

Correction du TP N°6 : Implémentation de l'algorithme de Ricart et Agrawala en utilisant la plateforme JADE

Classe site :

```
import java.util.ArrayList;
import jade.core.AID;
import jade.core.Agent;
import jade.core.behaviours.*;
import jade.lang.acl.ACLMessage;

public class site extends Agent{
    int h, last, id;
    boolean priorité;
    String etat;
    ArrayList Ri = new ArrayList(),
    attendu = new ArrayList(),
    differe = new ArrayList();

    public void setup(){
        System.out.println("Agent " + this.getLocalName());

        Object [] args = this.getArguments(); //récupération des arguments
        if (args != null){

            id = Integer.parseInt(args[0].toString()); //récupérer l'identité

            for( int i = 1; i < args.length; i++) // récupérer l'ensemble Ri
                Ri.add(args[i].toString());
            //System.out.println("Agent " + this.getLocalName() + " id " + id + " Ri " + Ri.toString());
        }

        this.addBehaviour(new Dehors()); //rajouter le 1er comportement
    } //setup

    public class Dehors extends Behaviour{
        public void action(){
            etat = "dehors";
            System.out.println("Agent " + getLocalName() + " je suis dans l'état dehors");
            block((int) (Math.random() * 1000));
            consulterBoite();
        }
        public boolean done(){
            int val = (int) (Math.random() * 2); //generer une variable aléatoire qui prend soit la valeur 0 soit la valeur 1
            if (val == 0) return false; //je reste dans l'état dehors
            else { //(val == 1) //je vais devenir demandeur
                addBehaviour(new DevenirDemandeur()); //on rajoute ce comportement pour devenir dans l'état demandeur
                return true;
            }
        }
    }
}
```

```

public class DevenirDemandeur extends OneShotBehaviour{
    public void action(){
        etat = "demandeur";
        h++;
        last = h;

        //attendu = Ri;
        for( int i = 0; i < Ri.size(); i++){

            attendu.add(Ri.get(i)); // copier Ri dans attendu

            ACLMessage msgEnvoi = new ACLMessage (ACLMessage.INFORM); //envoyer une requete à tous les sites qui sont dans Ri
            msgEnvoi.addReceiver(new AID(Ri.get(i).toString(),AID.ISLOCALNAME));
            msgEnvoi.setContent("requete"+"&"+last+"&"+id);
            send(msgEnvoi);
        }

        System.out.println("Agent "+getLocalName()+"je suis dans l'état demandeur,j'ai envoyé le message:"+ "requete"+"&"+last+"&"+id);
        block((int) (Math.random() * 1000));

        addBehaviour(new AttendreVecteurAttenduDevientVide()); //on rajoute ce comportement pour attendre
                                                                    //que la taille du vecteur attendu == 0

        consulterBoite(); //pour récupérer des requêtes/permissions envoyées par les autres sites
    }
}

public class AttendreVecteurAttenduDevientVide extends Behaviour{
    public void action(){

        System.out.println("Agent " + getLocalName()+ " je teste la taille du vecteur attendu : "+ attendu.size());
        block((int) (Math.random() * 1000));
        consulterBoite(); //pour récupérer des requêtes/permissions envoyées par les autres sites

    }
    public boolean done(){

        if (attendu.size()==0) { //on a reçu toutes les permissions
            etat = "dedans";
            addBehaviour(new EnSC()); //on appelle le comportement EnSC pour entrer en Section Critique
            return true;
        }
        else{ //attendu.size()!=0
            return false; // on refait l'exécution de la méthode action du comportement AttendreVecteurAttenduDevientVide
                          //pour tester une autre fois le vecteur attendu
        }
    }
}

public class EnSC extends OneShotBehaviour {
    public void action(){
        for (int i = 0; i < 5; i++)
            System.out.println("***** Agent " + getLocalName() + " je suis en SC");

        block((int) (Math.random() * 1000));
        addBehaviour(new Liberer()); //on rajoute ce comportement pour libérer la Section Critique
        consulterBoite(); //pour récupérer des requêtes/permissions envoyées par les autres sites
    }
}

```

```

public class Liberer extends OneShotBehaviour {
    public void action(){
        etat = "dehors";

        for (int i = 0 ; i < differe.size(); i++){ //envoyer une permission aux sites dont on a différé l'envoi de permission
            ACLMessage msgEnvoi = new ACLMessage (ACLMessage.INFORM);
            msgEnvoi.addReceiver(new AID(differe.get(i).toString(),AID.ISLOCALNAME));
            msgEnvoi.setContent("permission"+"&"+"id");
            send(msgEnvoi);
        }
        differe = new ArrayList();//vider le vecteur differe

        System.out.println("Agent " + getLocalName() + " je libère la SC ");
        addBehaviour(new Dehors());
        consulterBoite(); //pour récupérer des requêtes/permissions envoyées par les autres sites
    }
}

public void consulterBoite(){
    ACLMessage msgRecu = receive();

    if (msgRecu != null){
        String msgContenu = msgRecu.getContent();

        System.out.println("----- Agent " + getLocalName() + " j'ai reçu " + msgContenu + " de la part de " + msgRecu.getSender().getLocalName());

        String [] DiviserMsgContenu = msgContenu.split("&"); //Diviser le message pour extraire les informations
        String TypeMessage = DiviserMsgContenu[0]; //récupérer le type du message c-à-d requete ou permission

        if (TypeMessage.equals("requete")){ // traitement de la requete reçue

            int k, j;

            k = Integer.parseInt(DiviserMsgContenu[1]); // récupérer l'horloge de la requete reçue
            j = Integer.parseInt(DiviserMsgContenu[2]); //récupérer l'identité du site qui a envoyé la requete
            h = Math.max(h, k); //maximiser l'horloge

            priorité = ((etat.equals("dedans"))||((etat.equals("demandeur"))&&((last<k)||((last==k)&&(id<j))))); //calcul de la priorité
            //System.out.println("+++++++ Agent " + getLocalName() + " priorité " + priorité);

            if (priorité){ // si je suis prioritaire ---> je diffère (retarde) l'envoi de permission

                differe.add(msgRecu.getSender().getLocalName()); //on sauvegarde le nom du site qui a envoyé la requete dans
                //le vecteur differe
            }
            else{ //// si je ne suis pas prioritaire ---> j'envoi ma permission
                ACLMessage msgEnvoi = new ACLMessage (ACLMessage.INFORM);
                msgEnvoi.addReceiver(msgRecu.getSender());
                msgEnvoi.setContent("permission"+"&"+"id");
                send(msgEnvoi);
            }
        } //requete
        else{
            if (TypeMessage.equals("permission")){ // traitement de la permission reçue

                attendu.remove(msgRecu.getSender().getLocalName()); // on supprime du vecteur attendu le nom du site qui
                //a envoyé la permission
            } //if (TypeMessage.equals("permission"))
        } //else
    } //if (msgRecu != null){
} // méthode consulterBoite
} // class site

```

Classe test :

```
public class test {  
    public static void main(String[] args) {  
        String [] commande = new String[3];  
        String argument = "";  
  
        argument = argument+ "a:site(1, b, c)";  
        argument = argument+";b:site(2, a, c)";  
        argument = argument+";c:site(3, a, b)";  
  
        commande [0]="-cp";  
        commande [1]="jade.boot";  
        commande [2]= argument;  
        jade.Boot.main(commande);  
    }  
}
```