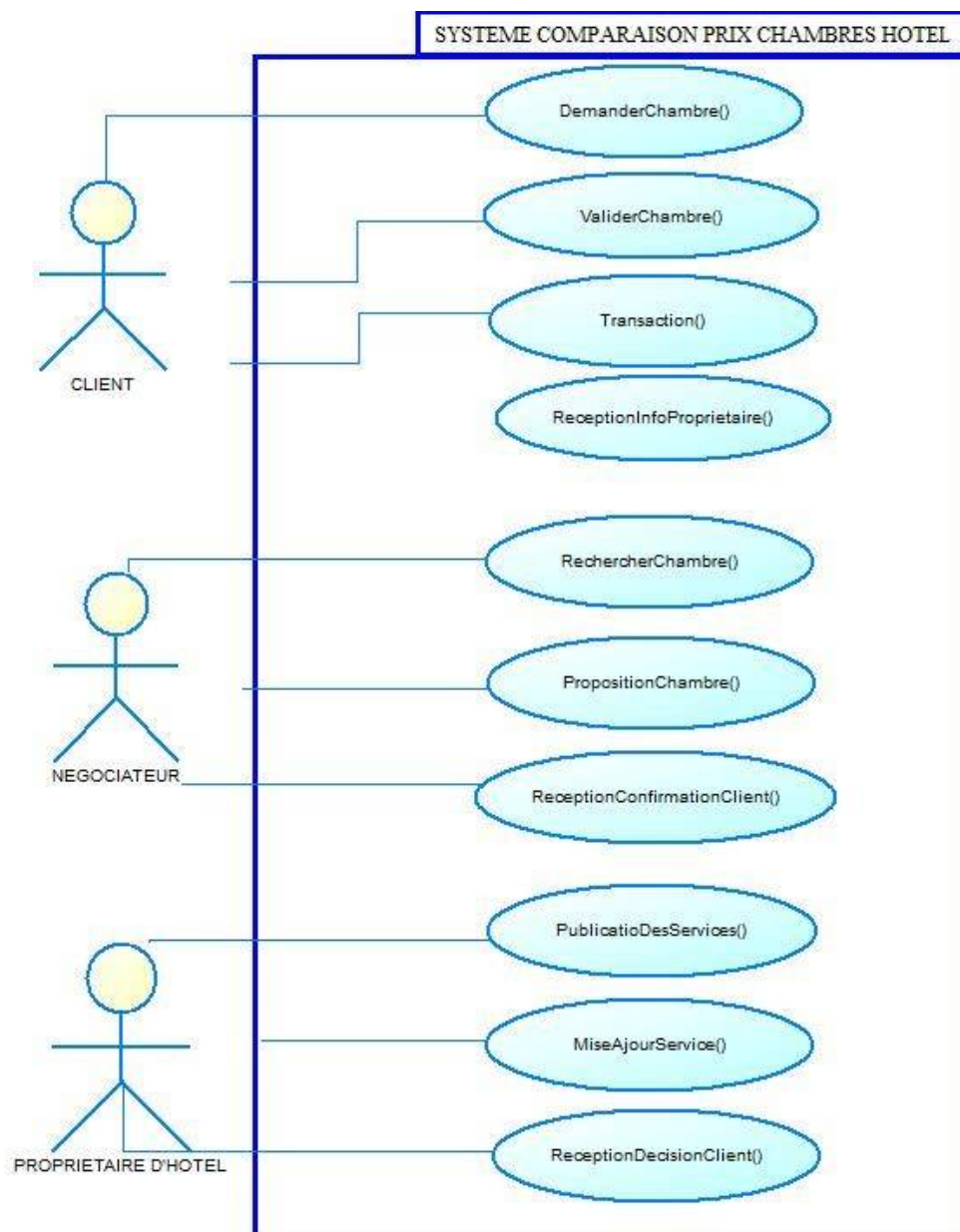


FIGURE 1 – Diagramme des cas d'utilisations

2.0.2 Description des cas d'utilisations

Le (TABLEAU 1) nous donne les descriptions de chaque cas de la (FIGURE 1) de notre projet.

Numéro	Acteur	Nomination	Description
Cas 1	Client	Demander prix d'une chambre d'hôtel	Demander le prix d'une chambre d'hôtel à un négociateur
Cas 2	Client	Valider une chambre d'hôtel	Valider une chambre d'hôtel proposer par un négociateur
Cas 3	Client	Réception information propriétaire	Réception des informations du propriétaire de la chambre d'hôtel
Cas 4	Client	Initialiser transaction	Initialiser la transaction avec le propriétaire de la chambre d'hôtel
Cas 5	Négociateur	Recherche chambre	Recherche une chambre d'hôtel à partir de la liste des propriétaires de chambre d'hôtels disponible
Cas 6	Négociateur	Proposer chambre	Proposer chambre des prix de chambre d'hôtels
Cas 7	Négociateur	Proposer chambre	Proposer chambre des prix de chambre d'hôtels
Cas 8	Négociateur	Récupérer liste propriétaire	Récupérer la liste des propriétaires disponibles
Cas 9	Propriétaire	Publier service	Publication du service dans l'annuaire
Cas 10	Propriétaire	Mise à jour des services	Modifier et supprimer un service
Cas 11	Propriétaire	Réception décision client	Réception décision du client par le négociateur (REJETER/VALIDER)

TABLE 1 – Description des cas utilisations

3 Analyse du sujet

L'analyse du sujet consiste à identifier la liste des agents avec ses fonctions principales et les interactions qu'ils effectuent avec les autres agents.

4 Conception

4.1 Conception architecturale

La conception architecturale quant à elle doit aboutir au raffinement des modèles de rôle et d'interaction par l'analyse des structures organisationnelles. Cette dernière sera répartie en trois (03) parties :

- ☞ Présentation des modèles des agents ;
- ☞ Présentation des modèles des services ;
- ☞ Présentation des modèles d'acointances.

4.1.1 Modèle des agents

Nous avons trois agents pour notre application à savoir :

- ☞ **Client** : donc le rôle joué est celui de demandeur (Nombre : 1 à n).
- ☞ **Négociateur** : le rôle est l'intermédiaire entre le client et le propriétaire (Nombre : 1).

- ☞ **Propriétaire** : (propriétaire de l'hôtel) : donc le rôle est celui de vendeur de chambre (Nombre : 1 à n).

Ces agents et leurs rôles cités ci-dessus sont illustrés par la (FIGURE 2).

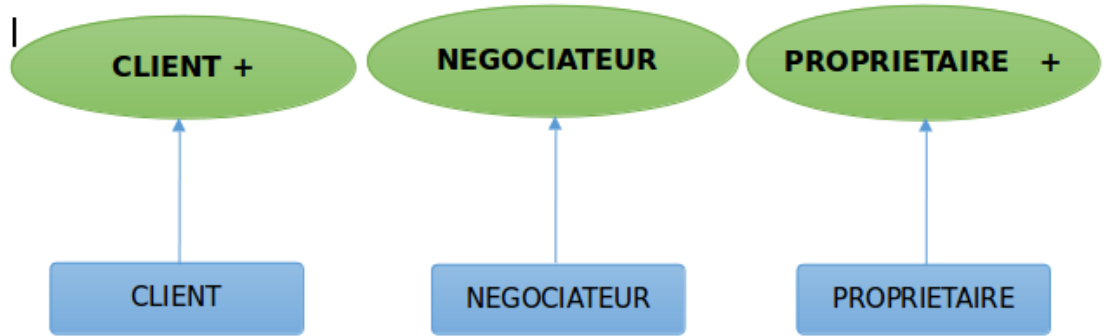


FIGURE 2 – Modèle des agents de notre application

4.1.2 Modèle des services

Nous distinguerons trois différents types de services à partir de notre application. Ceux-ci sont entre autre :

- ☞ Les services sur le client ;
- ☞ Les services sur le négociateur ;
- ☞ Les services sur le propriétaire.

Les services sur le client : (TABLE 2)

Service 1 : Demande de chambre	
Inputs	Caractéristiques chambres : Prix
Outputs	Liste des prix de chambre dans un intervalle
Pré-condition	Critère : Le prix doit être dans l'intervalle
Post-condition	Envoi de la requête a Négociateur
Service 2 : Récupération liste chambres	
Inputs	Liste des chambres disponible
Outputs	Réception liste prix
Pré-condition	Choix selon la liste reçue
Post-condition	Envoi décision
Service 3 : Récupération informations du propriétaire	
Inputs	Décision favorable client
Outputs	Réception informations sur le propriétaire
Pré-condition	Transaction à effectuer avec le propriétaire retenu
Post-condition	TransactionInit

TABLE 2 – Liste des service sur le client

Les services sur le négociateur : (TABLE 3)

Service 1 : Récupération de la liste des chambres disponible	
Inputs	Critère : Prix dans l'intervalle
Outputs	Listes des chambres selon intervalle prix
Pré-condition	Réception des critères du client
Post-condition	Répondre au client
Service 2 : Informer aux propriétaires de(s) offre(s)	
Inputs	REQUEST Client au Négociateur
Outputs	Réponse des propriétaires
Pré-condition	Existence du propriétaire (Enregistrement)
Post-condition	Envoi proposition aux propriétaires
Service 3 : Tri des prix	
Inputs	Les propositions des propriétaires
Outputs	Tri des prix selon requête du client
Pré-condition	Intervalle du prix
Post-condition	Exécution du tri
Service 4 :	
Inputs	Choix du client
Outputs	Confirmation choix du client
Pré-condition	CONFIRM Client
Post-condition	Envoyer décision aux propriétaires et information du propriétaire choisi au client

TABLE 3 – Liste des service sur le négociateur

Les services sur le propriétaire : (TABLE 4)

Service 1 : Proposition des prix des chambres disponible	
Inputs	Critère : Offre envoyée par le Négociation
Outputs	Réponse à l'offre (Proposition des prix)
Pré-condition	Existence propriétaire et de l'offre
Post-condition	Envoi proposition

TABLE 4 – Liste des service sur le propriétaire

4.1.3 Modèle d'acointances

Compte tenu des modèles des agents et leur interaction, nous avons le modèle d'acointances (TABLE 5).

Agent	Client	Négociateur	Propriétaire
Client	-	Interaction3	-
Négociateur	Interaction1	-	Interaction5
Propriétaire	Interaction2	Interaction4	-

TABLE 5 – Modèle d'acointances

☞ Interaction1

- ❶ Initialiser transaction

- ☞ Interaction2
 - ❶ Demande de prix de chambre
 - ❷ Confirmer choix
- ☞ Interaction3
 - ❶ Donner liste chambre
 - ❷ Donner informations propriétaire
- ☞ Interaction4
 - ❶ Demande de prix de chambre
 - ❷ Confirmer choix client
- ☞ Interaction5
 - ❶ Donner de prix de chambre

L'interaction entre les clients est mise pour permettre aux clients de communiquer entre eux et de savoir à combien un autre client donné avait payé pour une chambre donnée du même hôtel.

4.2 Conception détaillée

Dans cette partie, il sera question pour nous de présenter deux parties indispensables à notre projet à savoir :

- ☞ Les modèles des classes ;
- ☞ Les modèles de protocoles.

4.2.1 Les modèles des classes

Le diagramme de classes constitue la structure interne. En d'autres termes, une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation. Cependant, Il s'agit d'une vue statique, car les facteurs temporels ne sont pas pris en compte.

Nous allons présenter les différentes classes de notre application

- ☞ **Classe des agents** : Client, Propriétaire, Négociateur.
- ☞ **Classe des comportements** : NegotiationInit, NegotiationResponse, TransactionInit, TransactionResponse, CommunicationInit, CommunicationResponse, ConsultationInit, ConsultationResponse.
- ☞ **Classes utilisées de JADE** : DFService, Behaviour, CyclicBehaviour, ACLMessage.

Diagramme de classe client : (FIGURE 3).

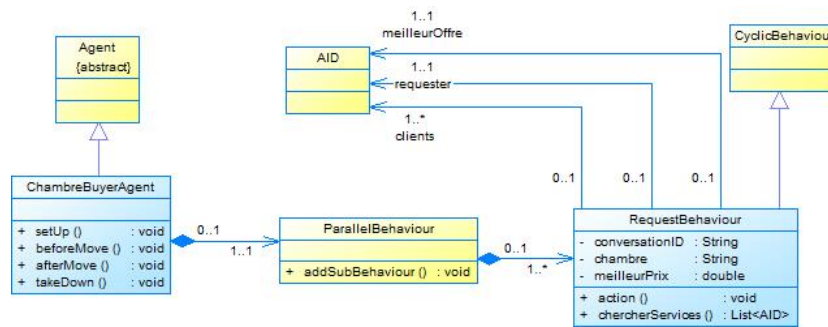


FIGURE 3 – Diagramme de classe client

Diagramme de classe propriétaire : (FIGURE 4).

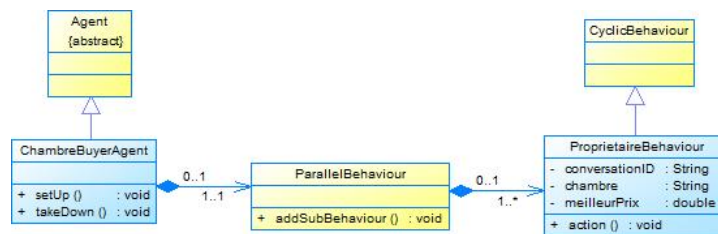


FIGURE 4 – Diagramme de classe propriétaire

Diagramme de classe système : (FIGURE 5).

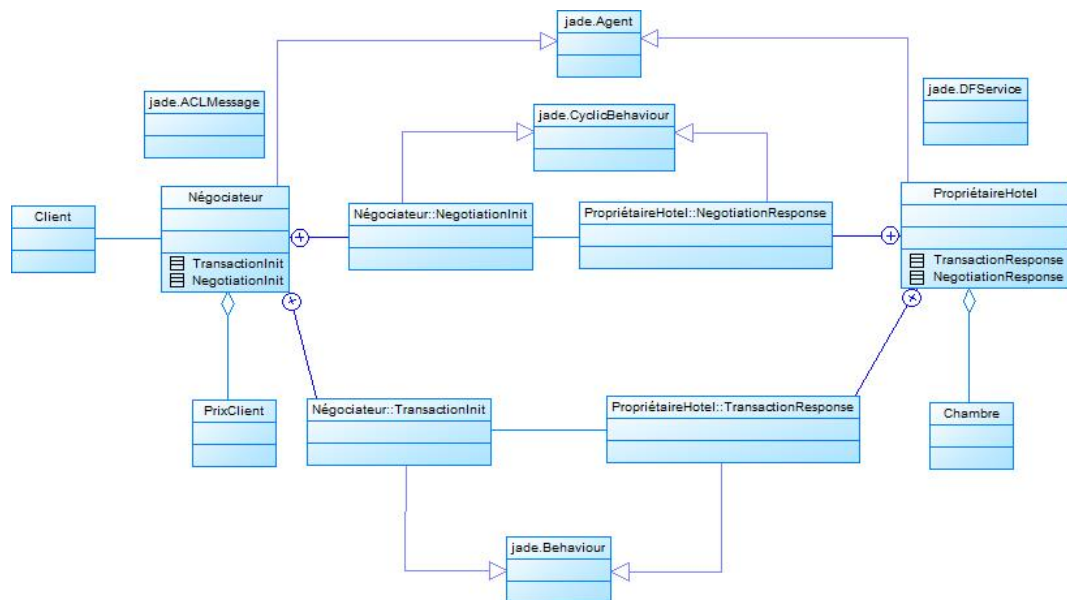


FIGURE 5 – Diagramme de classe système

4.2.2 Les modèles de protocoles

Dans cette dernière, nous allons présenter quelques descriptions des modèles de protocoles de notre application et celui du système en général.

- ☞ **Modèle de protocole : Négociation ;**
- ☞ **Modèle de protocole : Transaction.**

Modèle de protocole : « Négociateur » (FIGURE 6).

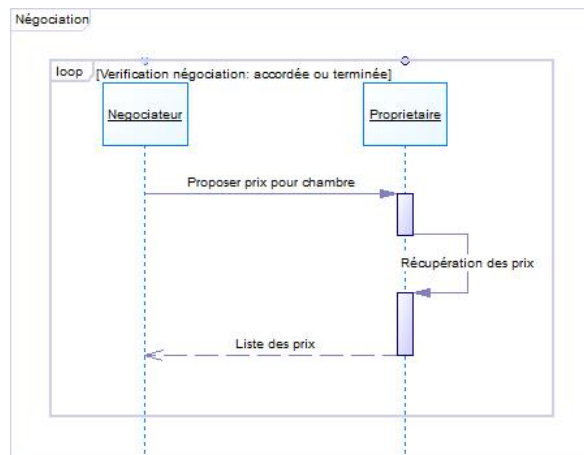


FIGURE 6 – Modèle de protocole du négociateur

Initiation du modèle négociation : (FIGURE 7).

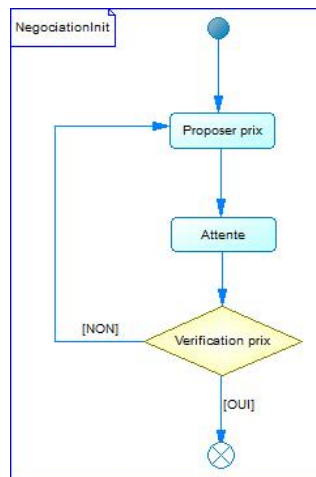


FIGURE 7 – Initiation du modèle négociateur

Réponse du négociateur : (FIGURE 8).

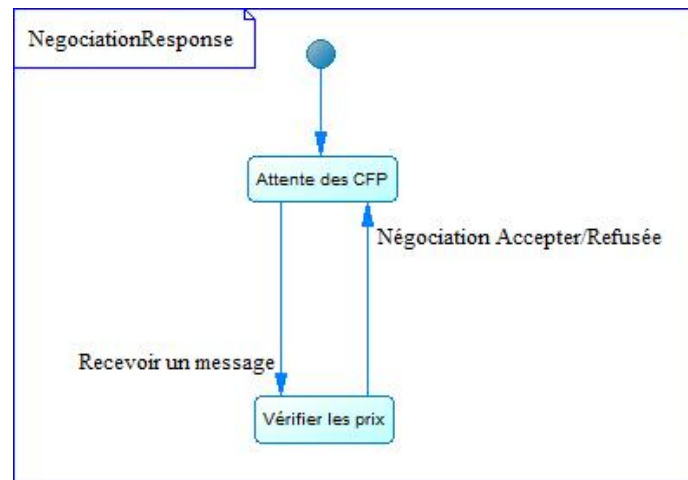


FIGURE 8 – Réponse négociateur

Modèle de protocole : « Transaction » (FIGURE 9).

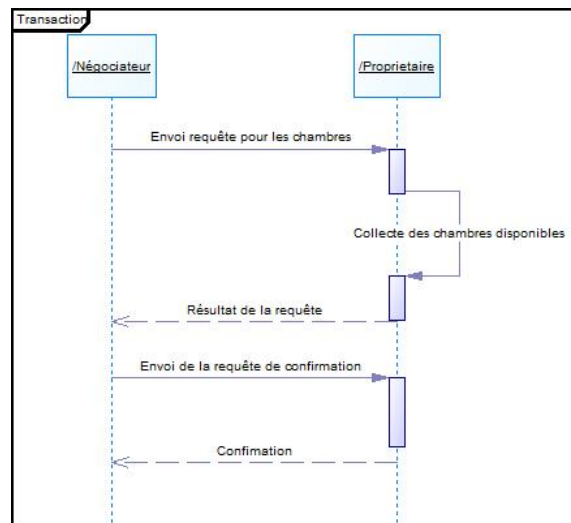


FIGURE 9 – Modèle de protocole du transaction

Initiation du modèle transaction : (FIGURE 10).

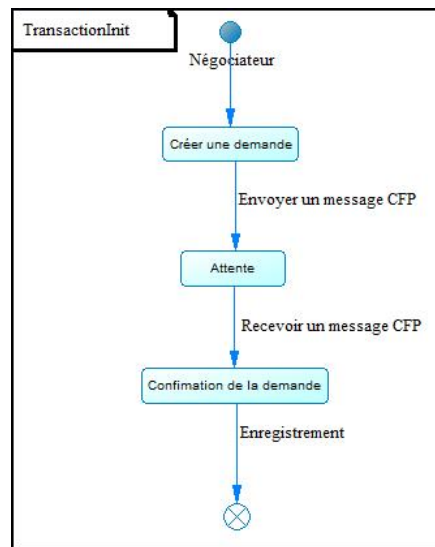


FIGURE 10 – Initiation du modèle transaction

Transaction réponse : (FIGURE 11).

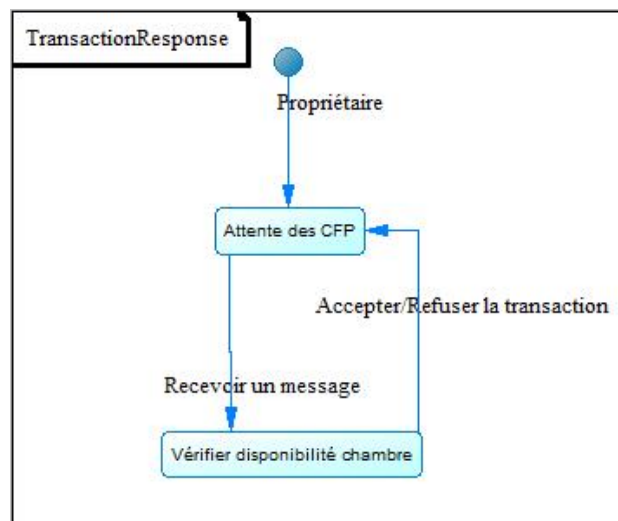


FIGURE 11 – Transaction réponse

4.2.3 Diagramme d'activité du système

Les diagrammes d'activité vont nous permettre d'identifier avec le plus de détails possible les comportements de chaque agent pendant l'exécution de notre programme du début jusqu'à la fin de la simulation. Le diagramme de notre application (FIGURE 12).

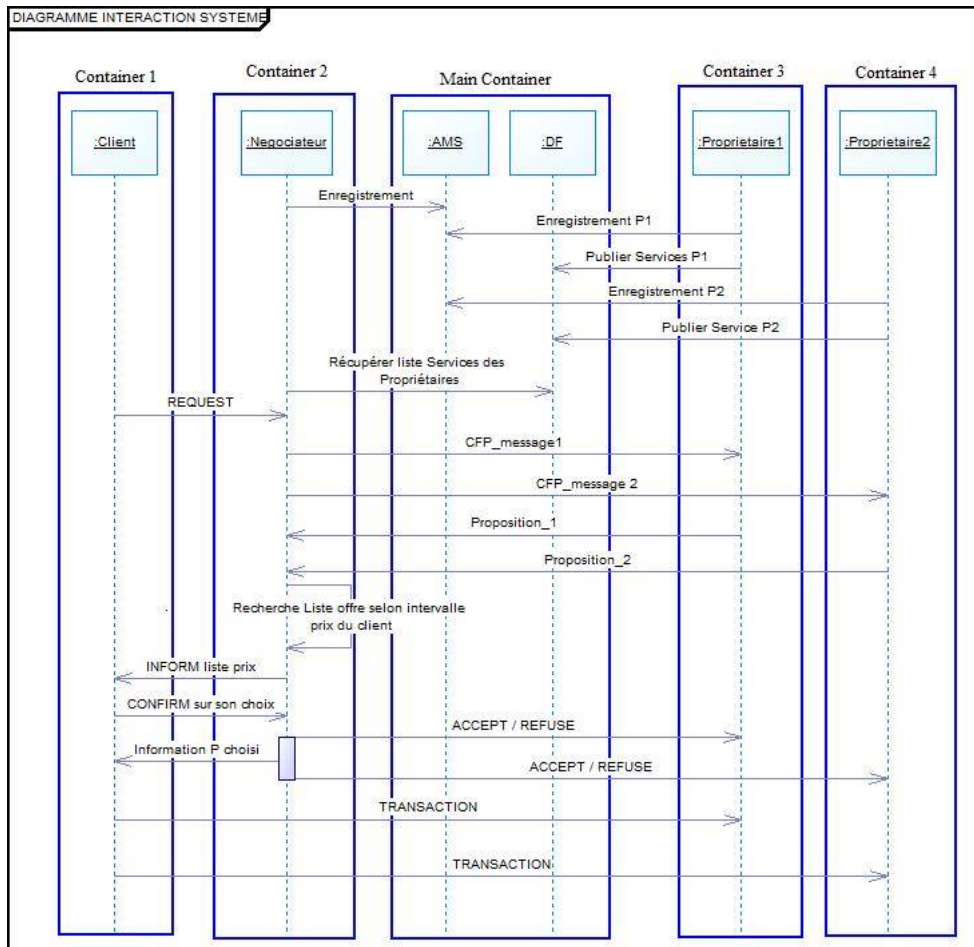


FIGURE 12 – Diagramme d'activité du système

5 Implémentation

Dans cette partie de rapport, nous allons vous présenter l'architecture adopter ainsi que les outils utilises durant la phase de développement de notre application.

Étant donné que le système exige une communication entre les agents, donc nous devrions implémenter une architecture distribuée orienté agent.

5.1 Présentation de quelques plateforme SMA

Dans cette partie, nous allons présenté quelques plateforme SMA avec ses descriptions :

CORMAS (COMmon Resources Multi-Agent System) est un framework de développement de systèmes multi- agents, open-source et basé sur le langage de programmation orientée objet SmallTalk. Il est centré sur des

problématiques de recherche en sciences du développement et de négociation entre acteurs.

DoMIS est un outil permettant la conception de Systèmes Multi-agents (orientés "pilotage opérationnel de systèmes complexes") . Utilisé Pour l'analyse décisionnelle des systèmes complexes.

JACK est un langage de programmation et un environnement de développement pour agents cognitifs, développé par la société Agent Oriented Software comme une extension orientée agent du langage Java.

JADE (Java Agent DEvelopment) est un framework de développement de systèmes multi-agents, open-source et basé sur le langage Java. Il offre en particulier un support avancé de la norme FIPA-ACL, ainsi que des outils de validation syntaxique des messages entre agents basé sur les ontologies.

5.2 Structure des données

Dans le cadre de notre projet, aucune structure de données n'est indispensable. Étant donné le nombre minimale de données à traiter seulement le prix des chambres d'hôtels.

5.3 Choix et justification des outils utilisés

Présentation de JADE : C'est une plate-forme multi-agent créée par le laboratoire TILAB

Avantages de JADE :

- ☞ C'est un framework qui suit les normes FIPA (Foundation for Intelligent Physical Agents).
- ☞ Utilisation des 3 modules dans la norme FIPA :
 - ❖ **DF** : « Directory Facilitator » annuaire de service de la plate-forme ;
 - ❖ **ACC** : « Agent Communication Channel » gestionnaire de communication entre les agents ;
 - ❖ **AMS** : « Agent Management System » responsable de la sécurité des services (enregistrement de toutes les fournisseurs de services).
- ☞ JADE dispose de plusieurs composants qui facilitent le développement des agents :
 - ❖ **Un Runtime Environment** : environnement de développement des agents ;
 - ❖ **Une librairie de classes (skeleton)** : qui facilite la création des agents ;
 - ❖ **Une suite d'outils graphiques** qui permet de gérer facilement les agents et les 3 modules : DF, ACC, AMS.

5.4 Environnement de développement

- ☞ Ubuntu 16.04 comme système d'exploitation ;
- ☞ Github pour la gestion des versions ;
- ☞ Eclipse Neon ;
- ☞ app.genmymodel.com pour la conception (UML).

5.5 Démarrage de l'application

Pour démarré notre application, il faut suivre les étapes suivantes :

- ❶ Démarrer le main container,
- ❷ Démarrer 1 ou plusieurs propriétaires avec le prix de leur service ;
- ❸ Démarrer 1 négociateur pour récupérer la liste des services publier par les propriétaires,
- ❹ Démarrer 1 ou plusieurs client qui demande une chambre d'hôtel.

5.6 Flux d'interaction des agents

La (FIGURE 13) nous montre un exemple de le flux d'interaction des messages entre les agents. L'outil « agent sniffer » de JADE permet de visualiser toutes les conversations de chaque agents.

Dans cet exemple, on a utilisé :

- ☞ 4 agents propriétaires : P1 prix 100, P2 prix 150, P3 prix 200, P4 prix 250
- ☞ 1 agent négociateur : negociateur
- ☞ 1 agent client : client prix entre 100 et 200
- ☞ ajouter avec les agent du main container qui sont DF et AMS

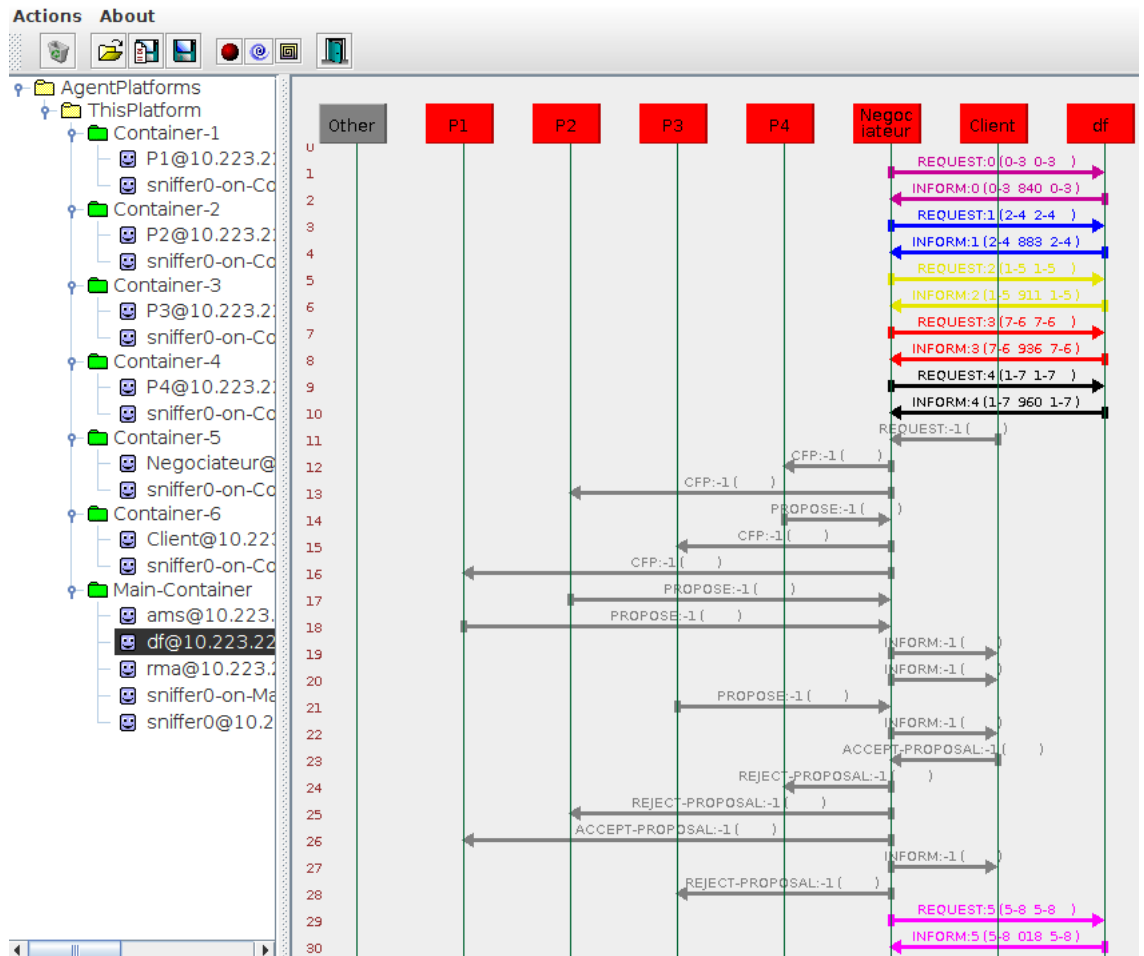


FIGURE 13 – Flux de conversations avec Sniffer Agent

5.6.1 Explications et interprétations des messages obtenus

- ☞ les interactions entre agent négociateur et DF représentent la récupération des informations ou liste des services disponible dans DF (effectué tous les 10 secondes).
- ☞ les interactions entre l'agent client et l'agent négociateur :
 - ❶ Requête du client qui demande une chambre d'hôtel avec une intervalle de prix (prix minimale et prix maximale)
 - ❷ Réponse du négociateur avec la proposition des prix de chambre disponible.
 - ❸ Validation du client pour la chambre
 - ❹ Réponse du négociateur avec la soumission des informations personnels du propriétaire de la chambre pour la transaction.
- ☞ les interactions entre les agents propriétaires et l'agent négociateur :
 - ❶ l'agent négociateur fait des appels d'offre aux agents propriétaires

- ② les agents propriétaires répondent au offre du négociateur.
 - ③ à partir des exigences du client, le négociateur répond aux propositions des agents propriétaires (ACCEPTER/REFUSER)
- une seule interactions entre l'agent propriétaire et l'agent client ; après avoir reçu les informations personnels du propriétaire, le client initie la transaction.

6 Présentation de l'application

6.1 Agent Main

Présentation de l'agent main container principale (FIGURE 14).

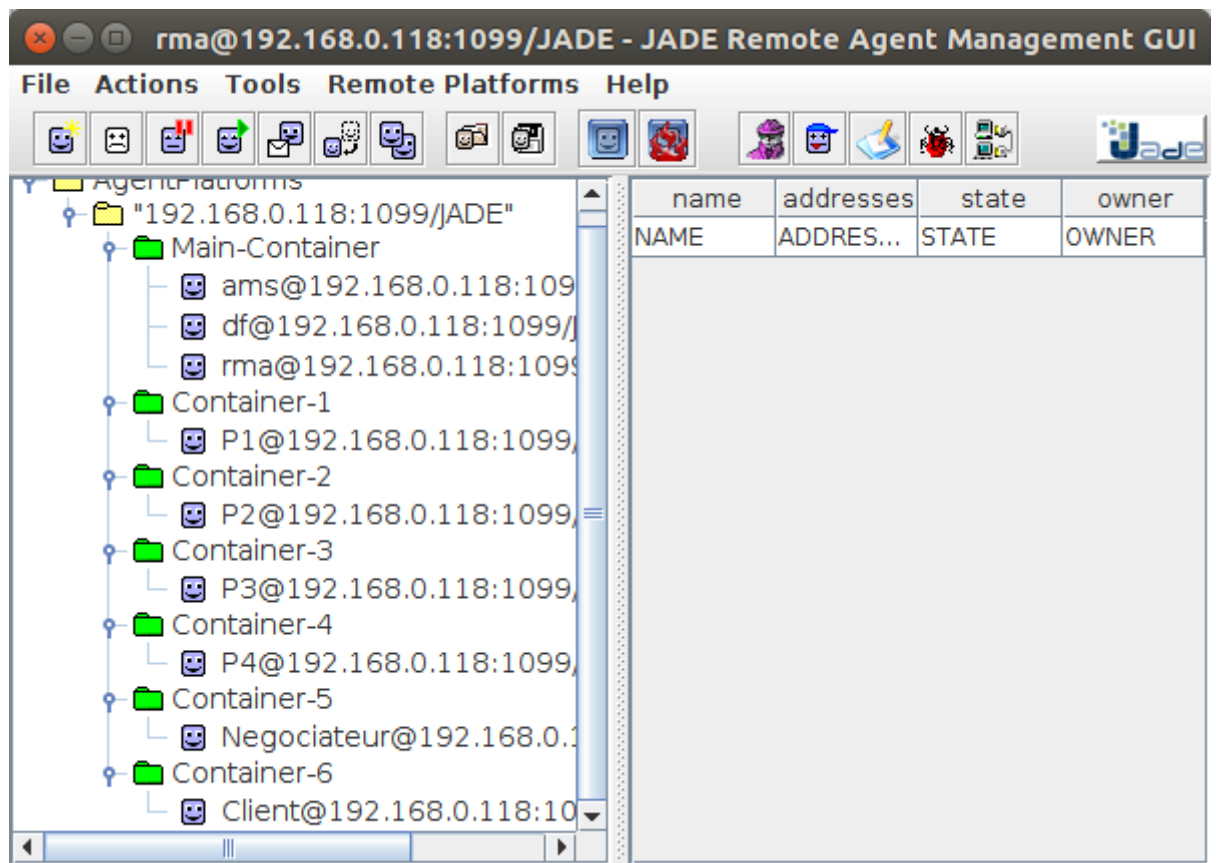


FIGURE 14 – Présentation de l'agent main

6.2 Agent Propriétaire

Présentation du container de l'agent propriétaire (FIGURE 15).

×

–

□

Proprietaire

Nom:

P3

Deploy

Prix:

200

Ajouter

Type	Content	Sender
CFP	Demande d...	Negociateur
REJECT_PR...	Refuser	Negociateur

FIGURE 15 – Présentation de l’agent propriétaire

6.3 Agent Client

Présentation du container de l’agent client (FIGURE 16).

×

–

□

Client

Prix Min:

Prix Max:

Ajouter

Vider

Prix:

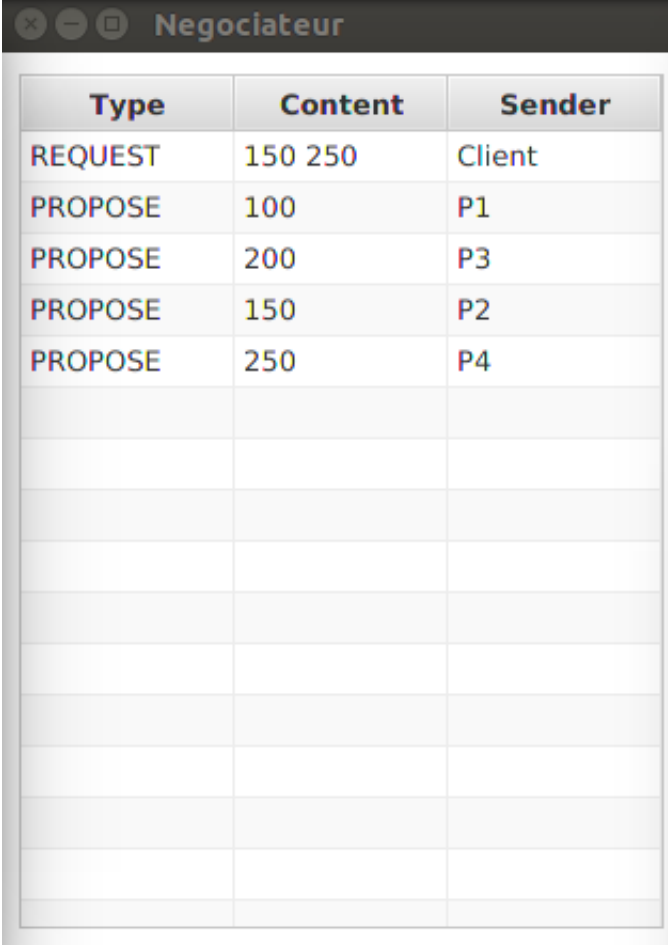
Valider

Type	Content	Sender
INFORM	200	Negociateur
INFORM	150	Negociateur
INFORM	250	Negociateur
INFORM	P4	Negociateur

FIGURE 16 – Présentation de l'agent client

6.4 Agent Négociateur

Présentation du container de l'agent négociateur (FIGURE 17).



The image shows a window titled "Negociateur" with a table containing negotiation messages. The table has three columns: "Type", "Content", and "Sender". The messages are as follows:

Type	Content	Sender
REQUEST	150 250	Client
PROPOSE	100	P1
PROPOSE	200	P3
PROPOSE	150	P2
PROPOSE	250	P4

FIGURE 17 – Présentation de l’agent négociateur

7 Difficultés rencontrés

- Le temps pour la compréhension et de maîtriser les outils ;
- Comprendre comment modéliser et implémenter un système orienté agent répondant aux besoins de notre sujet ;
- La coordination du travail en équipe pour la réalisation des tâches assignées à chaque membre ;
- Difficulté liée à la compréhension de la plate-forme JADE ;
- Difficultés liées à la compréhension du modèle de conception GAIA.

8 Conclusion

Parvenu au terme de notre projet qui a été d'une importance capitale pour l'équipe car, nous a permis de mettre en pratique toutes les connaissances théoriques reçues pendant le cours. Ce projet dans sa globalité nous a permis de comprendre comment modéliser un projet orienté agent et de mettre en œuvre ce dernier avec JADE (Java Agent Development), de comprendre son architecture (AMS, ACC et DF), la configuration et le démarrage d'un container de JADE, de comprendre le cycle de vie des agents, le comportement des agents jade, la communication entre les agents avec ACL et de comprendre le phénomène de mobilité des agents pour ne citer que ces parties. Notre logiciel permettra aux clients qui sont représentés par des agents du système de pouvoir s'offrir ou pas une chambre dans un hôtel à partir d'un intervalle définissant le montant ou la somme d'argent dont ils disposent. Réservé à toute maintenance, notre application pourra permettre aussi celle évolutive car toute modification allant dans le sens de l'amélioration des fonctionnalités, de la performance et même l'interaction homme machine sera la bienvenue car le logiciel est extensible.

Références

- LEDANG, Hung (2001). “Des cas d’utilisation à une spécification B”. In : *Approches Formelles dans l’Assistance au Développement de Logiciels-AFADl’2001*, 10–p.
- AUDIBERT, Laurent (2009). *UML 2 : de l’apprentissage à la pratique : présentation des diagrammes UML (cas d’utilisation, classes, objets, états-transitions, activités, structures composites, communication, séquence, composants, déploiement), langage de contraintes OCL, introduction aux patrons de conception (design patterns), mise en oeuvre d’UML*. Ellipses.

Books only

- AUDIBERT, Laurent (2009). *UML 2 : de l’apprentissage à la pratique : présentation des diagrammes UML (cas d’utilisation, classes, objets, états-transitions, activités, structures composites, communication, séquence, composants, déploiement), langage de contraintes OCL, introduction aux patrons de conception (design patterns), mise en oeuvre d’UML*. Ellipses.