



Introducción a JADE

Java Agent DEvelopment Framework

Jade Tutorials, <http://jade.tilab.com/doc/index.html>
Agentes Inteligentes: JADE. J.F. Garamendi, Curso de doctorado URJC, 2004
Introducción a JADE. Curso de doctorado Sistemas Multiagente UPV

1. Introducción

2. Entorno JADE

**RMA, Dummy, Sniffer, DF GUI,
Introspector**

Iniciar JADE

3. Estructura de un AGENTE

Comportamientos

4. Ejemplo

1. Introducción

JADE = herramienta de desarrollo de Sistemas Multiagente (TILAB)

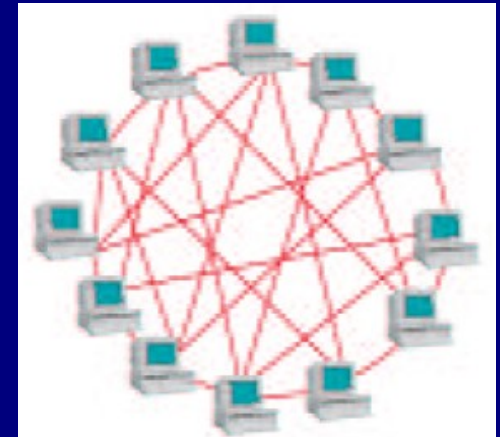
<http://jade.tilab.com/doc/index.html>

Compuesto por:

- Entorno de ejecución
- Librerías JAVA para implementar agentes
- Herramientas gráficas

Características:

- Modelo *peer-to-peer* P2P
- Interoperabilidad: basado en el estándar FIPA
(Foundation for Intelligent Physical Agents)
- Portabilidad
- Intuitiva

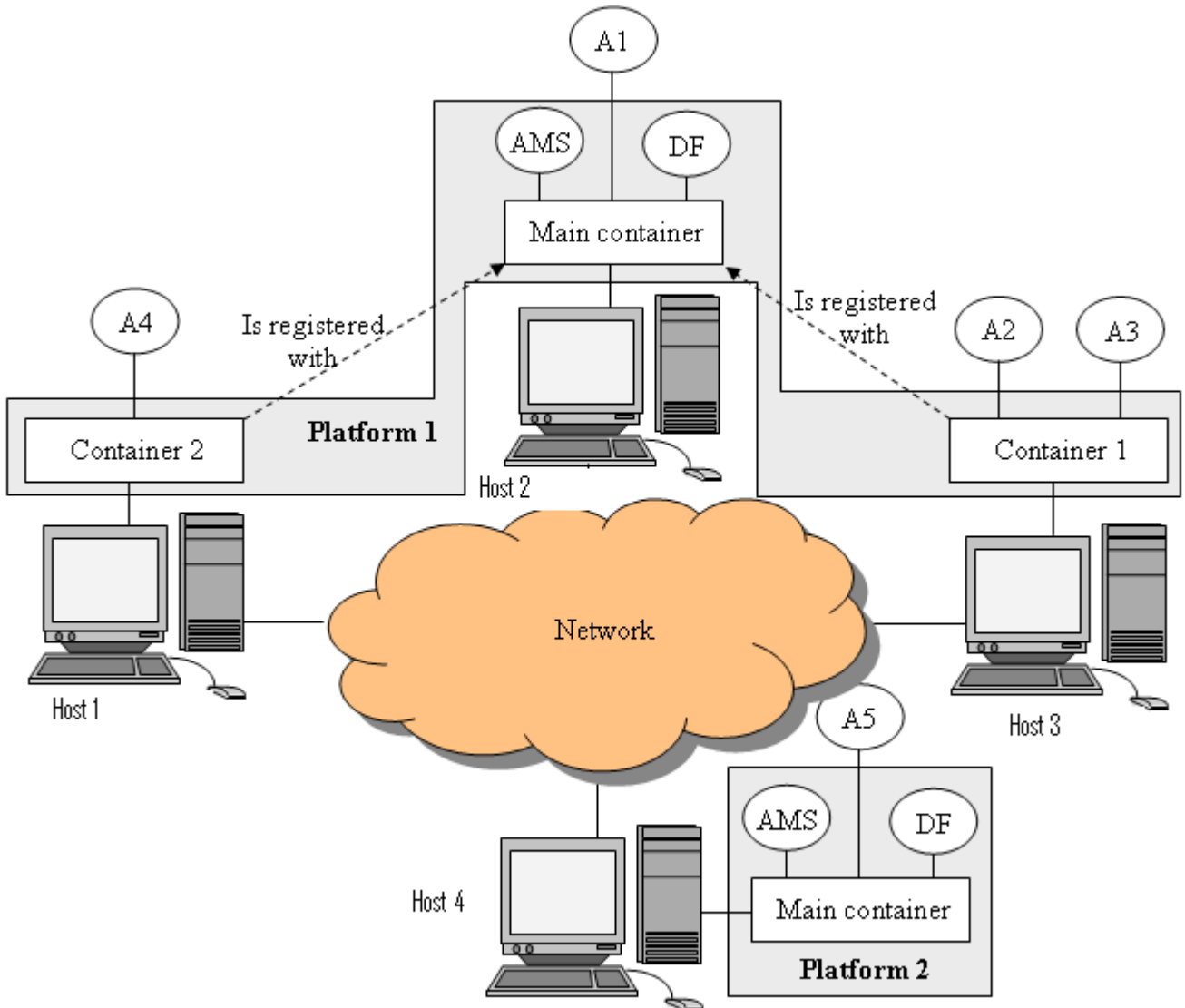


Características

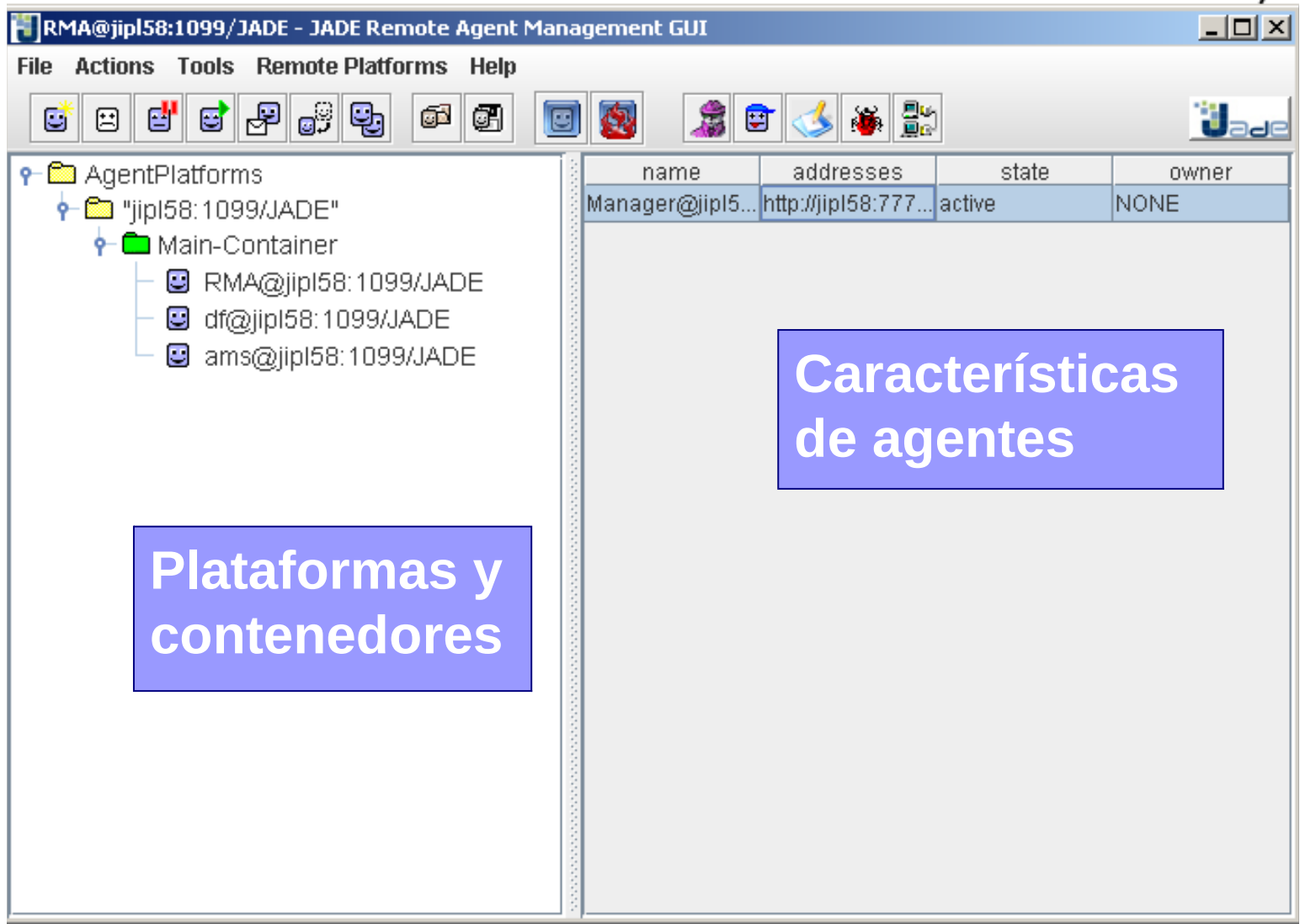
- Los agentes tienen nombres únicos y proporcionan **servicios**.
- Utilizan mensajes asíncronos, FIPA-ACL (Agent Communication Language) como lenguaje de comunicación.
- Cada agente se implementa como un *thread*
- Distintas plataformas pueden comunicarse via protocolos http o iiop.
- Incorpora un conjunto de agentes auxiliares

2. Entorno JADE

- *Los agentes necesitan del entorno de ejecución para poder “vivir”.*
- **Contenedor:**
 - *instancia del entorno de ejecución.*
- **Plataforma:**
 - capa que oculta el entorno donde se ha decidido ejecutar la aplicación,
 - *conjunto de contenedores activos,*
 - **Main Container + Agentes Auxiliares.**



■ Plataforma



The screenshot shows the JADE Remote Agent Management GUI. The window title is "RMA@jipl58:1099/JADE - JADE Remote Agent Management GUI". The menu bar includes "File", "Actions", "Tools", "Remote Platforms", and "Help". The toolbar contains various icons for agent management. The left pane shows a tree view of the agent environment:

- AgentPlatforms
 - "jipl58:1099/JADE"
 - Main-Container
 - RMA@jipl58:1099/JADE
 - df@jipl58:1099/JADE
 - ams@jipl58:1099/JADE

The right pane displays a table with the following data:

| name | addresses | state | owner |
|------------------|----------------------|--------|-------|
| Manager@jipl5... | http://jipl58:777... | active | NONE |

Two blue callout boxes are overlaid on the screenshot:

- A box in the bottom-left area of the tree view contains the text "Plataformas y contenedores".
- A box in the center-right area of the table contains the text "Características de agentes".

Plataforma

The screenshot shows the JADE Remote Agent Management GUI. The window title is "RMA@jipl58:1099/JADE - JADE Remote Agent Management GUI". The menu bar includes "File", "Actions", "Tools", "Remote Platforms", and "Help". The toolbar contains various icons for agent management. The main area is divided into a tree view on the left and a table on the right.

Tree View:

- AgentPlatforms
 - "jipl58:1099/JADE"
 - Main-Container
 - RMA@jipl58:1099/JADE
 - df@jipl58:1099/JADE
 - ams@jipl58:1099/JADE

Table:

| name | addresses | state | owner |
|------------------|----------------------|--------|-------|
| Manager@jipl5... | http://jipl58:777... | active | NONE |

Annotations:

- 2. RMA: Remote Management System:** Points to the "Main-Container" folder in the tree view.
- 3. DF: Directory Facilitator:** Points to the "df@jipl58:1099/JADE" agent in the tree view.
- 1. AMS: Agent Management System:** Points to the "ams@jipl58:1099/JADE" agent in the tree view.
- Main Container:** A blue box pointing to the "Main-Container" folder in the tree view.

- Plataforma:

1. **AMS** (*Agent Management System*)

Proporciona el servicio de nombres

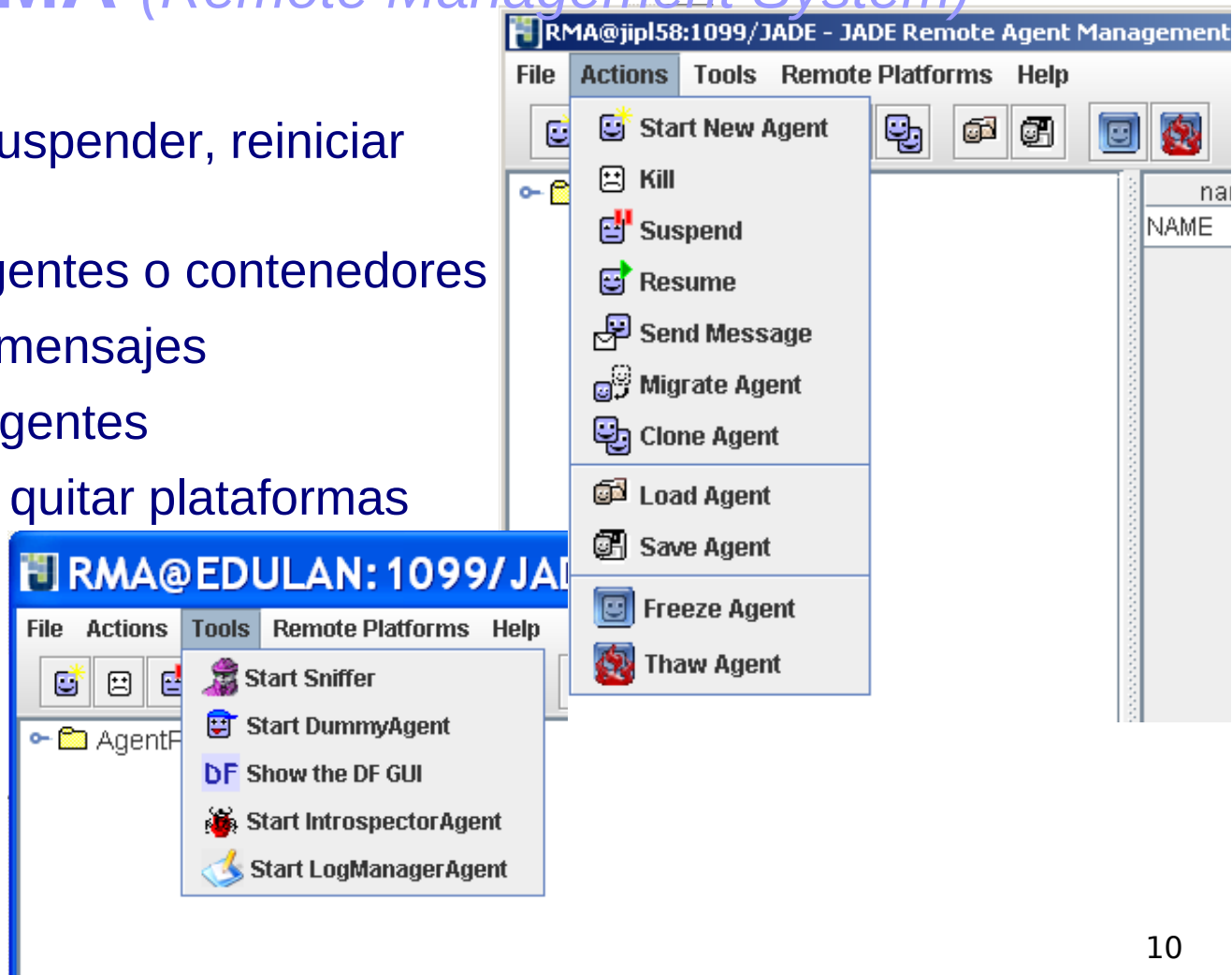
- Asegura que cada agente tiene un nombre único
- Representa la “autoridad”: es posible crear o matar agentes en contenedores remotos a través del AMS.

■ Plataforma:

2.RMA (*Remote Management System*)

Permite:

- Iniciar, suspender, reiniciar agentes
- Matar agentes o contenedores
- Mandar mensajes
- Clonar agentes
- Añadir o quitar plataformas remotas



■ Plataforma:

3. DF (*Directory Facilitator*)

- Es un interfaz del DF
- Permite:
 - Ver descripciones de los agentes registrados
 - Registrar y eliminar agentes
 - Modificar registros
 - Buscar descripciones
- Puede ser iniciado desde el RMA

■ Plataforma: *Directory Facilitator GUI*

The screenshot displays the Directory Facilitator GUI. A menu is open with options: Start Sniffer, Start DummyAgent, DF Show the DF GUI, Start Intro, and Start Logh. The main window shows a table of registrations with the following data:

| Agent name | DF F |
|--------------------------|----------|
| A6@EDULAN:1099/JADE | http://E |
| A2@EDULAN:1099/JADE | htp://E |
| A4@EDULAN:1099/JADE | http://E |
| Manager@EDULAN:1099/JADE | http://E |
| A3@EDULAN:1099/JADE | http://E |

A callout box labeled "Servicios registrados" points to the table. The right-hand pane shows the "DF description" for agent E2@EDULAN:1099/JADE, with sections for Languages, Interaction-protocols, Agent services (containing "Medic_axis"), and Lease Time (set to "unlimited").

Plataforma: Agentes auxiliares

The screenshot displays the JADE Remote Agent Management GUI. The main window shows a tree view of AgentPlatforms and a table of active agents. A secondary window shows a menu for starting auxiliary agents.

| name | addresses | state | owner |
|------------------|----------------------|--------|-------|
| Manager@jipl5... | http://jipl58:777... | active | NONE |

AgentPlatforms structure:

- AgentPlatforms
 - "jipl58: 1099/JADE"
 - Main-Container
 - RMA@jipl58: 1099/JADE
 - df@jipl58: 1099/JADE
 - ams@jipl58: 1099/JADE

c. Introspector

b. Dummy Agent

a. Sniffer Agent

RMA@EDULAN: 1099/JADE

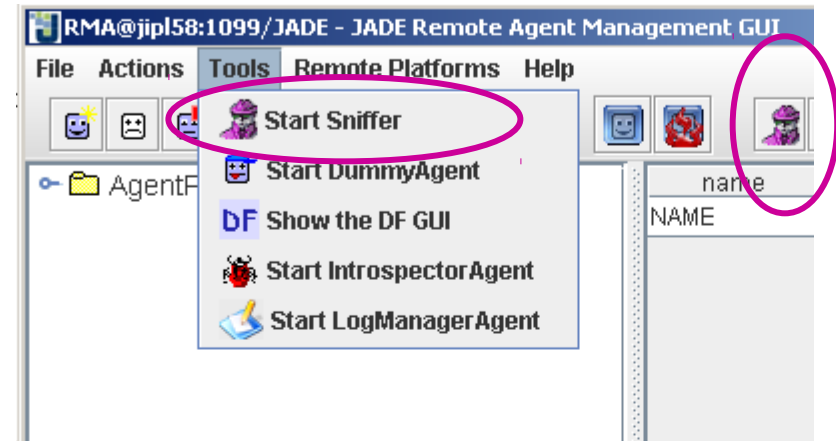
File Actions Tools Remote Platforms Help

- Start Sniffer
- Start DummyAgent
- Show the DF GUI
- Start IntrospectorAgent
- Start LogManagerAgent

- Plataforma: *a. Sniffer Agent*



- ❖ Muestra las interacciones que se producen
- ❖ Puede ser iniciado desde el RMA
- ❖ El usuario selecciona los agentes que desea monitorizar
- ❖ Permite ver el contenido de cada mensaje



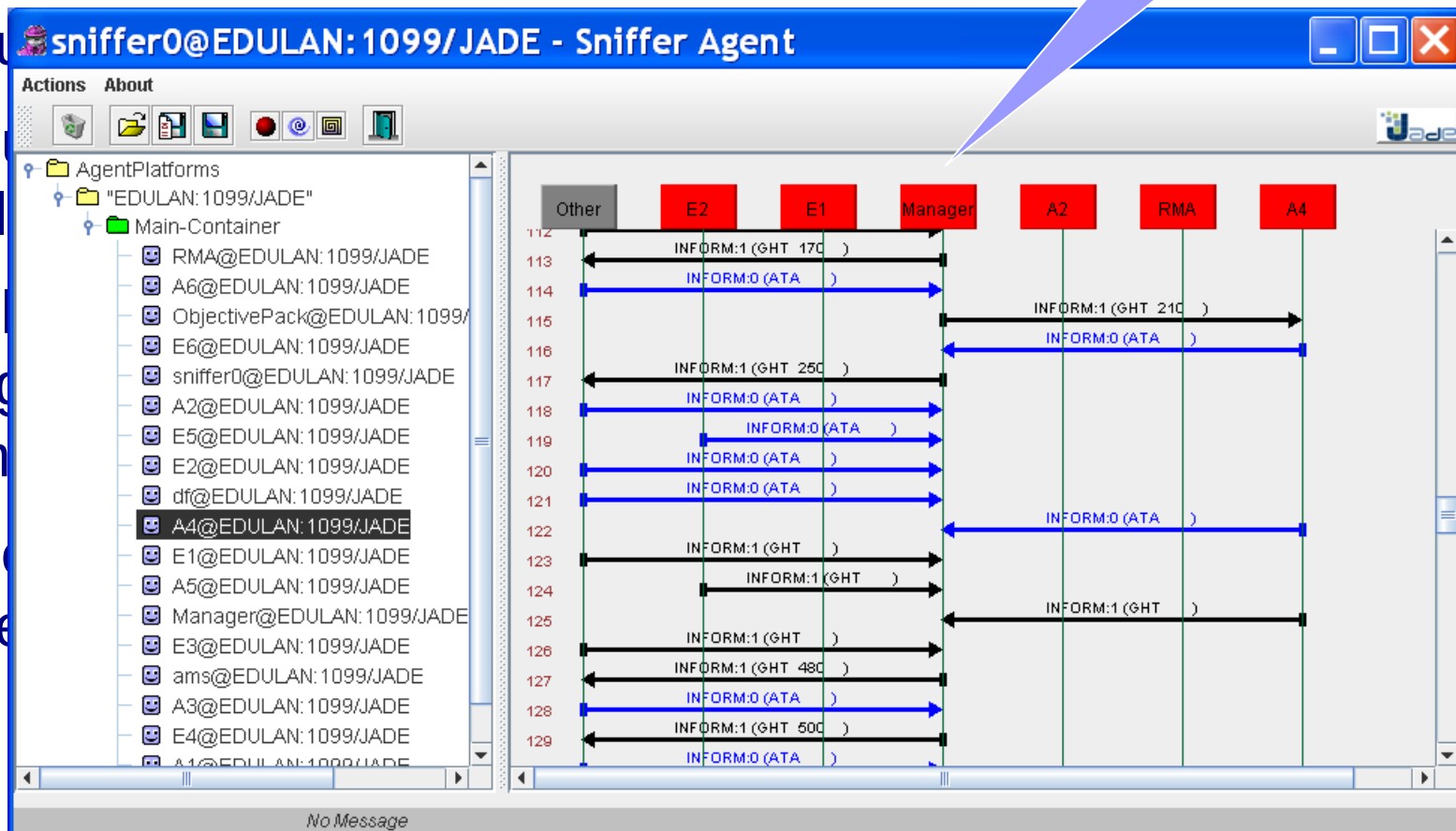
■ Plataforma: *a.Sniffer Agent*



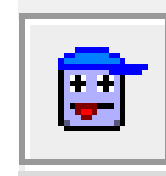
Agentes Controlados

❖ Muestra las interacciones

❖ P
el
E
ag
m
❖ P
de



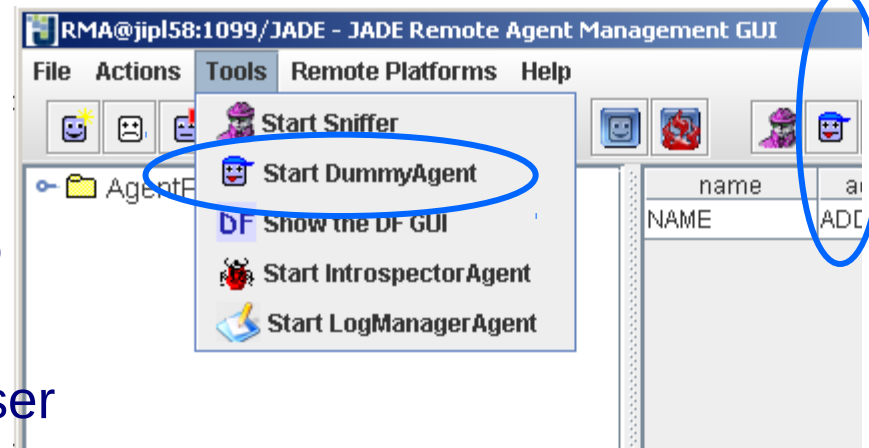
■ Plataforma: *b.Dummy Agent*



❖ Permite interactuar con agentes de forma sencilla.

- Componiendo y enviando mensajes ACL
- Estos mensajes pueden ser almacenados y empleados posteriormente

❖ Puede ser iniciado desde el RMA.



■ Plataforma: *b.Dummy Agent*



❖ Permite
agentes
sencilla

- Comp

Campos del
mensaje

❖ Puede
desde

The screenshot shows a window titled "da0@EDULAN: 1099/JADE - DummyAgent". It has tabs for "General", "Current message", and "Queued message". The "Current message" tab is active, showing fields for "Sender" (Set da0@EDULAN:1099/JADE), "Receivers" (E5@EDULAN:1900/JADE), "Reply-to", "Communicative act" (cancel), and "Content". Below these are fields for "Language", "Encoding", "Ontology", "Protocol" (Null), "Conversation-id", "In-reply-to", "Reply-with", "Reply-by" (Set), and "User Properties".

To the right, a "Queued message" list shows a sequence of messages:

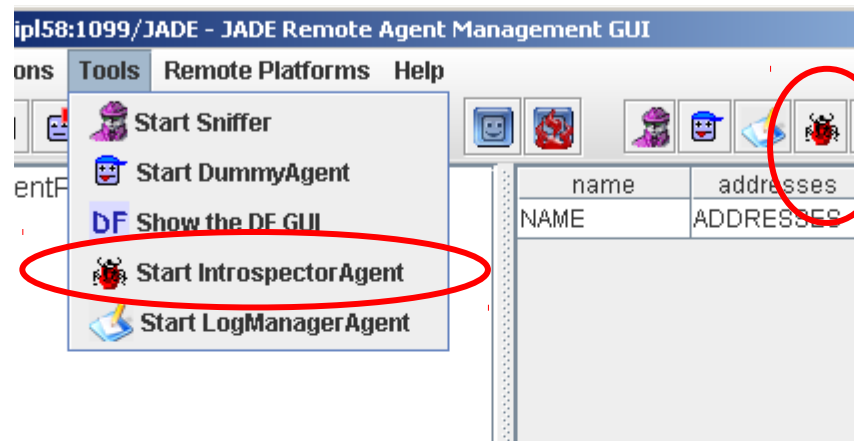
| | | | |
|----------------|-----------------|----------|----------------------|
| 6/03/06 14:16: | FAILURE | rcv from | ams@EDULAN:1099/JADE |
| 6/03/06 14:16: | CANCEL | sent to | E5@EDULAN:1900/JADE |
| 6/03/06 14:15: | ACCEPT-PROPOSAL | sent to | E6@EDULAN:1099/JADE |

Secuencia
mensajes

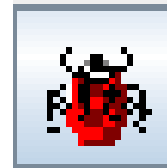
- Plataforma: *c.Introspector Agent*



- ❖ Permite monitorizar y controlar el ciclo de vida de un agente
- ❖ Puede ser iniciado desde el RMA



- Plataforma: *c./Introspector Agent*



Cola de entrada de mensajes

Cola de salida de mensajes

The screenshot shows the JADE Introspector interface. On the left, a list of agents is visible, with 'E1@jipl58:1099/JADE' selected. The main window displays the 'View State Debug' panel for this agent. The 'Current State' is 'Active'. Below this, there are buttons for 'Suspend', 'Wait', 'Wake Up', and 'Kill'. The 'Incoming Messages' panel shows a list of five 'INFORM' messages in the 'Pending' state. The 'Outgoing Messages' panel is currently empty. At the bottom, the 'Behaviours' panel shows a tree structure with 'FSMBehaviour' expanded, listing states like 'STATE_QUIT', 'STATE_TARGET', etc. A metadata panel on the right shows 'Name: 3', 'Class: Classes.CTroop\$3', and 'Kind: CyclicBehaviour'.

Estado del agente

Comportamientos del agente

■ Iniciar JADE

```
java jade.Boot [options] [AgentSpecifier list]
```

- container** (*si es distinto del principal*) –
- host** (*nombre del host*)
- port** (*puerto, por defecto es el 1099*)
- name** (*nombre simbólico de la plataforma*)
- gui** (*lanza el RMA*)
- mtp** (*permite añadir protocolos externos además del iiop*)

3. Estructura agente

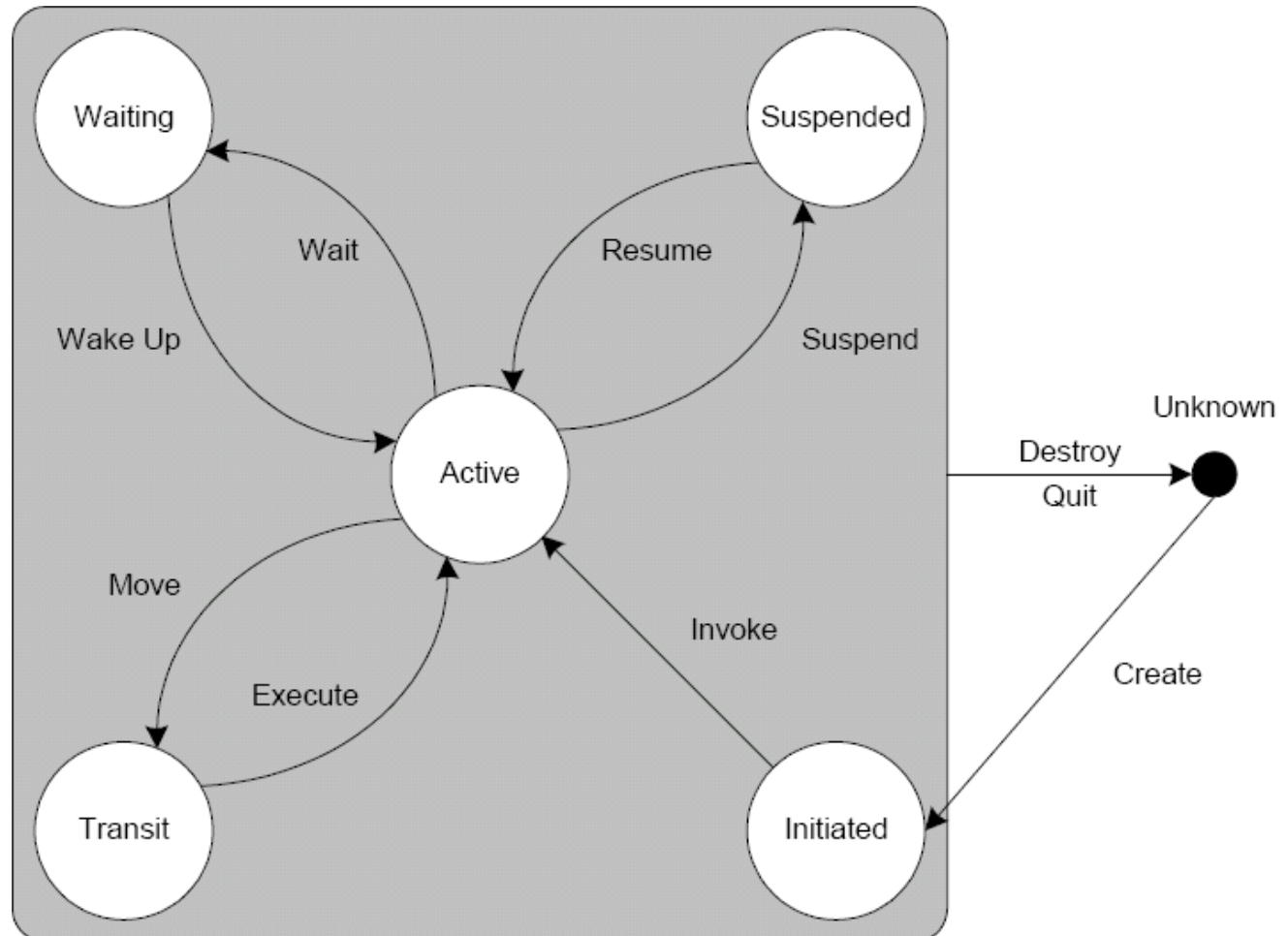
- Agente JADE es una instancia de una clase de usuario JAVA que extiende la clase básica **Agent**.

```
public class Mi_Agente extends Agent {  
    ...  
}
```

- **Ciclo de vida** de agente JADE es el ciclo de FIPA

3. Estructura agente

- **Ciclo de vida** de agente JADE es el ciclo de FIPA



3. Estructura

Métodos de la clase:

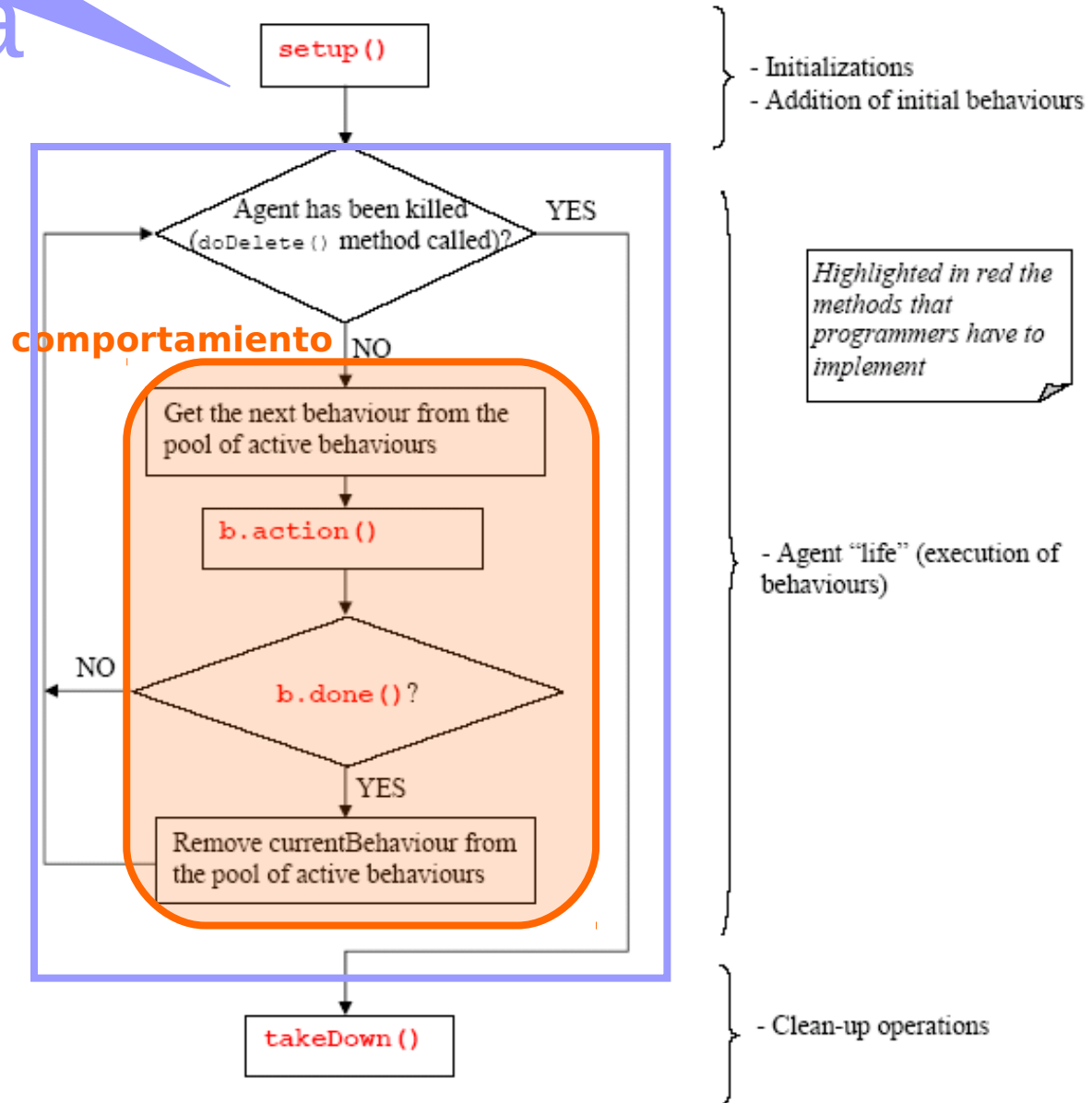
A sobrecargar:

- **setup()** = inicializa el agente
- **takeDown()** = método que se activa cuando se finaliza el agente

Finales:

- **doDelete()** = orden para finalizar un agente.
- **getAID()** = el AID-Identificador del agente

`getAID().getName() = nombre del AID del agente`



3. Estructura agente

- Tareas que realiza un agente JADE
=> se estructuran en **comportamientos**

- Métodos de la clase *Agent*:
 - **addBehaviour()**: Añade al agente la instancia de comportamiento que recibe como argumento

 - **removeBehaviour()**: Elimina el comportamiento de la lista de comportamientos

3. Estructura agente

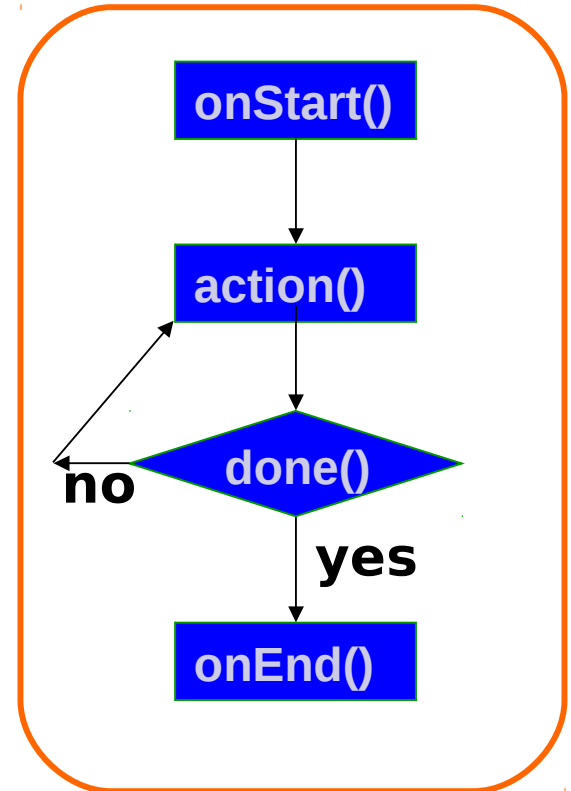
- Métodos de la clase *Behaviour*:

- Fundamentales:

- **action()**: Lo que hace realmente el agente. Se pasa a realizar otro comportamiento, sólo cuando éste método planificado del comportamiento actualmente finaliza.
 - **done()**: Si ha acabado o no el comportamiento del agente

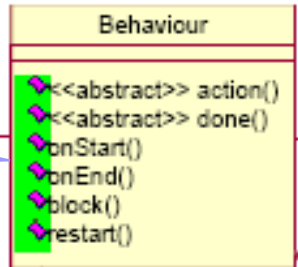
- Adicionales:

- **onStart()**: se invoca sólo una vez antes de la primera ejecución del método **action()**.
 - **onEnd()**: se invoca una vez sólo después de que el método **done()** devuelve **true**



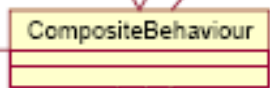
Conjunto de acciones que realiza un agente para lograr su objetivo

0..*

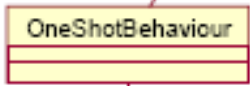
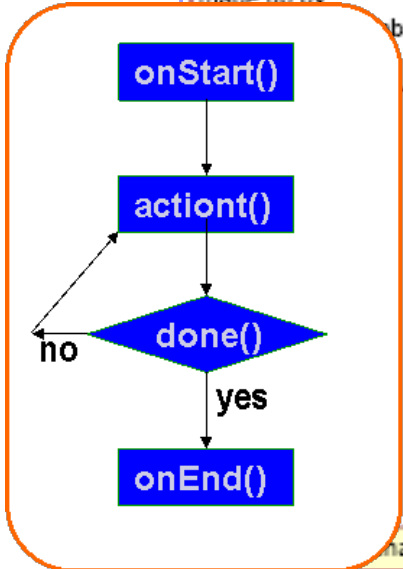
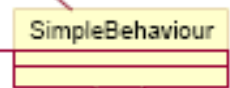


Models a generic task

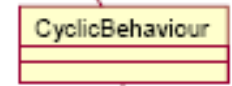
Models a complex task i.e. a task that is made up by



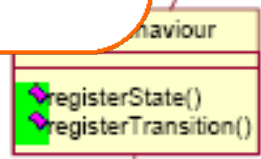
Models a simple task i.e. a task that is not composed of sub-tasks



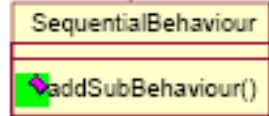
Models an atomic task (its done() method returns true)



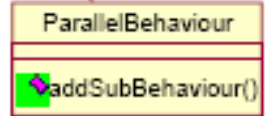
Models a cyclic task (its done() method returns false)



Models a complex task whose sub-tasks corresponds to the activities performed in the states of a Finite State Machine



Models a complex task whose sub-tasks are executed sequentially



Models a complex task whose sub-tasks are executed concurrently

do

4. Ejemplos *A.HelloWorldAgent*

```
package examples.hallo;
import jade.core.Agent;

public class HelloWorldAgent extends Agent {

    protected void setup() {
        System.out.println(" Hola a todos! Mi
                           nombre es "+getLocalName());
    }
}
```

4. Ejemplo *B.HelloWorldAgentBye*

```
public class HelloWorldAgentBye extends Agent {  
  
    protected void setup() {  
        System.out.println(" Hola a todos!  
                             Mi nombre es " +getLocalName());  
        doDelete();  
    }  
  
    protected void takeDown() {  
        System.out.println(  
            "Adios, ya no tengo nada mas que hacer.");  
    }  
}
```