SMA-SA

TP2

# Introduction

Ce TP a pour but de mettre en application nos connaissances sur les systèmes multi-agents dans un contexte particulier. Dans le cadre de ce TP, l’objectif est de réaliser un système multi-agents qui est capable de négocier automatiquement. Ce système a pour but de réaliser la négociation de la vente de billets de transport entre des fournisseurs et des négociateurs. On considère que dans notre système, deux négociateurs ainsi que trois fournisseurs cohabitent.

# Partie 1

L’objectif est donc la négociation entre différents agents. Pour cela, nous avons deux types d’agents différents.

Tout d’abord, il y a les fournisseurs. Ceux-ci possèdent des billets en quantité limitée. Ces billets ont un prix minimum en dessous duquel le fournisseur ne veut pas les vendre. Mais cet agent cherchera à les vendre un certain pourcentage plus cher que l’on pourra déterminer.

Ensuite, il y a les négociateurs. Ceux-ci ont des appels d’offre qui leur sont alloués. Leur mission est d’acheter le billet correspondant à cet appel d’offre dans le budget demandé. Cet agent va lui essayer d’obtenir le billet un pourcentage moins cher que son budget, pourcentage que l’on pourra également déterminer.

Les négociateurs vont d’abord demander à tous les fournisseurs s’ils possèdent des billets correspondant à leurs appels d’offre. Si un fournisseur possède un billet correspondant, il répond au négociateur en lui proposant un prix. Sinon il lui renvoie un refus.

Durant la négociation, si un négociateur achète à un billet à un fournisseur, celui-ci doit décrémenter le nombre de billet de ce type qu’il lui reste. Ainsi, à chaque fois qu’il recevra une proposition ou une demande d’un autre négociateur, il devra d’abord vérifier si le billet est toujours disponible. S’il l’est, il continue la négociation. Sinon, il renvoie un refus.

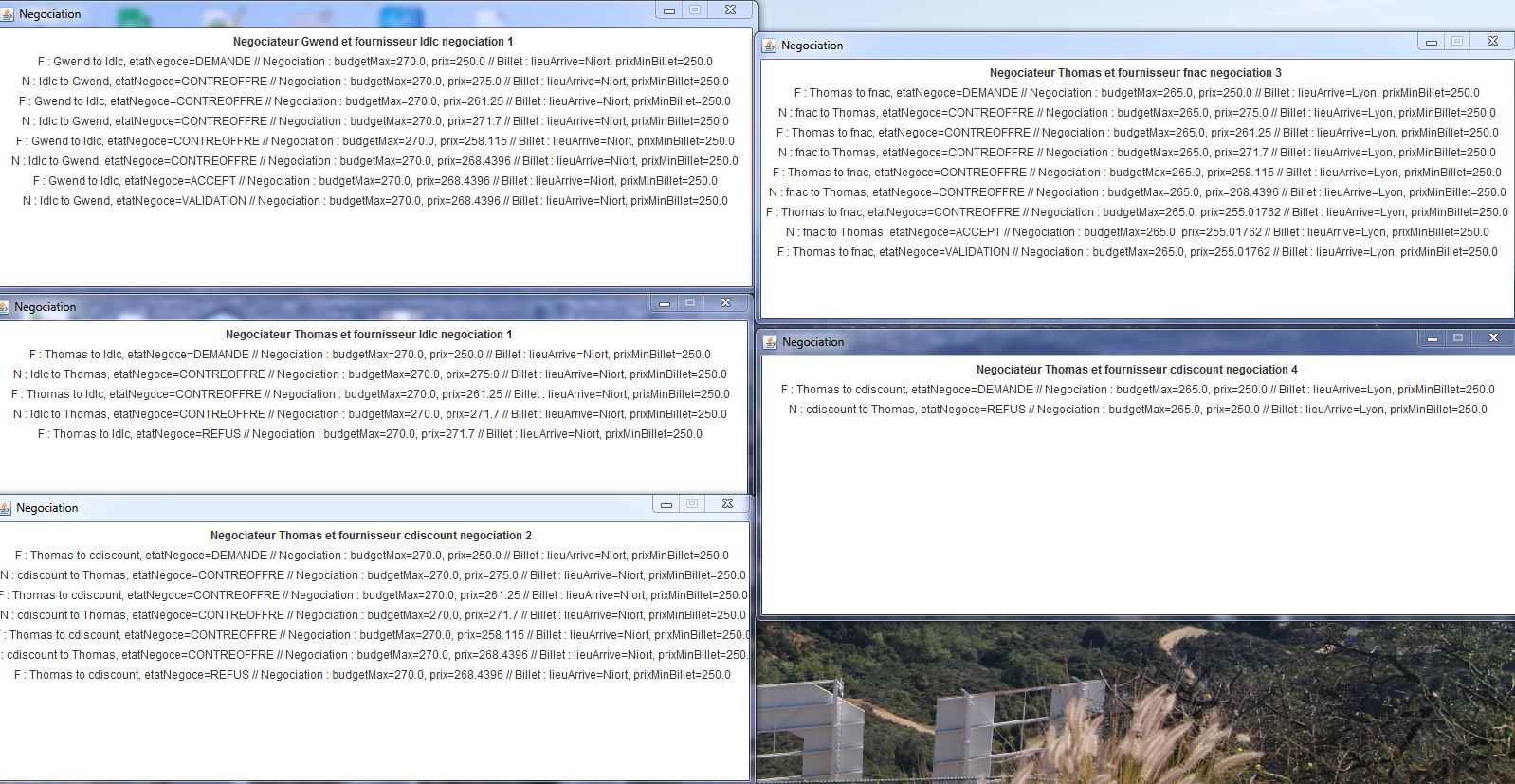
Au début de la négociation, le fournisseur envoie son prix majoré pour le billet. Si celui-ci est trop important par rapport au budget du négociateur, celui va faire une proposition un peu en dessous de son budget afin de faire baisser le prix. Le fournisseur et le négociateur vont alors à tour de rôle répondre par des propositions jusqu’à ce que le prix du billet convienne au deux, c’est-à-dire qu’il est compris dans le budget du négociateur et qu’il est supérieur au prix minimum demandé par le fournisseur. Si les deux agents n’arrivent pas à tomber sur un accord en six propositions, on considère qu’ils n’y arriveront pas et l’un des deux renvoie un refus à l’autre.

Si la proposition d’un agent convient à l’autre agent, celui-ci va envoyer une validation à l’autre. Comme notre système comporte plusieurs négociateurs et fournisseurs et dialoguent tous en même temps, il est nécessaire de mettre en place un accusé de réception de la validation. En effet, entre le moment où un fournisseur envoie une proposition et reçoit une acceptation de cette proposition, il a pu vendre le billet à un autre négociateur. Il faut donc qu’il confirme au négociateur qu’il le lui a bien vendu. De même dans l’autre sens.

Comme le TP1, nous avons choisi de faire communiquer les agents grâce à une liste de message. Ceux-ci comportent toutes les informations relatives à la négociation en cours, c’est-à-dire les parties prenantes, la négociation, le type de message (proposition, validation ou refus) et l’état du message. Chaque agent va tout d’abord lire la liste des messages et repérer ou non un message le concernant. S’il y en a un, il va traiter le message en répondant à son interlocuteur, puis changer l’état du message pour que celui-ci soit marqué comme traité. On peut donc suivre l’historique des messages. La liste des messages est visible par tous les agents.

# Partie 2

Voici une capture d’écran de notre application. Celle-ci met du temps à fonctionner. Il faut attendre quelques minutes afin que toutes les négociations se fassent. Un délai important a été repéré lors de l’ajout d’un message par un agent dans la liste des messages. Cet ajout étant synchronisé entre les thread, ceux-ci doivent attendre que cette action soit faite avant de continuer leur fonctionnement, ce qui ralentit considérablement le déroulement de l’application.

Comme visible sur l’image, chaque négociation fait l’objet d’une fenêtre. Dans cette fenêtre, les lignes correspondent aux messages envoyés par les agents. On peut alors suivre facilement le déroulement des négociations avec toutes les informations nous intéressant.

# Partie 3

Voici le diagramme de classe de notre application.

Les fournisseurs et les négociateurs sont tous les deux des agents qui ont des caractéristiques communes : ils possèdent un nom et sont tous les deux exécutés dans un thread. Nous avons donc créé une classe mère, Agent, qui implémente la classe Runnable.

Le négociateur possède une liste d’appel d’offres qui contiennent les informations relatives aux demandes des clients. Le fournisseur quant à lui possède une liste de billet.

Nous avons décidé de créer une classe Negociation qui permet de suivre les négociations en cours. Elle conserve les informations principales relatives à la négociation comme les parties prenantes, le dernier prix proposé, les détails du billet …

La classe IHM nous permet d’afficher les fenêtres pour chaque négociation. Elle facilite le suivi des négociations en cours.

