

Architecture logicielle et qualité

RAPPORT DU TP

Système multi-agent
Market Pit Simulation

Réalisé par :

Youssef NIDABRAHIM

Mohamed Reda BENCHRAA

Encadré par :

Pr. Jérémie BOUNY

Table des illustrations

Figure 1: Diagramme de séquence - initialisation	7
Figure 2: Diagramme de séquence - Négociation	8
Figure 3: Diagramme de classe.....	8
Figure 4: Diagramme d'activité	9
Figure 5: Lancement de Main container	13
Figure 6: Lancement de l'Admin container	13
Figure 7: Lancement du vendeur container	14
Figure 8: Lancement de l'Acheteur container	15
Figure 9: Containers et ses agents	15
Figure 10: Les services publiés par les vendeurs.....	16
Figure 11: Extrait de messages reçus par les vendeurs.....	16
Figure 12: Extrait de messages reçus par les acheteurs.....	17
Figure 13: Interaction entre les vendeurs et admin.....	18
Figure 14: Négociation entre les acheteurs et les vendeurs.....	18
Figure 15: Log des transactions.....	19

Table des matières

Table des illustrations.....	ii
Chapitre I : Contexte général du projet	1
1. Objectif du projet	2
2. Règles de gestion.....	2
2.1 Déroulement du jeu	2
2.2 Traitements	2
2.3 Agents.....	3
3. Résultats, interprétation et discussion.....	4
Chapitre II : Analyse et conception	6
1. Scénarios	7
2. Diagramme de classe.....	8
3. Diagramme d'activité	9
Chapitre III : Réalisation et Mise en œuvre	10
1. Outils et technologies.....	11
1.1 Eclipse.....	11
1.2 Java	11
1.3 Jade.....	11
1.4 Git et GitHub.....	12
1.5 PowerAMC.....	13
2. Prise en main	13
Conclusion et Perspective	20

CHAPITRE I

Contexte général du projet

1. Objectif du projet

Ce jeu est dédié pour toute personne intéressée sous le cadre d'une culture Entrepreneuriat. Tout en modélisant une situation réelle d'un marché de vente et d'achat, en analysant les comportements des acheteurs et des vendeurs. Ainsi participer concrètement à des transactions marchandes simples permet en revanche de faciliter l'accès à l'étude de la coordination marchande.

2. Règles de gestion

2.1 Déroulement du jeu

Au début de chaque jeu, l'organisateur devra préparer un certain nombre de cartes pour les acheteurs et les vendeurs, ici la distribution se fait de manière aléatoire à leurs demandes.

Chaque participant au début de chaque partie ou à la fin d'une transaction obtiendra une nouvelle carte numérotée.

Chaque carte représente la valeur d'une unité d'un bien qui peut être achetée par les acheteurs et vendue par les vendeurs.

Chaque participant doit garder sa carte de manière à ce que personne ne puisse la voir.

Après avoir enregistré la Transaction de la vente; l'acheteur et le vendeur vont alors rendre leurs cartes, reçoivent de nouvelles cartes et retournent sur le marché pour conclure de nouvelles transactions.

Il est possible que certains acheteurs ou certains vendeurs ne parviennent pas à réaliser de transaction. Ils ne doivent cependant pas se décourager puisque de nouvelles cartes leur seront distribuées au début de la manche suivante. Les gains sont de 0 pour chaque unité du bien non vendue ou non achetée.

A la fin de la période d'échange qui dure 10min, il fera enlever les cartes correspondant aux unités du bien non négociées.

2.2 Traitements

Les acheteurs et les vendeurs pourront négocier pendant une période d'échange.

Lorsqu'un acheteur et un vendeur se seront mis d'accord sur un prix, ils doivent confirmer ensemble et enregistrer leur transaction.

Le prix négocié et la valeur de gain ou de perte seront enregistrés sur la fiche de transaction des joueurs.

Le gain ou la perte de la transaction est la différence entre le prix de négociation et le coût réelle sur la carte.

Lorsque les transactions de la première manche sont terminées, les joueurs ne peuvent commencer la deuxième partie qu'après la confirmation de l'administrateur.

Les résultats de la période passée sont alors enregistrés sur la fiche de transaction. Un nouveau cycle de décisions peut alors être joué.

2.3 Agents

- **Acheteur :**

Il n'est pas conseillé d'acheter à un prix supérieur à la valeur sur la carte. Si l'acheteur accepte d'acheter à un prix supérieur à sa valeur, alors la transaction sera validée avec une perte.

Le gain pour l'achat est la différence entre la valeur portée sur la carte et le prix de la négociation. Si jamais aucune achat n'est enregistré, il n'y aura ni gain ni perte non plus.

L'acheteur peut approcher tout vendeur pour commencer la négociation.

Négocier le prix avec un vendeur avec la possibilité de suggérer un prix, accepter le prix suggéré par le vendeur ou bien le refuser.

- **Vendeur :**

Le gain ou la perte de la vente est la différence entre le prix de négociation et le coût réelle sur la carte. Si jamais aucune vente n'est enregistré, il n'y aura ni gain ni perte non plus. s

Le vendeur peut approcher tout acheteur en vue de commencer la négociation.

Négocier le prix avec un acheteur avec la possibilité de suggérer un prix, accepter le prix suggéré par l'acheteur ou bien le refuser.

- **Administrateur :**

L'administrateur peut lancer le jeu, il doit attendre la demande des vendeurs ou acheteurs pour leur attribuer une carte.

- **Facilitateur :**

Agent DF, Directory Facilitator, fournit un service de pages jaunes à la plate-forme.

- **Gestionnaire :**

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

3. Résultats, interprétation et discussion

Si les vendeurs proposent des prix trop élevés ou si les acheteurs proposent des prix trop faibles, les gains totaux réalisés par l'ensemble des élèves seront faibles et le marché n'aboutira donc pas à une coordination efficace.

Cette expérience est intéressante. Elle permet d'introduire, à partir d'un contact direct avec le marché, les problèmes de négociation et de coordination, la question de la formation des prix, la loi de l'offre et de la demande. La construction de courbes d'offre et de demande ainsi que la compréhension du concept d'équilibre sont facilitées. En Première, on peut également s'appuyer sur cette expérience pour présenter la notion d'efficacité, ce qui permet aisément d'introduire les concepts de surplus du vendeur, de surplus de l'acheteur et de surplus collectif.

Les coûts de production des vendeurs et les valeurs des acheteurs déterminent les courbes d'offre et de demande sur le marché ainsi que les prix et quantité de l'équilibre concurrentiel.

Les négociations des participants aboutissent presque systématiquement à une convergence vers ce prix et cette quantité d'équilibre, cette convergence étant de plus en plus marquée au fur et à mesure de la répétition des périodes d'échange. La loi de l'offre et de la demande se vérifie donc et les prix remplissent leur rôle informatif.

La convergence vers l'équilibre concurrentiel permet également de maximiser la somme des gains des participants à l'échange et assure ainsi de l'efficacité de la coordination.

On peut partir directement de l'observation des séries de prix négociés et leur demander de réfléchir sur les raisons de la convergence vers un prix unique.

La discussion peut se prolonger sur la pertinence de la loi de l'offre et de la demande dans des contextes de marché plus complexes, sur les raisons pour lesquelles la flexibilité des prix peut être remise en cause.

CHAPITRE II

Analyse et conception

1. Scénarios

Les scénarios représentent une succession particulière d'enchaînements, s'exécutant du début à la fin du cas d'utilisation, un enchaînement étant l'unité de description de séquences d'actions. Un cas d'utilisation contient en général un scénario nominal et plusieurs scénarios alternatifs (qui se terminent de façon normale) ou d'erreur (qui se terminent en échec).

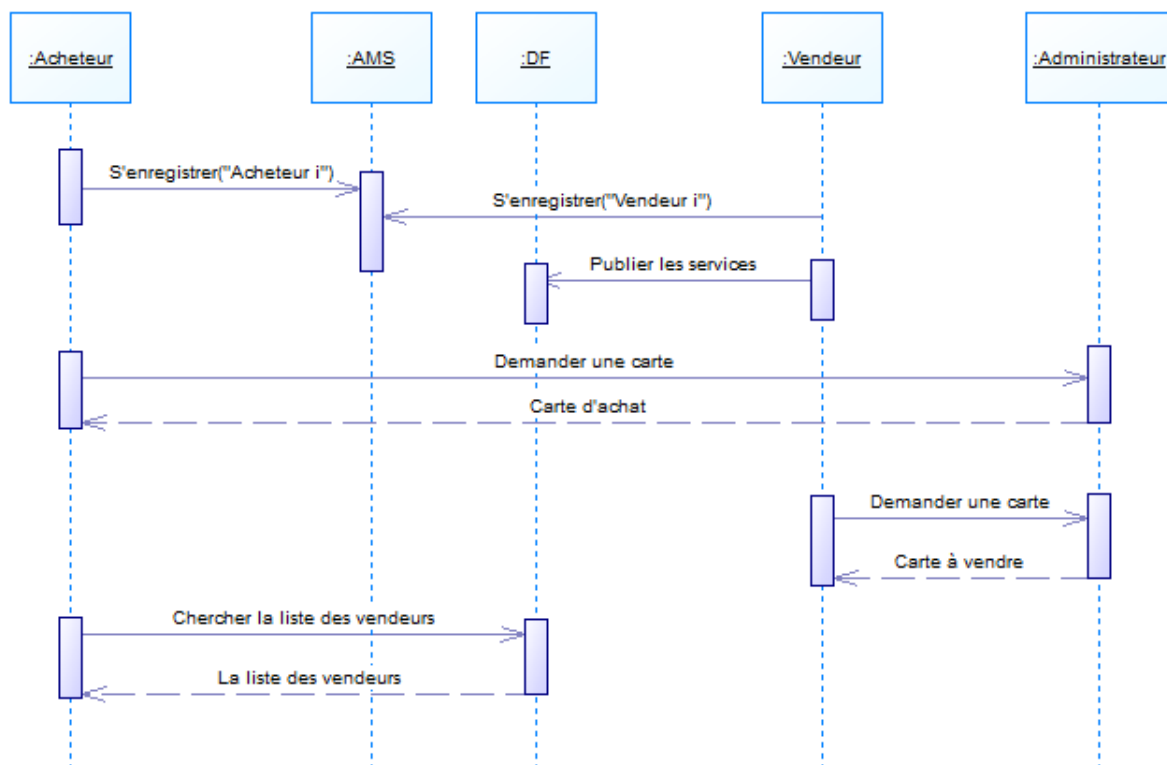


Figure 1: Diagramme de séquence - initialisation

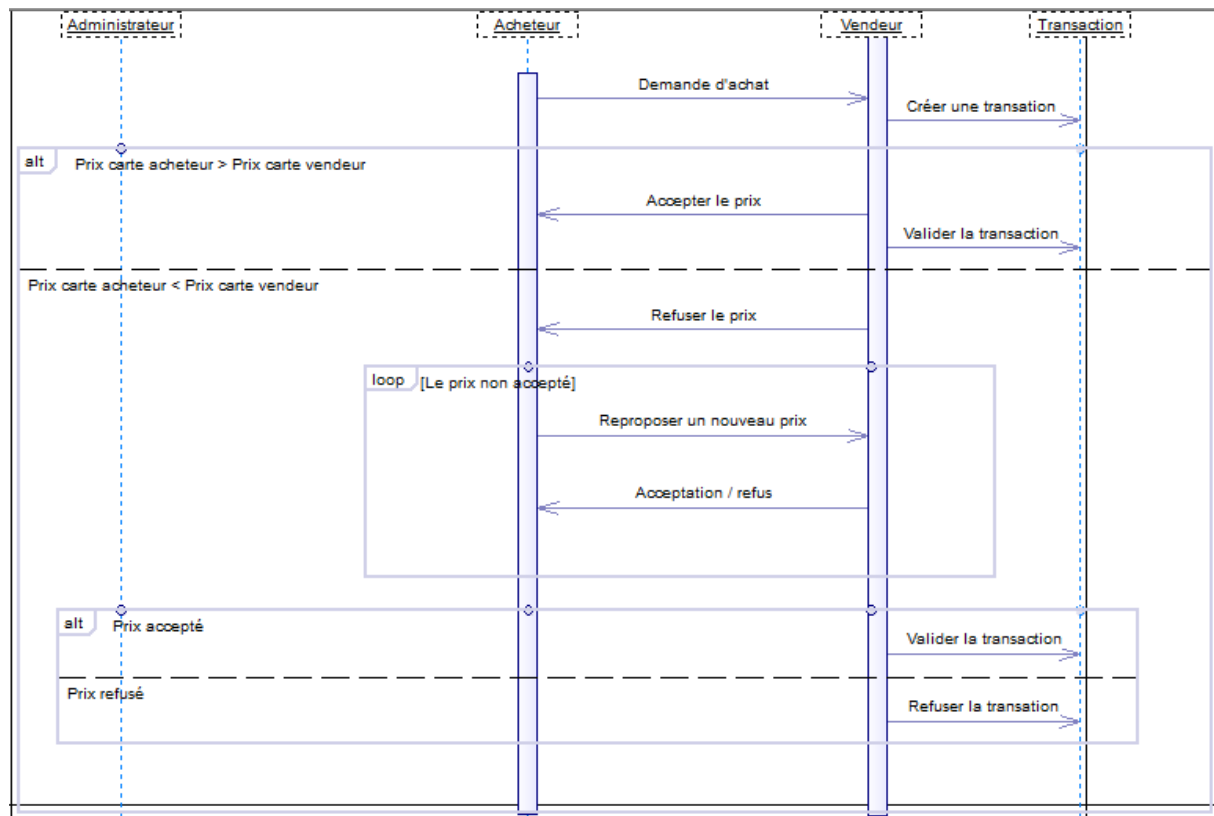


Figure 2: Diagramme de séquence - Négociation

2. Diagramme de classe

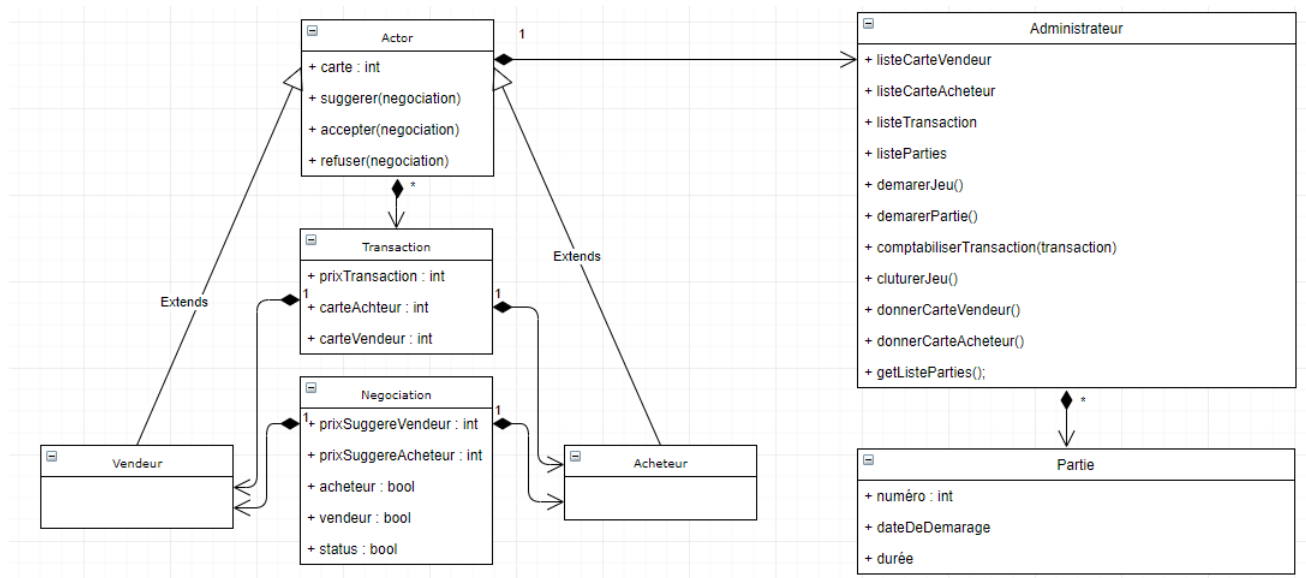


Figure 3: Diagramme de classe

3. Diagramme d'activité

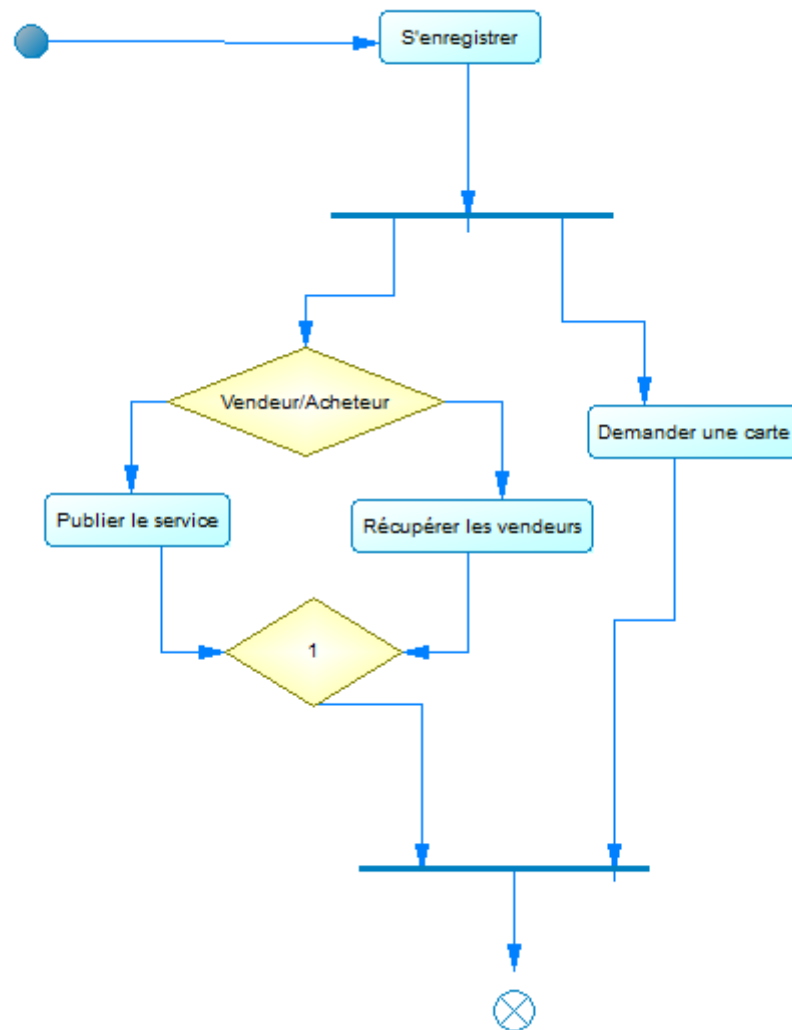


Figure 4: Diagramme d'activité

CHAPITRE III

Réalisation et Mise en œuvre

1. Outils et technologies

1.1 Eclipse

Eclipse est un IDE, environnement de développement intégré, écrit en Java, extensible par des greffons, multi-langages et multi-plates-formes.

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 comme C/C++, Python, PHP, Ruby, etc.

Toutes les fonctions qu'on peut attendre de ce genre de logiciel sont présentes ou existent sous forme de greffons (coloration syntaxique, complétion, débogage, gestion de projets, intégration aux gestionnaires de versions, ...).

1.2 Java

Java est un langage de programmation orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems.

La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitation avec peu ou pas de modifications. Pour cela, divers plateformes et frameworks associés visent à guider, sinon garantir, cette portabilité des applications développées en Java.

1.3 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.

C'est un framework qui permet le développement de systèmes multiagents et d'applications conformes aux normes FIPA, Foundation for Intelligent Physical Agents.

JADE possède trois modules principaux (nécessaire aux normes FIPA).

- DF : Directory 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.

Ces trois modules sont activés à chaque démarrage de la plate-forme.

JADE est un middleware qui facilite le développement des systèmes multi agents (SMA). JADE contient :

- Un *Runtime Environment* : l'environnement où les agents peuvent vivre.
- Une librairie de classes : que les développeurs utilisent pour écrire leurs agents
- Une suite d'outils graphiques : qui facilitent la gestion et la supervision de la plateforme des agents

Chaque instance du JADE est appelée conteneur ou Container, et peut contenir plusieurs agents.

Un ensemble de conteneurs constitue une plateforme.

Chaque plateforme doit contenir un conteneur spécial appelé main-container et tous les autres conteneurs s'enregistrent auprès de celui-là dès leur lancement.

1.4 Git et GitHub

Git est un logiciel de gestion de versions décentralisé. Il est conçu pour être efficace tant avec les petits projets, que les plus importants.



Contrairement à des outils comme SVN ou CVS, GIT fonctionne de façon décentralisée, c'est-à-dire que le développement ne se fait pas sur un serveur centralisé, mais chaque personne peut développer sur son propre dépôt, il facilite ensuite la fusion (merge) des différents dépôts.

GITHUB est un service web d'hébergement et de gestion de développement logiciel utilisant les logiciels de gestion de versions GIT. Il propose des comptes professionnels payants, ainsi que des comptes gratuits pour les projets de logiciels libres. En tant qu'étudiant, nous avons bénéficié d'un pack student qui nous offre des hébergements privés.



1.5 PowerAMC

POWERAMC est un logiciel de conception qui permet de modéliser les traitements informatiques et leurs bases de données associées. Il se destine aux architectures de données, aux architectures d'informations et aux architectures d'entreprise.

2. Prise en main

```
INFOS: -----  
Agent container Main-Container@192.168.142.1 is ready.  
-----
```

Figure 5: Lancement de Main container

Lancement du main container qui contient les agents système de Framework Jade.

```
INFOS: -----  
Agent container Container-1@192.168.142.1 is ready.  
-----  
Admin Admin@192.168.142.1:1099/JADE
```

Figure 6: Lancement de l'Admin container

Lancement de l'admin container d'où appartient notre agent administrateur qui va s'occuper de préparer le terrain pour le jeu.


```
INFOS: -----  
Agent container Container-2@192.168.142.1 is ready.  
-----  
Vendeur Vendeur2@192.168.142.1:1099/JADE  
Vendeur Vendeur3@192.168.142.1:1099/JADE  
Vendeur Vendeur1@192.168.142.1:1099/JADE  
Vendeur Vendeur4@192.168.142.1:1099/JADE  
Vendeur Vendeur5@192.168.142.1:1099/JADE  
-----  
> Réception d'une carte :82  
Expéditeur :Admin@192.168.142.1:1099/JADE  
-----  
> Réception d'une carte :52  
Expéditeur :Admin@192.168.142.1:1099/JADE  
Block  
Block  
-----  
> Réception d'une carte :74  
Expéditeur :Admin@192.168.142.1:1099/JADE  
Block  
-----  
> Réception d'une carte :30  
Expéditeur :Admin@192.168.142.1:1099/JADE  
Block  
-----  
> Réception d'une carte :62  
Expéditeur :Admin@192.168.142.1:1099/JADE  
Block
```

Figure 7: Lancement du vendeur container

Vendeur container contient tous les agents vendeurs qui vont être lancés au lancement du container, pour notre test nous avons lancé cinq agents vendeur. Dès qu'un agent vendeur est lancé, il va demander à l'agent Admin une carte pour la vendre.

```

INFOS: -----
Agent container Container-3@192.168.142.1 is ready.
-----

Acheteur : Acheteur3@192.168.142.1:1099/JADE
Acheteur : Acheteur1@192.168.142.1:1099/JADE
Acheteur : Acheteur2@192.168.142.1:1099/JADE
Acheteur : Acheteur4@192.168.142.1:1099/JADE
Acheteur : Acheteur5@192.168.142.1:1099/JADE
-----

> Réception d'une carte :76
From : Admin@192.168.142.1:1099/JADE
Acheteur : Envoie d'une demande
-----

> Réception d'une carte :83
From : Admin@192.168.142.1:1099/JADE
Acheteur : Envoie d'une demande
-----

> Réception d'une carte :80
From : Admin@192.168.142.1:1099/JADE
Acheteur : Envoie d'une demande
-----

> Réception d'une carte :54
From : Admin@192.168.142.1:1099/JADE
Acheteur : Envoie d'une demande
-----

> Réception d'une carte :64
From : Admin@192.168.142.1:1099/JADE
Acheteur : Envoie d'une demande
-----

```

Figure 8: Lancement de l'Acheteur container

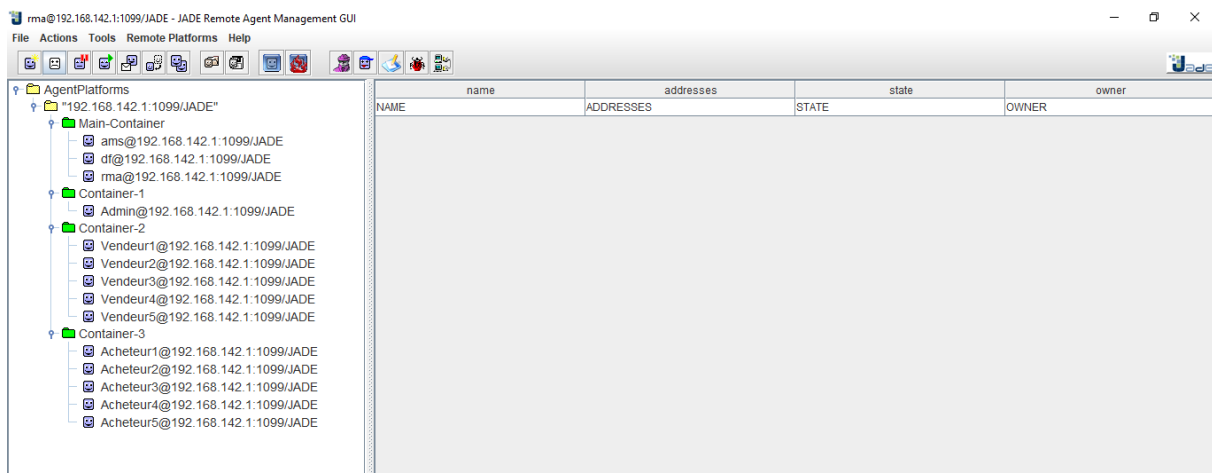


Figure 9: Containers et ses agents

Comme vous pouvez le voir, les containers contiennent bien des agents. On a choisi de créer cinq agents vendeurs et cinq agents acheteur et un seul administrateur.

Registrations with this DF	Search Result	DF Federation
Agent name	Addresses	Resolvers
Vendeur4@192.168.142.1:10...	http://DESKTOP-4HGIOQQ:77...	
Vendeur5@192.168.142.1:10...	http://DESKTOP-4HGIOQQ:77...	
Vendeur1@192.168.142.1:10...	http://DESKTOP-4HGIOQQ:77...	
Vendeur2@192.168.142.1:10...	http://DESKTOP-4HGIOQQ:77...	
Vendeur3@192.168.142.1:10...	http://DESKTOP-4HGIOQQ:77...	

Figure 10: Les services publiés par les vendeurs

Chaque agent vendeur doit publier ses services, c'est-à-dire d'indiquer à l'agent DF, le facilitateur, qu'il a une carte à vendre, cela permet aux agents acheteur de récupérer le contact des vendeurs.

```

-----
> Réception d'un refus
From :Acheteur5@192.168.142.1:1099/JADE
Contenu : Carte proposée est trop chère
-----
> Validation de la transaction
-----
> Demande d'achat
From :Acheteur5@192.168.142.1:1099/JADE
Contenu : Je vous demande le prix de ta carte !
acheteur : 64
vendeur : 74
-----
> Réception d'un refus
From :Acheteur5@192.168.142.1:1099/JADE
Contenu : Carte proposée est trop chère
Block
-----

```

Figure 11: Extrait de messages reçus par les vendeurs

```
-----  
> Réception d'une annulation  
From :Vendeur4@192.168.142.1:1099/JADE  
Contenu : J'ai déjà vendu ma carte  
Acheteur : Envoie d'une demande  
-----  
> Réception de l'offre  
From :Vendeur5@192.168.142.1:1099/JADE  
Contenu : Voilà ma proposition  
Prix : 52  
-----  
> Réception de la confirmation  
From :Vendeur5@192.168.142.1:1099/JADE  
Contenu : Vendeur5:52:Acheteur3:80:52  
-----  
> Réception d'une annulation  
From :Vendeur4@192.168.142.1:1099/JADE  
Contenu : J'ai déjà vendu ma carte  
Acheteur : Envoie d'une demande  
-----  
> Réception d'une annulation  
From :Vendeur5@192.168.142.1:1099/JADE  
Contenu : J'ai déjà vendu ma carte  
Acheteur : Envoie d'une demande  
-----  
> Réception d'un refus  
From :Vendeur1@192.168.142.1:1099/JADE  
-----
```

Figure 12: Extrait de messages reçus par les acheteurs

Les agents acheteurs demandent à l'agent DF de leurs envoyer les vendeurs qui ont déjà publié leurs services, le DF leurs transmet la liste des vendeurs et à partir de cette liste que les acheteurs pourront avoir le contact afin de négocier et d'acheter la carte proposée par un vendeur.

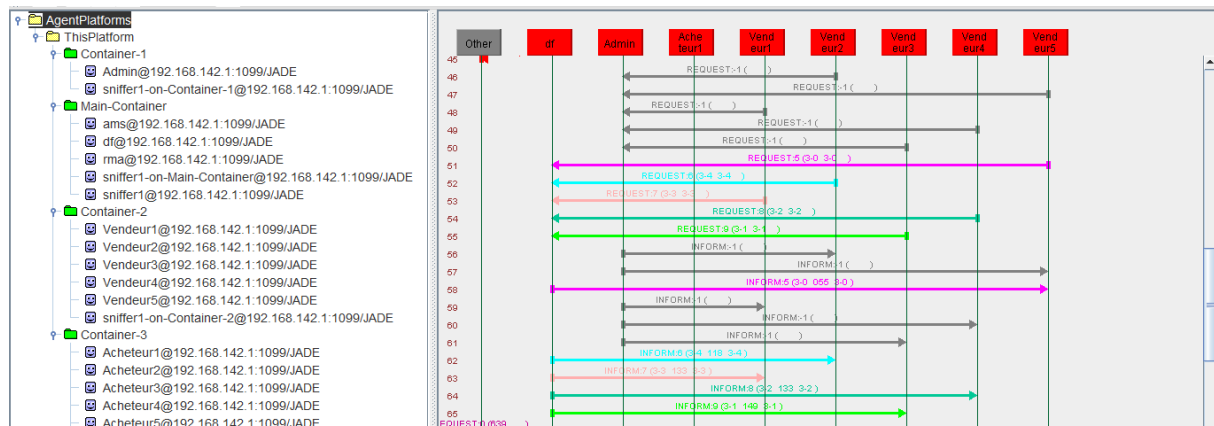


Figure 13: Interaction entre les vendeurs et admin

Cette figure illustre les demandes des cartes par les vendeurs à l'administrateur.

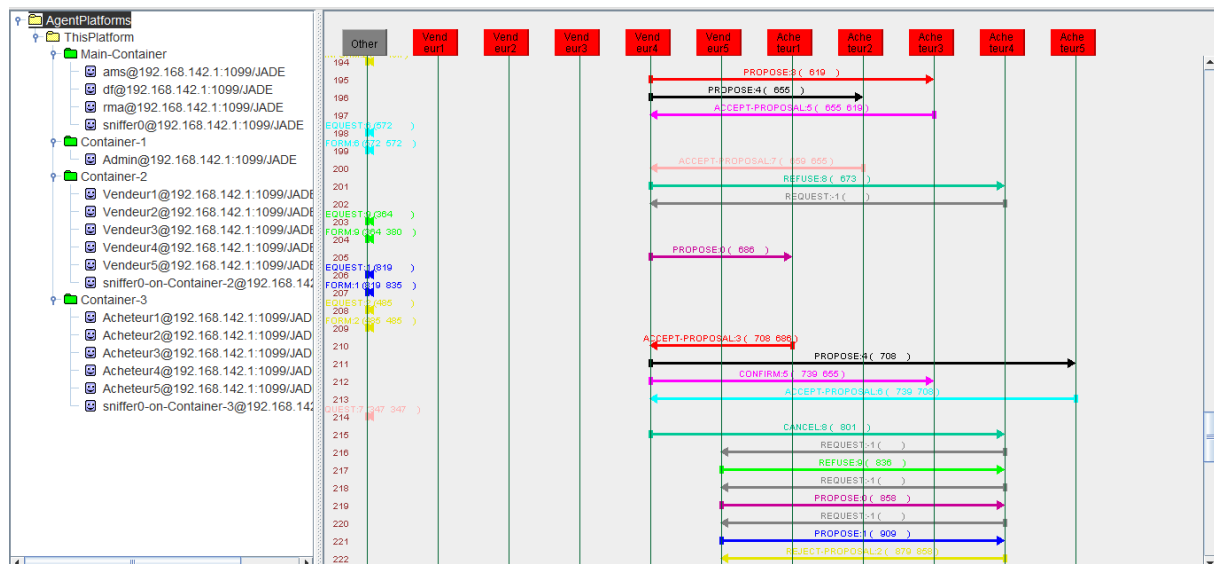
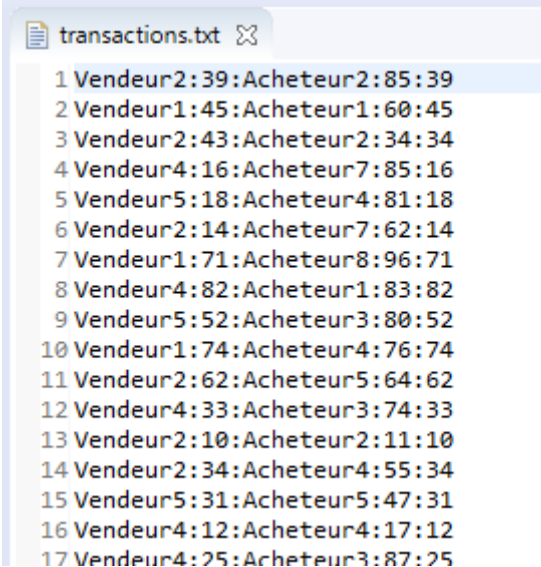


Figure 14: Négociation entre les acheteurs et les vendeurs

Cette figure représente l'enchaînement des messages qui circulent entre les acheteurs et les vendeurs.



```
transactions.txt
1 Vendeur2:39:Acheteur2:85:39
2 Vendeur1:45:Acheteur1:60:45
3 Vendeur2:43:Acheteur2:34:34
4 Vendeur4:16:Acheteur7:85:16
5 Vendeur5:18:Acheteur4:81:18
6 Vendeur2:14:Acheteur7:62:14
7 Vendeur1:71:Acheteur8:96:71
8 Vendeur4:82:Acheteur1:83:82
9 Vendeur5:52:Acheteur3:80:52
10 Vendeur1:74:Acheteur4:76:74
11 Vendeur2:62:Acheteur5:64:62
12 Vendeur4:33:Acheteur3:74:33
13 Vendeur2:10:Acheteur2:11:10
14 Vendeur2:34:Acheteur4:55:34
15 Vendeur5:31:Acheteur5:47:31
16 Vendeur4:12:Acheteur4:17:12
17 Vendeur4:25:Acheteur3:87:25
```

Figure 15: Log des transactions

Si une négociation est réussite, le programme sauvegarde la transaction dans un fichier. Ce dernier donne une idée sur ce qui s'est passé réellement et des informations sur la transaction à savoir le vendeur, l'acheteur, le montant de la carte du vendeur, le montant de la carte de l'acheteur ainsi que le prix de l'achat de la carte. Ce fichier sera la base d'une analyse du résultat de cette simulation.

Conclusion et Perspective

L'objectif visé à travers ce rapport est de présenter le programme réalisé dans le cadre de notre TP en architecture et qualité logiciel qui consiste à concevoir et à développer un système multi-agent pour simuler un marché, *Market Pit Simulation*.

Nous avons pu développer un vrai système multi-agent qui met en place un certain nombre d'agents et qui commencent à se communiquer et négocier pour avoir un bon prix. Ce programme est développé en utilisant le framework Jade qu'on a près l'occasion de le découvrir au cours de ce travail.

Nous pourrions envisager à améliorer ce travail en y apportant d'autres ingrédients par exemple une analyse du résultat de notre simulation avec des graphes.