Laboratorio 4
Agentes Móviles

Técnicas Avanzadas de Inteligencia Artificial

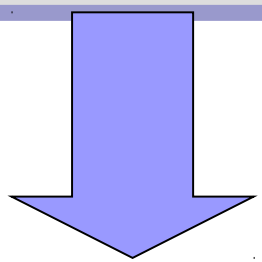
Dpt. Lenguajes y Sistemas Informáticos.

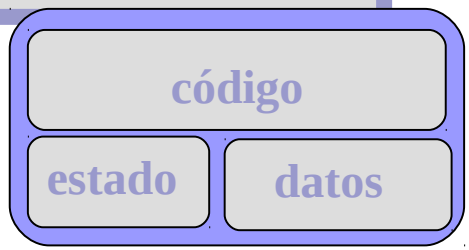
FISS. UPV-EHU

5.1. Introducción

Desde el punto de vista de sistemas distribuidos,

un **agente móvil** es un programa con identidad única que puede mover su *código*, *datos* y *estados* entre máquinas de una misma red.





Para conseguirlo los agentes móviles son capaces de suspender su ejecución en cualquier momento y continuar una vez que son residentes en otro localización.

Paradigmas

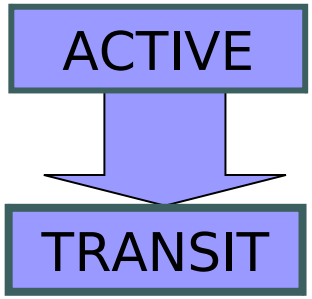
Cliente-servidor, Ejecuciones remotas y Agentes móviles

5.1. Introducción

MOVILIDAD INTRA-PLATAFORMA

- JADE proporciona servicios llamados ***Agent Mobility Service*** que implementan la movilidad intra-plataformas.
- Proporciona la habilidad de moverse entre diferentes contenedores de la misma plataforma.
- No permite a los agente moverse entre contenedores de diferentes plataformas

MOVER y CLONAR



5.1. Introducción

ASPECTOS MOVER:

- void **doMove**(Location destination) // jade.core Location
  - ContainerID/PlatformID
- void **beforeMove**()
- void **afterMove**()

MOVER y CLONAR

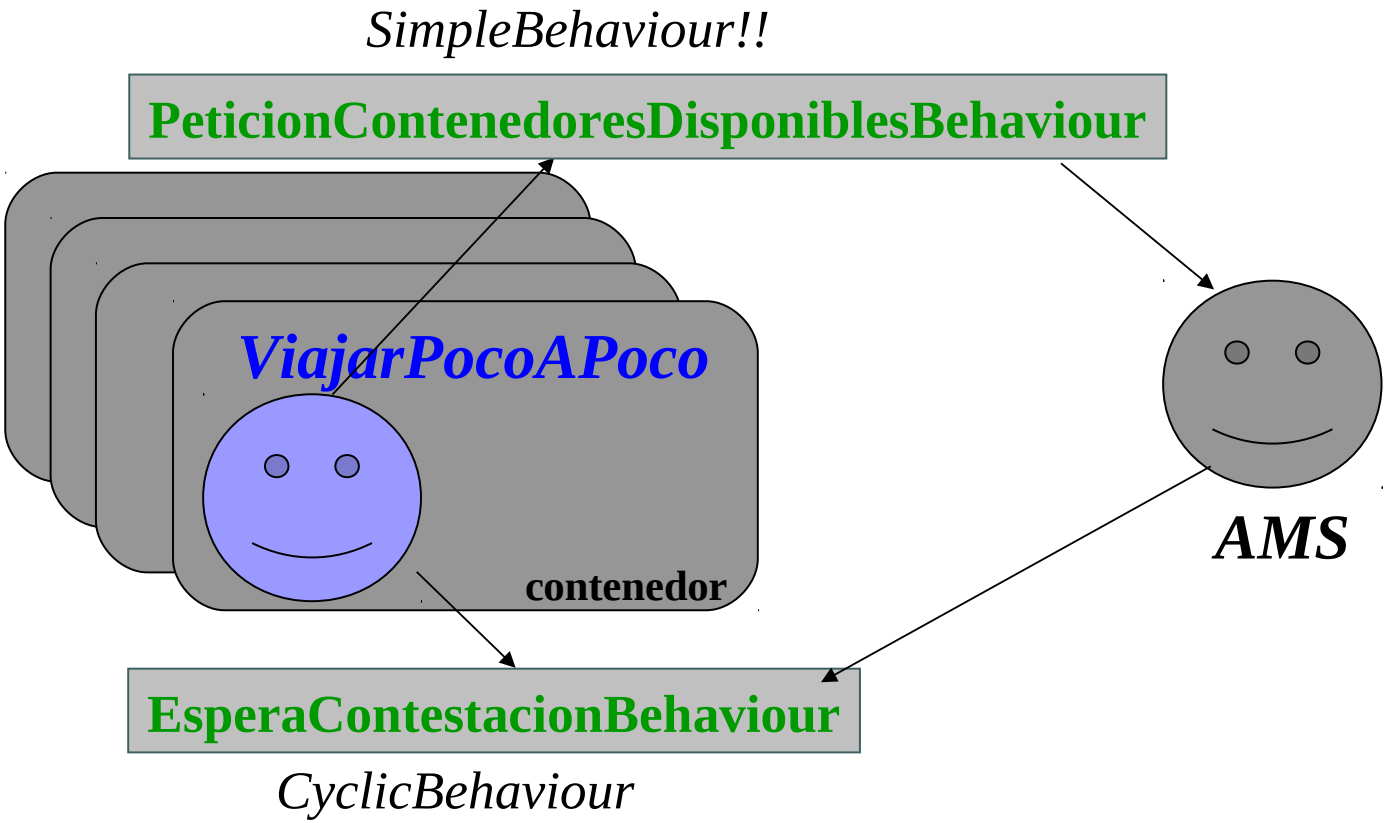
ASPECTOS CLONAR:

- public void **doClone**(Location destination, String newName) // jade.core Location
  - ContainerID/PlatformID

5.2. ViajarPocoAPoco, Descripción

- ***ViajarPocoAPoco*** es un agente móvil que se va desplazando poco a poco a través de todos los contenedores accesibles de la plataforma.
- Se ha considerado tan solo la movilidad intra-plataformas.
- Un agente puede navegar a través de contenedores diferentes situados en el mismo o diferentes ordenadores, pero esta limitado a una sola plataforma de JADE.

5.2. ViajarPocoAPoco, Visión General



5.2. ViajarPocoAPoco, Setup

```
public void setup() {
// register the SL0 content language
getContentManager().registerLanguage(new SLCodec(),
                                FIPANames.ContentLanguage.FIPA_SL0);
// register the mobility ontology
getContentManager().registerOntology(
                                        MobilityOntology.getInstance());

// get the list of available locations and show it in the GUI
addBehaviour(new
            PeticionContenedoresDisponiblesBehaviour(this));
addBehaviour(new EsperaContestacionBehaviour(this));

}
```

5.2. ViajarPocoAPoco,

comportamiento **PeticionContenedoresDisponiblesBehaviour**

public PeticionContenedoresDisponiblesBehaviour(ViajarPocoAPoco viajar) …

// fills all parameters of the request ACLMessage

```
request.clearAllReceiver();
request.addReceiver(viajar.getAMS());
request.setLanguage(FIPANames.ContentLanguage.FIPA_SL0);
request.setOntology(MobilityOntology.NAME);
request.setProtocol(FIPANames.InteractionProtocol.FIPA_REQUEST);
…
```

5.2. ViajarPocoAPoco,

Comportamiento **PeticionContenedoresDisponiblesBehaviour**

```
// creates the content of the ACLMessage
try {
Action action = new Action();
action.setActor(viajar.getAMS());
action.setAction(new QueryPlatformLocationsAction());
viajar.getContentManager().fillContent(request, action);
} catch (Exception fe) {
fe.printStackTrace();
}
// System.out.println("Buscando sitios a los que ir");
send(request);
```

5.2. ViajarPocoAPoco,

Comportamiento **EsperaContestacionBehaviour**

```
class EsperaContestacionBehaviour extends CyclicBehaviour {
// protected void tratarRespuesta() {
// System.out.println("Esperando mensaje");
public EsperaContestacionBehaviour(ViajarPocoAPoco viajar) {
super(viajar);
}
```

public void action() {

1. Espera mensaje INFORM del AMS (receive con template INFORM)
2. Contenido de mensaje pasa a variable listaDeContenedoresDisponibles
3. Busca uno no visitado (no en listaDeContenedoresVisitados)
4. donde es el nuevo sitio al que ir
5. doMove

5.2. ViajarPocoAPoco,

Comportamiento **EsperaContestacionBehaviour**

```
protected void beforeMove() {
listaDeContenedoresVisitados.add(donde);
// Solo puesto para conseguir detener el comportamiento del
agente y
// visualizar todo adecuadamente
try {
      System.out.println("\nIntroduce un numero cualquiera");
      BufferedReader buff = new BufferedReader(new
      InputStreamReader(System.in));
      int num = buff.read();
}
catch (java.io.IOException io) {
System.out.println(io);
}

}
```

5.2. ViajarPocoAPoco,

Comportamiento **EsperaContestacionBehaviour**

```
protected void afterMove() {
// System.out.println(getLocalName() + " Despues de moverme.");
// //////
// BEEP(), pita 10 veces con cierta distancia entre uno y otro
// //////

// Register again SL0 content language and JADE mobility ontology,
// since they don't migrate.
getContentManager().registerLanguage(new SLCodec(),
FIPANames.ContentLanguage.FIPA_SL0);
getContentManager().registerOntology(MobilityOntology.getInstanc
e());

addBehaviour(new
PeticionContenedoresDisponiblesBehaviour(this));

}
```

5.2. ViajarPocoAPoco, Ejecución 1

Carpeta Mobility→ OnlyMove

1. Compilar las clases, 1.compila.bat: javac *.java
2. Lanzar el GUI, 2.gui.bat: java jade.Boot -gui
3. Lanzar contenedor alfa, 3.alfa.bat:

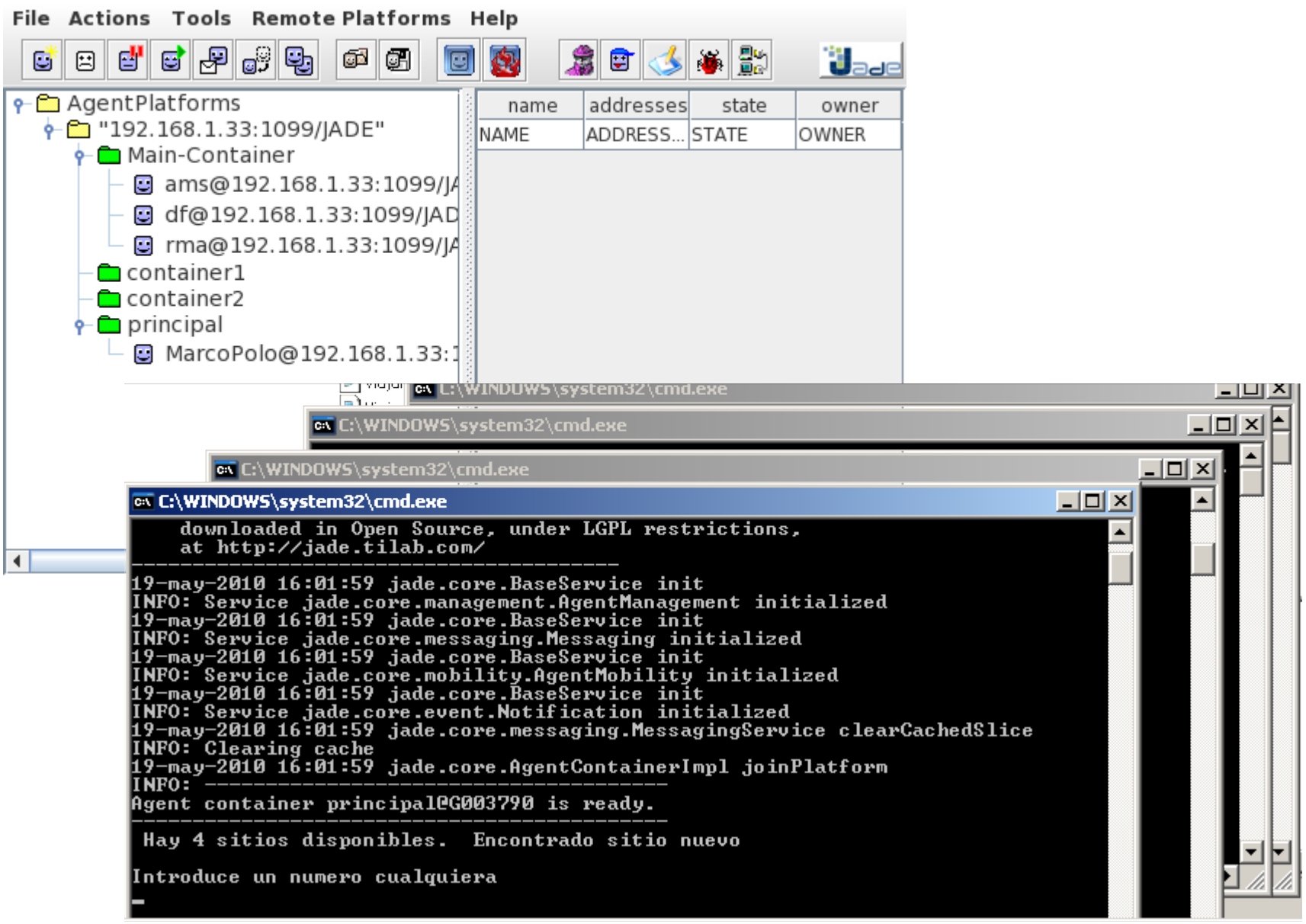
java jade.Boot -container -container-name alfa

4. Lanzar contenedor beta, 4.beta.bat:

java jade.Boot -container -container-name beta

5. Lanzar a MarcoPolo, 5.ViajarPocoAPoco.bat: introducir un número cada vez

java jade.Boot -container -container-name principal MarcoPolo:ViajarPocoAPoco


File Actions Tools Remote Platforms Help
AgentPlatforms
"192.168.1.33:1099/JADE"
Main-Container
ams@192.168.1.33:1099/JA
df@192.168.1.33:1099/JAD
rma@192.168.1.33:1099/JA
container1
container2
principal
MarcoPolo@192.168.1.33:1
name
addresses
state
owner
NAME
ADDRESS...
STATE
OWNER
C:\WINDOWS\system32\cmd.exe
C:\WINDOWS\system32\cmd.exe
C:\WINDOWS\system32\cmd.exe
C:\WINDOWS\system32\cmd.exe
downloaded in Open Source, under LGPL restrictions,
at http://jade.tilab.com/
19-may-2010 16:01:59 jade.core.BaseService init
INFO: Service jade.core.management.AgentManagement initialized
19-may-2010 16:01:59 jade.core.BaseService init
INFO: Service jade.core.messaging.Messaging initialized
19-may-2010 16:01:59 jade.core.BaseService init
INFO: Service jade.core.mobility.AgentMobility initialized
19-may-2010 16:01:59 jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
19-may-2010 16:01:59 jade.core.messaging.MessagingService clearCachedSlice
INFO: Clearing cache
19-may-2010 16:01:59 jade.core.AgentContainerImpl joinPlatform
INFO:
Agent container principal@G003790 is ready.
Hay 4 sitios disponibles. Encontrado sitio nuevo
Introduce un numero cualquiera

5.3. ViajarTodo, Ejecución 2

Carpeta Mobility→ OnlyMove

1. Lanzar el GUI, 2.gui.bat: java jade.Boot -gui
2. Lanzar contenedor alfa, 3.alfa.bat:

java jade.Boot -container -container-name alfa

3. Lanzar contenedor beta, 4.beta.bat:

java jade.Boot -container -container-name beta

4. Lanzar a ElCano, 6.ViajarTodo.bat: el sólo se irá moviendo

jade.Boot -container -container-name principal ElCano:ViajarSinLectura

5.2. ViajarTodo

Ejecución 3- lanzar contenedores desde vuestros ordenadores

Carpeta Mobility→ OnlyMove

1. Lanzo el GUI en G002601: java jade.Boot -gui
2. Lanzar contenedores contra mi ordenador G002601:

   7.nombre.bat, sustituir nombre por algún nombre único

   java jade.Boot –host G002601
   -container -container-name nombre

3. Yo lanzo a ElCano para que os visite,

   6.ViajarTodo.bat

   jade.Boot -container -container-name principal
   ElCano:ViajarSinLectura

5.2. ViajarTodo

Ejecución 4- visualizar y controlar la plataforma desde vuestros PCs

Carpeta Mobility→ OnlyMove

- **Visualizar el Gui desde cualquier ordenador.**
- **Ejecutar todo como en el caso anterior**
- **Editar GuiRemoto.bat:**

  java jade.Boot -container -host G002601
  nombreRMA:jade.tools.rma.rma

  Copia del Gui lanzado en **G002601**

- **Lanzar ElCano, uno, otro, … ¿qué pasa?**

5.3. MobileAgent, Clases Principales

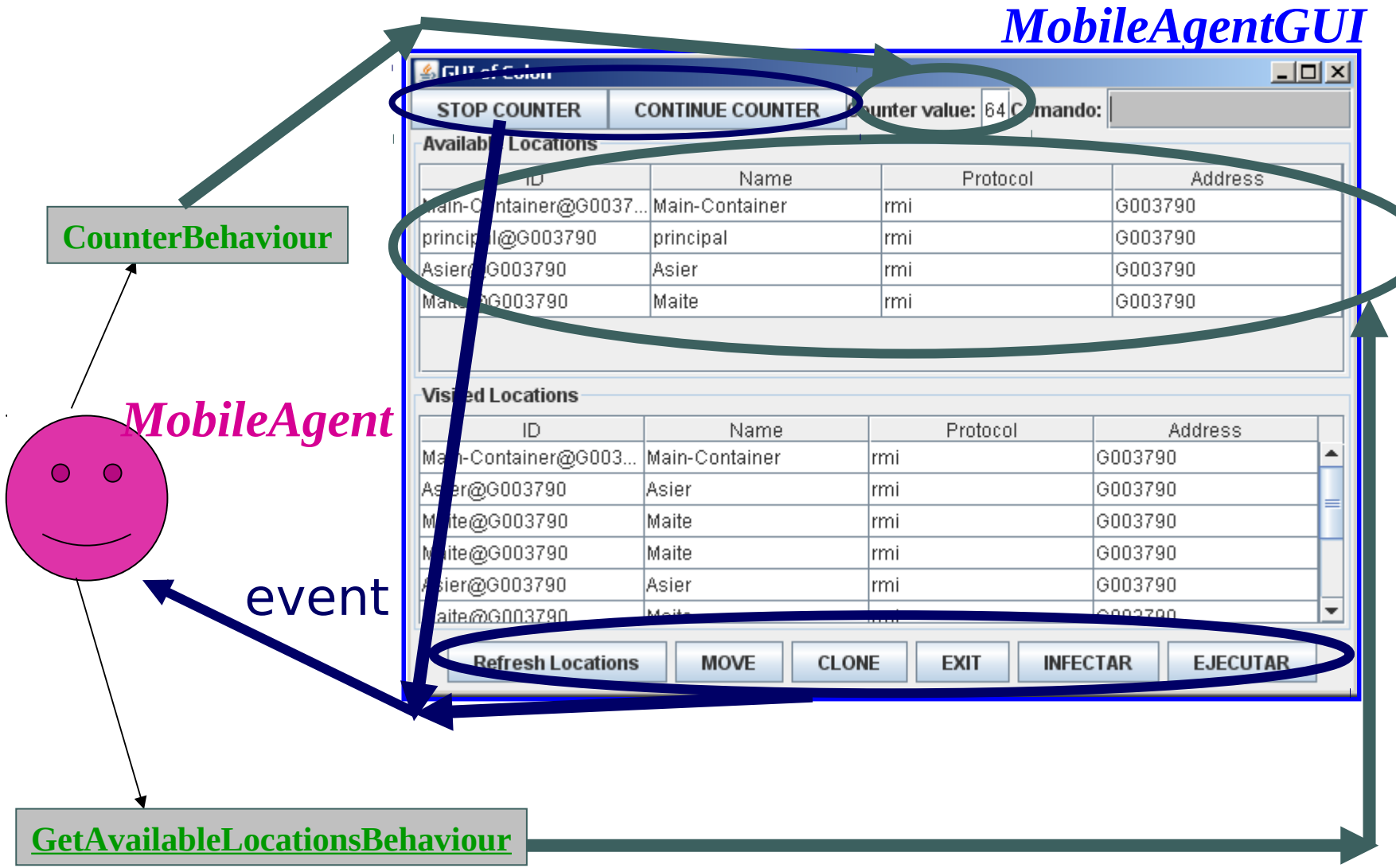
MobileAgent.java: Es un ***GuiAgent***, un agente móvil que extiende de un agente con GUI asociado y es el que se mueve entre los diferentes contenedores, situados en distintos/mismos ordenadores dentro de una red local.

GetAvailableLocationsBehaviour: Esta clase define un comportamiento en el cual se obtienen todos los contenedores disponibles dentro de la red local.

CounterBehaviour: es un "SimpleBehaviour" que utiliza los datos cnt y cntEnable del MobileAgente. Incrementa el contador, lo visualiza, se bloquea durante 2 segundos y se repite indefinidamente.

MobileAgentGui: Es un ***JFrame,*** el GUI de MobileAgent. Como en java se crea un nuevo thread por cada GUI, la comunicación con el thread del agente se realiza mediante eventos.

5.3. MobileAgent, Clases Principales



5.3. MobileAgent, Variables de la clase

```
int     cnt;   // this is the counter
public boolean cntEnabled;  // this flag indicates if counting is enabled
transient  protected MobileAgentGui gui;  // this is the gui
Location nextSite;  // this variable holds the destination site

// These constants are used by the Gui to post Events to the Agent
public static final int EXIT = 1000;
public static final int MOVE_EVENT = 1001;
public static final int STOP_EVENT = 1002;
public static final int CONTINUE_EVENT = 1003;
public static final int REFRESH_EVENT = 1004;
public static final int CLONE_EVENT = 1005;

// this vector contains the list of visited locations
Vector visitedLocations = new Vector();
```

5.3. MobileAgent, Setup

```
// register the SL0 content language
getContentManager().registerLanguage(new SLCodec(),
                                     FIPANames.ContentLanguage.FIPA_SL0);
// register the mobility ontology
```

```
getContentManager().registerOntology(MobilityOntology.getInstance());
```

```
// Crea la interfaz de usuario
gui = new MobileAgentGui(this);
gui.setVisible(true);
// Añade el behaviour que obtiene la lista de sitios disponibles
addBehaviour(new GetAvailableLocationsBehaviour(this));
// Añade el behaviour del contador y de la recepción de mensajes
Behaviour b1 = new CounterBehaviour(this);
addBehaviour b1;
```

5.3. MobileAgent, Método **onGuiEvent**

```
protected void onGuiEvent(GuiEvent ev){
switch(ev.getType()) {
      case EXIT:
            gui.dispose(); doDelete(); break;
      case MOVE_EVENT:
            Iterator moveParameters = ev.getAllParameter();
            nextSite =(Location)moveParameters.next();
            doMove(nextSite); break;
      case CLONE_EVENT:
            Iterator cloneParameters = ev.getAllParameter();
            nextSite =(Location)cloneParameters.next();
            doClone(nextSite,"clone"+cnt+"of"+getName()); break;
      case STOP_EVENT:
            stopCounter(); break;
      case CONTINUE_EVENT:
            continueCounter(); break;
      case REFRESH_EVENT:
            addBehaviour(new
               GetAvailableLocationsBehaviour(this));
            break;
}
```

5.3. MobileAgent, Métodos **move**

beforeMove()

```
gui.dispose();
gui.setVisible(false);
System.out.println(getLocalName()+" se va a otro sitio.");
```

afterMove()

```
// creates and shows the GUI
gui = new MobileAgentGui(this);

if (nextSite != null) {
    visitedLocations.addElement(nextSite);
     for (int i=0; i<visitedLocations.size(); i++)
         gui.addVisitedSite((Location)visitedLocations.elementAt(i));
}
gui.setVisible(true);
```

```
// register again the SL0 content language and JADE mobility ontology
  getContentManager().registerLanguage(new SLCodec(),
FIPANames.ContentLanguage.FIPA_SL0);
  getContentManager().registerOntology(MobilityOntology.getInstance());
```

```
addBehaviour(new GetAvailableLocationsBehaviour(this));
```

5.3. MobileAgent, Ejecución

Carpeta Mobility→ MoveClone

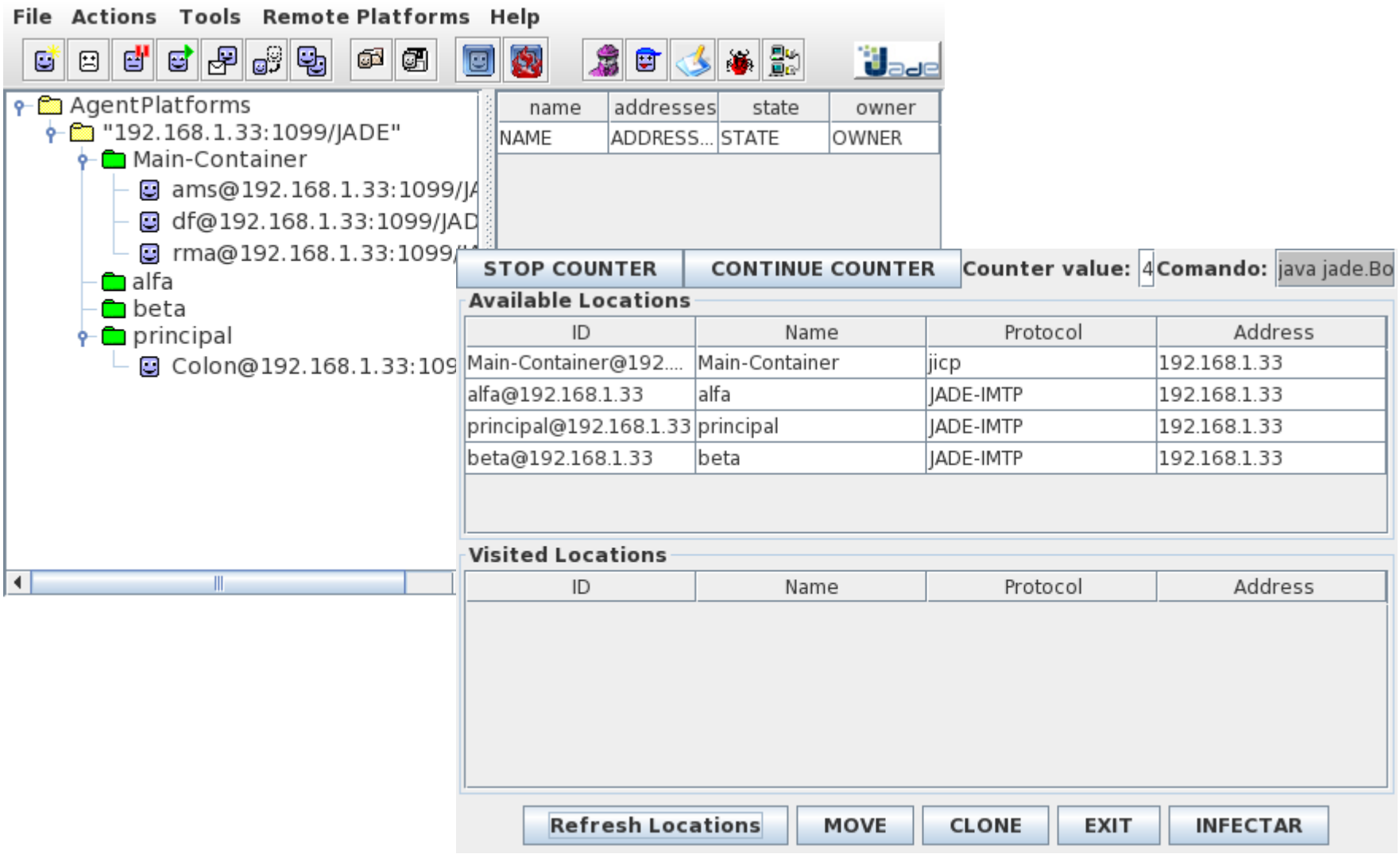
1. Compilar las clases: javac *.java
2. Lanzar el GUI: java jade.Boot -gui
3. Lanzar contenedor alfa:

   java jade.Boot -container -container-name alfa
4. Lanzar contenedor beta:

   java jade.Boot -container -container-name beta
5. Lanzar MobileAgent

   java jade.Boot -container -container-name principal
   Colon:MobileAgent

5.3. MobileAgent, Ejecución



5.3. MobileAgent, Ejecución- Tareas

Actividades a realizar:

- **MOVER** a un contenedor seleccionado
- **INFECTAR**, mover por todos los contenedores existentes.
- **CLONAR**, duplicar en otros contenedores.
- **EJECUTAR** por todos los contenedores: mspaint, notepad, shutdown, …
- …