



INSTITUT
DE LA FRANCOPHONIE
POUR L'INNOVATION

CONCEPTION ET IMPLEMENTATION D'UN
SYSTEME MULTI-AGENTS SUR LA PLATE-FORME
GAIA

Rapport de Travail Pratique N2 : SIM-IA

Gestion de CrossDocking par
agents réactifs

Auteurs :

ADOUM OKIM BOKA
OLEMBO RÉEL DEVIN RICHMOND
OUEDRAOGO ABDOULFATAO
BOHINBO TIMOTHÉE

Encadrant :

Dr. Manh Hung NGUYEN

8 mars 2019

Table des matières

1	Introduction	4
2	La définition de Cross-docking	5
2.1	Les avantages du cross-docking	6
2.2	CROSS DOCKING LOGISTIQUE : L'OFFRE	6
3	Outils et technologies utilisés	6
4	Analyse de l'application	7
4.1	Modèle de rôles de l'application	7
4.1.1	Modèle de rôle de l'agent wagon	8
4.1.2	Modèle de rôle de l'agent entrepôt	8
4.1.3	Modèle de rôle de l'agent camion	9
4.1.4	Modèle de rôle de l'agent client	9
4.2	Modèle de protocoles de l'application	10
4.2.1	Modèle de service Camion	10
4.2.2	Modèle de service Entrepôt	11
4.2.3	Modèle de service Fournisseur	12
4.2.4	Modèle de service wagon	12
4.3	Modèle d'acointance	13
5	Conception	13
5.1	Modèle d'agents	13
5.2	Modèle de classe	13
5.3	Protocole détaillé	13
5.4	Diagramme d'inter-action	16

6	Implémentation	17
6.1	Architecture de notre application	18
6.2	Interface principale de gestion de système	19
6.3	Une vue dà l’initialisation de la communication inter-agent	20
7	Conclusion	21

Résumé

Ce travail est une suite logique de cours de Intelligence artificiel et système multi-agent. Dont l'un de ses objectifs est d'apprendre aux étudiants à Concevoir et d'implémenter des systèmes multi-agents inter-actifs en utilisant la methodology GAIA et la plate-forme Jade.

1 Introduction

Pour cloturer le cours de Intelligence Artificiel(IA) et Système Multi-agent, nous avons eu l'occasion de travailler sur les thèmes liés au domaine dont l'option de choix des thèmes dépendent de la pertinence du point de vue de groupe travail. Pour ceux, nous avons décidé de concevoir et d'implémenter une application de gestion de cross-docking par des agents réactifs. En effet le cross docking est un processus qui permet le décharger les marchandises des véhicules entrants dans un entrepot directement sur les véhicules sortants. Ce système a beaucoup d'avantages car il permet de réduire les coûts liés aux stockage des produits dans l'entrepôts, rapide et avec un gain de profit considerable.

Ce travail est structuré autour des points suivants. Ou nous allons commencer par l'etat de l'art de la technogie de cross-docking question de s'impregner de son mode de fonctionnement, decrire les outils et la technologie à utiliser, analyser le problème, concevoire, mettre en oeuvre et faire quelques tests.

2 La définition de Cross-docking

le cross-docking est l'action de faire passer des marchandises des quais d'arrivée aux quais de départs, sans passage par le stock. Il est particulièrement intéressant d'utiliser le cross docking lorsque le niveau d'activité de l'entrepôt est élevé. Il permet en effet d'optimiser la manutention et l'entreposage des marchandises dans l'entrepôt. grâce à de fréquentes livraisons de marchandises.

Les marchandises réceptionnées le jour même doivent quitter l'entrepôt au plus tard le lendemain.

La méthode est donc particulièrement adaptée au traitement des commandes à priorité élevée.

Pour que sa mise en place soit couronnée de succès, il convient d'adopter une organisation d'une grande précision..

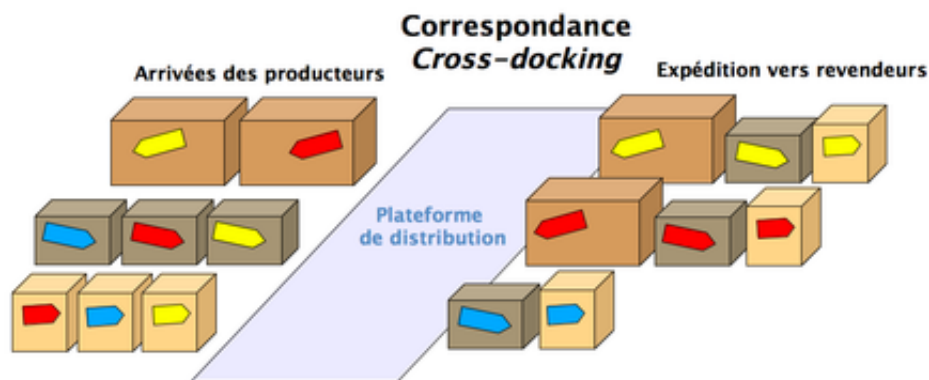


FIGURE 1 – schema illustratif du système

2.1 Les avantages du cross-docking

Le cross docking est une solution avantageuse en termes de réactivité et de coût. L'organisation du cross docking permet d'optimiser la gestion des approvisionnements et des stocks tout au long de la chaîne logistique. Eviter la phase de stockage offre un gain de temps précieux. Ce type de préparation est idéal pour les commandes urgentes ou à priorité élevée (produits périssables, événementiels...).

2.2 CROSS DOCKING LOGISTIQUE : L'OFFRE



FIGURE 2 – Image de la logistique cross-docking

Le service cross-docking, optimise les flux logistiques selon une organisation rigoureuse et précise :

- Réception contrôlée des marchandises ;
- Préparation des commandes ;
- Chargement des camions/containers ;
- Expédition, transport.

3 Outils et technologies utilisés

Pour la réalisation de notre système nous avons utilisé :

- La méthodologie GAIA pour l'analyse et la conception de notre système ;
- La librairie Jade pour créer les agent, les contenaires, et affecter des comportements aux agents ;
- ECLIPSE, comme environnement de développement ;

- Java, comme langage de programmation.

4 Analyse de l'application

Les phases des analyses et des spécifications des besoins permettent de d'écrire les fonctionnalités de la plate-forme et les contraintes sous lesquelles celle-ci doit être réalisé. Pour ce, nous allons énumérer les agents, leur role, leur activité ...

Pour notre application, nous avons identifié quatre (4) différents agents qui interviendront lors de la mise en place de ladite application.

Agent fournisseur : cet agent joue le rôle de fournisseur c'est le point qui fournit les produits à livrer à l'entrepôt pour ensuite être transféré au client final.

Agent wagon : c'est l'agent représentant les véhicules de livraison entrant et son rôle est de transporter des produits jusqu'au lieu de déchargement c'est-à-dire à l'entrepôt.

Agent entrepôt : c'est celui qui représente le lieu de livraison des véhicules entrants c'est-à-dire agent wagon. Son rôle est de vérifier les produits entrants et ensuite charger ces produits dans d'autres véhicules appelé agent camion et l'ai expédiés ensuite.

Agent camion : cet agent représente les véhicules sortants. le rôle de et agent est de transporter les produits chargés jusqu'à la destination finale.

4.1 Modèle de rôles de l'application

Dans cette partie nous allons décrire nos différents agents, leur protocole, leurs activités et leur permissions.

4.1.1 Modèle de rôle de l'agent wagon


Rôle schéma: wagon
Description: au nom de wagon, s'occupe du transporteur du produit de chez le fournisseur jusqu'au quai de déchargement.
Protocole: Communication, Consultation, Validation
Activités: Transporter, décharger, charger les produit
Permission: lire et écrire les attributs du produit 

FIGURE 3 – schema illustratif modèle de rôle de l'agent wagon

4.1.2 Modèle de rôle de l'agent entrepôt


Rôle schéma: entrepôt
Description: Au nom de l'entrepôt, est le lieu où se passe travaux de déchargement et le chargement des produits vers la destination entre le Wagon et le camion
Protocole: Communication, Consultation, Validation, Transaction
Activités: Décharger, vérifier les attributs, charger les produits
Permission: lire et écrire: <ul style="list-style-type: none">• les attributs des produits• les informations sur le stock• la liste de ses produits disponibles avec leur prix 

FIGURE 4 – schema illustratif modèle de rôle de l'agent entrepôt

4.1.3 Modèle de rôle de l'agent camion

Rôle schéma: camion
Description: au nom de camion, s'occupe du transport du produit jusqu'à la destination.
Protocole: Communication, Consultation, Validation
Activités: Transporter, charger, livrer les produit, vérifier le client
Permission: lire et écrire les attributs du produit et du client

FIGURE 5 – schema illustratif modèle de rôle de l'agent camion

4.1.4 Modèle de rôle de l'agent client

Rôle schéma: client
Description: Au nom de client, est le responsable à qui est doit livrer le produit
Protocole: Communication, Validation
activités: Modifier les attributs, Confirmer la livraison
Permission: Lire et écrire les attributs des produits

FIGURE 6 – schema illustratif modèle de rôle de l'agent wagon

4.2 Modèle de protocoles de l'application

4.2.1 Modèle de service Camion

Service: Livraison du produit	
inputs	Information sur le client
outputs	-
Pré-condition	connaître l'itinéraire
Post-condition	-

FIGURE 7 – Modèle de service Camion

4.2.2 Modèle de service Entrepôt

Service 1: Réception du produit	
inputs	Informations sur le produit
Outputs	-
Pré-condition	Vérification du produit
Post-condition	valider la réception du produit
Services 3: Information sur le destinataire(client)	
inputs	Information sur le client
Outputs	-
Pré-condition	Vérifier l'existence du
Post-condition	Accord entre les deux agents
Services 4: Valider l'envoi du produit	
inputs	information sur le camion chargé de l'envoi du produits
Outputs	-
Pré-condition	Vérifier la disponibilité du camion
Post-condition	envoi du produit a client

FIGURE 8 – Modèle de service Entrepôt

4.2.3 Modèle de service Fournisseur

Services: Enregistrement du produit	
Inputs	prépare la commande
outputs	-
pré-condition	accord avec l'agent entrepôt <input type="checkbox"/>
post-condition	

FIGURE 9 – Modèle de service Fournisseur

4.2.4 Modèle de service wagon

Service: Déchargement du produit à l'entrepôt	
Inputs	Information sur le produits
outputs	-
pré-condition	Accord avec l'agent fournisseur
post-condition	transport des produits vers l'entrepôt

FIGURE 10 – Modèle de service Wagon

4.3 Modèle d'acointance

Agents	Fournisseur	Wagon	Entrepôt	Camion	Client
Fournisseur	Non	Interaction	Interaction	Non	Non
Wagon	Interaction	Non	Interaction	Non	Non
Entrepôt	Interaction	Interaction	Non	Interaction	Non
Camion	Non	Non	Interaction	Non	Interaction
Client	Interaction	Non	Non	Interaction	Non

FIGURE 11 – Modèle d'acointance

5 Conception

5.1 Modèle d'agents

5.2 Modèle de classe

Ce modèle de classe illustre les relations entre les cas d'utilisation du système et l'ensemble des agents qui inter-agissent sur le système. Ainsi le diagramme décrit le comportement du système de point de vue utilisateur.

5.3 Protocole détaillé

Nous avons ressortie au moins quatre protocole dans notre système, dont quelques uns sont ressortis dans le modèle de classe ci-dessus :

Transaction Dans cette phase, le fournisseur demande un service de livraison d'un colie et Main lance le processus de livraison ;

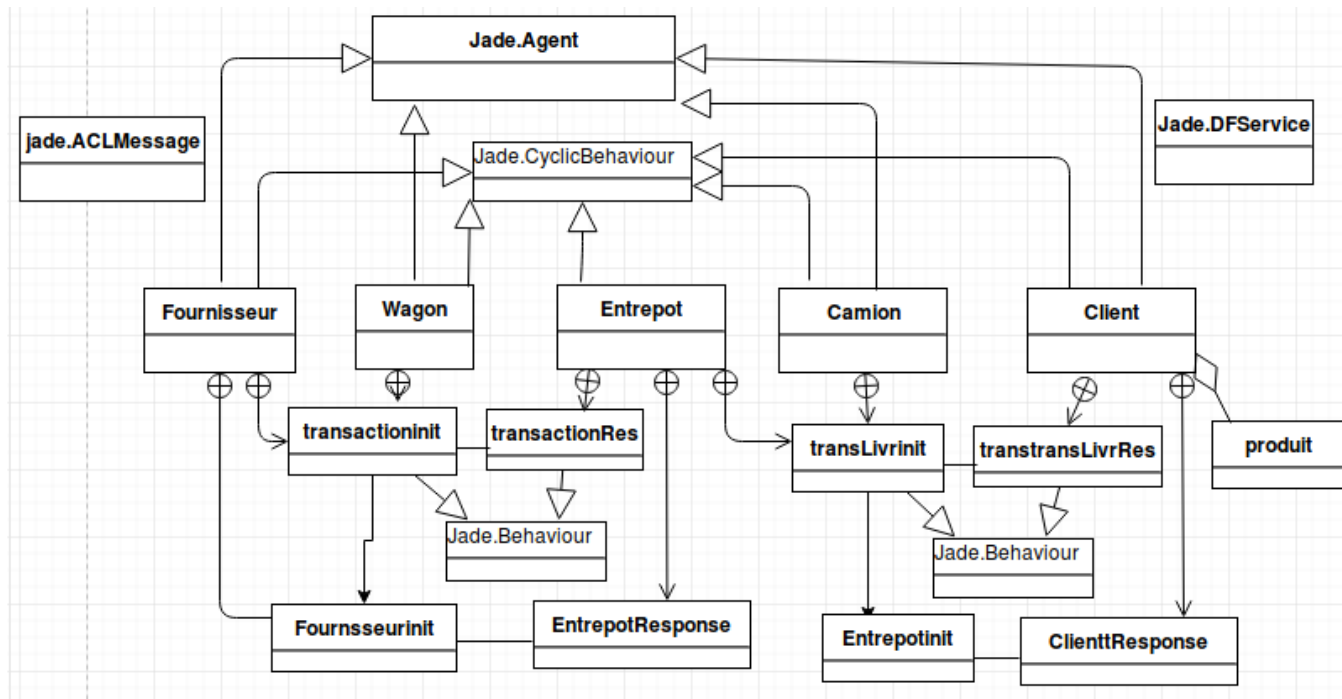


FIGURE 12 – modèle de classe de notre système

Transactioninit initialisation du processus de livraison de produit ;

TransactionRes Reponse de l'entrepôt au fournisseur a près la reception de produit ;

Fournisseurinit demande de service de livraison ;

EntrepotResponse reponse de l'entrepôt au fournisseur ;

TransLivinit initialisation de processus de livraison de produit de l'entrepot pour le client ;

TranstransLivrRes Reponse du client après la reception du produit livré ;

Entrepôtinit l'entrepôt informe le client qu'il a un colis et l'informe aussi du temps possible de livraison ;

ClientResponse Response du client et donne sa position et le conduit à tenir.

Entrepôtinit l'entrepôt informe le client qu'il a un colis à l'entrepôt ;

5.4 Diagramme d'interaction

Ainsi se présente notre diagramme d'interaction.

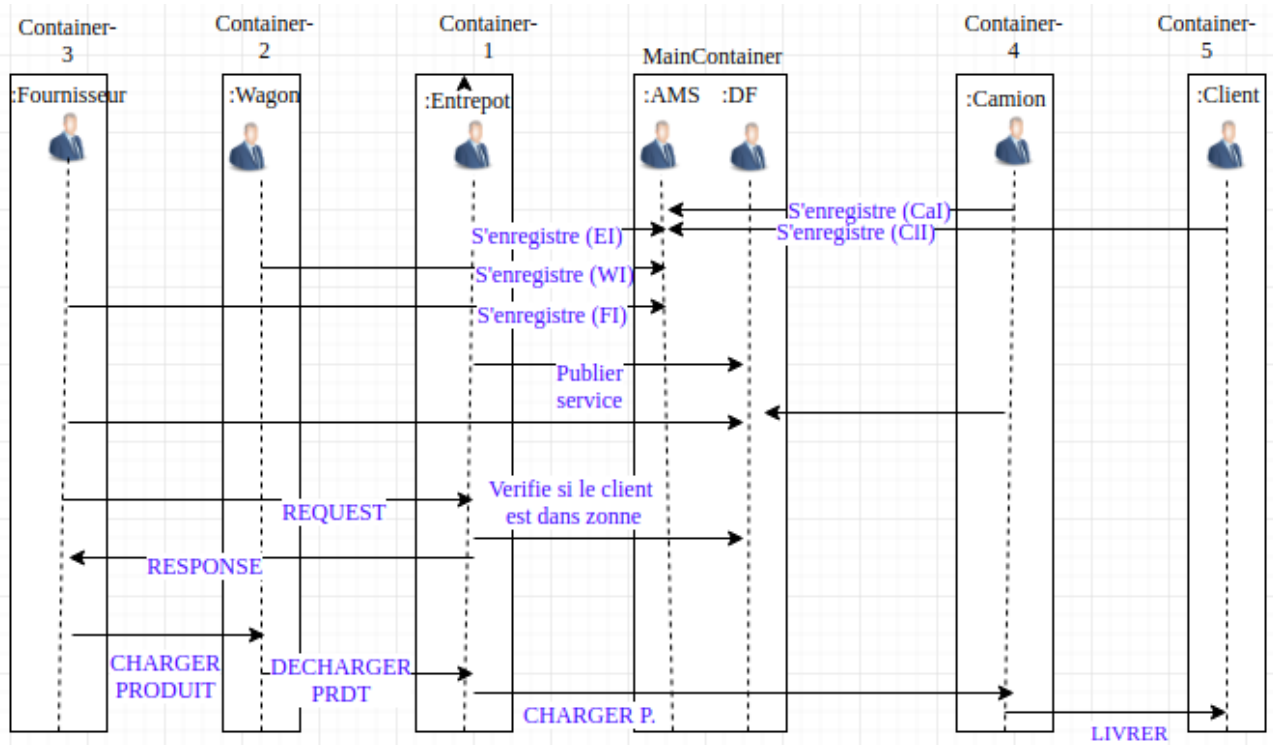


FIGURE 13 – diagramme d'interaction

6 Implémentation

Cette partie concerne la mise en œuvre de notre application qui consiste à effectuer le codage et faire les tests nécessaires. Nous présenterons captures d'écrans et vous joindre le code source.

6.1 Architecture de notre application

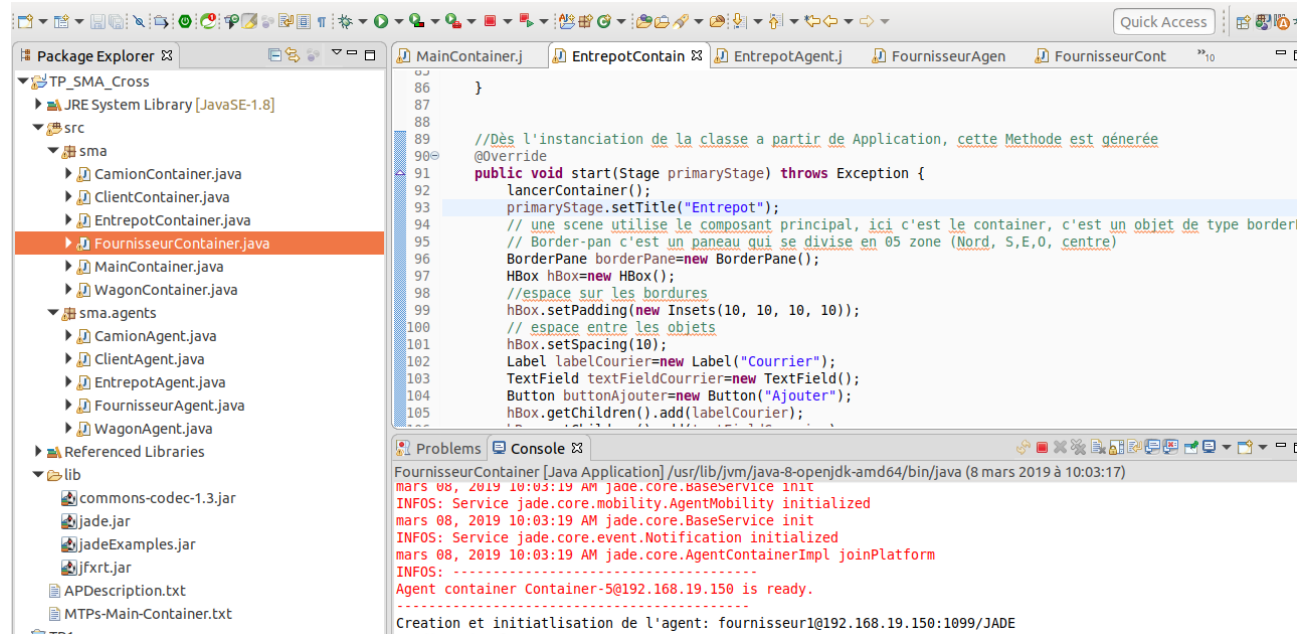


FIGURE 14 – Architecture de notre application

6.2 Interface principale de gestion de système

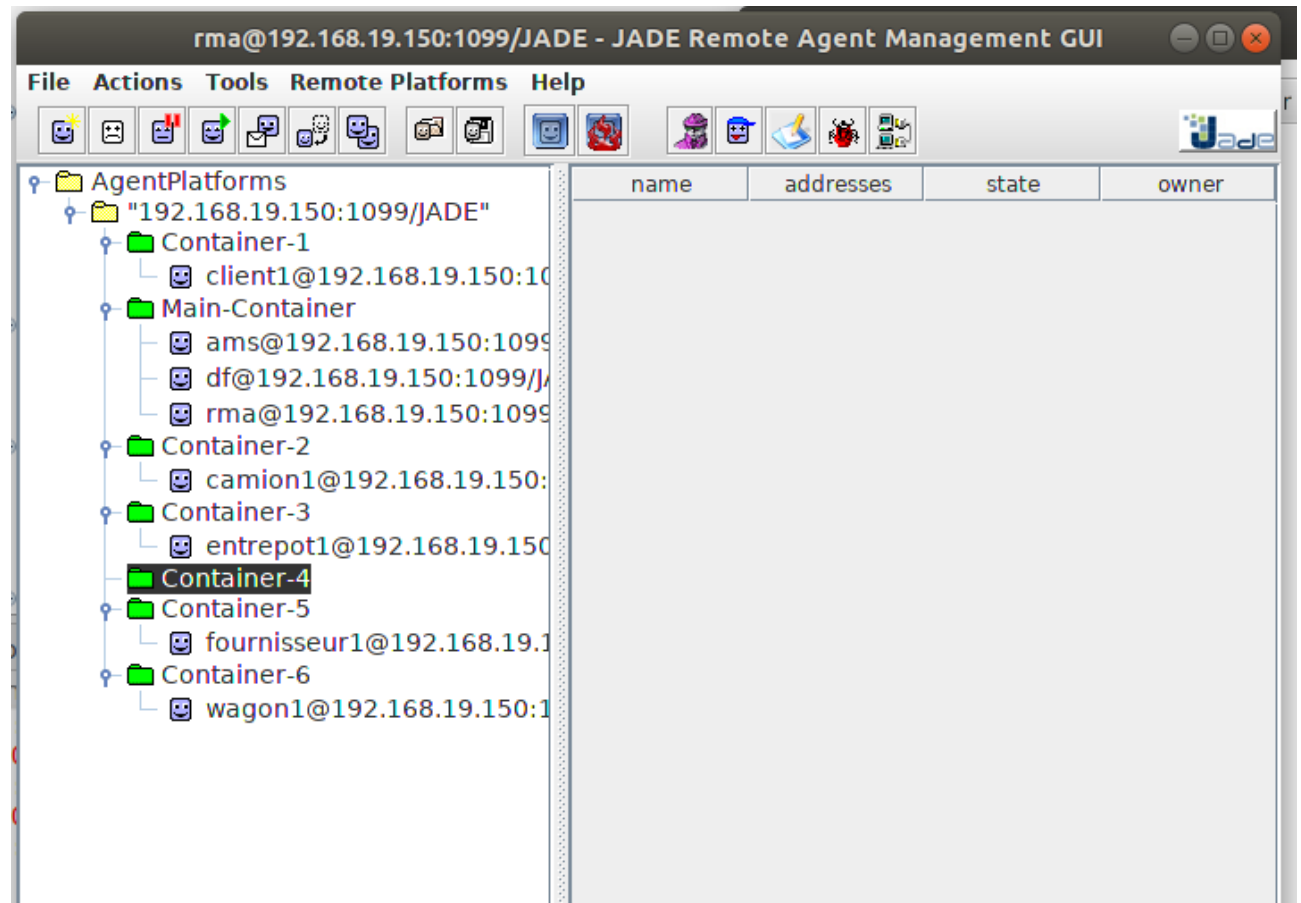


FIGURE 15 – Interface principale de gestion de système

6.3 Une vue dès l'initialisation de la communication inter-agent

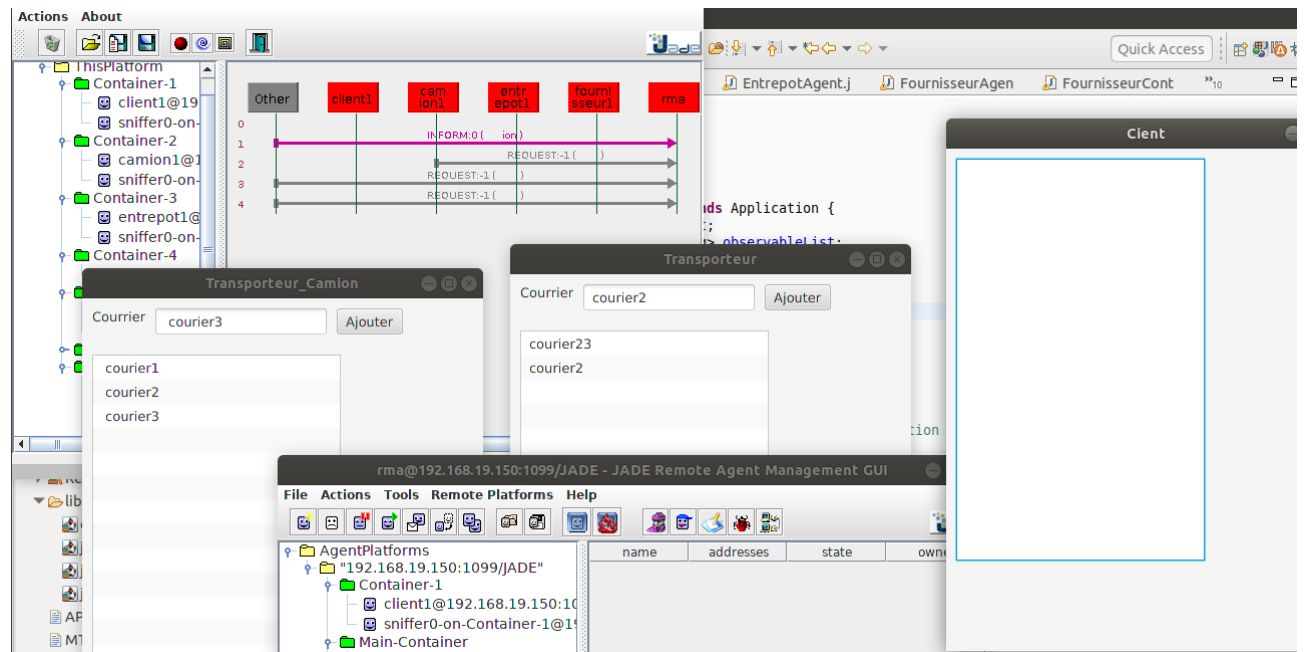


FIGURE 16 – Interface principale de gestion de système

7 Conclusion

Au terme de notre travail, nous avons pu nous familiariser avec les outils tels GAIA et le framework JADE sous l'environnement Eclipse dans le but de concevoir et implémenter une application du cross docking par agents réactifs. Cette application orientée agent permettra d'éviter les opérations dites manuelles dans le processus d'acheminement d'un produit au niveau de la plateforme jusqu'à sa livraison finale. En ce qui concerne les perspectives, nous améliorerons son efficacité dans le souci de régler des problèmes de grande portée.

Références

- [1] <https://www.semanticscholar.org/paper/Optimization-and-simulation-of-a-cross-docking-et-Zhang/9c971dd2a7c10e84da28a3576c369f6e6a8e56c9>
- [2] [http://www.ame.org/sites/default/files/target_articles/04 – 20 – 3 – Crossdocking.pdf](http://www.ame.org/sites/default/files/target_articles/04%20-%203-%20Crossdocking.pdf)