



Universitat de Girona

Departament d'Electrònica, Informàtica i Automàtica

**Assignatura:
Disseny de sistemes de supervisió**

**Pràctica 7:
“SISTEMES MULTIAGENT”**

Pràctica 7:
“SISTEMES MULTIAGENT”

Objectius de la pràctica:

- Comprensió dels conceptes bàsics de sistemes multiagents
- Utilització de la plataforma de sistemes multiagent JADE
- Disseny i desenvolupament d'una aplicació senzilla basada en la comunicació entre diferents agents.
- Entendre quina és l'aplicabilitat dels sistemes multiagents.

Contingut:

INTRODUCCIÓ.....	3
JADE	3
JADE: INTERFÍCIE GRÀFICA D'USUARI	4
JADE: PAQUETS.....	4
CONSTRUCCIÓ D'AGENTS	4
<i>Comportaments</i>	5
EXEMPLES.....	5
ENVIAR I REBRE MISSATGES	5
PROTOCOLS FIPA	6
SCHEDULING DE L'AGENDA DE DUES PERSONES	6
EXERCICI	6
PRIMERA PART.	6
SEGONA PART.	7
BIBLIOGRAFIA.....	7

En aquesta pràctica aprendrem a analitzar, descriure i dissenyar entorns en els quals els agents poden operar i interaccionar entre ells de forma efectiva i productiva. L'entorn proporciona la infraestructura per les interaccions a través protocols de comunicació i d'interacció entre agents.

JADE

- 1) Baixar-ho del web de l'escola (alternativament podeu usar la pàgina oficial indicada a la bibliografia)
- 2) Descomprimir el fitxer
- 3) Testejar la instal·lació mitjançant la instrucció
`java -classpath lib\jade.jar;lib\jadeTools.jar;lib\Base64.jar jade.Boot -gui`

JADE Remote Agent Management GUI

File Actions Tools Help

White Pages

⊖ JADE

agent-name	agent-address	agent-type
AGENT-NAME	AGENT-ADDR...	AGENT-TYPE

a) Modificar l'autoexec.bat per incloure la sentència

```
set CLASSPATH=%CLASSPATH%;.;c:\jade\lib\jade.jar;  
c:\jade\lib\jadeTools.jar; c:\jade\lib\Base64.jar
```

(Si JADE l'heu copiat en c:\jade)

- b) Reiniciar l'ordinador
- c) Testejar la instal·lació amb
java jade.Boot -platform -gui

JADE: interfície gràfica d'usuari

En el moment d'engegar Jade:

- Es pot escollir o no d'usar la interfície gràfica d'usuari amb l'opció -gui
- Es poden indicar els agents que interaccionaran
- Es pot indicar una plataforma amb nom o sense nom.

Per exemple, per indicar una plataforma sense nom, cal teclejar la comanda:

```
java jade.boot -platform -gui a:t1 a2:t2
```

On a1 i a2 són els noms dels agents de tipus t1 i t2 respectivament (t1 i t2 són dos classes Java on s'ha codificat els agents).

Una plataforma pot tenir diversos contenidors. Un contenidor pot estar ubicat en un únic ordinador o contenir agents que s'executen en diferents ordinadors. La interfície d'usuari mostrada per Jade està estructurada per containers.

Un agent principal, l'AMS, és l'encarregat de portar el control de tota la aplicació. A més, per desenvolupar una aplicació, a més JADE proporciona un conjunt d'eines que estan empaquetades en agents diferents:

- 1) L'agent RMA (Remote Monitoring Agent): Agent que s'engega amb l'opció -gui i proporciona el control de la interfície gràfica d'usuari.
- 2) L'agent DA (Dummy Agent): Agent que permet monitoritzar la interacció dels agents i ajudar en la detecció de fallides.
- 3) L'agent DF (Directory Facilitator), que permet tractar amb una interfície còmoda el control dels agents enregistrats.
- 4) L'agent sniffer que permet interceptar els missatges i veure'ls en una interfície usant una notació pròxima als diagrames d'UML.

Els agents AMS, DF i RMA es generen de forma automàtica per a cada contenidor. Els agents DA i sniffers es poden escollir segons les necessitats.

JADE: paquets

JADE proporciona un conjunt de paquets JAVA que permeten crear els agents.

- jade.core: nucli del sistema. Proporciona la classe Agent
- jade.core.behaviours: conté la jerarquia de comportaments, i penja de la classe Behaviour.
- jade.domain: elements bàsics de gestió del sistema (p.e. AMS; DF, ACC)
- jade.gui: eines per construir interfícies gràfiques
- jade.lang.acl: manipulació del llenguatge de comunicació entre agents. Té la classe ACLMessage
- jade.proto: control de protocols, com per exemple, contract-net

Construcció d'agents

Per construir un agent cal fer una subclasse de la classe Agent i instanciar-la

```
import jade.core.*;  
public class meuAgent extends Agent {}  
java jade.Boot -platform -gui a:meuAgent
```

Hi ha dos mètodes obligatoris a especificar en cada agent

Setup(): procediment d'inicialització. En ell cal indicar quins seran els comportaments de l'agent mitjançant addBehaviour();

TakeDown(): procediment d'eliminació de l'agent.

Comportaments

L'activitat de l'agent ha d'estar implementada en els seus comportaments. Cada agent té un scheduler intern que va fent que s'executin tots els comportaments de l'agent.

Els comportaments es programen mitjançant alguna de les classes de la jerarquia Behaviours. Hi ha un mètode associat a cada comportament, anomenat action(), que és el que conté les instruccions concretes associades al comportament.

Jerarquia de comportaments:

- SimpleBehaviour: comportament simple que es pot executar sense interrupcions. Té dues subclasses:
 - OneShotBehaviour: que s'executa només un cop
 - CyclicBehaviour: que es va executant de forma cíclica
- ReceiverBehaviour: comportament que serveix per rebre missatges. Aquest comportament es bloqueja fins que hi ha un missatge disponible, sense aturar a tot l'agent.
- ComplexBehaviour: comportaments complexos compostos de diferents fills. Es pot interrompre entre l'execució de dos fills. Pot ser de dos tipus:
 - SequentialBehaviour: executa els fills seqüencialment
 - NonDeterministicBehaviour: executa els fills de manera no-determinista

Exemples

Es tracta de que executeu els exemples proporcionats per Jade, entenent el funcionament de la plataforma per poder, després, realitzar una implementació.

Abans d'executar els exemples, caldrà que els compileu.

S'han d'executar des del directori jade/src, i, com estan estructurats per paquets, cal fer ús de noms qualificats.

Treballeu bàsicament tres exemples:

- 1) Enviar i rebre missatges.
- 2) Protocols FIPA.
- 3) Scheduling de l'agenda de dues persones

Es tracta que primer executeu els exemples, usant totes les eines que proporciona la plataforma per monitoritzar la comunicació i finalment visualitzeu el codi tractant d'entendre com estan implementats els agents.

Enviar i rebre missatges

Director: jade/src/examples/receivers

Paquet: examples.receivers.*

Aquest exemple mostra els mètodes que pot usar un agent per rebre un missatge. Està basat en dos agents:

- AgentSender: envia missatges INFORM
- AgentReceiver: reb els missatges

Executeu en primer lloc l'exemple sense GUI i després en GUI.

En l'execució amb GUI:

- 1) Primer engegueu la plataforma sense cap agent
- 2) Engegueu un sniffer
- 3) Engegueu l'agent receptor, i activeu-lo en l'sniffer
- 4) Engegueu l'agent emissor i activeu-lo en l'sniffer

Analitzeu els missatges enviat pels agents usant el diagrama de l'sniffer:

- Si posicioneu el ratolí sobre una fletxa, podreu llegir en la barra d'estats el tipus de missatge enviat
- Si feu un doble-click amb el ratolí, podeu veure el missatge complet.

Protocols FIPA

Director: jade/src/examples/protocols

Paquet: examples.protocols.*

En aquest exemple s'usen les preformatives query i request de FIPA. També hi ha dos agents:

- AgRequestInitiator: és el que inicia la conversació mitjançant un request.
- AgRequestResponder: és el que respon als requests.

Com en darrer cas de l'exemple anterior, engegueu primer la plataforma i després als agents, usant un sniffer per visualitzar els missatges enviats.

Scheduling de l'agenda de dues persones

Director: jade/src/demo/MeetingScheduling

Per executar-lo cal usar run.bat des del directori indicat. També hi ha un fitxer compile.bat per ajudar a la compilació.

En aquest exemple un agent ajuda a un usuari en la planificació de les reunions o cites amb altres persones.

Quan s'inicia un agent, cal posar-li un nom (login) i un password. A continuació es visualitza un calendari sobre el que es poden establir compromisos per l'agent (Appointments → fix).

Si hi ha dos agents, es poden fer cites conjunes. Primer cal que l'agent conegui quines persones hi ha connectades, per la qual cosa cal seleccionar "Update known persons with the facilitator".

Us animeu a provar l'aplicació des de dos PC diferents? Mireu el README de Jade per trobar exemples de com executar un agent des d'un altre màquina.

(Nota: potser no funcionarà si no teniu Regional Setting del PC posat en English).

Exercici

Primera part.

Es tracta de desenvolupar dos agents.

- Agent brosa:
 - Genera de forma aleatòria una brosa en un tauler de dimensions nxn.
 - Envia una petició perquè els robots li recullin la brosa.

- Selecciona el robot que estigui més a prop de la brossa.
- Agent robot:
 - Inicialment està en una posició aleatòria i espera algun missatge per tenir feina, és a dir, per recollir una brossa.
 - Quan rep una petició, calcula la distància des de la seva posició actual a la brossa, i és el bid amb que respon a la petició.
 - Si la seva proposta és acceptada, es desplaça fins a la posició on està la brossa i la destrueix.

Segona part.

Imagina't un domini de dues dimensions que consisteix en paquets i destinacions. En aquest domini, els robots poden moure els paquets fins a la seva destinació. Els robots només poden moure un paquet en cada moment, i no està permès caminar per sobre d'un paquet (és a dir, si un robot es troba un paquet al davant, l'ha de rodejar). Hi ha un cost associat al moviment del paquet. Un robot rep una recompensa per moure paquets cap a posicions més pròximes al seu destí.

Implementa els robots i respon a les qüestions següents:

- a) És possible que els robots desenvolupin una convencions per decidir la direcció dels paquets que són un obstacle en el seu moviment.
- b) Es podran formar camins sense obstacles (paquets com a obstacles)?
- c) Els punts de destinació es poden congestionar si tots els robots van acostant els paquets cap a ells. Podria ser que els robots s'especialitzessin en la seva tasca de forma que hi hagués robots que portessin els paquets a prop de la seva destinació i altres que acabessin la feina?
- d) Poden prendre avantatge els robots de les intencions dels altres?

Suggeriments: escollir una graella de $N \times N$, P paquets, R robots i D destins, amb per exemple: $N=1000$, $P=50$, $R=8$, $D=3$. Assumir que cada robot i cada paquet ocupen una cel·la de la graella, i que un robot es pot moure en qualsevol de les 8 posicions que li envolten en cada pas.

Aquest exercici correspon a l'exercici 11 de [Weiss].

Bibliografia

<http://www.fipa.org>

<http://sharon.cselt.it/projects/jade/>

Weiss, 1999

Multiagent Systems. A modern approach to Distributed Artificial Intelligence.

Ed. Gerhard Weiss

MIT Press, 1999

ISBN 0-262-23203-0