



**FACULTÉ
DES SCIENCES
D'ORSAY**

MASTER 2 RECHERCHE INFORMATIQUE IAC

Module : Systèmes Multi-Agent

Rapport du projet Jade

Réalisé par :

Salim Abdelfettah

2013 / 2014

Introduction

Ce projet a pour but la mise en place ainsi que l'implémentation de protocoles de coopération entre agents dans une simulation de cartographie de bâtiments. Dans ce but une structure de données a été définie pour la coopération et quelques mécanismes ont été employés.

Structure de données

Pour que les informations concernant les bâtiments puissent aisément être manipulées (par les agents ou le panel qui se charge de dessiner les bâtiments), mon choix c'est fait sur une structure de données dans une architecture centralisé. Cette dernière se présente sous la forme d'une liste (ArrayList) contenant des objets de type Building. Le type Building est quant à lui formé de 6 entiers (qui définissent la position verticale/horizontale, la longueur/largeur et le décalage vertical/horizontal d'un bâtiment), 2 booléens (l'un définissant l'état de l'agent correspondant au bâtiment, i.e. qu'il soit actif ou non et l'autre le fait que le bâtiment soit visible ou non) et d'une liste contenant les bâtiments qui serait potentiellement en conflit avec lui (ceux qui sont effectivement en conflit avec lui et ceux qui pourrait l'être si l'un et/ou l'autre serait amené à se décaler).

Protocoles

Le programme commence par créer (outre l'interface) un conteneur principal (Main Container) qui contiendra les agents. Lorsque la simulation est lancée, 2 agents sont créés et lancés immédiatement. Il s'agit de l'agent générateur de bâtiments (AgentGenerator) et de l'agent IHM (AgentHCI).

Protocole 1

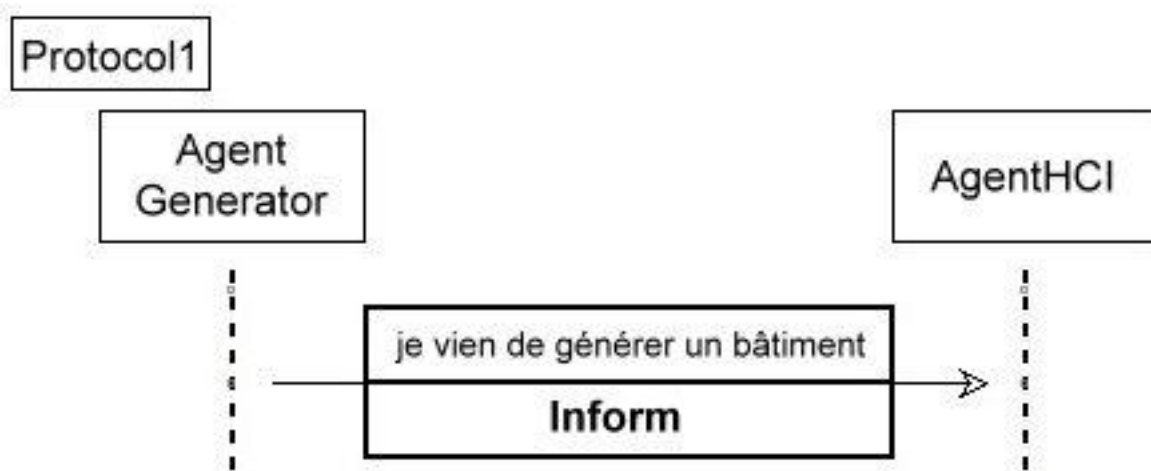


Figure 1 : Protocole 1

L'agent générateur génère des bâtiments avec des positions et des dimensions aléatoires (les valeurs sont effectivement aléatoires mais ont une valeur maximale fixée). Le protocole 1 est utilisé lors de chaque génération de bâtiment. A la réception d'un message l'informant de la génération d'un nouveau

bâtiment, l'agent IHM calcul les bâtiments qui sont potentiellement en conflit avec le nouveau bâtiment généré et les met dans sa liste. Une fois tous les bâtiments générés, les agents correspondant à ces bâtiments sont lancés.

Protocole 2

Lorsqu'un agent détecte que le bâtiment lui correspondant est en conflit avec un autre bâtiment qui est visible et dont l'agent correspondant est actif, il envoie un message à ce dernier l'informant que les bâtiments qui leur correspondent sont en conflit.

Lorsqu'un agent reçoit un message l'informant qu'il est en conflit (par abus de langage) avec un autre agent, il vérifie la possibilité de décalage entre eux (i.e. si en se décalant d'une distance non supérieure à la valeur maximale fixée, ils ne seront plus en conflit). Si c'est le cas, alors il lui propose de se pousser (et si possible et nécessaire le fait lui aussi ; dans la direction inverse). Si cela n'est pas le cas alors il lui propose de sortir de l'affichage.

Lorsqu'un agent reçoit une proposition de se pousser et s'il voit qu'en le faisant il n'entre pas en conflit avec plus d'agents qu'il ne l'était déjà, alors il accepte la proposition et se décale de la valeur demandée (sauf si cette valeur le contraint à dépasser la valeur maximale de décalage autorisé ; dans ce cas il ne se décalera que jusqu'à ce qu'il atteigne cette dernière) sinon il la rejette.

Lorsqu'un agent reçoit un rejet de proposition de se décaler, il calcul s'il peut le faire lui. Si c'est le cas, alors il le fait. Si cela n'est pas le cas, alors il lui demande de sortir de l'affichage.

Lorsqu'un agent reçoit une proposition de sortir de l'affichage, il demande à l'émetteur de cette proposition de lui indiquer le nombre de bâtiments avec lequel il est en conflit ainsi que sa surface. Lorsqu'il reçoit ces informations en retour, il compare le nombre de bâtiments en conflit entre lui et l'émetteur de la proposition. Si sa valeur dépasse celle de l'agent émetteur de la proposition alors il accepte la proposition et sort de l'affichage. Si les 2 valeurs précédentes sont égales alors il fera la comparaison entre 2 nouvelles valeurs calculées en divisant pour chacun la surface du bâtiment sur le logarithme du nombre de bâtiment en conflit ; et si sa valeur est inférieure ou égale à celle de l'agent émetteur, alors il fera de même que décrit précédemment. Dans tous les autres cas il rejettera la proposition.

Lorsqu'un agent reçoit une acceptation de quelque nature que ce soit, le protocole est terminé.

Lorsqu'un agent reçoit un message de refus de sortir de l'affichage alors c'est lui qui sort de l'affichage.

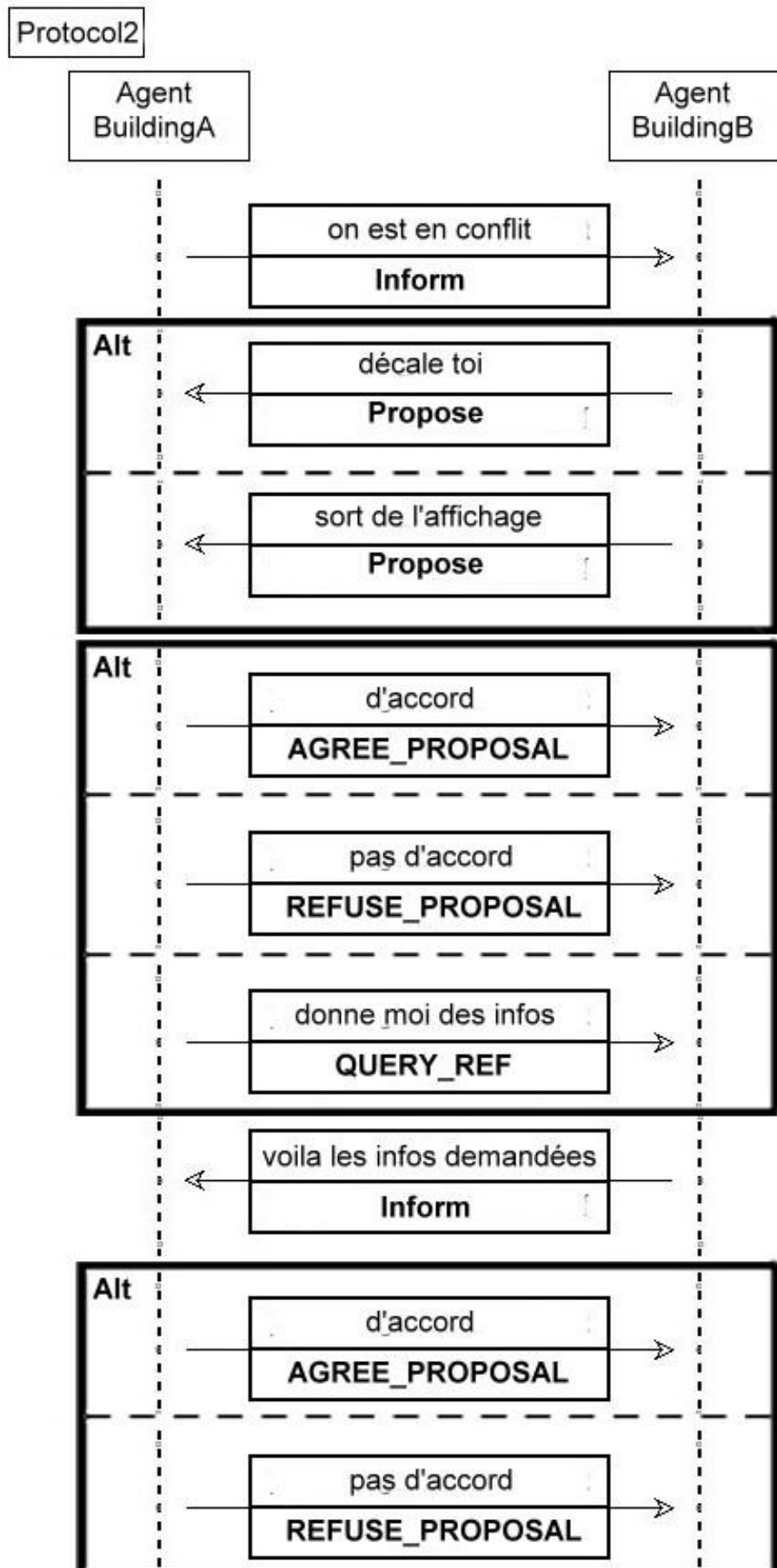


Figure 2 : Protocole 2

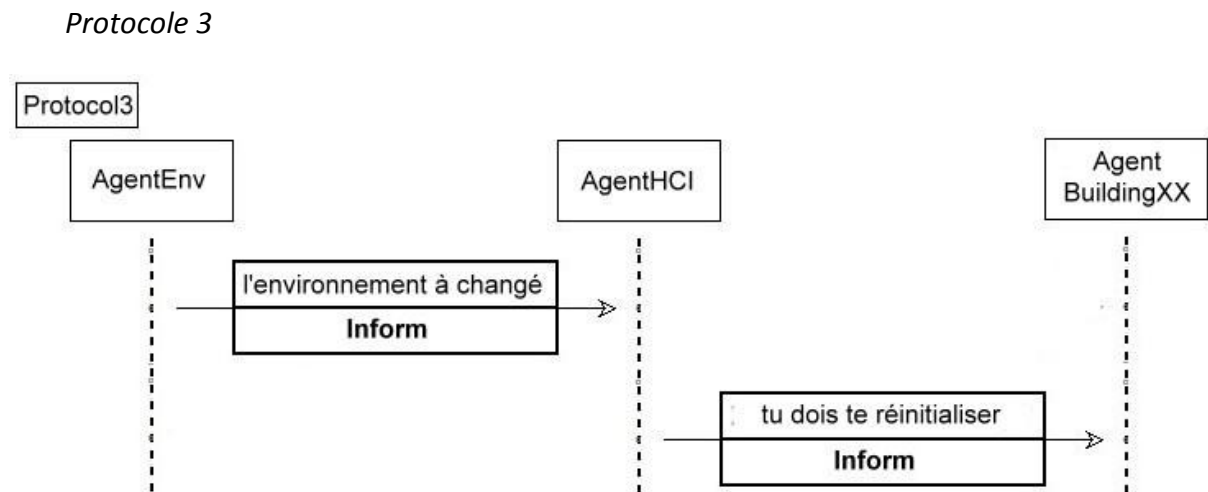


Figure 3 : Protocole 3

Lors d'un zoom ou d'un déplacement, un agent (AgentEnv) envoie un message à l'agent IHM l'informant de la modification de l'environnement. Ce dernier envoie un message de réinitialisation à tous les agents building qui font partie du nouvel environnement (l'ensemble des agents peuvent être concernés comme il se peut que seulement un sous-ensemble ne le soit ; par exemple le zoom en avant) et désactive les autres.

Mécanismes

Pour pallier aux problèmes obtenus dans certains cas, quelques mécanismes ont été employés.

Le premier est utilisé pour éviter la redondance de messages ; i.e. lorsqu'un agent envoie un message informant un autre agent qu'ils sont en conflit, on ne doit plus retrouver ce message après (sauf si le protocole est terminé ; dans ce cas le protocole pourra être relancé). Pour cela, l'utilisation d'un booléen qui indique si un agent est en cours de coopération avec un autre (et vice versa) suffit.

Un second mécanisme est celui de vérifier que lorsque l'on est non affiché, que l'on est plus en conflit avec les autres agents (affichés) alors le bâtiment pourra être affiché de nouveau (cela arrive dans certains cas où par exemple l'agent qui nous a demandé à sortir de l'affichage est lui aussi sorti de l'affichage sous la demande d'un tiers agent).

Conclusion

Pour faire coopérer des agents, des protocoles doivent être mis en place pour assurer de bon fonctionnement du système. Néanmoins, des mécanismes peuvent être requis pour palier à certains problèmes (exemple du second mécanisme) dont il serait assez difficile de les mettre en place sous forme de protocoles.