

JADE

Java Agent DEvelopment Framework - Part I

Integrated Master's in Informatics Engineering

Intelligent Agents

2017/2018

Synthetic Intelligence Lab

Ricardo Ramos

Filipe Gonçalves

Paulo Novais



Useful Links

- <http://jade.tilab.com/>
- <http://jade.tilab.com/doc/tutorials/JADEProgramming-Tutorial-for-beginners.pdf>
- <http://jade.tilab.com/doc/api/jade/core/Agent.html>
- <http://jade.tilab.com/doc/api/jade/core/behaviours/Behaviour.html>
- <http://jade.tilab.com/doc/api/jade/lang/acl/ACLMessage.html>

Features

- Agent platform with the purpose of simplifying the development of agents:
 - Execution environment.
 - Provides a class library for multi-agent development.
 - Graphical administration and management tools for agents.

Features

- Agent lifetime management.
- Message transport management.
- Message structure.
- Protocol Interaction.
- Ontologies.
- Safety.

Main concepts

Containers:

- Each instance of the execution environment is a Container.
- Each Container can have zero or more agents running simultaneously.
 - Main-Container is required to support the platform.
 - All other Containers have to register on it when they are created.
 - Containers can be remote.

Main concepts

Containers:

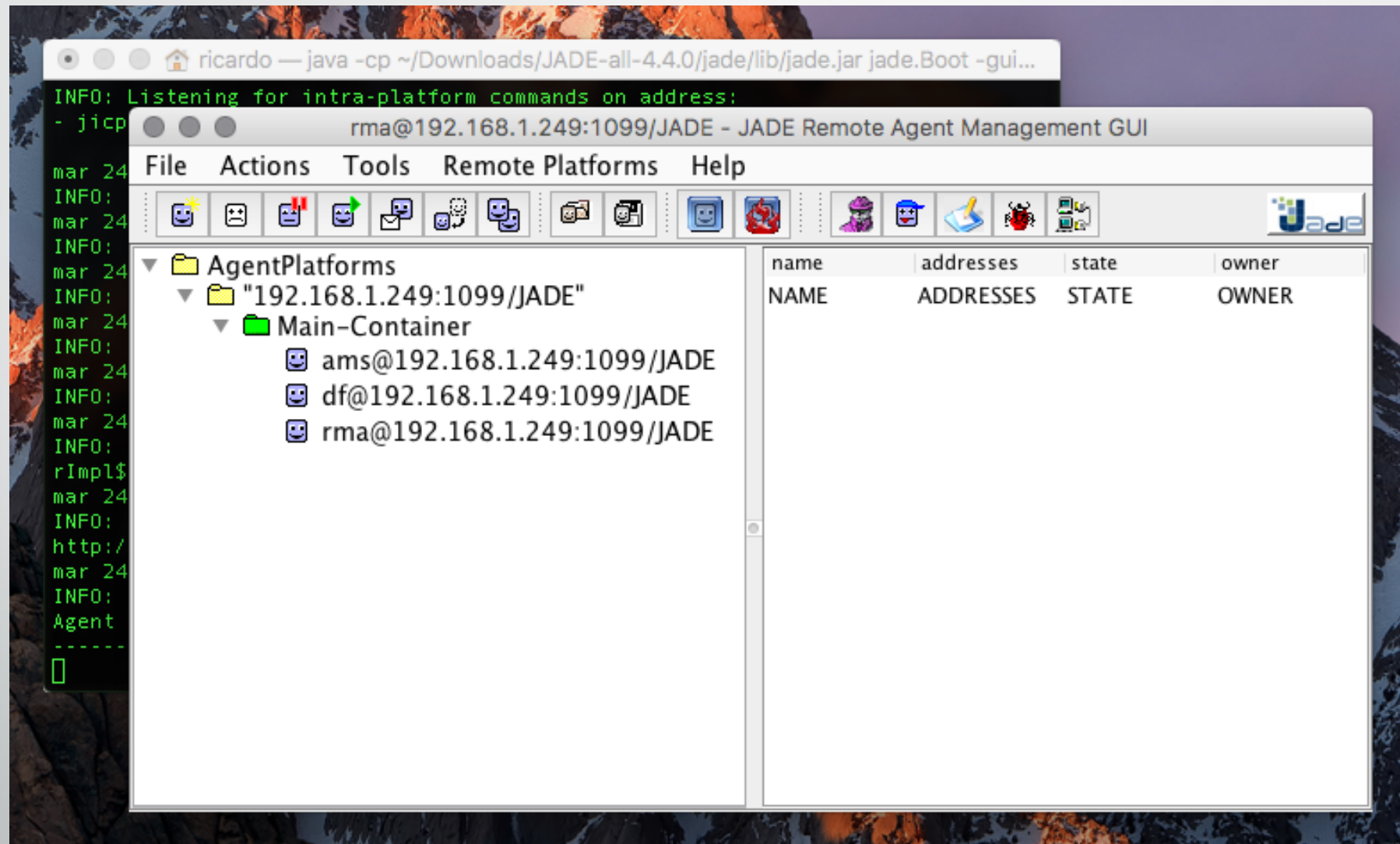
- **Main-Container:**

- Agent Management System (AMS): make sure there are no name collisions and lets you start and end agents.
- Directory Facilitator (DF): provides a yellow pages services through which an agent can search for other agents.

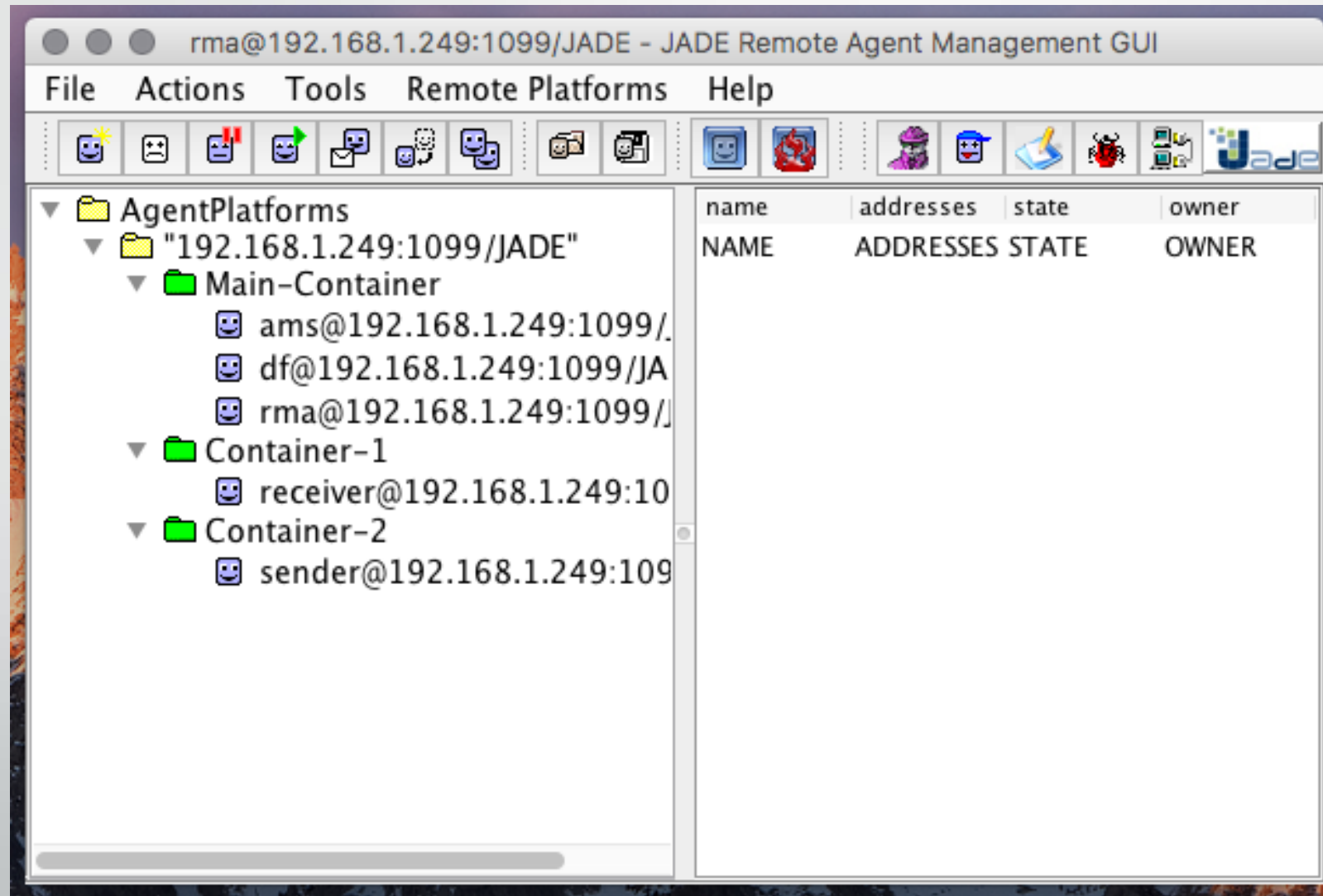
Optionally:

- Remote Monitoring Agent (RMA): allows to manage the platform through a GUI.

GUI without agents



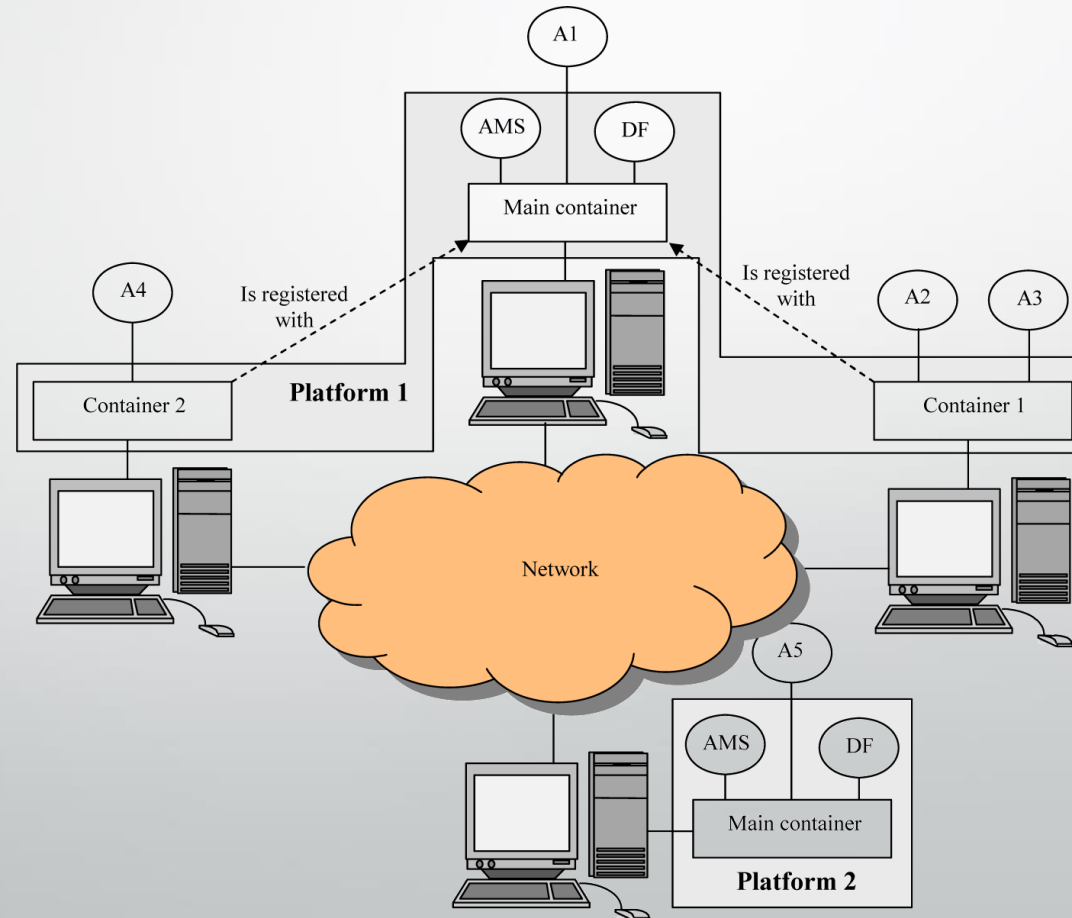
GUI with 2 agents



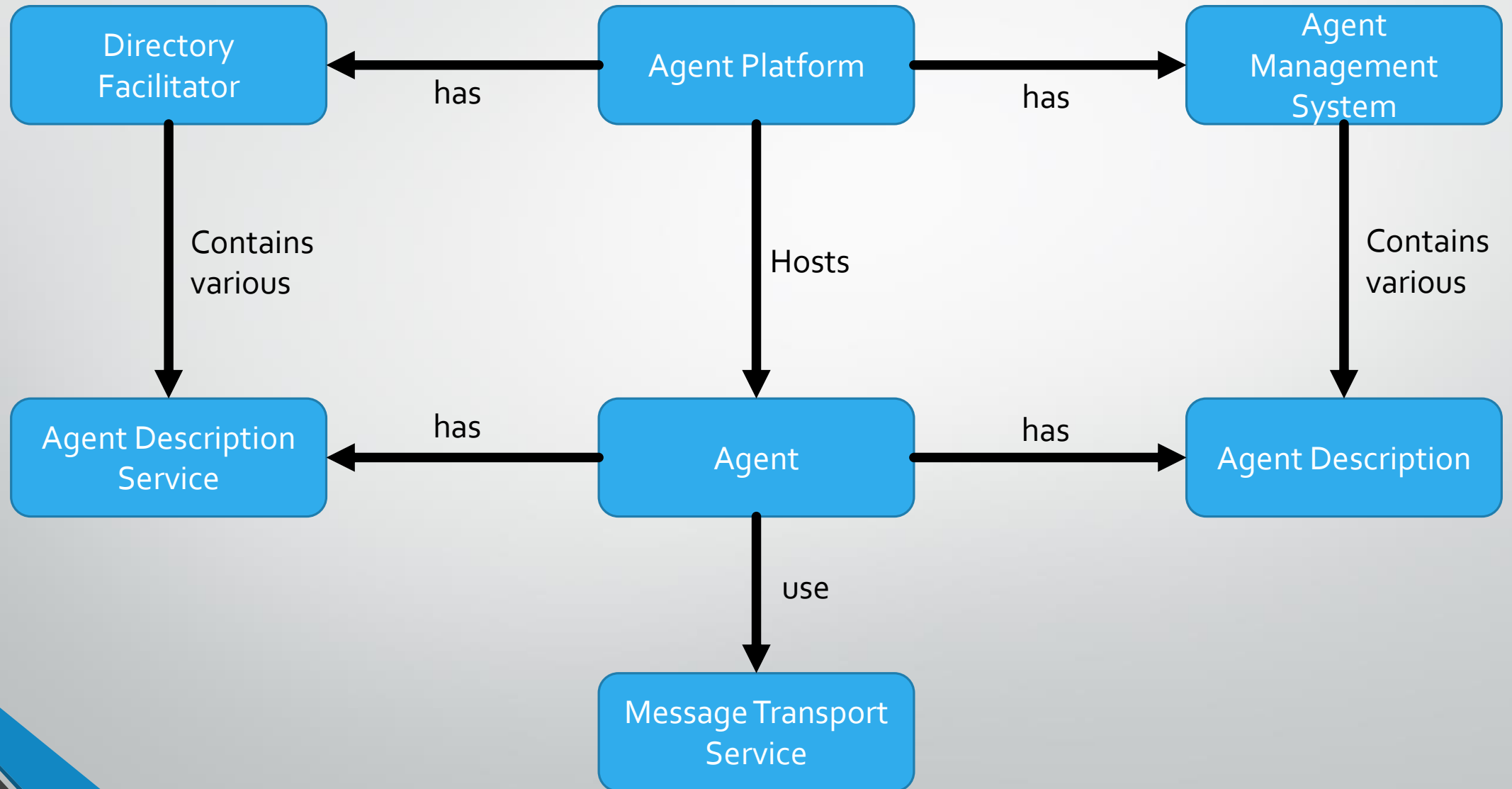
Main concepts

Platform:

- Set of distributed containers.



Agent UML Diagram



Main Concepts

- Performative-FIPA compliant:
 - FIPA - Foundation for Intelligent Physical Agents.
 - Promotes standards for agent development.
 - Delivers Messages Representation, Agents Structure, Messages Structure, etc.
 - FIPA Message Types: ASK, INFORM, REQUEST, PROPOSE, REFUSE, etc.

Behaviours

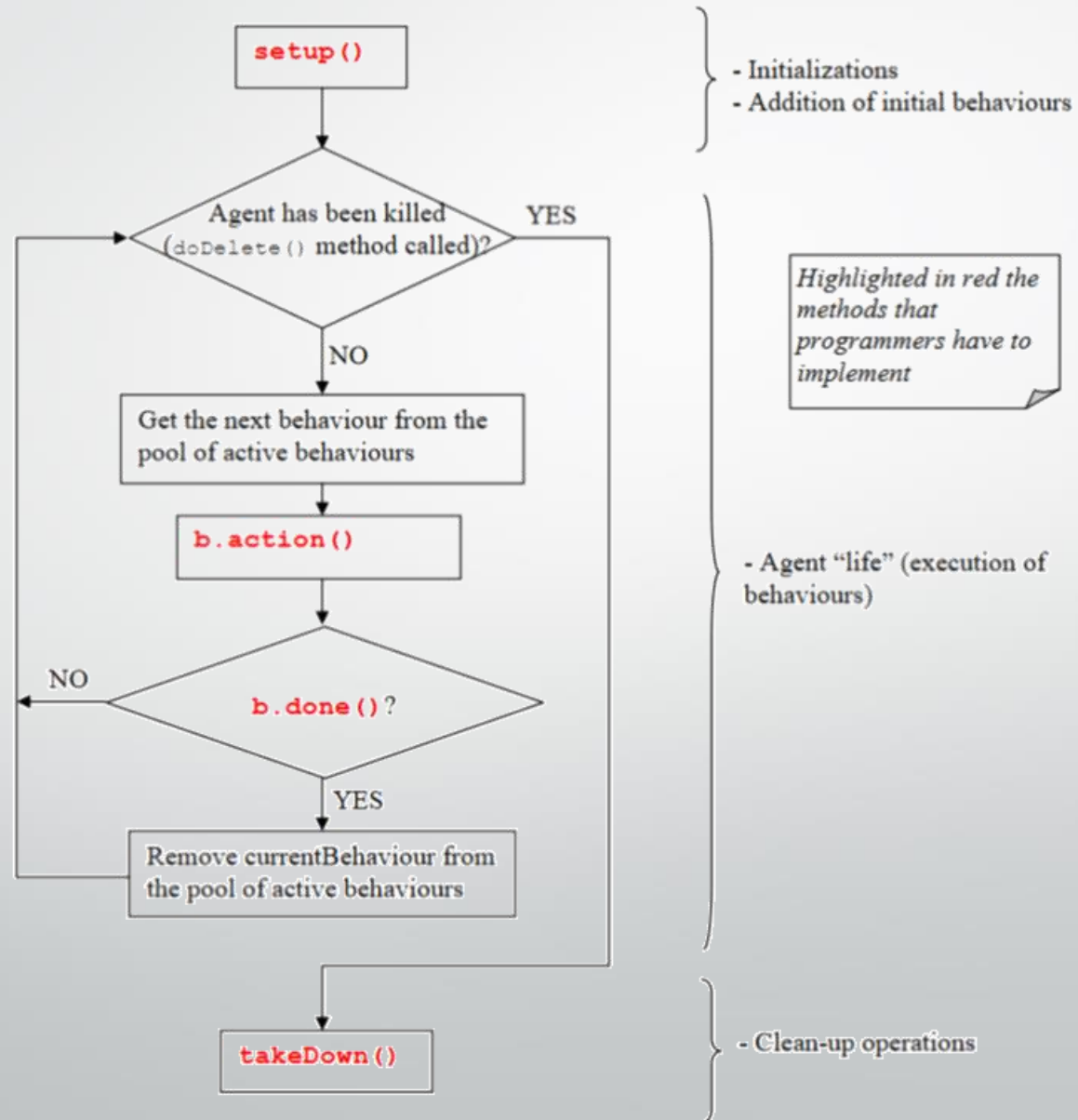
- Primitive:

- SimpleBehaviour
- CyclicBehaviour
- TickerBehaviour
- OneShotBehaviour
- WakerBehaviour
- SimpleAchieveREInitiator
- SimpleAchieveREResponder
- FSMBehaviour

- Complex:

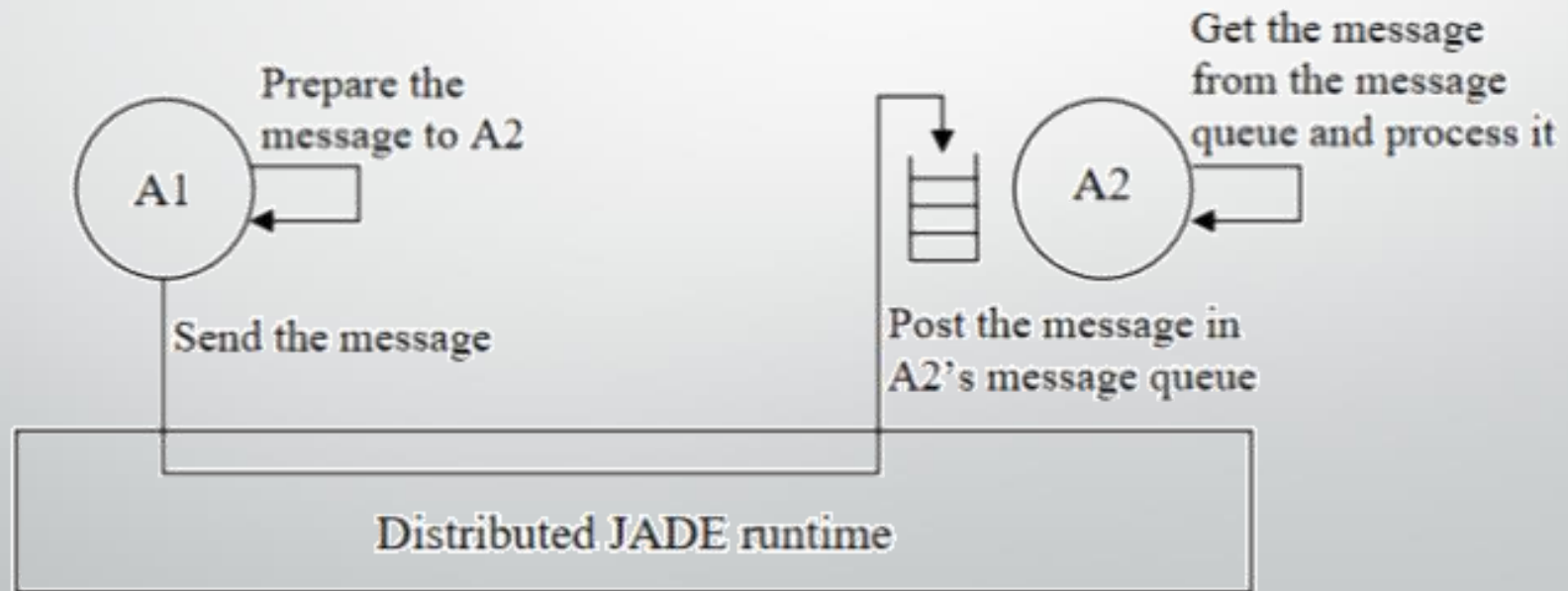
- ParallelBehaviour
- SequentialBehaviour

Life Cycle



Message Transfer

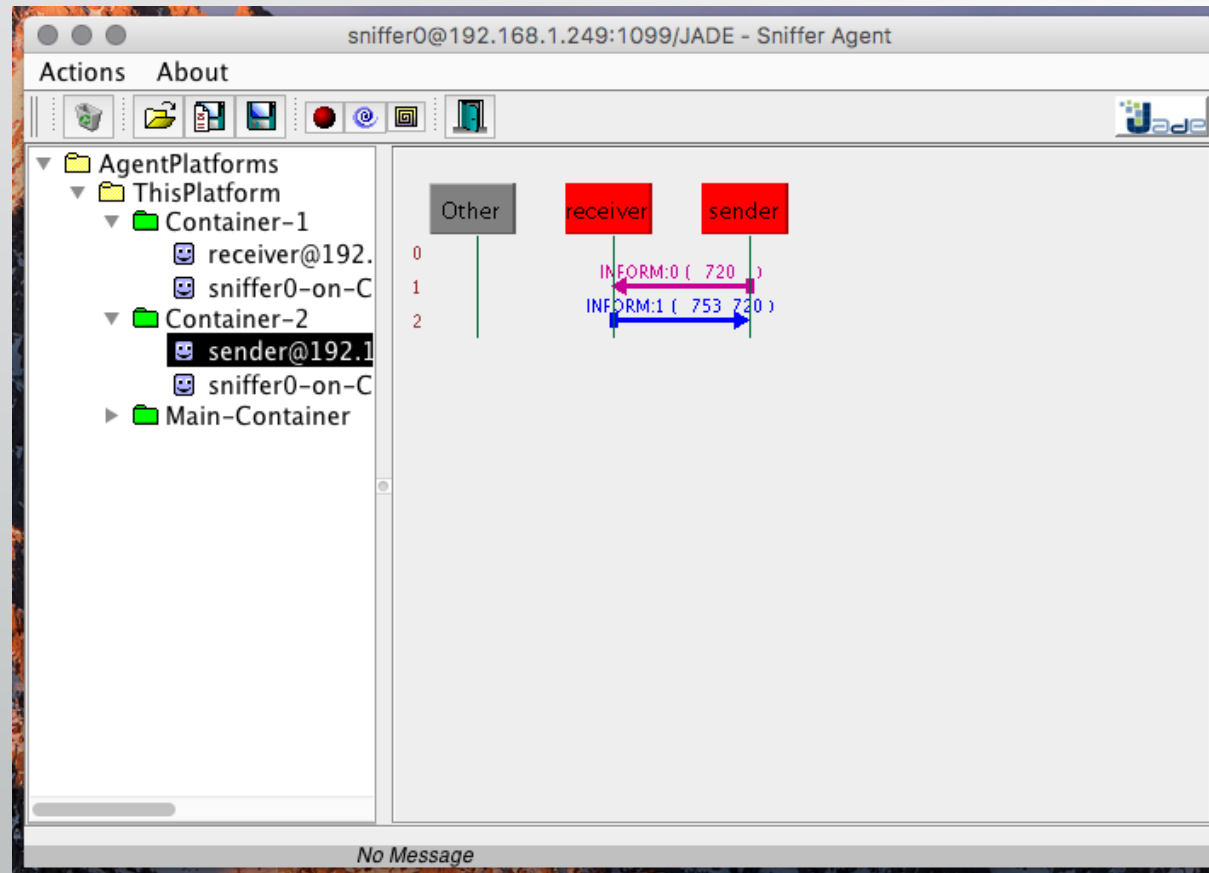
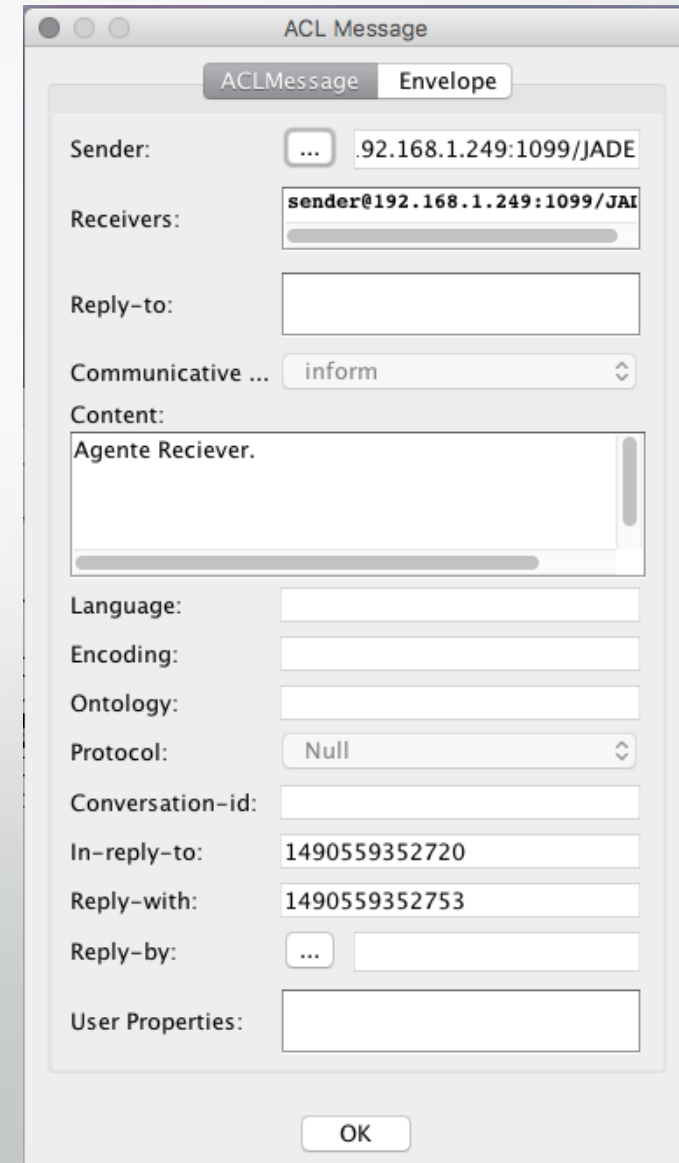
- Asynchronous message transfer.
- May have more than one receptor.



Message Example

```
(inform
  :sender      Admin@AdmM
  :receiver    AM0001@AmM
  :reply-with  1203302
  :language    XML
  :ontology    new_event
  :content     <new-event>
                <id>    0001      </id>
                <prio>  3         </prio>
                <sum>   "Visit dentist" </sum>
                <dateb>20090514   </dateb>
                <timeb>110000     </timeb>
                <datee>20090514   </datee>
                <timee>123000     </timee>
                </new-event>
  )
```

Sniffer

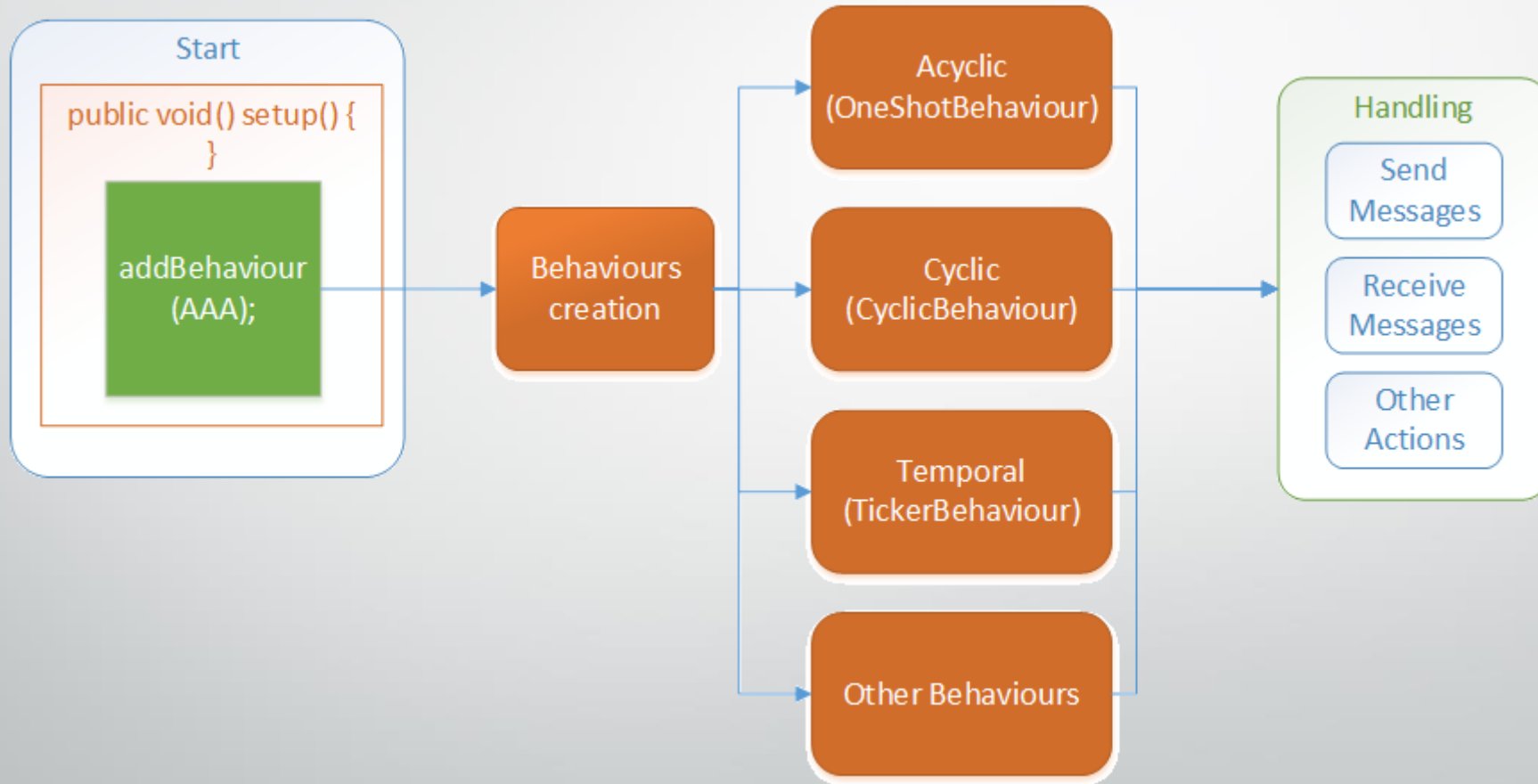



The screenshot shows the 'ACL Message' dialog box with the 'Envelope' tab selected. The fields are as follows:

- Sender:** ... 92.168.1.249:1099/JADE
- Receivers:** sender@192.168.1.249:1099/JADE
- Reply-to:** (empty)
- Communicative ...** inform
- Content:** Agente Reciever.
- Language:** (empty)
- Encoding:** (empty)
- Ontology:** (empty)
- Protocol:** Null
- Conversation-id:** (empty)
- In-reply-to:** 1490559352720
- Reply-with:** 1490559352753
- Reply-by:** ... (empty)
- User Properties:** (empty)

The 'OK' button is at the bottom right.

Development Process



Agent Skeleton

```
public class HelloAgent extends Agent {  
    public void setup() {  
        ...  
    }  
}
```

Behaviour Development

```
public class Hello extends Agent
{
    public void setup()
    {
        this.addBehaviour(do_something);
    }
}

private class do_something extends CyclicBehaviour {
    public void action()
    {
        // Actions that the behaviour will do
        // Such as actions to send/receive messages
    }
}
```

Agent Message

Create and send messages:

```
AID receiver = new AID();  
receiver.setLocalName("receiver");  
ACLMessage msg = new ACLMessage(ACLMessage.INFORM);  
msg.setContent("HelloWorld");  
msg.setConversationId(""+System.currentTimeMillis());  
msg.addReceiver(receiver);  
myAgent.send(msg);
```

Agent Message

Receive messages:

```
private ACLMessage msg = null;
if ((msg = myAgent.receive()) != null)
{
    System.out.println("New message received");
    System.out.println("Sender: " + msg.getSender().getLocalName());
    System.out.println("Name:" + msg.getSender().getName());
    System.out.println("Content: " + msg.getContent());
}
```

Start Platform

- Run the following command to start the Main-Container (console):
 - `java jade.Boot [Options] [Agents]`
- [Options]:
 - `-container`
 - `-host "HostName"` (default: localhost)
 - `-port "PortNumber"` (default: 1099)
 - `-gui` (RMA GUI)
- [Agents]:
 - Identifiers separated by spaces
 - Identifier is of type: `AgentName:JavaPackage.ClassName`

Start Platform Examples

Multiple containers

- In the same machine:

- Start JADE: `java -cp jade.Boot`
- Add a new container with an agent: `java -cp jade.Boot -container`

- In different machines:

- Start JADE in machine PCAna: `java -cp jade.Boot`
- Add new container in the machine PCJohn with an agent:
 - `java -cp jade.Boot -host PCAna -container a:ClasseA`
 - Agent a lives in the machine PCJohn, but belongs to the platform running in the machine PCAna; its GUID (Globally Unique ID) is `a@PCAna:1099/JADE`

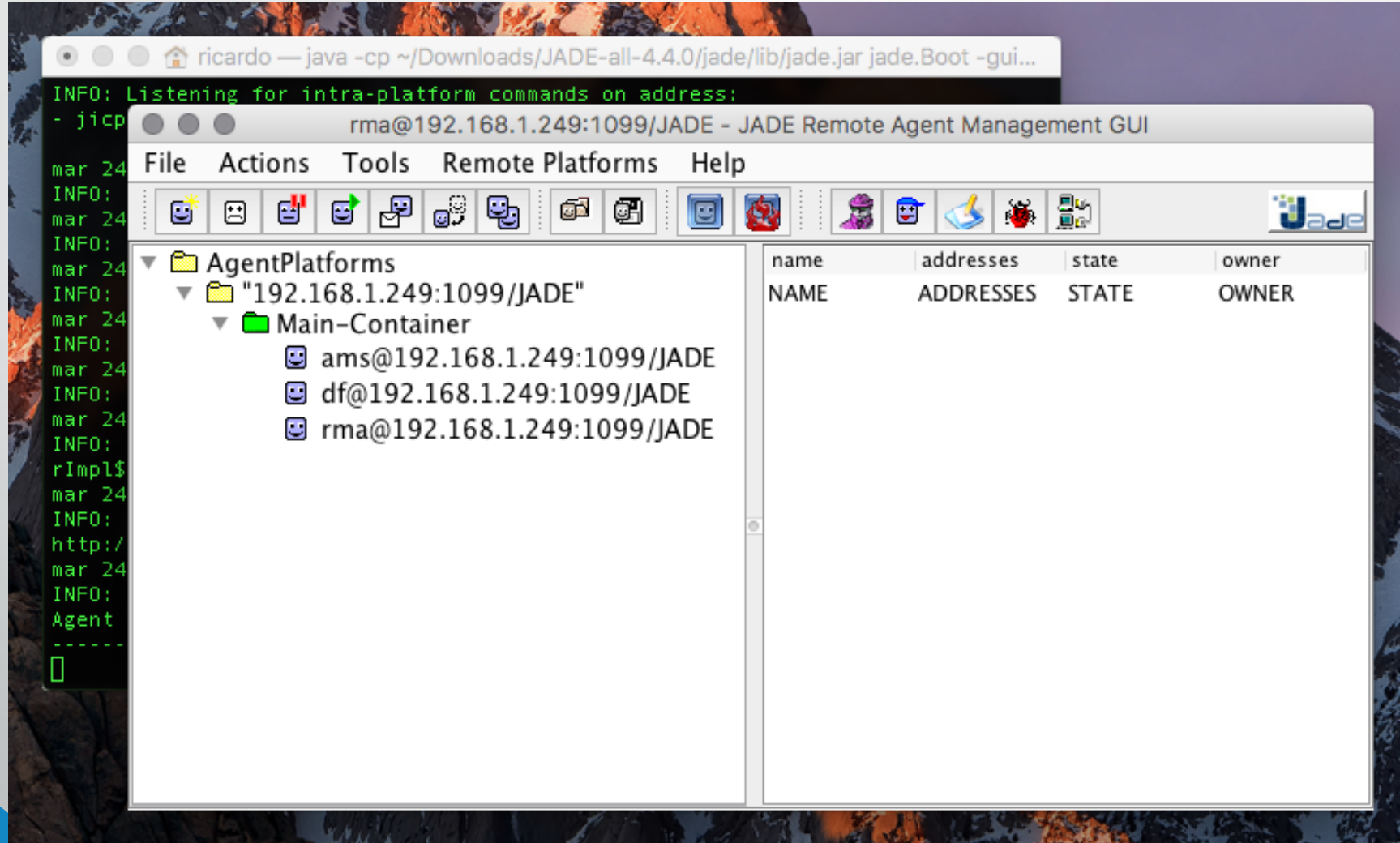
Start Platform

Tools that simplify platform administration and Application development:

- Remote Monitoring Agent;
- Dummy Agent;
- Sniffer Agent;
- Introspector Agent;
- Directory Facilitator GUI;

JADE Start

```
java -cp /Users/ricardo/Downloads/JADE-all-4.4.0/jade/lib/jade.jar jade.Boot -gui
```

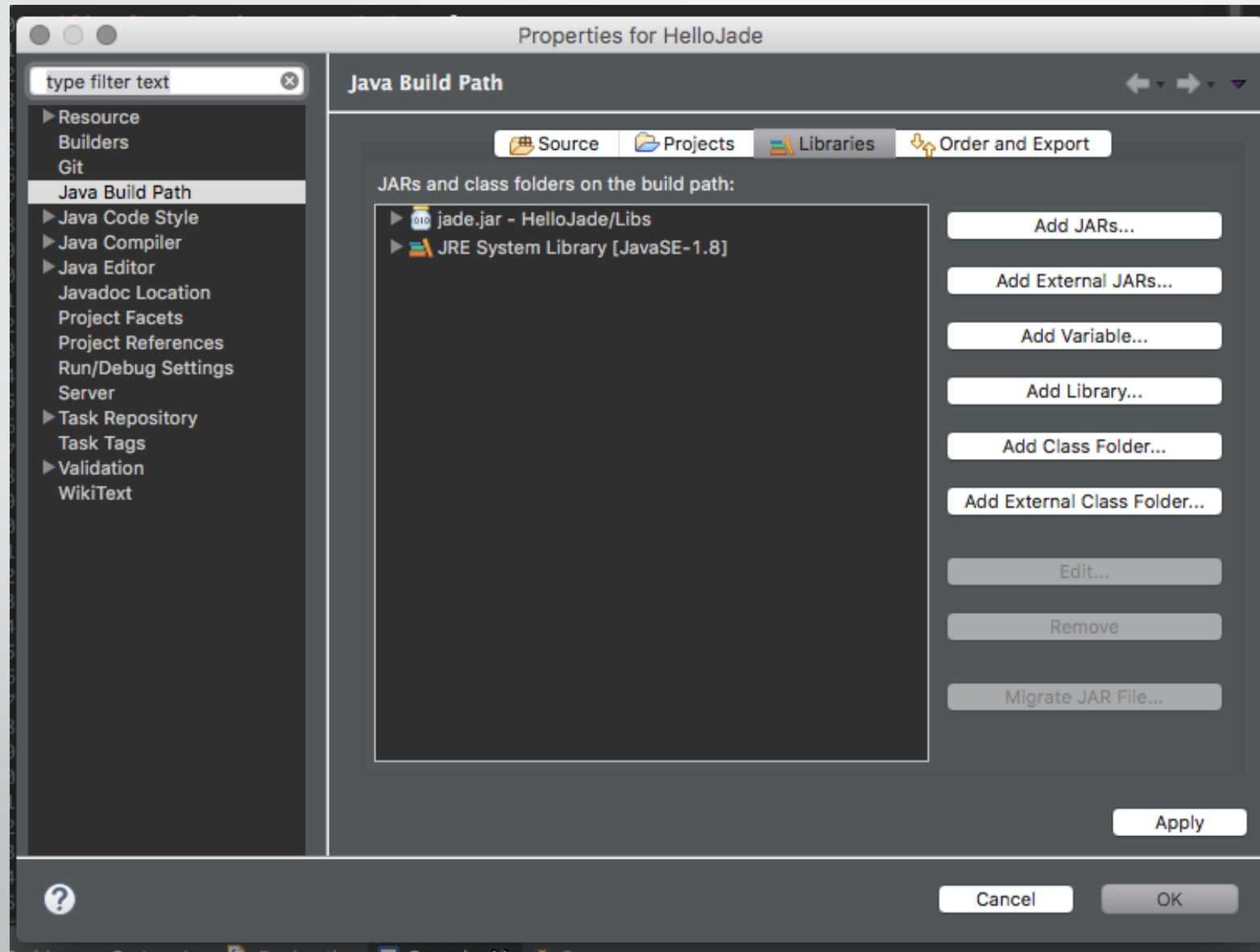


JADE Start

```
[iMac-de-Ricardo:~ ricardo$ java -cp /Users/ricardo/Downloads/JADE-all-4.4.0/jade/
/lib/jade.jar jade.Boot -gui
mar 26, 2017 9:25:42 PM jade.core.Runtime beginContainer
INFO: -----
      This is JADE 4.4.0 - revision 6778 of 21-12-2015 12:24:43
      downloaded in Open Source, under LGPL restrictions,
      at http://jade.tilab.com/
      -----
mar 26, 2017 9:25:42 PM jade.imtp.leap.LEAPIMTPManager initialize
INFO: Listening for intra-platform commands on address:
- jicp://192.168.1.249:1099

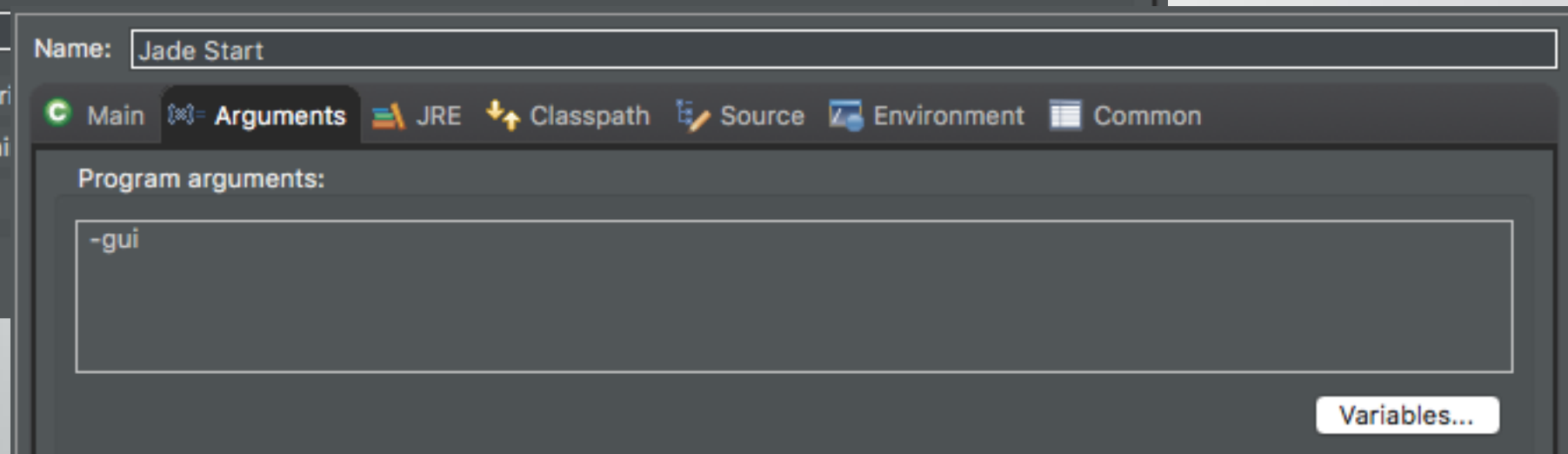
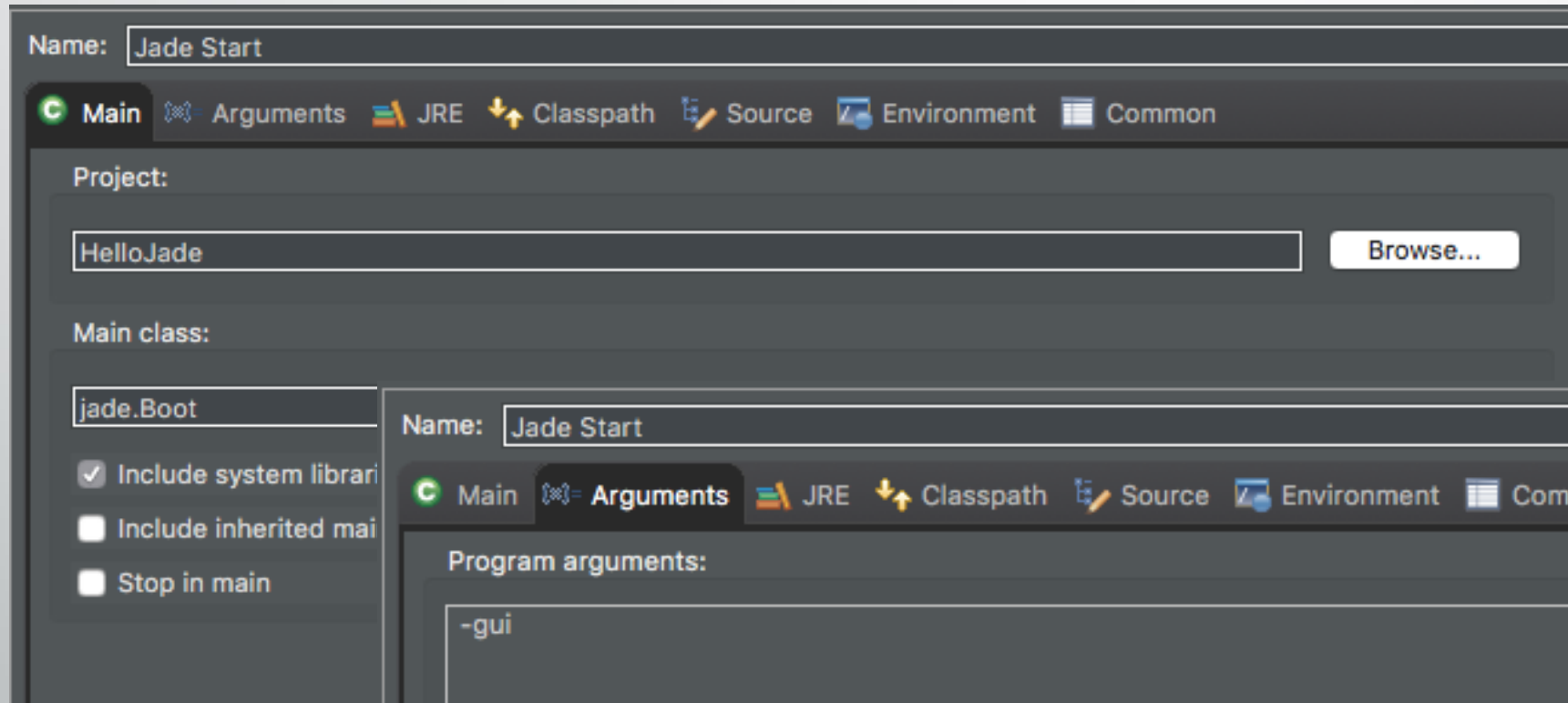
mar 26, 2017 9:25:43 PM jade.core.BaseService init
INFO: Service jade.core.management.AgentManagement initialized
mar 26, 2017 9:25:43 PM jade.core.BaseService init
INFO: Service jade.core.messaging.Messaging initialized
mar 26, 2017 9:25:43 PM jade.core.BaseService init
INFO: Service jade.core.resource.ResourceManagement initialized
mar 26, 2017 9:25:43 PM jade.core.BaseService init
INFO: Service jade.core.mobility.AgentMobility initialized
mar 26, 2017 9:25:43 PM jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
mar 26, 2017 9:25:43 PM jade.mtp.http.HTTPServer <init>
INFO: HTTP-MTP Using XML parser com.sun.org.apache.xerces.internal.jaxp.SAXParse
rImpl$JAXPSAXParser
mar 26, 2017 9:25:43 PM jade.core.messaging.MessagingService boot
INFO: MTP addresses:
http://192.168.1.249:7778/acc
mar 26, 2017 9:25:43 PM jade.core.AgentContainerImpl joinPlatform
INFO: -----
Agent container Main-Container@192.168.1.249 is ready.
-----
```

Eclipse Setup



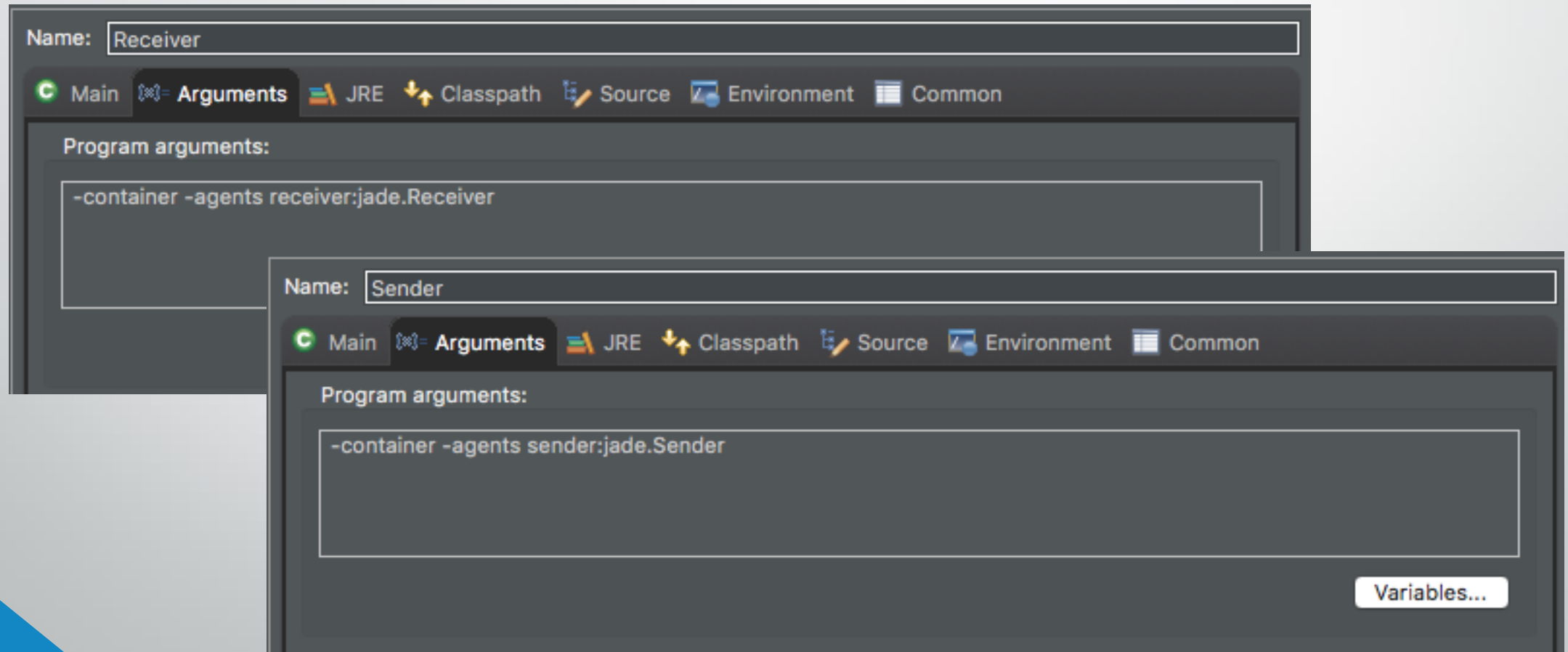
Launch Agent

Run Configurations:



Execute Agent

Run Configurations:



Basic Agent

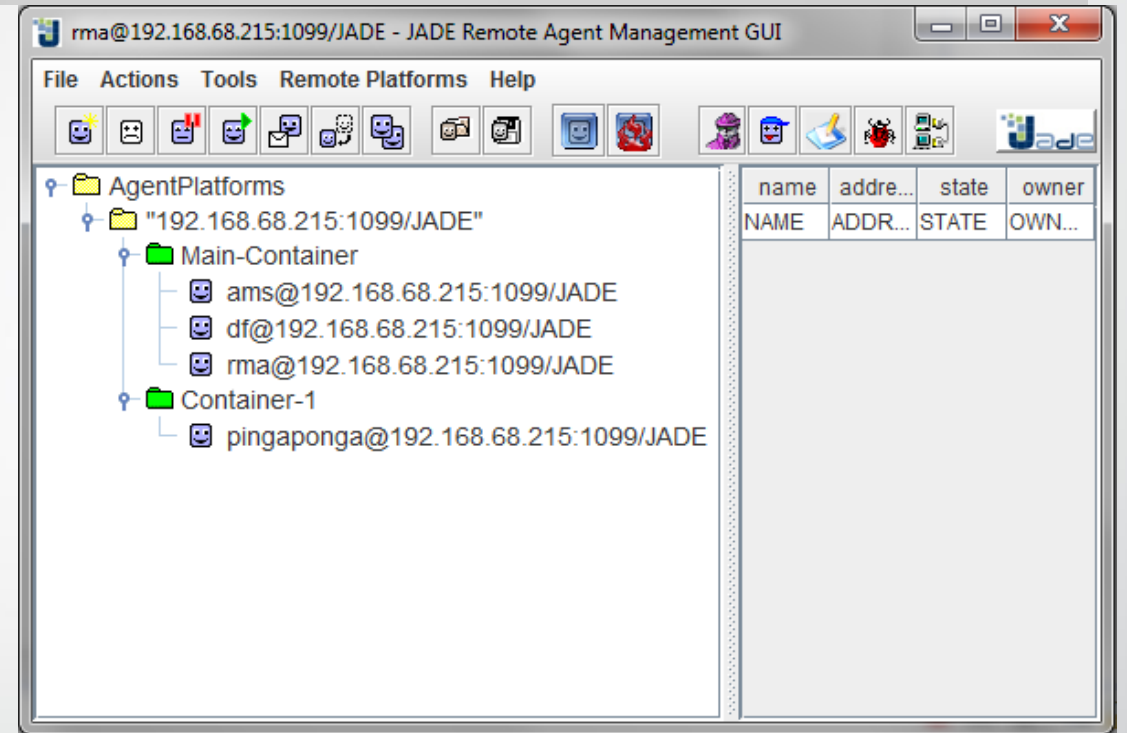
```
public class PingPong extends Agent
{
    @Override
    protected void setup()
    {
        super.setup();

        System.out.println(this.getLocalName()+" a começar!");
    }

    @Override
    protected void takeDown()
    {
        super.takeDown();

        System.out.println(this.getLocalName()+" a morrer...");
    }
}
```

```
set 15, 2015 10:17:58 AM jade.core.BaseService init
INFO: Service jade.core.management.AgentManagement initialized
set 15, 2015 10:17:58 AM jade.core.BaseService init
INFO: Service jade.core.messaging.Messaging initialized
set 15, 2015 10:17:58 AM jade.core.BaseService init
INFO: Service jade.core.resource.ResourceManagement initialized
set 15, 2015 10:17:58 AM jade.core.BaseService init
INFO: Service jade.core.mobility.AgentMobility initialized
set 15, 2015 10:17:58 AM jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
set 15, 2015 10:17:58 AM jade.core.AgentContainerImpl joinPlatform
INFO: -----
Agent container Container-1@192.168.68.215 is ready.
-----
pingaponga a começar!
```



In order to execute takeDown() method, the agent must invoke the doDelete() method inside the setup() method.

Message Receive

```
public class PingPong extends Agent
{
    @Override
    protected void setup()
    {
        super.setup();
        System.out.println(this.getLocalName()+" a começar!");

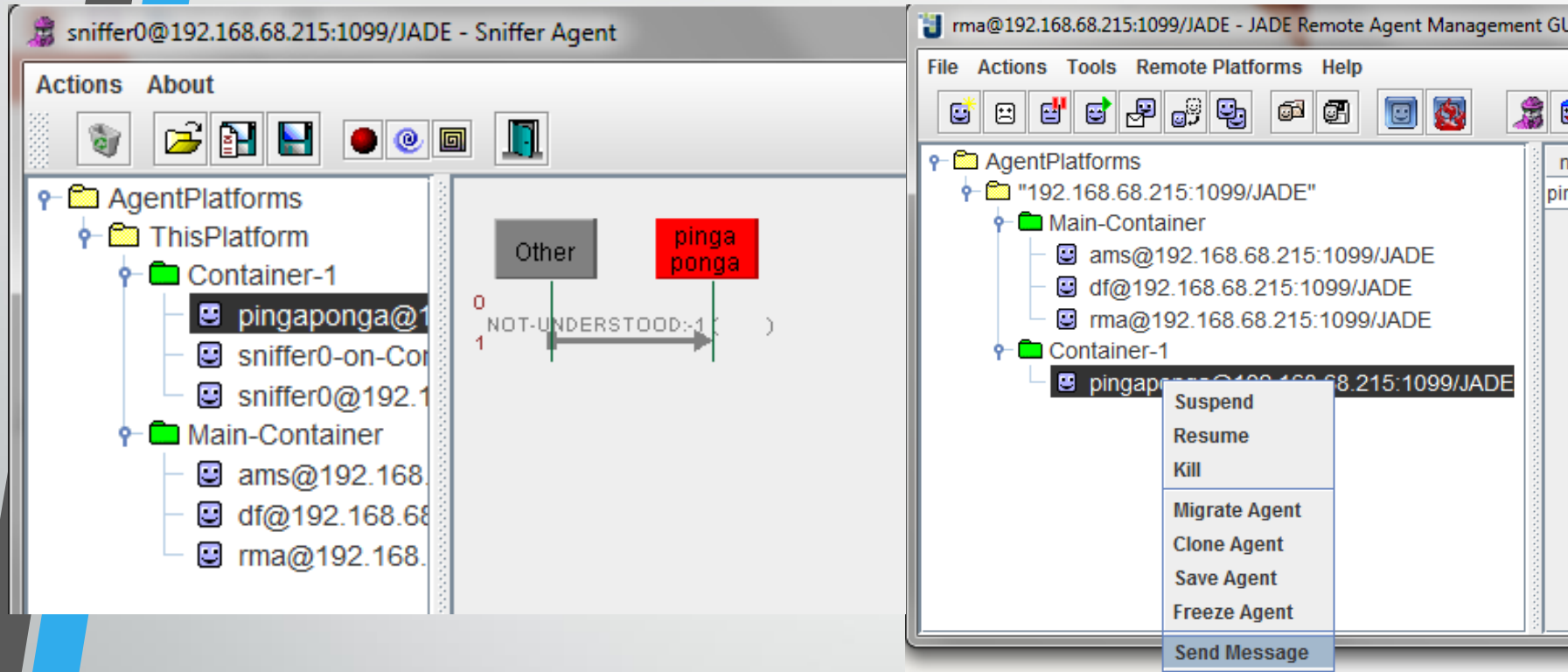
        this.addBehaviour(new ReceiveBehavior());
    }

    @Override
    protected void takeDown()
    {
        super.takeDown();

        System.out.println(this.getLocalName()+" a morrer...");
    }

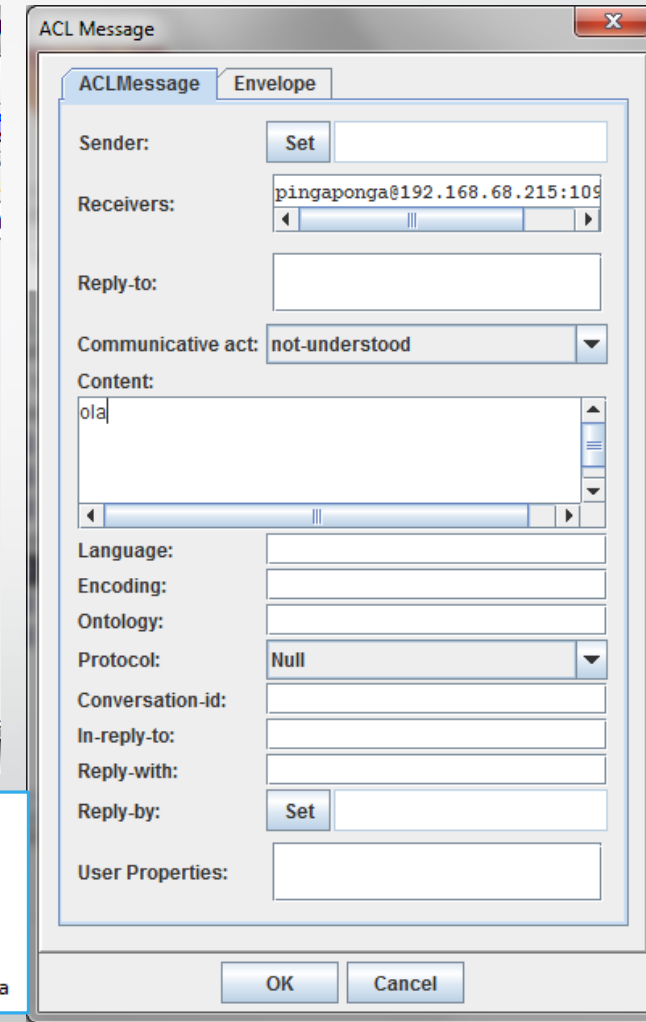
    private class ReceiveBehavior extends CyclicBehaviour
    {
        @Override
        public void action()
        {
            ACLMessage msg = receive();
            if (msg != null)
            {
                System.out.println("Recebi uma mensagem de "+msg.getSender()+" . Conteúdo: "+msg.getContent());
            }
            block();
        }
    }
};
```


Message Receive



The image shows two windows from the JADE (Java Agent Development Environment) interface. The left window, titled "sniffer0@192.168.68.215:1099/JADE - Sniffer Agent", displays a tree view of the agent platform structure. It includes "AgentPlatforms", "ThisPlatform", "Container-1", and "Main-Container". A diagram in the center shows a message flow from "Other" to "pinga ponga". The right window, titled "rma@192.168.68.215:1099/JADE - JADE Remote Agent Management GUI", shows a tree view of the remote agent platform. It includes "AgentPlatforms", "192.168.68.215:1099/JADE", "Main-Container", and "Container-1". A context menu is open over the "pinga ponga" agent, with options like "Suspend", "Resume", "Kill", "Migrate Agent", "Clone Agent", "Save Agent", "Freeze Agent", and "Send Message".

```
set 15, 2015 10:23:28 AM jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
pingaponga a começar!
set 15, 2015 10:23:28 AM jade.core.AgentContainerImpl joinPlatform
INFO: -----
Agent container Container-2@192.168.68.215 is ready.
-----
Recebi uma mensagem de ( agent-identifier :name rma@192.168.68.215:1099/JADE :addresses (sequence http://192.168.68.215:7778/acc ) ). Conteúdo: ola
```



The image shows the "ACL Message" dialog box. It has two tabs: "ACLMessage" and "Envelope". The "ACLMessage" tab is active. It contains fields for "Sender:", "Receivers:", "Reply-to:", "Communicative act:", "Content:", "Language:", "Encoding:", "Ontology:", "Protocol:", "Conversation-id:", "In-reply-to:", "Reply-with:", "Reply-by:", and "User Properties:". The "Receivers:" field is set to "pingaponga@192.168.68.215:1099". The "Communicative act:" is set to "not-understood". The "Content:" field contains "ola". The "Protocol:" is set to "Null". The "Reply-by:" field has a "Set" button. The "User Properties:" field is empty. There are "OK" and "Cancel" buttons at the bottom.

Message Reply

```
private class AnswerBehavior extends CyclicBehaviour
{
    @Override
    public void action()
    {
        ACLMessage msg = receive();
        if (msg != null)
        {
            System.out.println("Recebi uma mensagem de "+msg.getSender()+" Conteúdo: "+msg.getContent());
            ACLMessage resp = msg.createReply();

            if (msg.getContent().equals("ping"))
            {
                resp.setContent("pong");
                resp.setPerformative(ACLMessage.INFORM);
            }
            else
            {
                resp.setContent("Não percebi...");
                resp.setPerformative(ACLMessage.NOT_UNDERSTOOD);
            }

            send(resp);
        }
        block();
    }
}
```

```
@Override
protected void setup()
{
    super.setup();
    System.out.println(this.getLocalName()+" a começar!");

    //this.addBehaviour(new ReceiveBehavior());
    this.addBehaviour(new AnswerBehavior());
}
```

Cyclic Message Send

```
public class CyclicSender extends Agent
{
    @Override
    protected void setup()
    {
        super.setup();
        this.addBehaviour(new SendMessage());
    }

    private class SendMessage extends CyclicBehaviour
    {
        @Override
        public void action()
        {
            AID receiver = new AID();
            receiver.setLocalName("pingaponga");

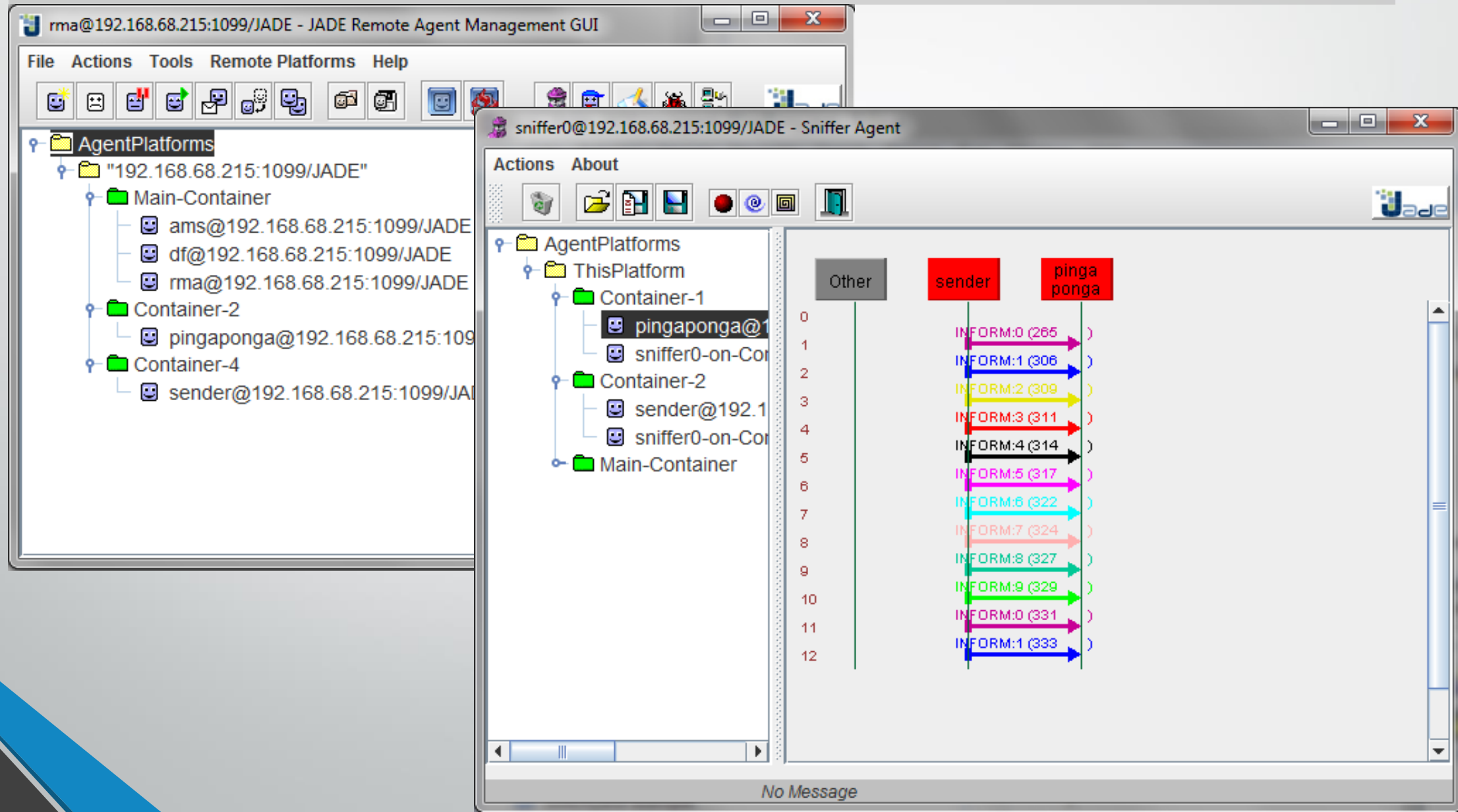
            ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
            long time = System.currentTimeMillis();
            msg.setConversationId(""+time);
            msg.addReceiver(receiver);

            if (time % 2 == 0)
                msg.setContent("ping");
            else
                msg.setContent("not ping");

            myAgent.send(msg);

            block(1000);
        }
    }
}
```

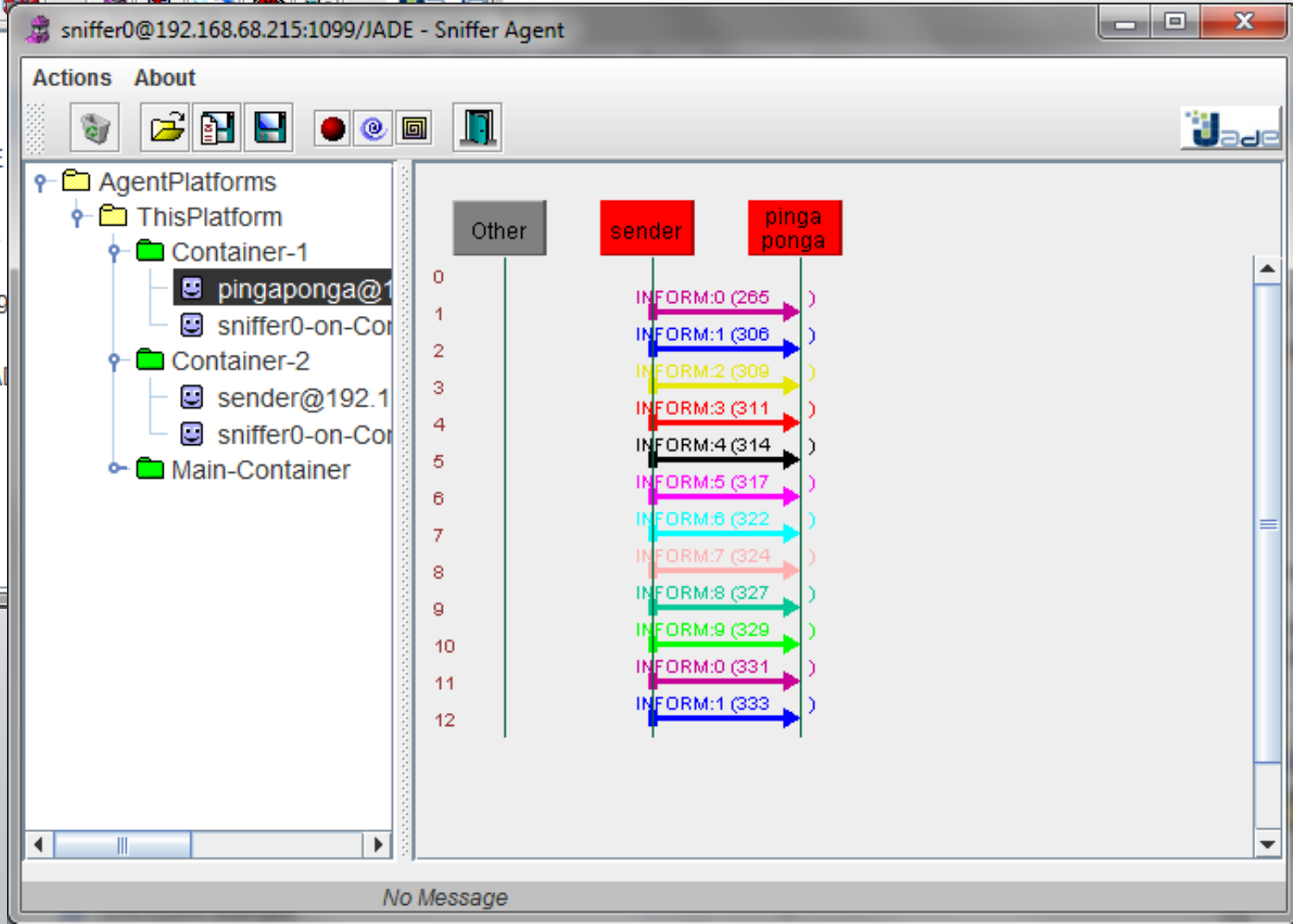
Cyclic Message Send



The image displays two windows from the JADE (Java Agent Development Environment) system. The background window is the 'JADE Remote Agent Management GUI' showing a hierarchy of agent platforms. The foreground window is the 'Sniffer Agent' interface, which is capturing and displaying messages between agents.

JADE Remote Agent Management GUI:

- Platform: rma@192.168.68.215:1099/JADE
- AgentPlatforms:
 - "192.168.68.215:1099/JADE"
 - Main-Container
 - ams@192.168.68.215:1099/JADE
 - df@192.168.68.215:1099/JADE
 - rma@192.168.68.215:1099/JADE
 - Container-2
 - pingaponga@192.168.68.215:1099/JADE
 - Container-4
 - sender@192.168.68.215:1099/JADE



The 'Sniffer Agent' window shows the following structure:

- AgentPlatforms:
 - ThisPlatform
 - Container-1
 - pingaponga@192.168.68.215:1099/JADE
 - sniffer0-on-Container-1
 - Container-2
 - sender@192.168.68.215:1099/JADE
 - sniffer0-on-Container-2
 - Main-Container

The message capture area shows a cyclic exchange between the 'sender' and 'pinga ponga' agents. The messages are as follows:

Index	From	To	Message
0	sender	pinga ponga	INFORM:0 (265)
1	pinga ponga	sender	INFORM:1 (306)
2	sender	pinga ponga	INFORM:2 (309)
3	pinga ponga	sender	INFORM:3 (311)
4	sender	pinga ponga	INFORM:4 (314)
5	pinga ponga	sender	INFORM:5 (317)
6	sender	pinga ponga	INFORM:6 (322)
7	pinga ponga	sender	INFORM:7 (324)
8	sender	pinga ponga	INFORM:8 (327)
9	pinga ponga	sender	INFORM:9 (329)
10	sender	pinga ponga	INFORM:0 (331)
11	pinga ponga	sender	INFORM:1 (333)
12	sender	pinga ponga	INFORM:2 (335)

The status bar at the bottom indicates 'No Message'.

Cyclic Message Send (limited run)

```
public class CyclicSender4Times extends Agent {

    int counter=0;

    @Override
    protected void setup(){

        super.setup();

        System.out.println(this.getLocalName()+" a começar!");

        this.addBehaviour(new sendMessage());

    }

    @Override
    protected void takeDown(){

        System.out.println(this.getLocalName()+" a morrer!");

    }

}
```

```
public class sendMessage extends SimpleBehaviour {

    @Override
    public int onEnd() {
        myAgent.doDelete();
        return 0;
    }

    @Override
    public void action(){

        AID receiver=new AID();
        receiver.setLocalName("pingaponga");

        ACLMessage msg =new ACLMessage(ACLMessage.INFORM);
        long time =System.currentTimeMillis();
        msg.setConversationId(""+time);
        msg.addReceiver(receiver);

        if(time % 2 == 0)
            msg.setContent("ping");
        else
            msg.setContent("not ping");

        myAgent.send(msg);

        counter++;

        block(1000);

    }

    @Override
    public boolean done() {
        if (counter>=4){
            return true;
        }else return false;
    }

}
```

Cyclic Message Send (limited run)

- Configurations:
 - container –agents pingaponga:agents.PingPong;sender:agents.CyclicSender4Times

```

out 23, 2016 2:31:48 PM jade.core.Runtime beginContainer
INFO: -----
      This is JADE 4.3.3 - revision 6726 of 2014/12/09 09:33:02
      downloaded in Open Source, under LGPL restrictions,
      at http://jade.tilab.com/
-----
out 23, 2016 2:31:48 PM jade.imtp.leap.LEAPIMTPManager initialize
INFO: Listening for intra-platform commands on address:
- jicp://192.168.1.76:62333

out 23, 2016 2:31:48 PM jade.core.BaseService init
INFO: Service jade.core.management.AgentManagement initialized
out 23, 2016 2:31:48 PM jade.core.BaseService init
INFO: Service jade.core.messaging.Messaging initialized
out 23, 2016 2:31:48 PM jade.core.BaseService init
INFO: Service jade.core.resource.ResourceManagement initialized
out 23, 2016 2:31:48 PM jade.core.BaseService init
INFO: Service jade.core.mobility.AgentMobility initialized
out 23, 2016 2:31:48 PM jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
sender a começar!out 23, 2016 2:31:48 PM jade.core.AgentContainerImpl joinPlatform
INFO: -----
Agent container Container-18@192.168.1.76 is ready.
-----

pingaponga a começar!
Recebi uma mensagem de ( agent-identifier :name sender@192.168.1.76:1099/JADE :addresses (sequence http://Laptop.lan:7778/acc )) .Conteúdo: ping
Recebi uma mensagem de ( agent-identifier :name sender@192.168.1.76:1099/JADE :addresses (sequence http://Laptop.lan:7778/acc )) .Conteúdo: ping
Recebi uma mensagem de ( agent-identifier :name sender@192.168.1.76:1099/JADE :addresses (sequence http://Laptop.lan:7778/acc )) .Conteúdo: not ping
Recebi uma mensagem de ( agent-identifier :name sender@192.168.1.76:1099/JADE :addresses (sequence http://Laptop.lan:7778/acc )) .Conteúdo: ping
sender a morrer!

```

Two-way Message Sending

```
private class SendMessage extends TickerBehaviour
{
    public SendMessage(Agent a, long timeout)
    {
        super(a, timeout);
    }

    @Override
    protected void onTick()
    {
        AID receiver = new AID();
        receiver.setLocalName("pingaponga");

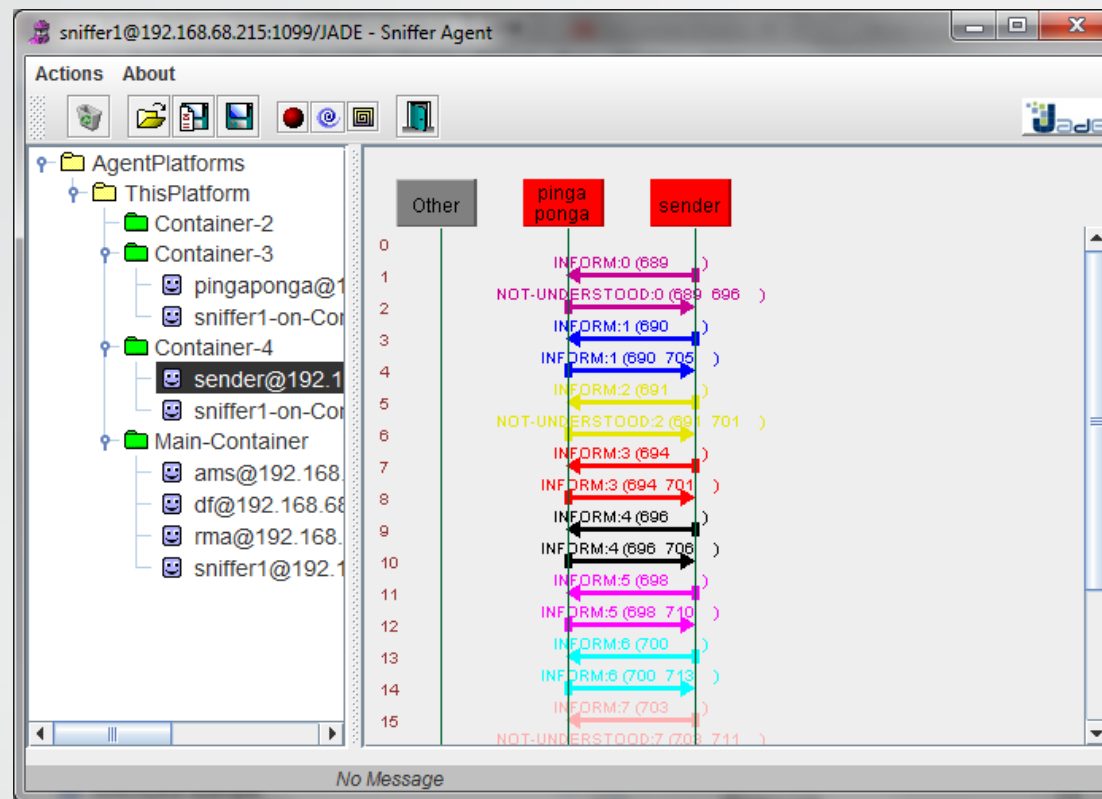
        ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
        long time = System.currentTimeMillis();
        msg.setConversationId(""+time);
        msg.addReceiver(receiver);

        if (time % 2 == 0)
            msg.setContent("ping");
        else
            msg.setContent("not ping");

        myAgent.send(msg);
    }
}
```

```
@Override
protected void setup()
{
    super.setup();
    this.addBehaviour(new SendMessage(this, 1000));
    this.addBehaviour(new ReceiveBehavior());
}
```

```
private class ReceiveBehavior extends CyclicBehaviour
{
    @Override
    public void action()
    {
        ACLMessage msg = receive();
        if (msg != null)
        {
            System.out.println("Recebi uma mensagem de "+msg.getSender()+" Conteúdo: "+msg.getContent());
        }
        block();
    }
};
```

```

Recebi uma mensagem de ( agent-identifier :name sender@192.168.68.215:1099/JADE :addresses (sequence http://192.168.68.215:7778/acc )) . Conteúdo: not ping
Recebi uma mensagem de ( agent-identifier :name sender@192.168.68.215:1099/JADE :addresses (sequence http://192.168.68.215:7778/acc )) . Conteúdo: ping
Recebi uma mensagem de ( agent-identifier :name sender@192.168.68.215:1099/JADE :addresses (sequence http://192.168.68.215:7778/acc )) . Conteúdo: ping
Recebi uma mensagem de ( agent-identifier :name sender@192.168.68.215:1099/JADE :addresses (sequence http://192.168.68.215:7778/acc )) . Conteúdo: ping
Recebi uma mensagem de ( agent-identifier :name sender@192.168.68.215:1099/JADE :addresses (sequence http://192.168.68.215:7778/acc )) . Conteúdo: ping
Recebi uma mensagem de ( agent-identifier :name sender@192.168.68.215:1099/JADE :addresses (sequence http://192.168.68.215:7778/acc )) . Conteúdo: not ping
Recebi uma mensagem de ( agent-identifier :name sender@192.168.68.215:1099/JADE :addresses (sequence http://192.168.68.215:7778/acc )) . Conteúdo: ping
Recebi uma mensagem de ( agent-identifier :name sender@192.168.68.215:1099/JADE :addresses (sequence http://192.168.68.215:7778/acc )) . Conteúdo: not ping
Recebi uma mensagem de ( agent-identifier :name sender@192.168.68.215:1099/JADE :addresses (sequence http://192.168.68.215:7778/acc )) . Conteúdo: ping
Recebi uma mensagem de ( agent-identifier :name sender@192.168.68.215:1099/JADE :addresses (sequence http://192.168.68.215:7778/acc )) . Conteúdo: not ping
Recebi uma mensagem de ( agent-identifier :name sender@192.168.68.215:1099/JADE :addresses (sequence http://192.168.68.215:7778/acc )) . Conteúdo: ping

Recebi uma mensagem de ( agent-identifier :name pingaponga@192.168.68.215:1099/JADE :addresses (sequence http://192.168.68.215:7778/acc )) . Conteúdo: Não percebi...
Recebi uma mensagem de ( agent-identifier :name pingaponga@192.168.68.215:1099/JADE :addresses (sequence http://192.168.68.215:7778/acc )) . Conteúdo: Não percebi...
Recebi uma mensagem de ( agent-identifier :name pingaponga@192.168.68.215:1099/JADE :addresses (sequence http://192.168.68.215:7778/acc )) . Conteúdo: pong
Recebi uma mensagem de ( agent-identifier :name pingaponga@192.168.68.215:1099/JADE :addresses (sequence http://192.168.68.215:7778/acc )) . Conteúdo: Não percebi...
Recebi uma mensagem de ( agent-identifier :name pingaponga@192.168.68.215:1099/JADE :addresses (sequence http://192.168.68.215:7778/acc )) . Conteúdo: Não percebi...
Recebi uma mensagem de ( agent-identifier :name pingaponga@192.168.68.215:1099/JADE :addresses (sequence http://192.168.68.215:7778/acc )) . Conteúdo: pong
Recebi uma mensagem de ( agent-identifier :name pingaponga@192.168.68.215:1099/JADE :addresses (sequence http://192.168.68.215:7778/acc )) . Conteúdo: Não percebi...
Recebi uma mensagem de ( agent-identifier :name pingaponga@192.168.68.215:1099/JADE :addresses (sequence http://192.168.68.215:7778/acc )) . Conteúdo: Não percebi...
Recebi uma mensagem de ( agent-identifier :name pingaponga@192.168.68.215:1099/JADE :addresses (sequence http://192.168.68.215:7778/acc )) . Conteúdo: Não percebi...
Recebi uma mensagem de ( agent-identifier :name pingaponga@192.168.68.215:1099/JADE :addresses (sequence http://192.168.68.215:7778/acc )) . Conteúdo: pong
Recebi uma mensagem de ( agent-identifier :name pingaponga@192.168.68.215:1099/JADE :addresses (sequence http://192.168.68.215:7778/acc )) . Conteúdo: Não percebi...

```

Multiple Behaviours

```
@Override
protected void setup()
{
    super.setup();
    this.addBehaviour(new ReceiveBehavior());
    this.addBehaviour(new NewSendMessage("ping", this, 5000));
    this.addBehaviour(new NewSendMessage("pong", this, 1000));
    this.addBehaviour(new NewSendMessage("ola!", this, 2500));
}

private class NewSendMessage extends TickerBehaviour
{
    String content;
    public NewSendMessage(String content, Agent a, long timeout)
    {
        super(a, timeout);
        this.content = content;
    }

    @Override
    protected void onTick()
    {
        AID receiver = new AID();
        receiver.setLocalName("pingaponga");

        ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
        msg.setConversationId(""+System.currentTimeMillis());
        msg.addReceiver(receiver);
        msg.setContent(content);

        myAgent.send(msg);
    }
}
```


Multiple Behaviours

[illegible]

Parallel Behaviour

```
public class SenderParallel extends Agent {

    int counter1=0;
    int counter2=0;
    int counter3=0;

    @Override
    protected void setup(){

        super.setup();

        System.out.println(this.getLocalName()+" a começar!");

        ParallelBehaviour b= new ParallelBehaviour(this,ParallelBehaviour.WHEN_ALL){

            @Override
            public int onEnd(){

                System.out.println("ParallelBehaviour finalizing");
                myAgent.delete();
                return 0;
            }
        };

        this.addBehaviour(b);
        b.addSubBehaviour(new SendMessage1());
        b.addSubBehaviour(new SendMessage2());
        b.addSubBehaviour(new SendMessage3());

    }

    @Override
    protected void takeDown(){

        System.out.println(this.getLocalName()+" a morrer!");

    }

}
```

```
public class SendMessage1 extends SimpleBehaviour {

    @Override
    public void action(){

        AID receiver=new AID();
        receiver.setLocalName("pingaponga");

        ACLMessage msg =new ACLMessage(ACLMessage.INFORM);
        long time =System.currentTimeMillis();
        msg.setConversationId(""+time);
        msg.addReceiver(receiver);

        if(time % 2 == 0)
            msg.setContent("ping1");
        else
            msg.setContent("not ping1");

        myAgent.send(msg);

        counter1++;

        System.out.println("Executing SendMessage1 for the "+counter1+" time!");

    }

    @Override
    public boolean done() {
        if (counter1>=3){
            System.out.println("Behaviour SendMessage1 finalizing!");
            return true;
        }else return false;
    }

}
```

Parallel Behaviour

```
public class SendMessage2 extends SimpleBehaviour {

    @Override
    public void action(){

        AID receiver=new AID();
        receiver.setLocalName("pingaponga");

        ACLMessage msg =new ACLMessage(ACLMessage.INFORM);
        long time =System.currentTimeMillis();
        msg.setConversationId(""+time);
        msg.addReceiver(receiver);

        if(time % 2 == 0)
            msg.setContent("ping2");
        else
            msg.setContent("not ping2");

        myAgent.send(msg);

        counter2++;

        System.out.println("Executing SendMessage2 for the "+counter2+" time!");
    }

    @Override
    public boolean done() {
        if (counter2>=5){
            System.out.println("Behaviour SendMessage2 finalizing!");
            return true;
        }else return false;
    }
}
```

```
public class SendMessage3 extends SimpleBehaviour {

    @Override
    public void action(){

        AID receiver=new AID();
        receiver.setLocalName("pingaponga");

        ACLMessage msg =new ACLMessage(ACLMessage.INFORM);
        long time =System.currentTimeMillis();
        msg.setConversationId(""+time);
        msg.addReceiver(receiver);

        if(time % 2 == 0)
            msg.setContent("ping3");
        else
            msg.setContent("not ping3");

        myAgent.send(msg);

        counter3++;

        System.out.println("Executing SendMessage3 for the "+counter3+" time!");
    }

    @Override
    public boolean done() {
        if (counter3>=13 ){
            System.out.println("Behaviour SendMessage3 finalizing!");
            return true;
        }else return false;
    }
}
```

Cyclic Message Send (limited run)

- Configurations:
 - container –agents pingaponga:agents.PingPong;sender:agents.SenderParallel

```
sender a começar!
out 23, 2016 3:46:27 PM jade.core.AgentContainerImpl joinPlatform
INFO: -----
Agent container Container-38@192.168.1.76 is ready.
-----
```

```
Executing SendMessage1 for the 1 time!
Executing SendMessage2 for the 1 time!
Executing SendMessage3 for the 1 time!
Executing SendMessage1 for the 2 time!
Executing SendMessage2 for the 2 time!
Executing SendMessage3 for the 2 time!
Executing SendMessage1 for the 3 time!
Behaviour SendMessage1 finalizing!
Executing SendMessage3 for the 3 time!
Executing SendMessage2 for the 3 time!
Executing SendMessage3 for the 4 time!
Executing SendMessage2 for the 4 time!
Executing SendMessage3 for the 5 time!
Executing SendMessage2 for the 5 time!
Behaviour SendMessage2 finalizing!
Executing SendMessage3 for the 6 time!
Executing SendMessage3 for the 7 time!
Executing SendMessage3 for the 8 time!
Executing SendMessage3 for the 9 time!
Executing SendMessage3 for the 10 time!
```

```
pingaponga a começar!
Recebi uma mensagem de ( agent-identifier :name sender@192.168.1.76:1099/JADE :addresses (sequence http://Laptop.lan:7778/acc )).Conteúdo: not ping1
Executing SendMessage3 for the 11 time!
Recebi uma mensagem de ( agent-identifier :name sender@192.168.1.76:1099/JADE :addresses (sequence http://Laptop.lan:7778/acc )).Conteúdo: ping2
Executing SendMessage3 for the 12 time!
Recebi uma mensagem de ( agent-identifier :name sender@192.168.1.76:1099/JADE :addresses (sequence http://Laptop.lan:7778/acc )).Conteúdo: not ping3
Executing SendMessage3 for the 13 time!
Recebi uma mensagem de ( agent-identifier :name sender@192.168.1.76:1099/JADE :addresses (sequence http://Laptop.lan:7778/acc )).Conteúdo: not ping1
Behaviour SendMessage3 finalizing!
ParallelBehaviour finalizing
sender a morrer!
```

Other ParallelBehaviour Arguments:

-*WHEN_ALL* (condition that terminates when all its children are done.)

-*WHEN_ANY* (condition that terminates when any of its child is done);

-*n* (number of child behaviours that must end before the composite behaviour ends)

Arguments Management

```
public class SenderArgument extends Agent{

    protected String message;

    @Override
    protected void setup(){

        super.setup();

        System.out.println(this.getLocalName()+" a começar!");

        Object[] args = getArguments();

        if (args != null && args.length > 0) {
            message = (String) args[0];
            System.out.println("Sending "+ message+"!");

            this.addBehaviour(new SendMessage());
        }

    }
}
```

```
public class SendMessage extends OneShotBehaviour {

    @Override
    public void action() {
        AID receiver=new AID();
        receiver.setLocalName("pingaponga");
        ACLMessage msg =new ACLMessage(ACLMessage.INFORM);
        long time =System.currentTimeMillis();
        msg.setConversationId(""+time);
        msg.addReceiver(receiver);
        msg.setContent(message);
        myAgent.send(msg);
    }

}
```


Arguments Management

- Configurations:
 - -container -agents pingaponga:agents.PingPong;sender:agents.SenderArgument(ping)

```

out 23, 2016 2:54:34 AM jade.core.Runtime beginContainer
INFO: -----
      This is JADE 4.3.3 - revision 6726 of 2014/12/09 09:33:02
      downloaded in Open Source, under LGPL restrictions,
      at http://jade.tilab.com/
-----
out 23, 2016 2:54:35 AM jade.imtp.leap.LEAPIMTPManager initialize
INFO: Listening for intra-platform commands on address:
- jicp://192.168.1.76:61474

out 23, 2016 2:54:35 AM jade.core.BaseService init
INFO: Service jade.core.management.AgentManagement initialized
out 23, 2016 2:54:35 AM jade.core.BaseService init
INFO: Service jade.core.messaging.Messaging initialized
out 23, 2016 2:54:35 AM jade.core.BaseService init
INFO: Service jade.core.resource.ResourceManagement initialized
out 23, 2016 2:54:35 AM jade.core.BaseService init
INFO: Service jade.core.mobility.AgentMobility initialized
out 23, 2016 2:54:35 AM jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
sender a começar!out 23, 2016 2:54:35 AM jade.core.AgentContainerImpl joinPlatform
INFO: -----
Agent container Container-1@192.168.1.76 is ready.
-----

Sending ping!
pingaponga a começar!
Recebi uma mensagem de ( agent-identifier :name sender@192.168.1.76:1099/JADE :addresses (sequence http://Laptop.lan:7778/acc )).Conteúdo: ping

```

Yellow Pages Service

- Register agent represented by the pingaponga agent in directory facilitator (DF)

```
super.setup();

DFAgentDescription dfd = new DFAgentDescription();
dfd.setName(getAID());
ServiceDescription sd = new ServiceDescription();
sd.setType("ping pong");
sd.setName("JADE-replies-pong");
dfd.addServices(sd);

try {
    DFService.register(this, dfd);
} catch (FIPAException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```

Remove log

```
super.takeDown();

try {
    DFService.deregister(this);
} catch (FIPAException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```

Yellow Pages Service

```
public class SenderFinder extends Agent{

    private AID[] pingpongAgents;

    @Override
    protected void setup(){

        super.setup();

        System.out.println(this.getLocalName()+" a começar!");

        this.addBehaviour(new SendMessage());

    }
}
```

```
public class SendMessage extends OneShotBehaviour {

    @Override
    public void action() {

        DFAgentDescription template = new DFAgentDescription();
        ServiceDescription sd = new ServiceDescription();
        sd.setType("ping pong");
        template.addServices(sd);

        DFAgentDescription[] result;
        try {

            result = DFService.search(myAgent, template);
            pingpongAgents = new AID[result.length];
            for (int i = 0; i < result.length; ++i) {
                pingpongAgents[i] = result[i].getName();
            }

            ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
            long time = System.currentTimeMillis();
            msg.setConversationId(""+time);
            msg.addReceiver(pingpongAgents[i]);
            msg.setContent("ping");
            myAgent.send(msg);
        } catch (FIPAException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

    }

}
```


Yellow Pages Service

- Configuration:
 - -container -agents pingaponga:agents.PingPong;sender:agents.SenderFinder

```

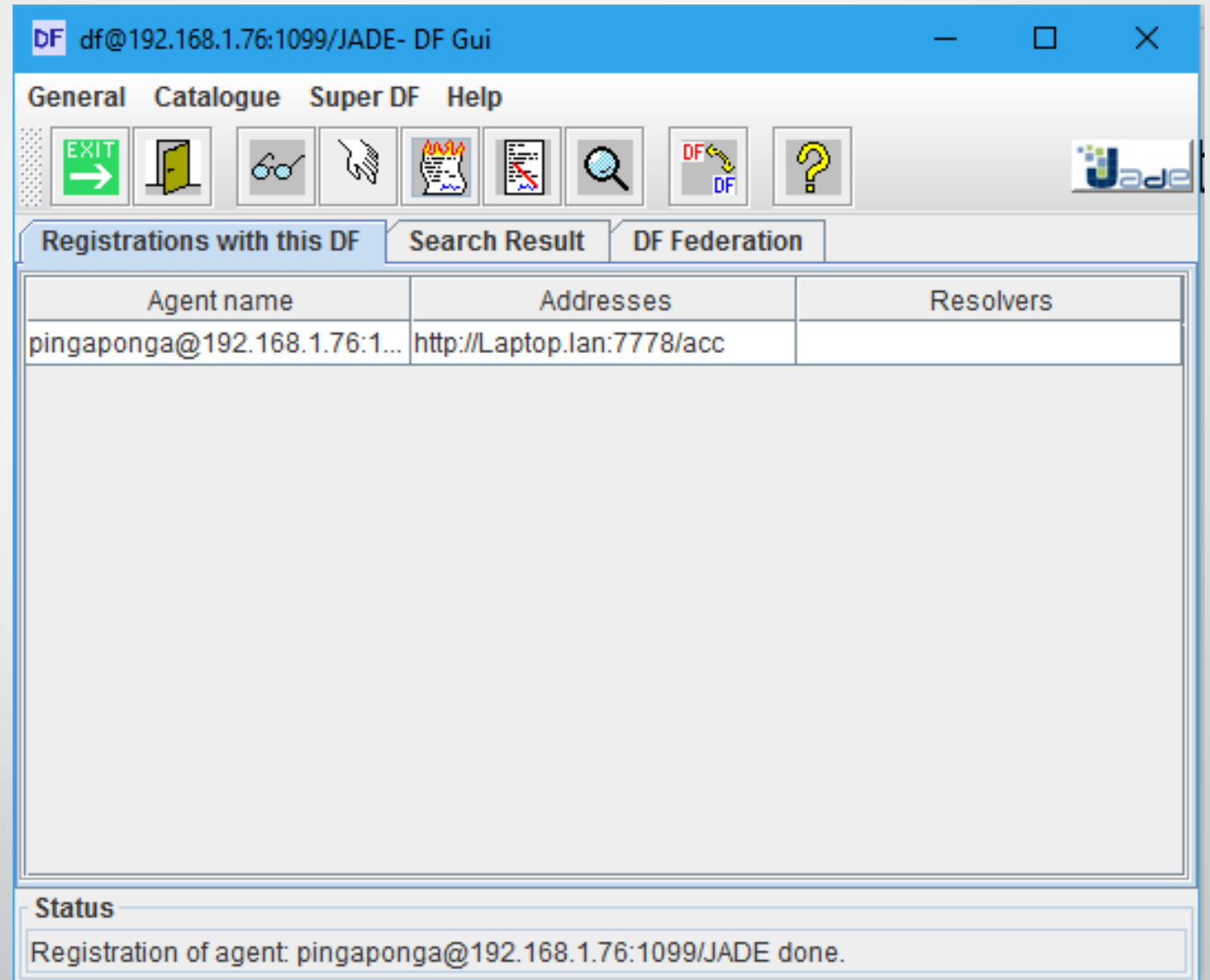
put 23, 2016 3:10:29 AM jade.core.Runtime beginContainer
INFO: -----
This is JADE 4.3.3 - revision 6726 of 2014/12/09 09:33:02
downloaded in Open Source, under LGPL restrictions,
at http://jade.tilab.com/
-----
out 23, 2016 3:10:29 AM jade.imtp.leap.LEAPIMTPManager initialize
INFO: Listening for intra-platform commands on address:
- jicp://192.168.1.76:61528

out 23, 2016 3:10:30 AM jade.core.BaseService init
INFO: Service jade.core.management.AgentManagement initialized
out 23, 2016 3:10:30 AM jade.core.BaseService init
INFO: Service jade.core.messaging.Messaging initialized
out 23, 2016 3:10:30 AM jade.core.BaseService init
INFO: Service jade.core.resource.ResourceManagement initialized
out 23, 2016 3:10:30 AM jade.core.BaseService init
INFO: Service jade.core.mobility.AgentMobility initialized
out 23, 2016 3:10:30 AM jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
out 23, 2016 3:10:30 AM jade.core.AgentContainerImpl joinPlatform
INFO: -----
Agent container Container-4@192.168.1.76 is ready.
-----
sender a começar!
pingaponga a começar!
Recebi uma mensagem de ( agent-identifier :name sender@192.168.1.76:1099/JADE :addresses (sequence http://Laptop.lan:7778/acc ) ).Conteúdo: ping

```

Yellow Pages Service

- Tools > Show the DF GUI



References

- JADE: Java Agent DEvelopment framework:
 - <http://jade.tilab.com/>
- Bellifemine F., Caire G., Greenwood D., Developing Multi-Agent Systems with JADE, John Wiley & Sons, ISBN: 978-0470057476, 2007.
- Giovanni, Caire; TUTORIAL, J. A. D. E. JADE programming for beginners., 2003.

JADE

Java Agent DEvelopment

Framework - Part I

Integrated Master's in Informatics Engineering

Intelligent Agents

2017/2018

Synthetic Intelligence Lab

Ricardo Ramos

Filipe Gonçalves

Paulo Novais

