



Laboratorio 1 Comunicación

Técnicas Avanzadas de Inteligencia Artificial

Dpt. Lenguajes y Sistemas Informáticos.

FISS. UPV-EHU



ÍNDICE

2. Comunicación-Comportamientos

2.1. Comunicación Ping-Pong

2.2. Comunicación filtrada

Recordar

- **Plataforma: Windows/Linux**
- **Con/sin Eclipse**
- **Versión: 4.3.0**
- **CLASSPATH**
- **Linux:**
 - **dos2unix**
 - **iconv**

2.1. Comunicación Agente Ping

Actividades a realizar:

- Activar Eclipse y Abrir workspace C:\LAB_TAIA
- Abrir proyecto TAIA
- Crear package **examples.PingAgent**
- Importar: **PingAgentReducido y EnviarPing.**
- Hay dos ejecuciones:
 - 3.PingAgentReducido.bat**
 - 4.EnviarPing.bat**

2.1.1. PingAgentReducido

```
public class PingAgentReducido extends Agent {  
  
    // Clase que describe el comportamiento que permite recibir un mensaje  
    // y contestarlo  
    class WaitPingAndReplyBehaviour extends SimpleBehaviour {  
        ...  
        public void action() {  
            ...  
        }  
        public boolean done() {  
            return finished;  
        }  
    } //Fin de la clase WaitPingAndReplyBehaviour  
  
    protected void setup() {  
        System.out.println("Inicializacion AgentePingReducido");  
        WaitPingAndReplyBehaviour PingBehaviour = new  
            WaitPingAndReplyBehaviour(this);  
        addBehaviour(PingBehaviour);  
    }  
}
```

2.1.1. PingAgentReducido

```
public void action() {  
    ACLMessage msg = receive();  
    if (msg != null){  
        if (msg.getPerformative() == ACLMessage.NOT_UNDERSTOOD) {  
            System.out.println("Mensaje NOT UNDERSTOOD recibido"); }  
        else {  
            ACLMessage reply = msg.createReply();  
            if (msg.getPerformative()== ACLMessage.QUERY_REF){  
                String content = msg.getContent();  
                if ((content != null) && (content.indexOf("ping") != -1)){  
                    System.out.println("Recibido mensaje QUERY-REF con contenido ping");  
                    reply.setPerformative(ACLMessage.INFORM);  
                    reply.setContent("pong");  
                }  
                else {  
                    System.out.println("Recibido mensaje QUERY-REF con contenido desconocido");  
                    reply.setPerformative(ACLMessage.NOT_UNDERSTOOD);  
                    reply.setContent("( UnexpectedContent (expected ping))");  
                }  
            }  
        }  
    }  
}
```

2.1.1. PingAgentReducido

```
public void action() {  
    ...  
    // Recibido un QUERY REF con contenido correcto  
    // Recibido un QUERY REF con contenido incorrecto  
    else {  
        System.out.println("Recibido mensaje no QUERY-REF");  
        reply.setPerformative(ACLMessage.NOT_UNDERSTOOD);  
        reply.setContent("( Unexpected-act "+ACLMessage.getPerformative  
            (msg.getPerformative())+" ) ( expected (query-ref :content ping)))");  
    }  
    send(reply);  
}  
}  
public boolean done() {  
    return finished;  
}  
} //Fin de la clase WaitPingAndReplyBehaviour  
...
```

2.1.1. PingAgentReducido

- **Métodos asociados a Agent:**
 - **receive()** → `ACLMessage msg = receive();`
 - **send()** → `send(reply);`
- **Métodos asociados a ACLMessage:**
 - **Crear respuesta:**
 - **createReply()**- `ACLMessage reply = msg.createReply();`
 - **Asignar valores:**
 - **setPerformative**- `msg.setPerformative(ACLMessage.QUERY_REF);`
 - **setContent**
 - **Obtener Valores:**
 - **getContent()**
 - **getPerformative()**
 - **Performativas:** `ACLMessage.NOT_UNDERSTOOD` ,
`ACLMessage.QUERY_REF`, `ACLMessage.INFORM`, ...

2.1.1. PingAgentReducido

1ª ejecución

- Crear interfaz del ejecutable: **PingAgentReducido**
- Main class: **jade.Boot**
- Arguments (copiar del archivo
3.PingAgentReducido.bat): **-gui**
Ping:examples.PingAgent.PingAgentReducido
- Ejecutar: **Run**

2.1.1. PingAgentReducido

The screenshot shows an IDE window with several tabs: `TimeAgent.java`, `ComplexBehaviourAgen`, `TimeAgent.java`, and `PingAgentReducido.ja`. The console window displays the following output:

```
PingAgent [Java Application] C:\Archiv...
23-feb-2009 14:27:40 je
INFO: -----
This is JADE 3.5 -
downloaded in Open
at http://jade.tilb
-----
23-feb-2009 14:27:41 je
INFO: Service jade.core
23-feb-2009 14:27:41 je
INFO: Service jade.core
23-feb-2009 14:27:41 je
INFO: Service jade.core
23-feb-2009 14:27:41 je
INFO: Service jade.core
23-feb-2009 14:27:41 je
INFO: Service jade.core
23-feb-2009 14:27:41 je
INFO: Clearing cache
23-feb-2009 14:27:41 je
INFO: HTTP-MTP Using XM
23-feb-2009 14:27:41 je
INFO: MTP addresses:
http://G003790.infor.or
23-feb-2009 14:27:41 je
INFO: -----
Agent container Main-Container@G003790 is ready.
-----
Inicializacion AgentePingReducido
```

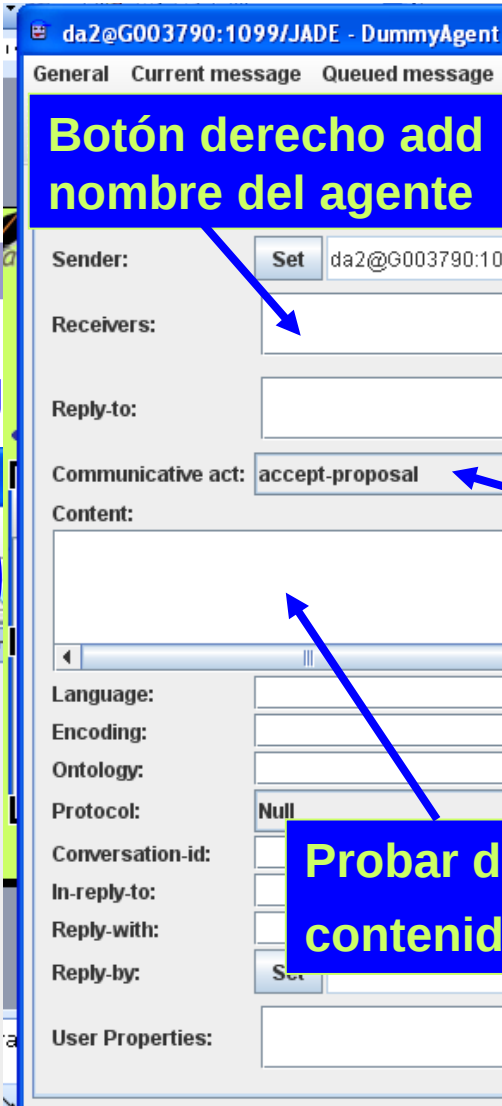
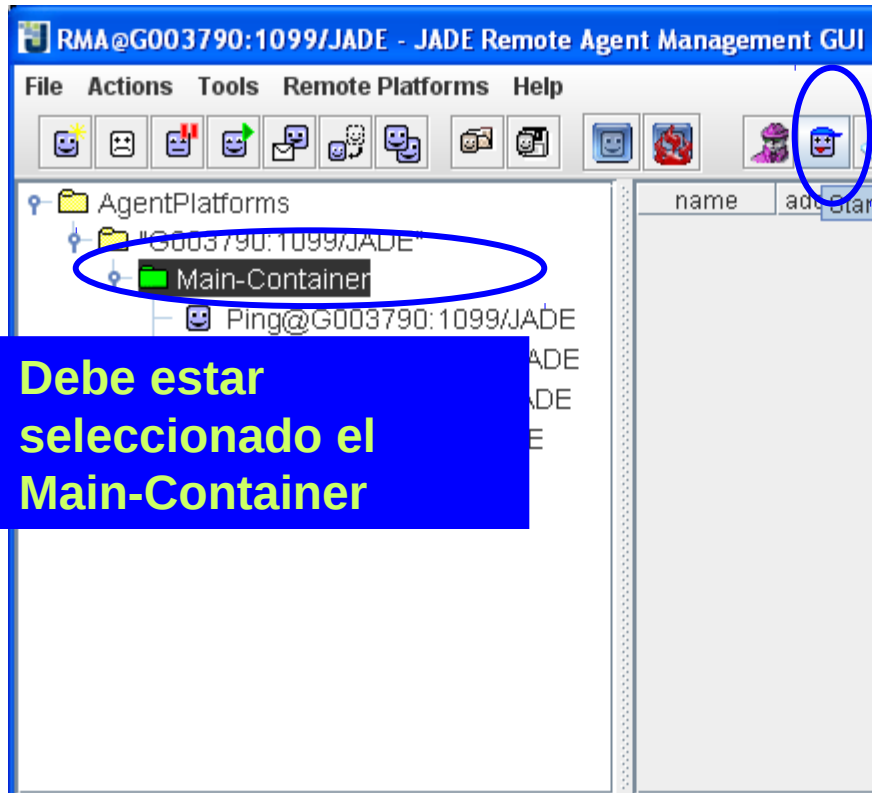
Overlaid on the IDE is the **JADE Remote Agent Management GUI** window. It features a menu bar with `File`, `Actions`, `Tools`, `Remote Platforms`, and `Help`. The main area shows a tree view of `AgentPlatforms` with a sub-entry for `"G003790:1099/JADE"` containing a `Main-Container` with four agents: `Ping@G003790:1099/JADE`, `RMA@G003790:1099/JADE`, `ams@G003790:1099/JADE`, and `df@G003790:1099/JADE`. To the right of the tree is a table with the following structure:

name	addresses	state	owner
NAME	ADDRES...	STATE	OWNER

2.1.1.

Actividad

- Start Dummy Agent, ag



General Current message Queued message



ACLMessage Envelope

ACLMessage Envelope

Sender: Vi... g@192.168.1.33:1099/JADE

Receivers: da0@192.168.1.33:1099/JADE

Reply-to:

Communicativ... inform

Content:
pong

Language:

Encoding:

Ontology:

Protocol: Null

Conversation-...

In-reply-to:

Reply-with: 8.1.33:1099/JADE1382304526433

Reply-by: Vi...

User Properti...

OK

- 10/20/13 11:28 PM: INFORM
- 10/20/13 11:28 PM: QUERY-REF
- 10/20/13 11:27 PM: NOT-UNDERSTOOD
- 10/20/13 11:27 PM: REQUEST
- 10/20/13 11:27 PM: NOT-UNDERSTOOD
- 10/20/13 11:27 PM: QUERY-REF

Seleccionar mensaje

2.1.2. EnviarPing

```
public class EnviarPing extends Agent {
```

```
    class EnviarMensajeBehaviour extends SimpleBehaviour {
```

```
        public void action() {  
            ...  
        }
```

```
        public boolean done() {  
            return finished;  
        }
```

```
    } // Fin de la clase EnviarMensajeBehaviour
```

```
protected void setup() {
```

```
    EnviarMensajeBehaviour EnviarBehaviour = new
```

```
        EnviarMensajeBehaviour(this);
```

```
    addBehaviour(EnviarBehaviour);
```

```
}
```

```
}
```

2.1.2. EnviarPing

```
public void action() {
    try {
        System.out.println("\nIntroduce el nombre del destinatario (e.g. da0@myhost): ");
        BufferedReader buff = new BufferedReader(new
                                                    InputStreamReader(System.in));
        String responder = buff.readLine()+":1099/JADE";
        AID r = new AID();
        r.setName(responder);           // crea el identificador del agente
        ACLMessage msg = new ACLMessage(ACLMessage.QUERY_REF);
        msg.setSender(getAID());
        msg.addReceiver(r);
        msg.setContent("ping");
        System.out.println("\nEnviado mensaje QUERY-REF con contenido ping.");
        send(msg);
    }
    catch (java.io.IOException io)
        {System.out.println(io);
        }
}
```

2.1.2. EnviarPing

2ª ejecución

- Crear interfaz del ejecutable: **EnviarPing**
- Main class: **jade.Boot**
- Arguments (copiar del archivo 4.EnviarPing.bat):
 - gui Ping1:examples.PingAgent.PingAgentReducido
 - Ping2:examples.PingAgent.PingAgentReducido
 - EnvioPing:examples.PingAgent.EnviarPing
- Ejecutar: **Run**

2.1.2. EnviarPing

- Crear
- Main
- Argu
- gu
- Ejecu

The screenshot shows the Eclipse IDE's 'Run' dialog box. The 'Name' field is set to 'EnviarPing'. The 'Program arguments' field contains the following text:

```
-gui Ping1:examples.PingAgent.PingAgentReducido  
Ping2:examples.PingAgent.PingAgentReducido  
EnvioPing:examples.PingAgent.EnvioPing
```

The 'VM arguments' field is empty. A blue arrow points from the 'Program arguments' field to a Notepad window titled '4. EnviarPing.bat - Bloc de notas'. The Notepad window contains the following command:

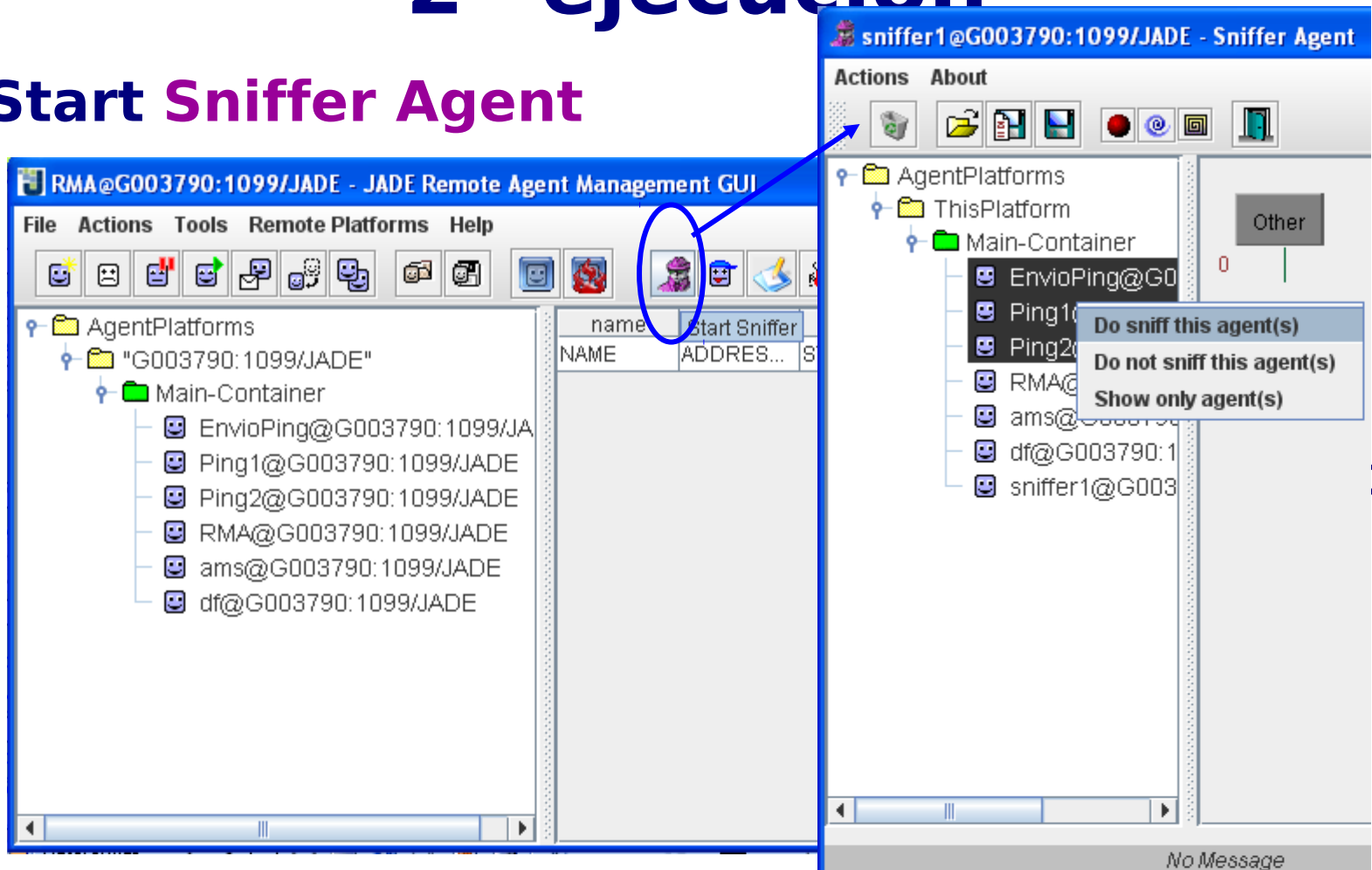
```
java -iade.Boot -gui Ping1:examples.PingAgent.PingAgentReducido  
Ping2:examples.PingAgent.PingAgentReducido  
EnvioPing:examples.PingAgent.EnvioPing
```

The Notepad window also shows a menu bar with 'Archivo', 'Edición', 'Formato', 'Ver', and 'Ayuda'.

2.1.2. EnviarPing

2ª ejecución

- **Start Sniffer Agent**

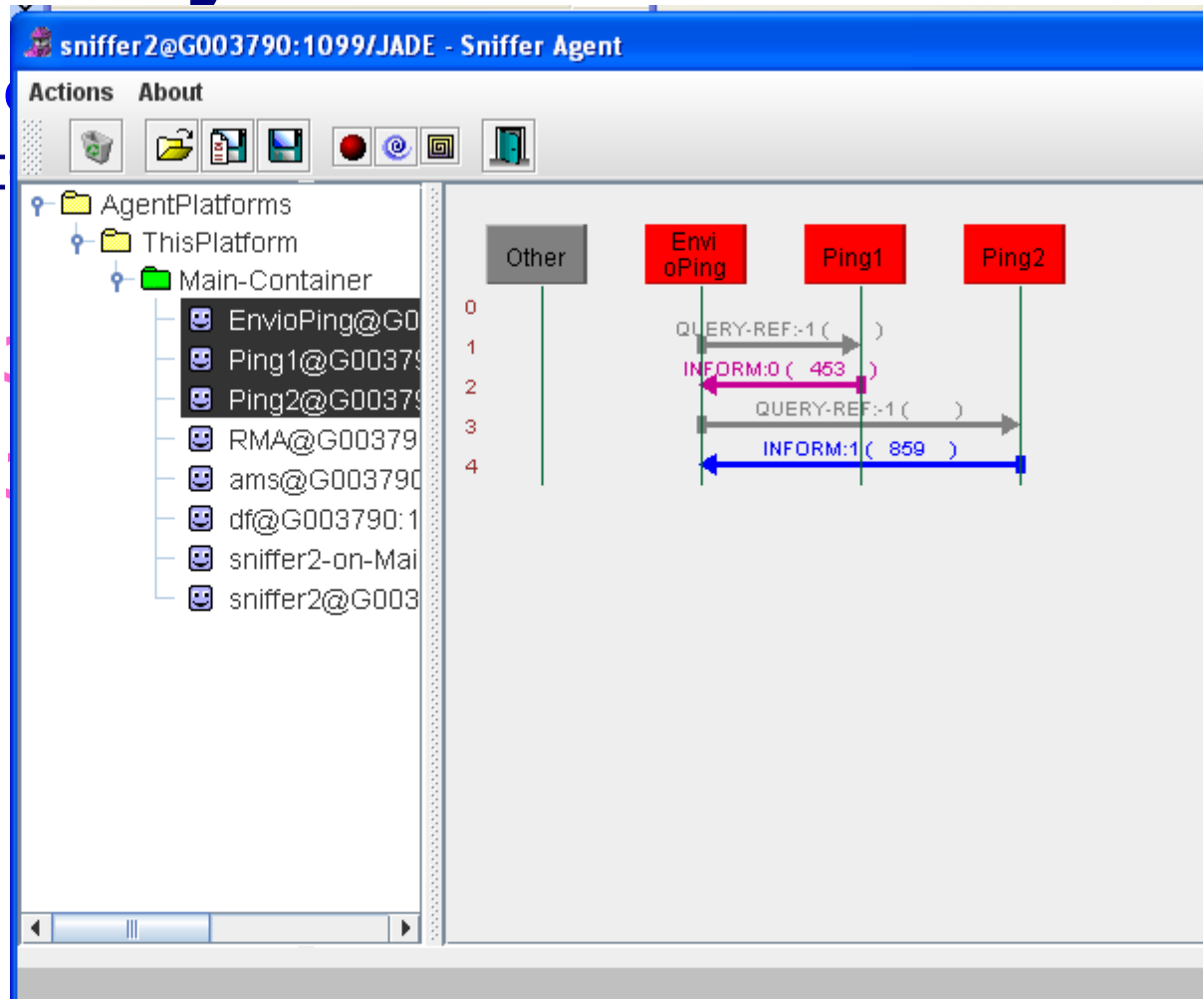


2.1.2. EnviarPing

2ª ejecución

- **Enviar mensaje** nombre del agente mensaje:

- **Ping1@G003790:1099/JADE**
- **Ping2@G003790:1099/JADE**



2.1.2. EnviarPing

• Vi

The screenshot displays the Jade IDE interface. On the left, a text editor shows the source code for the 'EnviarPing' Java application, including comments in Spanish and a URL. The central 'ACL Message' window is open, showing the details of a message sent from 'ioPing@G003790:1099/JADE' to 'Ping1@G003790:1099/JADE'. The message content is 'ping' and the communicative act is 'query-ref'. On the right, a sequence diagram illustrates the interaction between 'EnvioPing' and 'Ping1'. It shows a 'QUERY-REF:-1()' message from EnvioPing to Ping1, followed by an 'INFORM:0(854)' message from Ping1 back to EnvioPing. A second 'QUERY-REF:-1()' message is shown from EnvioPing to Ping2, followed by an 'INFORM:1(840)' message from Ping2 back to EnvioPing.

ACL Message

ACLMessage Envelope

Sender: ioPing@G003790:1099/JADE

Receivers: Ping1@G003790:1099/JADE

Reply-to:

Communicative act: query-ref

Content: ping

Language:

Encoding:

Ontology:

Protocol: Null

Conversation-id:

In-reply-to:

Reply-with:

EnvioPing Ping1 Ping2

QUERY-REF:-1()

INFORM:0(854)

QUERY-REF:-1()

INFORM:1(840)

null)

2.2. Comunicación Messaging

Actividades a realizar

- Crear paquete: **examples.messaging**
- Importar clases: **CustomTemplateAgent.java**
y BlockingReceiveAgent.java
- Hay tres ejercicios:
 - **5.CustomTemplateAgent.bat**
 - **6.BlockingReceiveAgent.bat**
- Crear los tres interfaces para ejecución.

2.2.1. CustomTemplateAgent

```
public class CustomTemplateAgent extends Agent {
```

Filtro de mensajes

```
private MessageTemplate template = MessageTemplate.and(  
    MessageTemplate.MatchPerformative(ACLMessage.REQUEST),  
    new MessageTemplate(new MatchXOntology()));
```

```
protected void setup() {
```

```
    System.out.println("El agente "+getLocalName()+" esta listo.");
```

Activar Dummy e Introspector

```
private class MatchXOntology implements  
    MessageTemplate.MatchExpression {  
  
    public boolean match(ACLMessage msg) {  
        String ontology = msg.getOntology();  
        return (ontology != null && ontology.startsWith("X"));  
    }  
} // END of inner class MatchXOntology+
```

2.2.1. CustomTemplateAgent

ejecución

- Crear interfaz del ejecutable: **Custom**
- Main class: **jade.Boot**
- Arguments (copiar del archivo
5.CustomTemplateAgent.bat): **-gui**
Template:examples.messaging.CustomTemplateAgent
- Ejecutar: **Run**

2.2.1. Custom Template Agent

Start Introspector Agent

- Activar el agente a inspección

The screenshot displays the Introspector GUI with the following components:

- Top Panel:** Shows the file structure: AgentPlatforms > ThisPlatform > Main-Container > Introspector0@G003790:1099/JADE.
- Left Panel:** A tree view of agents under 'Main-Container', including 'Template@G003790:1099/JADE', which is selected.
- Main Window:** Titled 'Template@G003790:1099/JADE', it features:
 - View State Debug:** A sidebar with state indicators (Active, Suspended, Idle, Waiting, Moving, Dead) and 'Change State' buttons (Suspend, Wait, Wake Up, Kill).
 - Incoming Messages:** A panel with 'Pending' and 'Received' tabs. The 'Pending' tab shows 'Incoming Messages -- Pending'.
 - Outgoing Messages:** A panel with 'Pending' and 'Sent' tabs. The 'Pending' tab shows 'Outgoing Messages -- Pending'. A blue arrow points to this area.
 - Behaviours:** A section showing a folder 'Behaviours' containing one item with a lightning bolt icon and the number '1'.
 - Properties:** A table with the following data:

Name:	1
Class:	examples.messaging.CustomTerr
Kind:	CyclicBehaviour

Start Introspector Agent: Visualización de mensajes recibidos

Introspector0@G003790:1099/JADE

File About

- Main-Contair
- Introspec
- Introspec
- RMA@G
- Template
- ams@G
- da0@G

AgentPlatforms.ThisPlatform.Mi

Template@G003790:1099/JADE

View State Debug

Current State

- Active
- Suspended
- Idle
- Waiting
- Moving
- Dead

Change State

- Suspend
- Wait
- Wake Up
- Kill

Incoming Messages

Pending Received

Incoming Messages -- Re

REQUEST

Behaviours

1

Name: 1
Class: examples.messaging.CustomTerr
Kind: CyclicBehaviour

General Current message Queued message



Jade

ACLMessage Envelope

Sender: Set da0@192.168.1.33:1099/JA

Receivers: Template@192.168.1.33:1099/JA

Reply-to:

Communicativ... request

Content:

Hola ...

Language:

Encoding:

Ontology: X-men

Protocol: Null

Conversation...

In-reply-to:

Reply-with:

Reply-by: Set

User Properti...

10/21/13 12:57 AM: REQUEST
10/21/13 12:54 AM: REQUEST
10/21/13 12:54 AM: ACCEPT-PROPOSAL

2.2.1. CustomTemplateAgent

Se ha recibido un mensaje que se identifica con el template diseñado:

(REQUEST

:sender (agent-identifier :name da0@192.168.1.33:1099/JADE)

:receiver (set (agent-identifier :name Template@192.168.1.33:1099/JADE))

:content "Hola ..."

:ontology X-men)

2.2.2. BlockingReceiveAgent

```
public class BlockingReceiveAgent extends Agent {  
  
    protected void setup() {  
        System.out.println("Agente "+getLocalName()+" : esperando un mensaje REQUEST...");  
  
        ACLMessage msg = blockingReceive(  
            MessageTemplate.MatchPerformative(ACLMessage.REQUEST));  
  
        System.out.println("Agente "+getLocalName()+" : he recibido mensaje REQUEST.");  
        System.out.println("Agente "+getLocalName()+" : responde con un mensaje  
            INFORM ...");  
  
        ACLMessage reply = new ACLMessage(ACLMessage.INFORM);  
        reply.addReceiver(msg.getSender());  
        reply.setContent("Finalizado");  
        send(reply);  
        doDelete();  
    }  
}
```

Activar Dummy e Introspector

2.2.2. BlockingReceiveAgent

ejecución

- Crear interfaz del ejecutable: **Blocking**
- Main class: **jade.Boot**
- Arguments (copiar del archivo 6.BlockingAgent.bat):
 - gui
 - Bloqueo:examples.messaging.BlockingReceiveAgent**
- Ejecutar: **Run**

Activar Dummy e Introspector

Introspector0@G003790:1099/JADE

File About

ThisPlatform

- Main-Containr
 - Introspec
 - RMA@G
 - ams@G0
 - da0@G0
 - df@G00

AgentPlatforms.ThisPlatform.M

da0@G003790:1099/JADE - DummyAgent

General Current message Queued message

5/11/09 11:15: INFORM
5/11/09 11:15: REQUEST

ACL Message

ACLMessage Envelope

Sender: View loqueo@G003790:1099/JADE

Receivers: da0@G003790:1099/JADE

Reply-to:

Communicative act: inform

Content: Finalizado

Language:

Encoding:

Ontology:

Protocol: Null

Conversation-id:

In-reply-to:

Reply-with:

Reply-by: View

User Properties:

OK

Resumen de métodos utilizados

- **Métodos asociados a **Agent**:**

- **receive()**- `ACLMessage msg = receive();`
- **receive(template)**
- **blockingReceive(template)**
- **send()**- `send(reply);`

Obtiene el
identificador del
agente: **getAID()**

- **Métodos asociados a **ACLMessage**:**

- **Crear respuesta:**
 - **createReply()**- `ACLMessage reply = msg.createReply();`
- **Asignar valores:**
 - **addReceiver**
 - **setSender**
 - **setPerformative**- `msg.setPerformative(ACLMessage.QUERY_REF);`
 - **setContent**
- **Obtener Valores:**
 - **getContent()**
 - **getPerformative()**

- **Performativas:**
`ACLMessage.NOT_UNDERSTOOD ,`
`ACLMessage.QUERY_REF,`
`ACLMessage.INFORM, ...`