

# LAB 3 (Agents: The innocent childhood)

Fatema Mirza & Mohammad Newaj Jamil

D7001D Network Programming and Distributed Applications





# LAB 3 (Agents: The innocent childhood)

by

Fatema Mirza & Mohammad Newaj  
Jamil

Submission Deadline: November 5, 2021

Teachers:

Dr. Evgeny Osipov,  
Ahmed Afif Monrat,

LULEÅ TEKNISKA UNIVERSITET  
LULEÅ TEKNISKA UNIVERSITET



# Contents

1	Part I – Trying is the best way of learning	1
1.1	Starting up Jade . . . . .	1
1.2	Creating Containers (default, User Defined Name) . . . . .	1
1.3	Creating Container in Different Machine . . . . .	2
1.4	Creating first Jade Agent (HelloWorldAgent.java), compiling Jade Agent . . . . .	2
1.5	Running HelloWorldAgent . . . . .	3
1.6	Get and diplay full Agent Identification Info . . . . .	5
1.7	Agent with Argument . . . . .	6
1.8	Agent Behavior . . . . .	6
1.9	SimpleBehaviour Class . . . . .	7
1.10	TickerBehaviour Class . . . . .	8
1.11	CyclicBehaviour. . . . .	10
1.12	Agent Communication . . . . .	10
1.13	Multi-agent System Example I . . . . .	12
1.14	Distributed Platform . . . . .	13
1.15	Agent with GUI . . . . .	15
2	Part II – Write your own agents	16

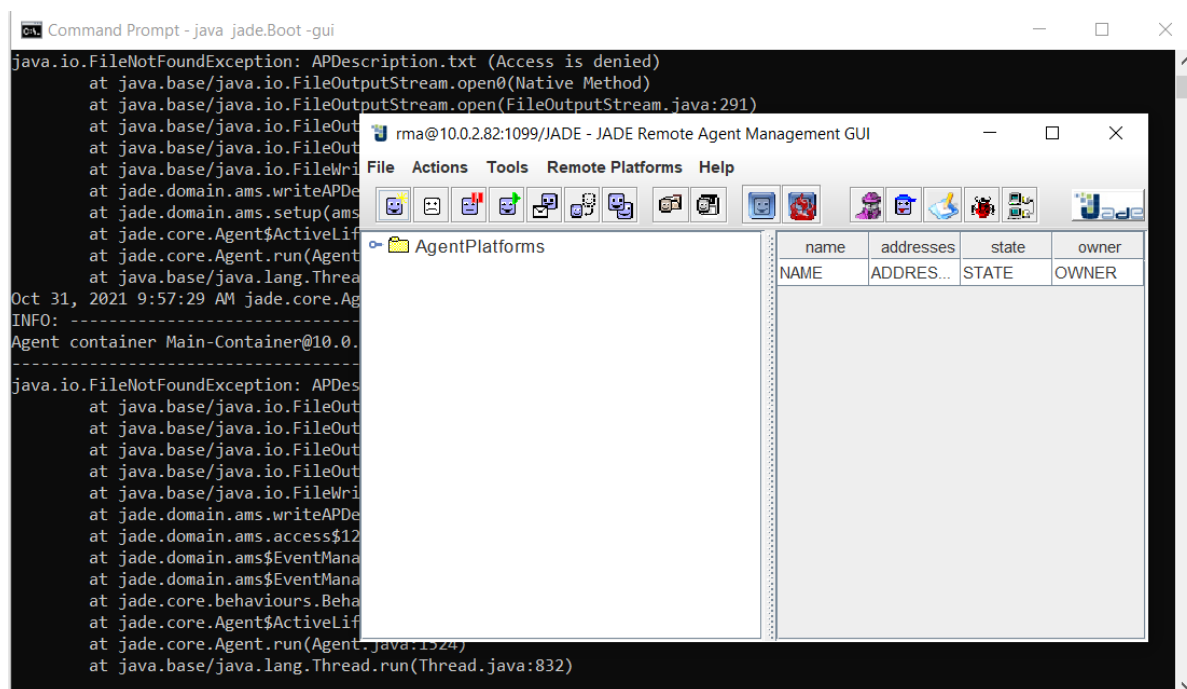
# 1

## Part I – Trying is the best way of learning

### 1.1. Starting up Jade

After setting up all the path correctly, type the following to check if it is running properly:

```
java jade.Boot -gui
```

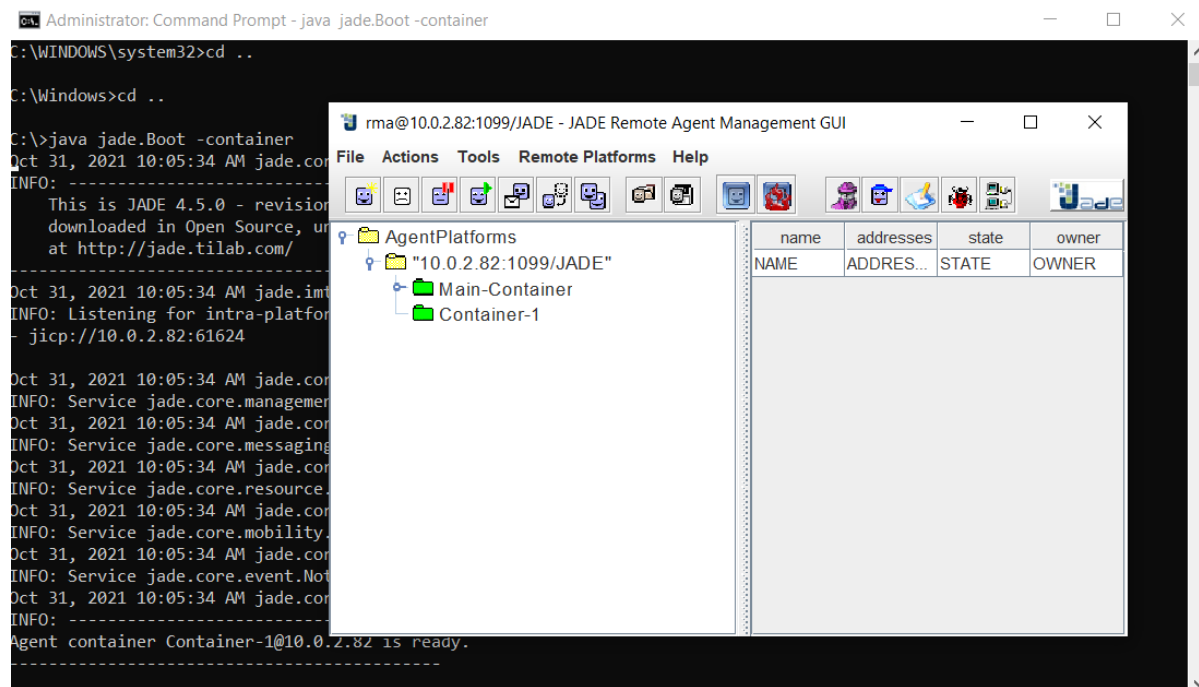


### 1.2. Creating Containers (default, User Defined Name)

Containers contain the agents created.

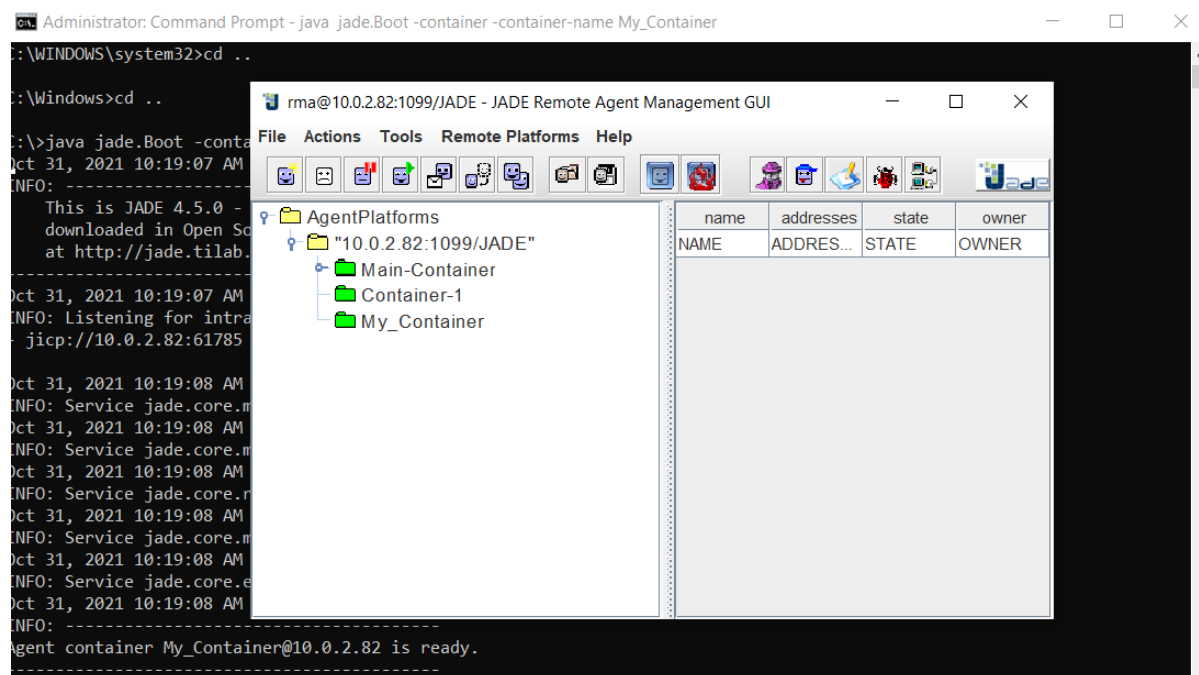
#### Creating default containers:

```
java jade.Boot -container
```



### Creating User defined containers:

java jade.Boot -container -container-name My\_Container



## 1.3. Creating Container in Different Machine

To create a container in another computer:

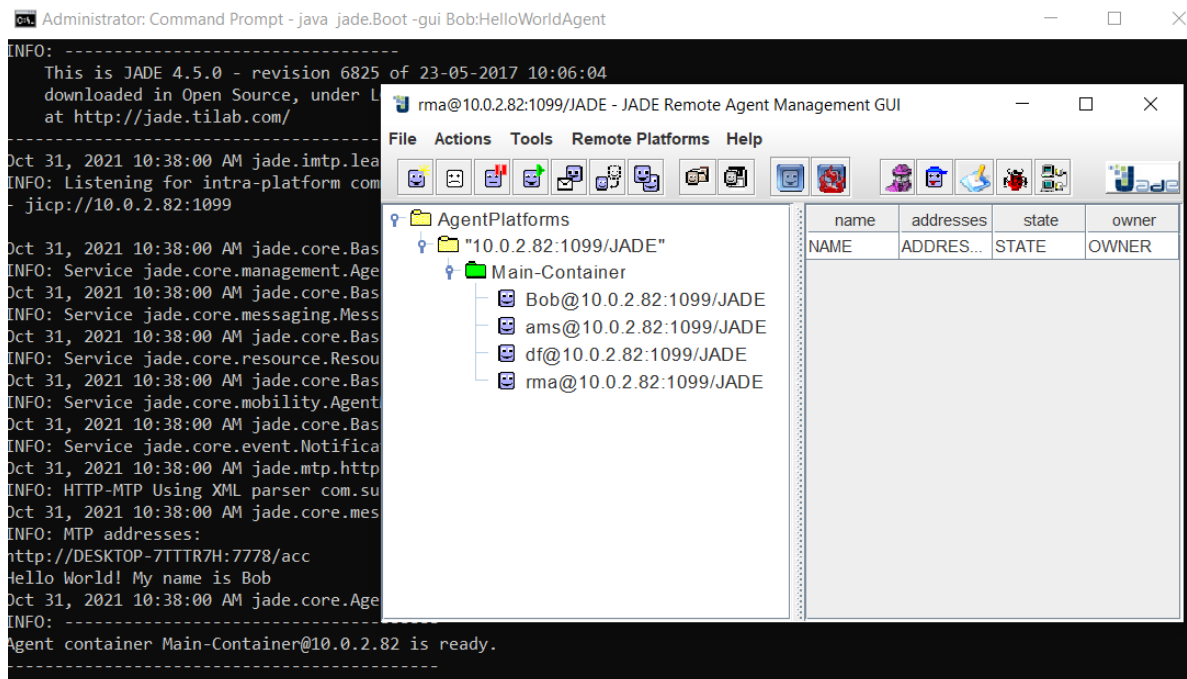
java jade.Boot -container -container-name My\_Container -host [IP/Computer Name] -port 1099

## 1.4. Creating first Jade Agent (HelloWorldAgent.java), compiling Jade Agent

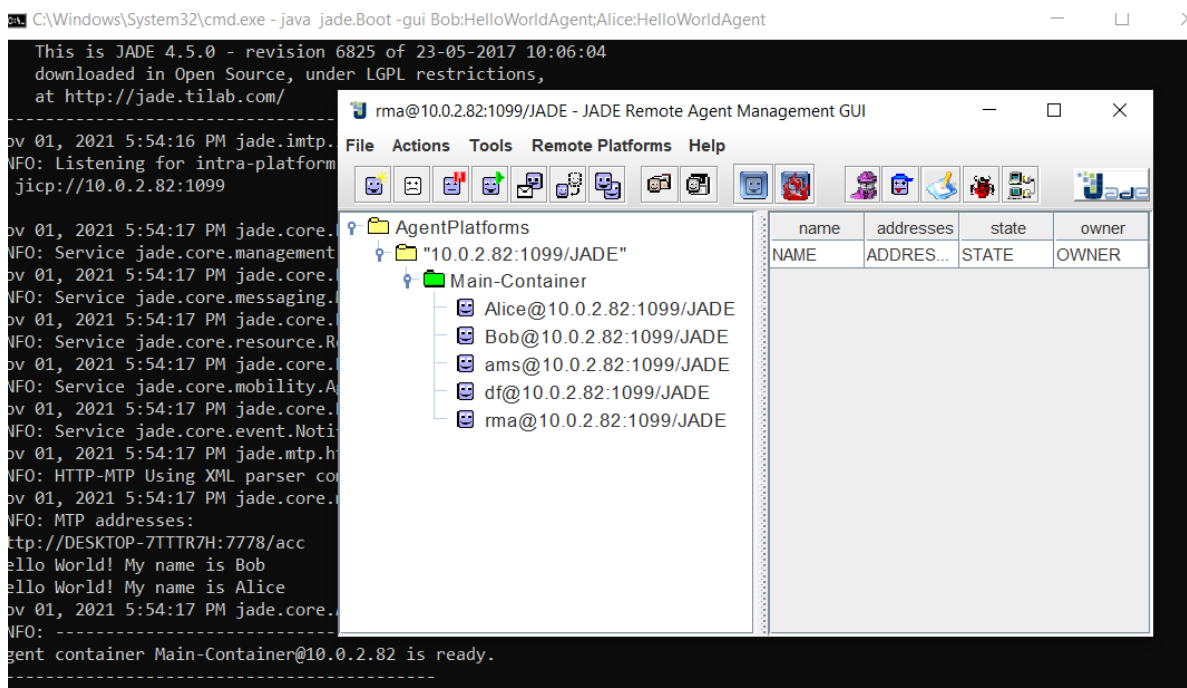
The instructions supplied in the manual have been followed.

## 1.5. Running HelloWorldAgent

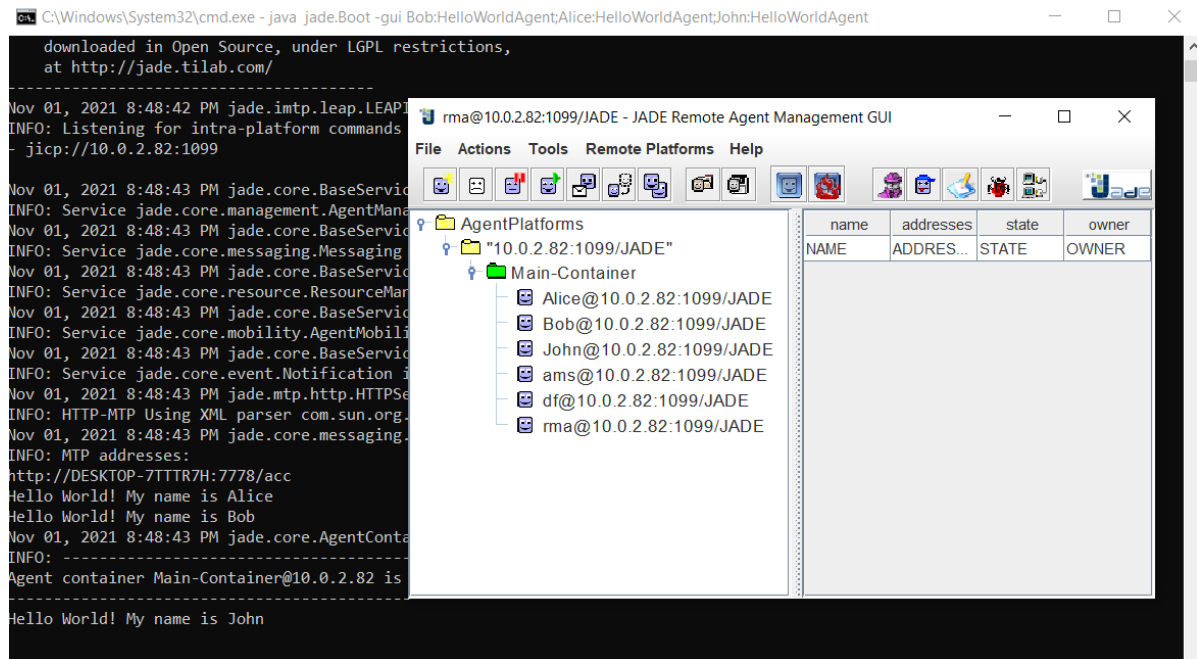
java jade.Boot -gui Bob:HelloWorldAgent



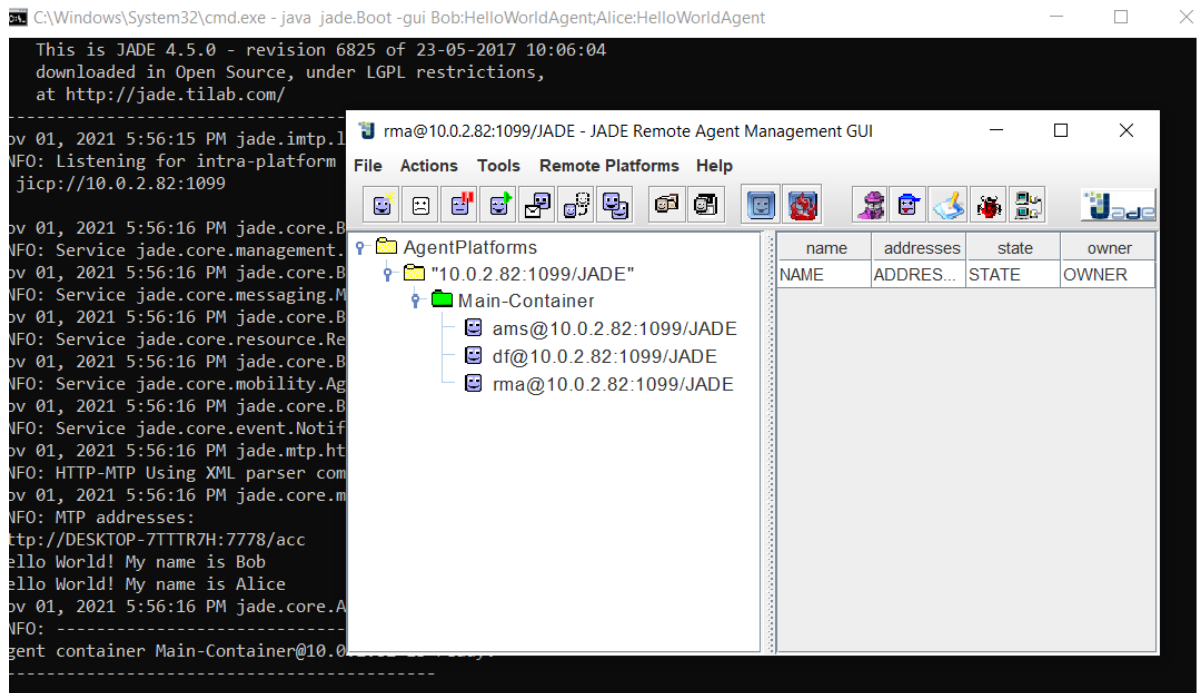
java jade.Boot -gui Bob:HelloWorldAgent;Alice:HelloWorldAgent



```
java jade.Boot -gui Bob:HelloWorldAgent;;Alice:HelloWorldAgent;John:HelloWorldAgent
```



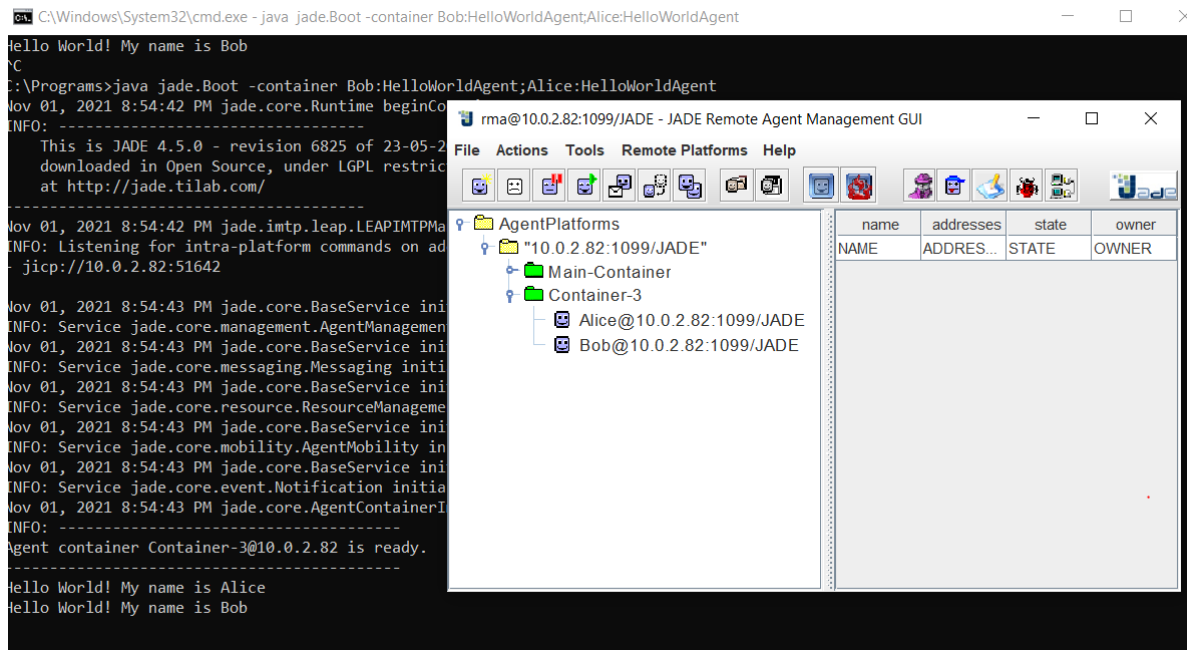
If the `doDelete()` function is activated, the agents get created and then immediately deleted. Therefore, we can observe that the initialization messages can be viewed but no agent can be seen in the GUI.



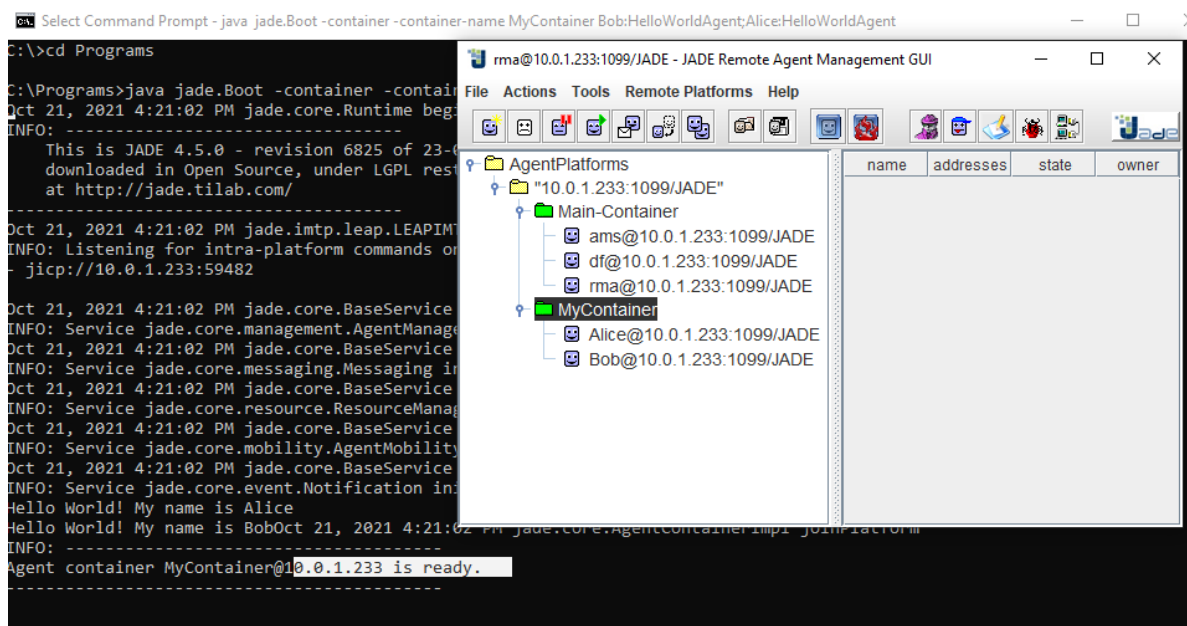


```
java jade.Boot -gui
```

```
java jade.Boot -container Bob:HelloWorldAgent;Alice:HelloWorldAgent
```



```
java jade.Boot -container -container-name MyContainer Bob:HelloWorldAgent;Alice:HelloWorldAgent
```



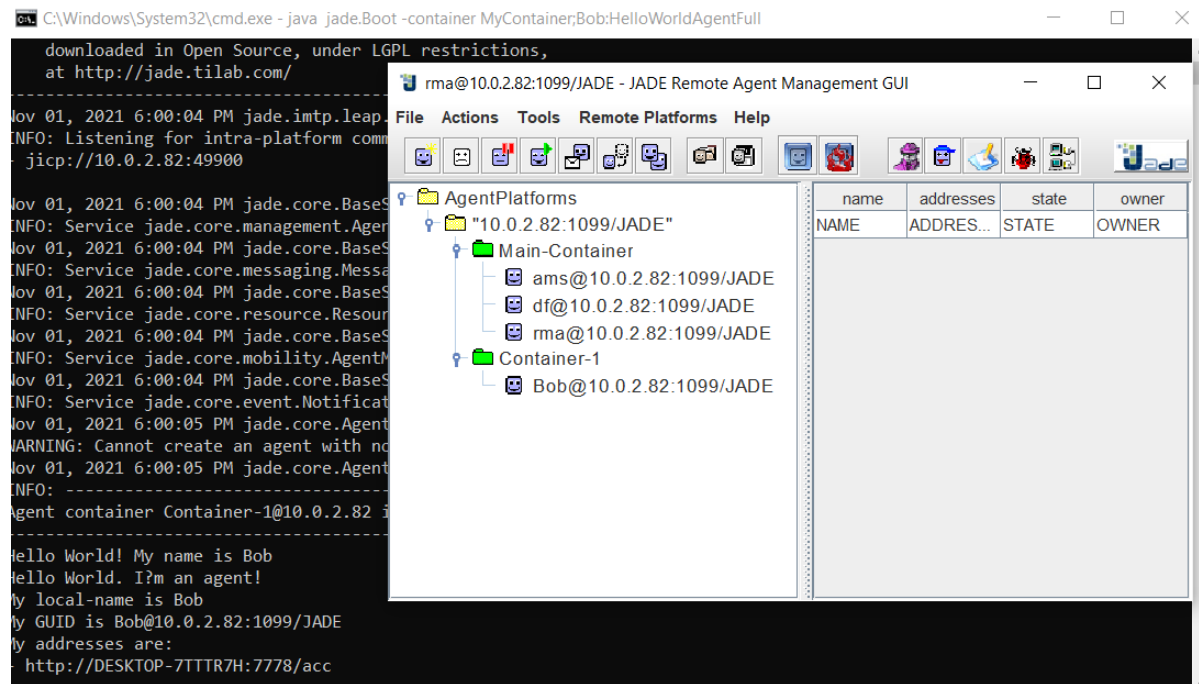
```
java jade.Boot -container -container-name LocalContainer Bob:HelloWorldAgent
```

```
java jade.Boot -container -container-name RemoteContainer -host "IP" -port 1099 Bob:HelloWorldAgent
```

## 1.6. Get and display full Agent Identification Info

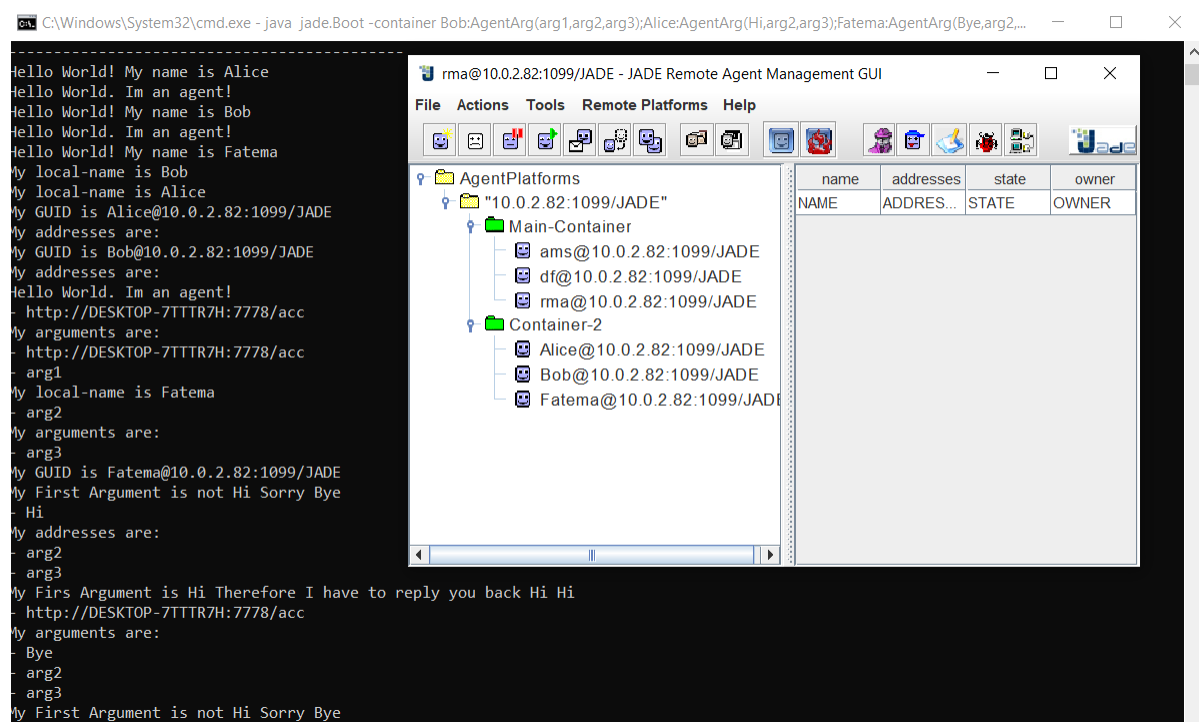
```
java jade.Boot -container MyContainer;Bob:HelloWorldAgentFull
```

Running this command will give the full detail of the agent as can be observed from the terminal.



## 1.7. Agent with Argument

`java jade.Boot -container Bob:AgentArg(arg1,arg2,arg3);Alice:AgentArg(Hi,arg2,arg3);Fatema:AgentArg(Bye,arg2,arg3)`

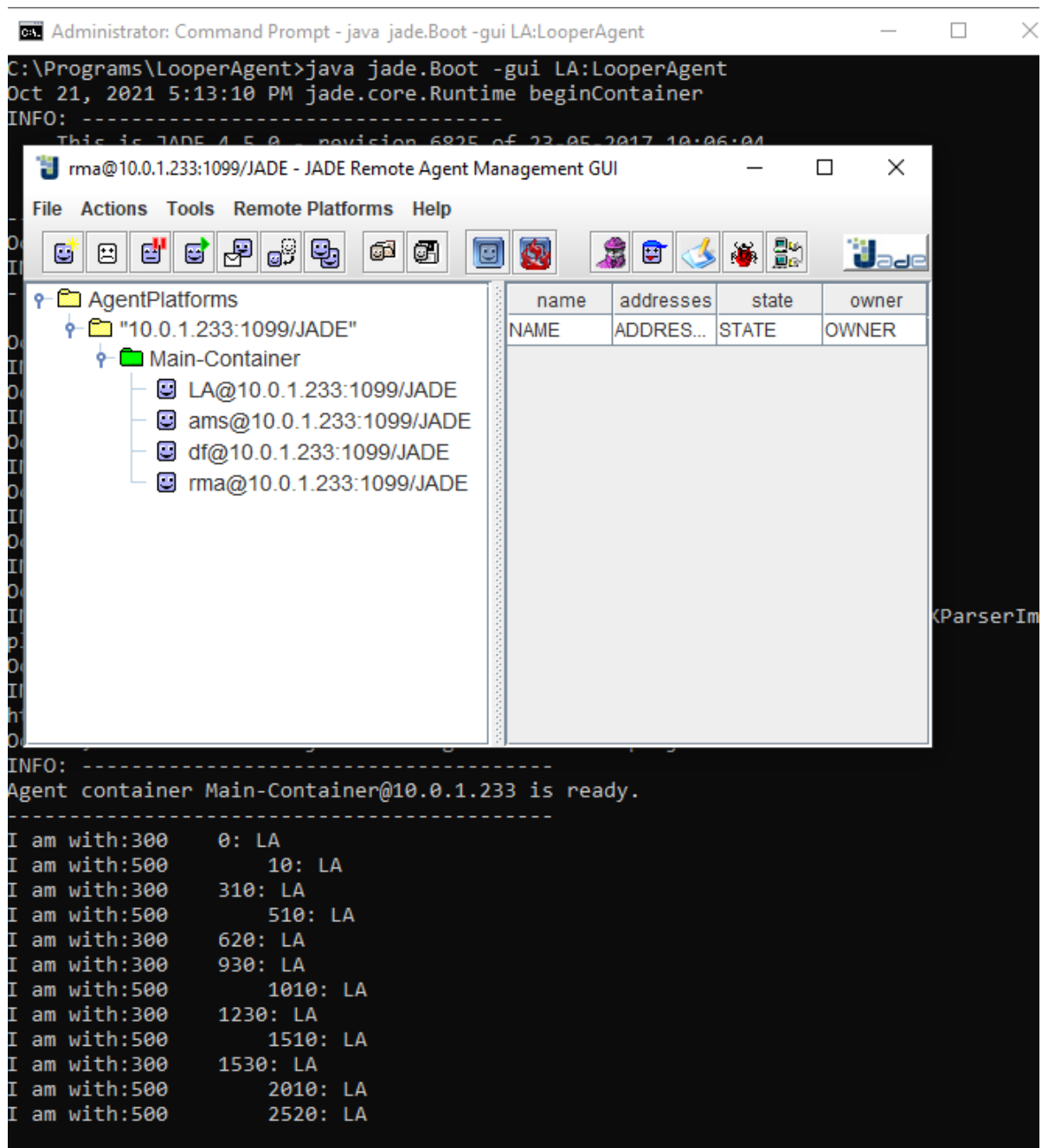


## 1.8. Agent Behavior

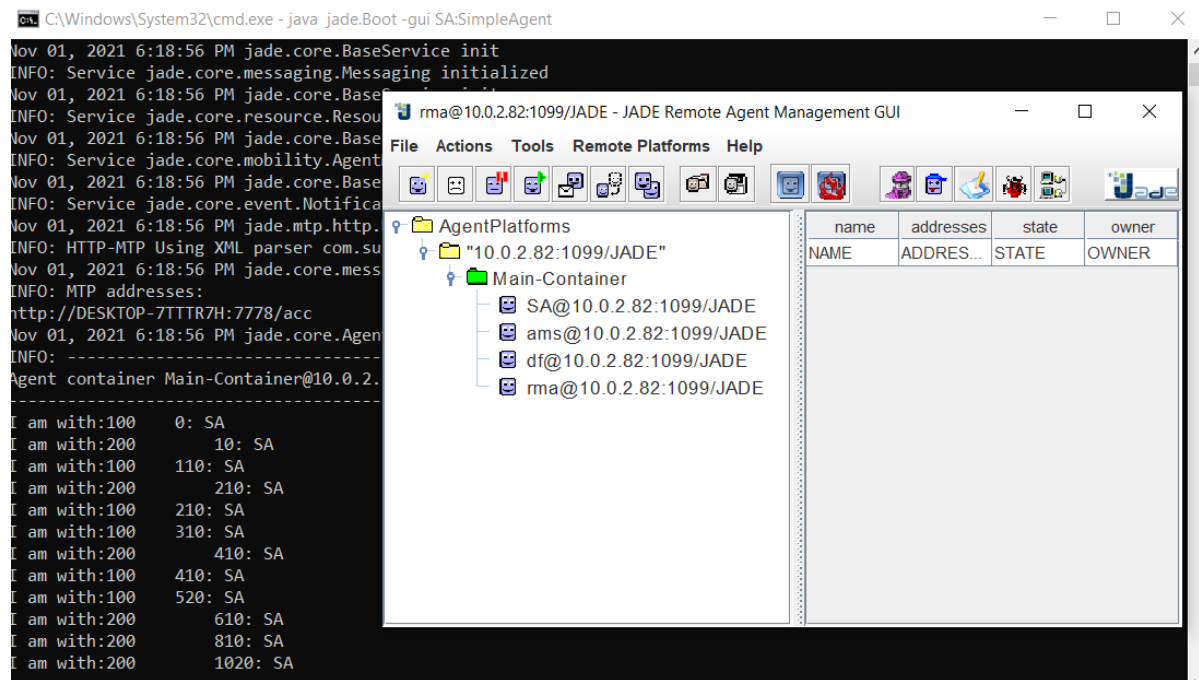
It has been demonstrated in the question already.

## 1.9. SimpleBehaviour Class

```
javac Looper.java
javac LooperAgent.java
java jade.Boot -gui LA:LooperAgent
```



A Simple Looper class has been written to extend this:



## 1.10. TickerBehaviour Class

```
javac TickerBehav1.java
```

```
javac TickerBehav2.java
```

```
javac TickerLooper.java
```

```
java jade.Boot -gui TB1:TickerBehav1
```

```
This is JADE 4.5.0 - revision 6825 of 23-05-2017 10:06:04
downloaded in Open Source, under LGPL restrictions,
at http://jade.tilab.com/
-----
Oct 21, 2021 6:41:55 PM jade.imtp.leap.LEAPIMTPManager initialize
INFO: listening for intra-platform commands on address:
- jicp://10.0.1.233:1099
-----
Oct 21, 2021 6:41:55 PM jade.core.BaseService init
INFO: Service jade.core.management.AgentManagement initialized
Oct 21, 2021 6:41:55 PM jade.core.BaseService init
INFO: Service jade.core.messaging.Messaging initialized
Oct 21, 2021 6:41:55 PM jade.core.BaseService init
INFO: Service jade.core.resource.ResourceManagement initialized
Oct 21, 2021 6:41:55 PM jade.core.BaseService init
INFO: Service jade.core.mobility.AgentMobility initialized
Oct 21, 2021 6:41:55 PM jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
Oct 21, 2021 6:41:55 PM jade.mtp.http.HTTPServer <init>
INFO: HTTP-MTP Using XML parser com.sun.org.apache.xerces.internal.jaxp.SAXParserImpl$JAXPSAXParser
Oct 21, 2021 6:41:55 PM jade.core.messaging.MessagingService boot
INFO: MTP addresses:
http://DESKTOP-BBAAEP1:7778/acc
Oct 21, 2021 6:41:55 PM jade.core.AgentContainerImpl joinPlatform
INFO: -----
Agent container Main-Container@10.0.1.233 is ready.
-----
I am Counting
I am Counting
I am Counting
```

```
java jade.Boot -gui Bob:TickerBehav2
```

```
C:\Programs\TickerAgent>java jade.Boot -gui Bob:TickerBehav2
Oct 21, 2021 6:42:38 PM jade.core.Runtime beginContainer
INFO: -----
      This is JADE 4.5.0 - revision 6825 of 23-05-2017 10:06:04
      downloaded in Open Source, under LGPL restrictions,
      at http://jade.tilab.com/
-----
Oct 21, 2021 6:42:38 PM jade.imtp.leap.LEAPIMTPManager initialize
INFO: Listening for intra-platform commands on address:
- jicp://10.0.1.233:1099

Oct 21, 2021 6:42:38 PM jade.core.BaseService init
INFO: Service jade.core.management.AgentManagement initialized
Oct 21, 2021 6:42:38 PM jade.core.BaseService init
INFO: Service jade.core.messaging.Messaging initialized
Oct 21, 2021 6:42:38 PM jade.core.BaseService init
INFO: Service jade.core.resource.ResourceManagement initialized
Oct 21, 2021 6:42:38 PM jade.core.BaseService init
INFO: Service jade.core.mobility.AgentMobility initialized
Oct 21, 2021 6:42:38 PM jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
Oct 21, 2021 6:42:39 PM jade.mtp.http.HTTPServer <init>
INFO: HTTP-MTP Using XML parser com.sun.org.apache.xerces.internal.jaxp.SAX
Oct 21, 2021 6:42:39 PM jade.core.messaging.MessagingService boot
INFO: MTP addresses:
http://DESKTOP-BBAAEP1:7778/acc
Oct 21, 2021 6:42:39 PM jade.core.AgentContainerImpl joinPlatform
INFO: -----
Agent container Main-Container@10.0.1.233 is ready.
-----
1003: Bob
2018: Bob
3019: Bob
```

```
java jade.Boot -gui Looper:TickerLooper
```

```
Oct 21, 2021 6:46:10 PM jade.core.BaseService init
INFO: Service jade.core.management.AgentManagement initialized
Oct 21, 2021 6:46:10 PM jade.core.BaseService init
INFO: Service jade.core.messaging.Messaging initialized
Oct 21, 2021 6:46:10 PM jade.core.BaseService init
INFO: Service jade.core.resource.ResourceManagement initialized
Oct 21, 2021 6:46:10 PM jade.core.BaseService init
INFO: Service jade.core.mobility.AgentMobility initialized
Oct 21, 2021 6:46:10 PM jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
Oct 21, 2021 6:46:10 PM jade.mtp.http.HTTPServer <init>
INFO: HTTP-MTP Using XML parser com.sun.org.apache.xerces.internal.jaxp.SAXParserImpl$JAXPSAXParser
Oct 21, 2021 6:46:10 PM jade.core.messaging.MessagingService boot
INFO: MTP addresses:
http://DESKTOP-BBAAEP1:7778/acc
I am with:300 0: Looper
I am with:500 0: Looper
Oct 21, 2021 6:46:10 PM jade.core.AgentContainerImpl joinPlatform
INFO: -----
Agent container Main-Container@10.0.1.233 is ready.
-----
I am with:300 310: Looper
I am with:500 500: Looper
I am with:300 610: Looper
I am with:300 910: Looper
1002: Looper
I am with:500 1010: Looper
I am with:300 1210: Looper
I am with:500 1510: Looper
I am with:300 1520: Looper
2017: Looper
I am with:500 2020: Looper
I am with:500 2520: Looper
3018: Looper
```

SimpleBehaviour: A class that can be extended in numerous ways to implement simple behaviours (Vaucher and Ncho, 2003).

CyclicBehaviour: A behaviour that remains active until its agent is killed and is called repeatedly after every event which presents to be helpful in handling message reception (Vaucher and Ncho, 2003).

OneShotBehaviour: A behaviour that is capable of executing only ONCE before dying (Vaucher and Ncho, 2003).

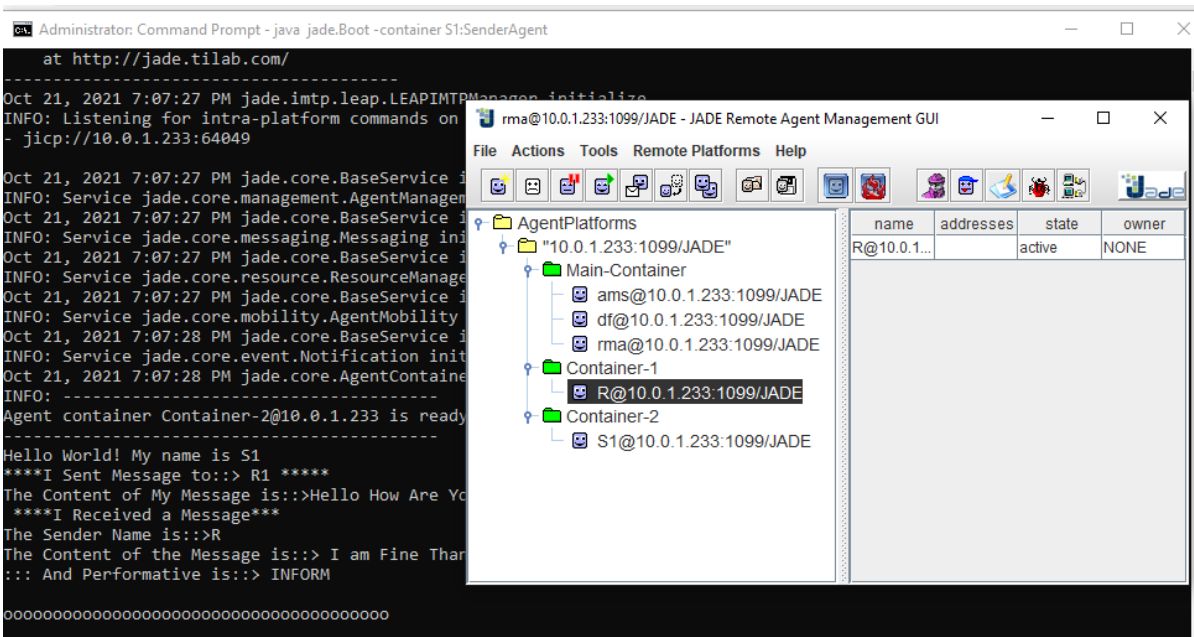
## 1.11. CyclicBehaviour

CyclicBehaviour: A behaviour that remains active until its agent is killed and is called repeatedly after every event which presents to be helpful in handling message reception (Vaucher and Ncho, 2003).

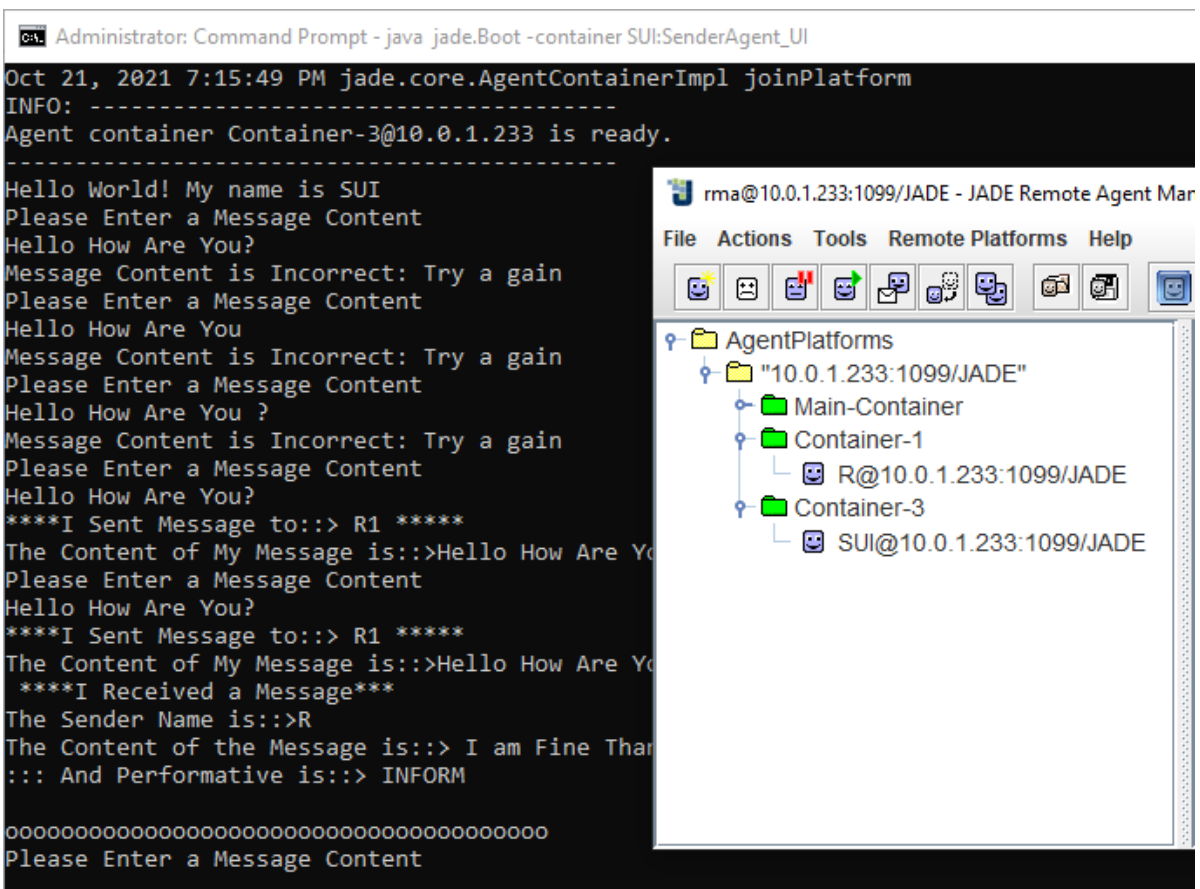
## 1.12. Agent Communication

```
java jade.Boot -gui
java jade.Boot -container R:ReceiverAgent
java jade.Boot -container S1:SenderAgent
```





**Communication with User Interface:** SenderAgent uses OneShotBehaviour to send and receive messages. SenderAgent\_UI uses CyclicBehaviour to send and receive messages. Furthermore, SenderAgent\_UI utilizes BufferedReader to implement the UI.



## 1.13. Multi-agent System Example I

Run the banking system code:

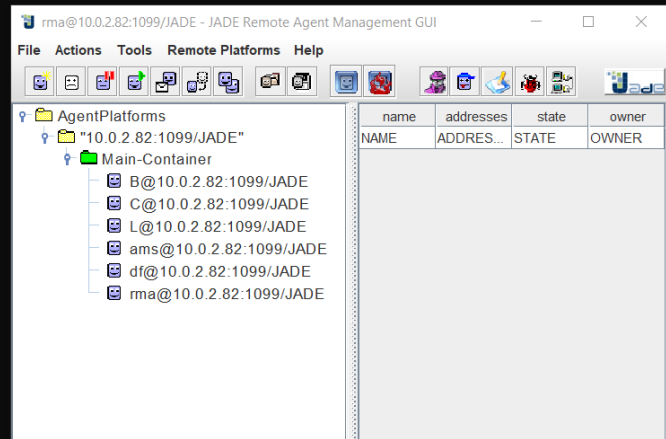
```
java jade.Boot -gui B:BankerAgent;C:CustomerAgent;L:LoanAgent
```

 C:\Windows\System32\cmd.exe - java jade.Boot -gui B:BankerAgent;C:CustomerAgent;L:LoanAgent

```
Nov 01, 2021 6:54:08 PM jade.mtp.http.HTTPServer <init>
INFO: HTTP-MTP Using XML parser com.sun.org.apache.xerces.internal.jaxp.SAXParserImpl$JAXPSAXParser
Nov 01, 2021 6:54:08 PM jade.core.messaging.MessagingService boot
INFO: MTP addresses:
http://DESKTOP-7TTTR7H:7778/acc
Nov 01, 2021 6:54:08 PM jade.core.AgentContainerImpl joinPlatform
INFO: -----
Agent container Main-Container@10.0.2.82 is ready.
-----
Hello World! My name is C
Hello World! My name is L
Hello World! My name is B
Please Enter Your Name for Loan:
FatemaJamil
****I Sent Message to:: C ****
The Content of My Message is::>FatemaJamil
***I Received a Message***
The Sender Name is:B
The Content of the Message is::> FatemaJamil
::: And Performative is:: REQUEST
Found Your Name::FatemaJamil
Record Alread Exists
****I Received a Message***
The Sender Name is::C
The Content of the Message is::> EXIST
::: And Performative is::> INFORM

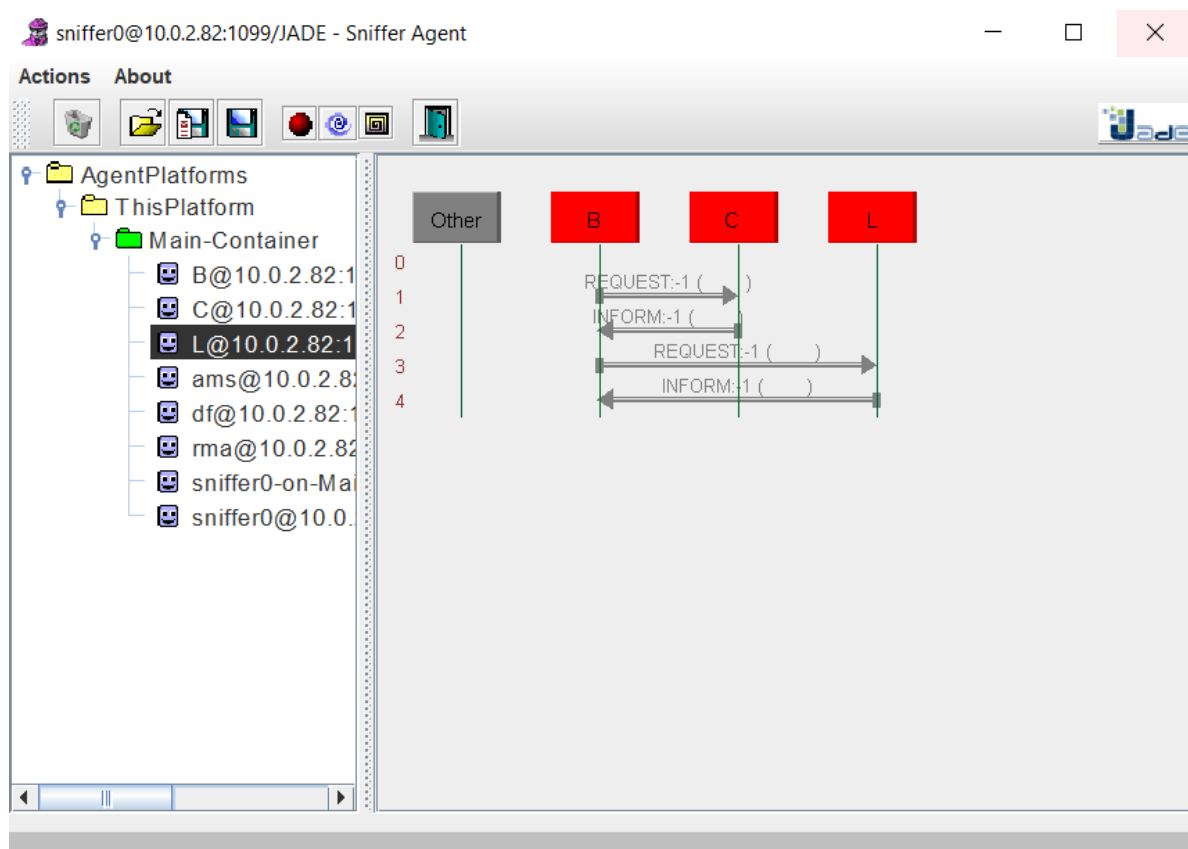
oooooooooooooooooooooooooooooooooooooooo
****I Replied to::> B***
You Are a Valid Customer and Eligible for Loan
A Message has Been Sent to LoanAgent
The Content of My Reply is:EXIST
oooooooooooooooooooooooooooooooooooooooo
***I Received a Message***
The Sender Name is:B
The Content of the Message is::> FatemaJamil
::: And Performative is:: REQUEST
Line Content::Arash Mousavi
Line Content::Fatema
Your file has been written
****I Received a Message***
The Sender Name is::L
The Content of the Message is::> Done
::: And Performative is::> INFORM

oooooooooooooooooooooooooooooooooooooooo
****I Replied to::> B***
Loan Recorded
The Content of My Reply is:Done
oooooooooooooooooooooooooooooooooooooooo
```





Run the sniffer agent by selecting the main container to start the sniffing. Run the code again and enter the customer name which has already been recorded and observe the sniffing action.



## 1.14. Distributed Platform

One machine was set up as the sender and the other as the receiver on the same network (achieved by using a hotspot to be on the same network). It can be observed from the images that the sending and receiving IP addresses are different - because they are from different computers.

To run the receiver:

```
java jade.Boot -container -host 192.168.43.247 -port 1099 R:ReceiverAgent
```

To run the sender:

```
java jade.Boot -container S1:SenderAgent
```

```
java jade.Boot -container S2:SenderAgent S3:SenderAgent S4:SenderAgent
```

AgentGui

**Message Performative**

REQUEST ↕

**Receiver**

R1

**Content**

Hi 2

**Agent List**

**Send**

**Cancel**

**Sent Messages**

Hi -> R1  
Hi 2 -> R1

**Received Messages**

**Agent List**

ams@192.168.43.247:1099/JADE  
R1@192.168.43.247:1099/JADE  
gAgent@192.168.43.247:1099/JADE  
df@192.168.43.247:1099/JADE  
rma@192.168.43.247:1099/JADE

```

INFO: service jade.core.event.notification initialized
Oct 23, 2021 11:22:33 AM jade.core.AgentContainerImpl joinPlatform
INFO: -----
Agent container S@192.168.43.170 is ready.
-----

Hello World! My name is S4
****I Sent Message to::> R1 ****
The Content of My Message is::>Hello How Are You?
****I Received a Message***
The Sender Name is::>R1@192.168.43.247:1099/JADE
The Content of the Message is::> I am Fine Thank You
::: And Performative is::> INFORM

oooooooooooooooooooooooooooooooooooooooooooooooooooo

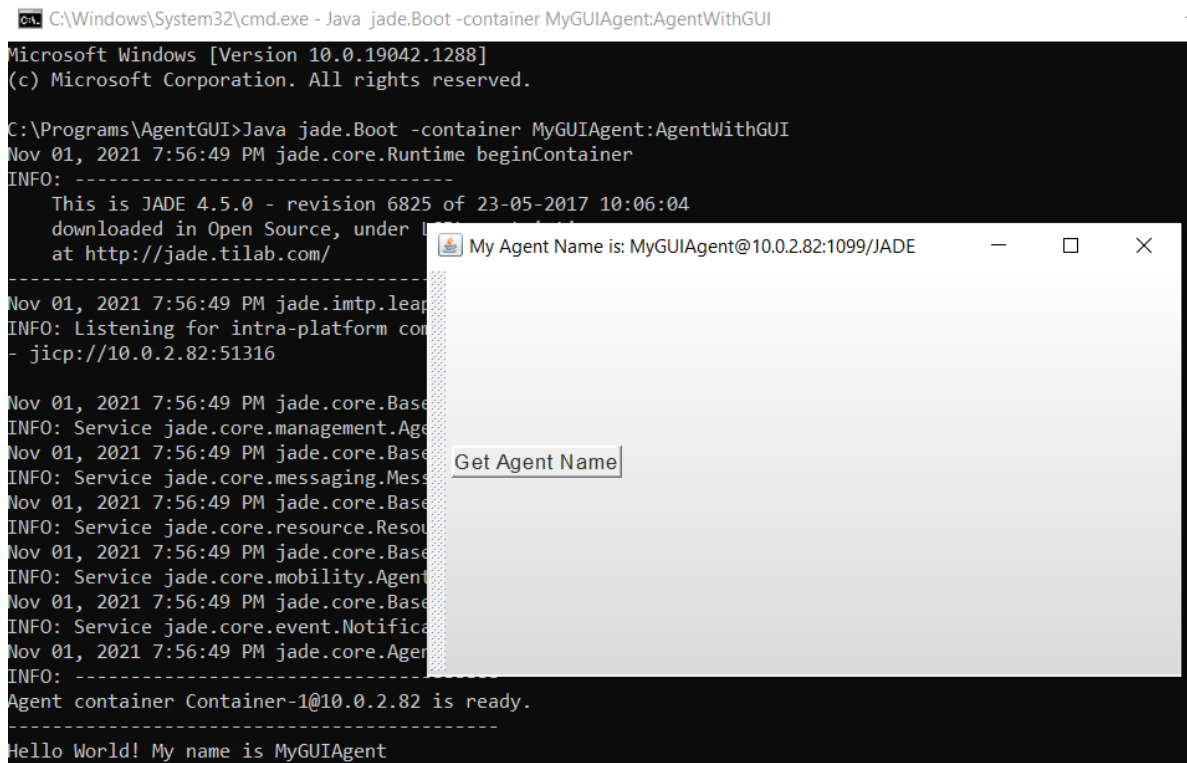
```

## 1.15. Agent with GUI

Run the following commands to view a JFrame interface where on clicking a button, the agent name is set to the title bar.

```
java jade.Boot -gui
```

```
java jade.Boot -container MyGUIAgent:AgentWithGUI
```



```

C:\Windows\System32\cmd.exe - Java jade.Boot -container MyGUIAgent:AgentWithGUI

Microsoft Windows [Version 10.0.19042.1288]
(c) Microsoft Corporation. All rights reserved.

C:\Programs\AgentGUI>Java jade.Boot -container MyGUIAgent:AgentWithGUI
Nov 01, 2021 7:56:49 PM jade.core.Runtime beginContainer
INFO: -----
This is JADE 4.5.0 - revision 6825 of 23-05-2017 10:06:04
downloaded in Open Source, under the LGPL license
at http://jade.tilab.com/
-----
Nov 01, 2021 7:56:49 PM jade.imtp.listener startListening
INFO: Listening for intra-platform communication on
- jicp://10.0.2.82:51316
Nov 01, 2021 7:56:49 PM jade.core.BaseManager start
INFO: Service jade.core.management.AgentManager started
Nov 01, 2021 7:56:49 PM jade.core.BaseManager start
INFO: Service jade.core.messaging.MessageBroker started
Nov 01, 2021 7:56:49 PM jade.core.BaseManager start
INFO: Service jade.core.resource.ResourceServer started
Nov 01, 2021 7:56:49 PM jade.core.BaseManager start
INFO: Service jade.core.mobility.AgentMover started
Nov 01, 2021 7:56:49 PM jade.core.BaseManager start
INFO: Service jade.core.event.NotificationManager started
Nov 01, 2021 7:56:49 PM jade.core.AgentContainer start
INFO: -----
Agent container Container-1@10.0.2.82 is ready.
-----
Hello World! My name is MyGUIAgent
  
```

# 2

## Part II – Write your own agents

In this assignment you have to write your own GUI Agent, which will communicate with another agent. An example of the interface is as shown below.

The image shows a window titled "Agent Smith" with a standard macOS-style title bar (red, yellow, and green buttons). The window is divided into two main sections. The left section contains three input fields: "Message performative:" with a dropdown menu, "Receiver:" with a text box, and "Content:" with a text box. Below these fields are two buttons: "Send" and "Cancel". The right section contains two large yellow rectangular boxes: "Sent messages:" at the top and "Received messages:" at the bottom.

You will get half of the lab points (and pass the lab) when you implement only the functionality as shown in the illustration above. Open your mind and implement extra features: For a nicer interface you will get additional 25%, for more advanced management of information (for example retrieve the recipients from a drop down list, a possibility to retrieve agents from a different platform, etc) you get another 25%.

We have created our own GUI Agent, named `SenderAgent`, which is capable of communicating with another agent, named `ReceiverAgent`. For `SenderAgent`, we have developed a GUI, named `SenderAgentGUI`. The code snippet for `SenderAgent.java`, `SenderAgentGUI.java`, and `ReceiverAgent.java` are shown below.

```

1  import jade.core.Agent;
2  import jade.core.AID;
3  import jade.core.behaviours.*;
4
5  import jade.domain.*;
6  import jade.domain.FIPAAgentManagement.*;
7
8  import jade.lang.acl.*;
9
10 import java.util.ArrayList;
11
12 import jade.gui.GuiAgent;
13
14 import jade.gui.GuiEvent;
15
16
17 public class SenderAgent extends GuiAgent {
18
19     private SenderAgentGUI senderAgentGUI;
20
21     public ArrayList<String> agentList;
22
23     private String perf = "";
24     private String rec = "";
25     private String cont = "";
26
27
28     private int req;
29
30
31
32     protected void setup() {
33
34
35         // Printout a welcome message
36         System.out.println("Sender agent " + getAID().getName() + " is ready.");
37
38
39         // query the AMS to get list of all active agents in all containers
40         agentList = new ArrayList<String>();

```

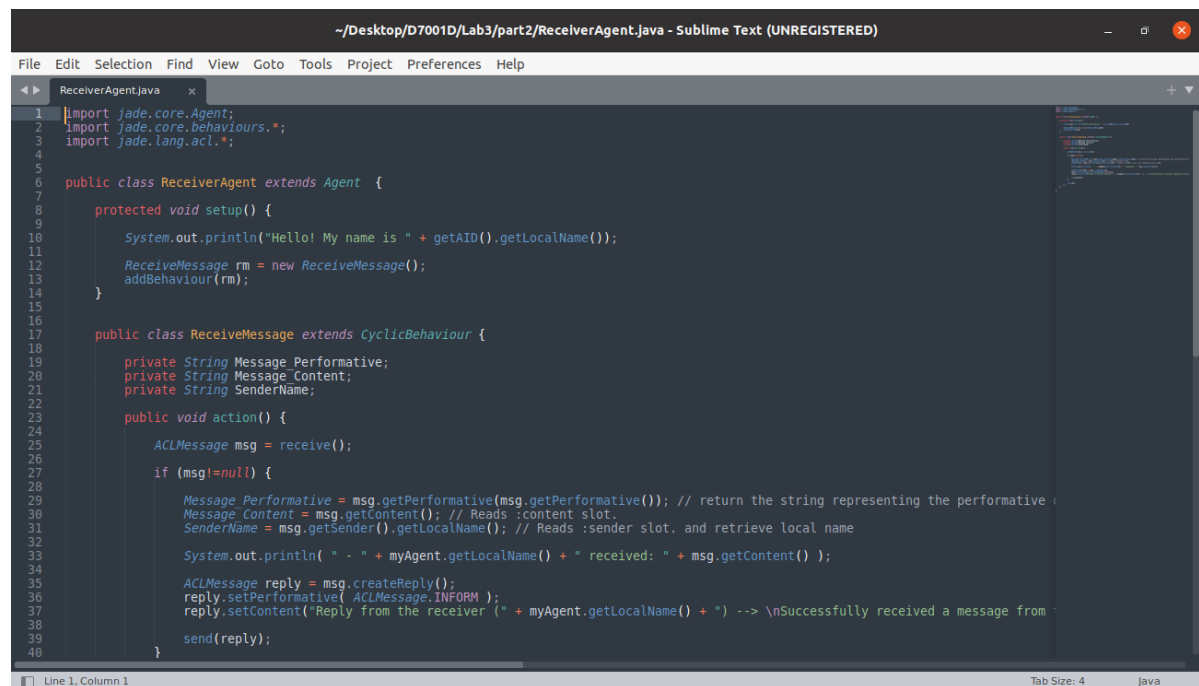
Line 83, Column 42      Tab Size: 4      java

```

1  import java.awt.*;
2  import java.awt.event.*;
3  import javax.swing.*;
4
5  import jade.gui.GuiEvent;
6
7  import java.util.ArrayList;
8
9
10 public class SenderAgentGUI extends JFrame {
11
12     private SenderAgent senderAgent;
13
14
15     JComboBox<String> messagePerformative;
16
17     JComboBox<Object> receivers;
18
19     JTextField content;
20
21     JTextArea sentMessage, receivedMessage;
22
23     JButton sendBtn, cancelBtn, RefreshBtn;
24
25
26     JLabel performativeLabel, receiverLabel, contentLabel;
27
28
29     JScrollPane scrollPanel, scrollPane2;
30
31
32     JPanel controlPanel;
33
34
35     ArrayList<String> receiversList;
36
37
38     String[] performat = {"INFORM", "REQUEST"};
39
40
41

```

Line 1, Column 1      Tab Size: 4      java



```

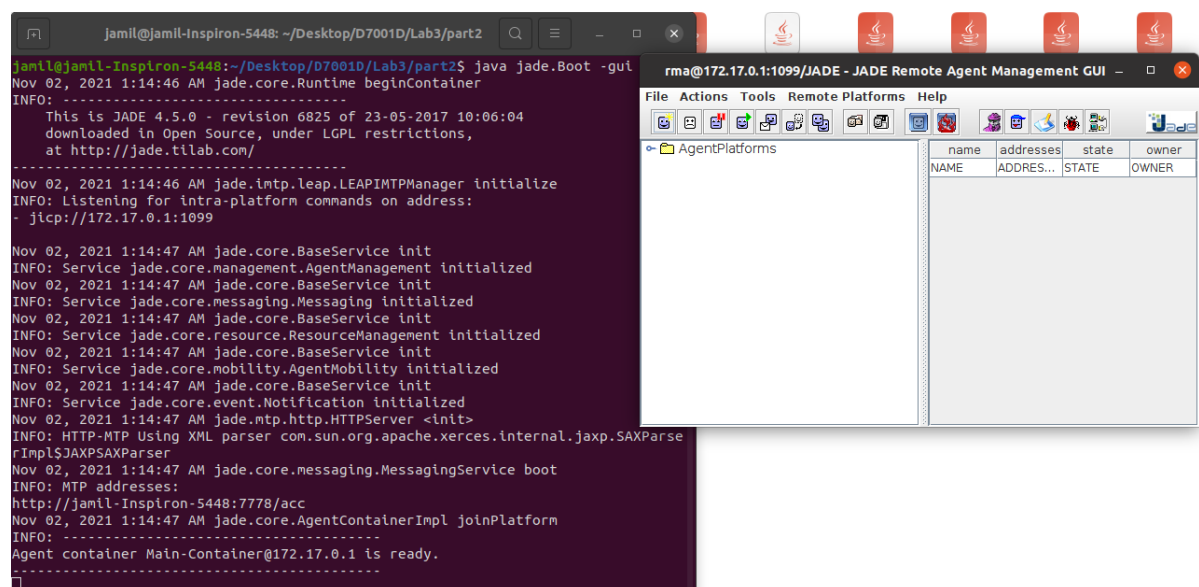
~/Desktop/D7001D/Lab3/part2/ReceiverAgent.java - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

ReceiverAgent.java
1 import jade.core.Agent;
2 import jade.core.behaviours.*;
3 import jade.lang.acl.*;
4
5
6 public class ReceiverAgent extends Agent {
7
8     protected void setup() {
9
10         System.out.println("Hello! My name is " + getAID().getLocalName());
11
12         ReceiveMessage rm = new ReceiveMessage();
13         addBehaviour(rm);
14     }
15
16     public class ReceiveMessage extends CyclicBehaviour {
17
18         private String Message_Performative;
19         private String Message_Content;
20         private String SenderName;
21
22         public void action() {
23
24             ACLMessage msg = receive();
25
26             if (msg!=null) {
27
28                 Message_Performative = msg.getPerformative(msg.getPerformative()); // return the string representing the performative
29                 Message_Content = msg.getContent(); // Reads :content slot.
30                 SenderName = msg.getSender().getLocalName(); // Reads :sender slot. and retrieve local name
31
32                 System.out.println( " - " + myAgent.getLocalName() + " received: " + msg.getContent() );
33
34                 ACLMessage reply = msg.createReply();
35                 reply.setPerformative( ACLMessage.INFORM );
36                 reply.setContent("Reply from the receiver (" + myAgent.getLocalName() + ") --> \nSuccessfully received a message from ");
37
38                 send(reply);
39             }
40         }
41     }
42 }

```

We have implemented all the functionality as mentioned in the question. As extra features, we have provided a nicer interface for GUI. We have added the functionality to retrieve the recipients from a drop-down list. Besides, it is possible to retrieve agents from a different platform (container) as well as from the same container. After clicking the "Refresh" button in the GUI, a user can see recipients from the same container as well as different containers. A demonstration is provided as follows.

First, we have to start up the JADE platform by running the following command as shown below.



```

jamil@jamil-Inspiron-5448: ~/Desktop/D7001D/Lab3/part2
jamil@jamil-Inspiron-5448:~/Desktop/D7001D/Lab3/part2$ java jade.Boot -gui
Nov 02, 2021 1:14:46 AM jade.core.Runtime beginContainer
INFO: -----
This is JADE 4.5.0 - revision 6825 of 23-05-2017 10:06:04
downloaded in Open Source, under LGPL restrictions,
at http://jade.tilab.com/
-----
Nov 02, 2021 1:14:46 AM jade.imtp.leap.LEAPIMTPManager initialize
INFO: Listening for intra-platform commands on address:
- jicp://172.17.0.1:1099
Nov 02, 2021 1:14:47 AM jade.core.BaseService init
INFO: Service jade.core.management.AgentManagement initialized
Nov 02, 2021 1:14:47 AM jade.core.BaseService init
INFO: Service jade.core.messaging.Messaging initialized
Nov 02, 2021 1:14:47 AM jade.core.BaseService init
INFO: Service jade.core.resource.ResourceManagement initialized
Nov 02, 2021 1:14:47 AM jade.core.BaseService init
INFO: Service jade.core.mobility.AgentMobility initialized
Nov 02, 2021 1:14:47 AM jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
Nov 02, 2021 1:14:47 AM jade.mtp.http.HTTPServer <init>
INFO: HTTP-MTP Using XML parser com.sun.org.apache.xerces.internal.jaxp.SAXParser
impl$JAXPSAXParser
Nov 02, 2021 1:14:47 AM jade.core.messaging.MessagingService boot
INFO: MTP addresses:
http://jamil-Inspiron-5448:7778/acc
Nov 02, 2021 1:14:47 AM jade.core.AgentContainerImpl joinPlatform
INFO: -----
Agent container Main-Container@172.17.0.1 is ready.
-----

```

After that, we can create a ReceiverAgent on Container1, named Fatima, as shown below.

```

jamil@jamil-Inspiron-5448: ~/Desktop/D7001D/Lab3/part2
jamil@jamil-Inspiron-5448:~/Desktop/D7001D/Lab3/part2$ java jade.Boot -container -container-name Container1 Fatima:ReceiverAgent
Nov 02, 2021 1:20:26 AM jade.core.Runtime beginContainer
INFO: -----
This is JADE 4.5.0 - revision 6825 of 23-05-2017 10:06:04
downloaded in Open Source, under LGPL restrictions,
at http://jade.tilab.com/
-----
Nov 02, 2021 1:20:26 AM jade.imtp.leap.LEAPIMTPManager initialize
INFO: Listening for intra-platform commands on address:
- jicp://172.17.0.1:45957

Nov 02, 2021 1:20:26 AM jade.core.BaseService init
INFO: Service jade.core.management.AgentManagement initialized
Nov 02, 2021 1:20:26 AM jade.core.BaseService init
INFO: Service jade.core.messaging.Messaging initialized
Nov 02, 2021 1:20:26 AM jade.core.BaseService init
INFO: Service jade.core.resource.ResourceManagement initialized
Nov 02, 2021 1:20:26 AM jade.core.BaseService init
INFO: Service jade.core.mobility.AgentMobility initialized
Nov 02, 2021 1:20:26 AM jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
Hello! My name is Fatima
Nov 02, 2021 1:20:26 AM jade.core.AgentContainerImpl joinPlatform
INFO: -----
Agent container Container1@172.17.0.1 is ready.
-----

```

Next, we can create a SenderAgent on Container2, named Jamil, as shown below.

```

jamil@jamil-Inspiron-5448: ~/Desktop/D7001D/Lab3/part2
jamil@jamil-Inspiron-5448:~/Desktop/D7001D/Lab3/part2$ java jade.Boot -container -container-name Container2 Jamil:SenderAgent
Nov 02, 2021 1:21:51 AM jade.core.Runtime beginContainer
INFO: -----
This is JADE 4.5.0 - revision 6825 of 23-05-2017 10:06:04
downloaded in Open Source, under LGPL restrictions,
at http://jade.tilab.com/
-----
Nov 02, 2021 1:21:51 AM jade.imtp.leap.LEAPIMTPManager initialize
INFO: Listening for intra-platform commands on address:
- jicp://172.17.0.1:34167

Nov 02, 2021 1:21:51 AM jade.core.BaseService init
INFO: Service jade.core.management.AgentManagement initialized
Nov 02, 2021 1:21:51 AM jade.core.BaseService init
INFO: Service jade.core.messaging.Messaging initialized
Nov 02, 2021 1:21:51 AM jade.core.BaseService init
INFO: Service jade.core.resource.ResourceManagement initialized
Nov 02, 2021 1:21:51 AM jade.core.BaseService init
INFO: Service jade.core.mobility.AgentMobility initialized
Nov 02, 2021 1:21:51 AM jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
Nov 02, 2021 1:21:51 AM jade.core.AgentContainerImpl joinPlatform
INFO: -----
Agent container Container2@172.17.0.1 is ready.
-----
Sender agent Jamil@172.17.0.1:1099/JADE is ready.

```

After running the above command, the following GUI will be launched.

The image shows a window titled "Agent Smith" with a standard macOS-style title bar (red, yellow, and green buttons). The window contains the following elements:

- Message Performative:** A label above a blue drop-down menu.
- Receiver:** A label above a light gray drop-down menu and a blue "Refresh" button.
- Content:** A label above a large white text input field.
- Buttons:** Two blue buttons labeled "Send" and "Cancel" are positioned below the content field.
- Sent Messages:** A label above a large white rectangular area for displaying sent messages.
- Received Messages:** A label above another large white rectangular area for displaying received messages.

In the GUI, first, we have to select the "Message Performative". For simplicity, only two message performative has been provided in the drop-down list, namely 'Inform' and 'Request'. Suppose we select the first message performative, named 'Inform'. After selecting this message performative, we have to select the recipient from the "Receiver" drop-down list. Since only one ReceiverAgent named 'Fatima' has been created, only this agent will be shown in the drop-down list. After selecting the recipient, we can write any content on the "Content" TextField. Then we can click on the "Send" button. After clicking the "Send" button, the content

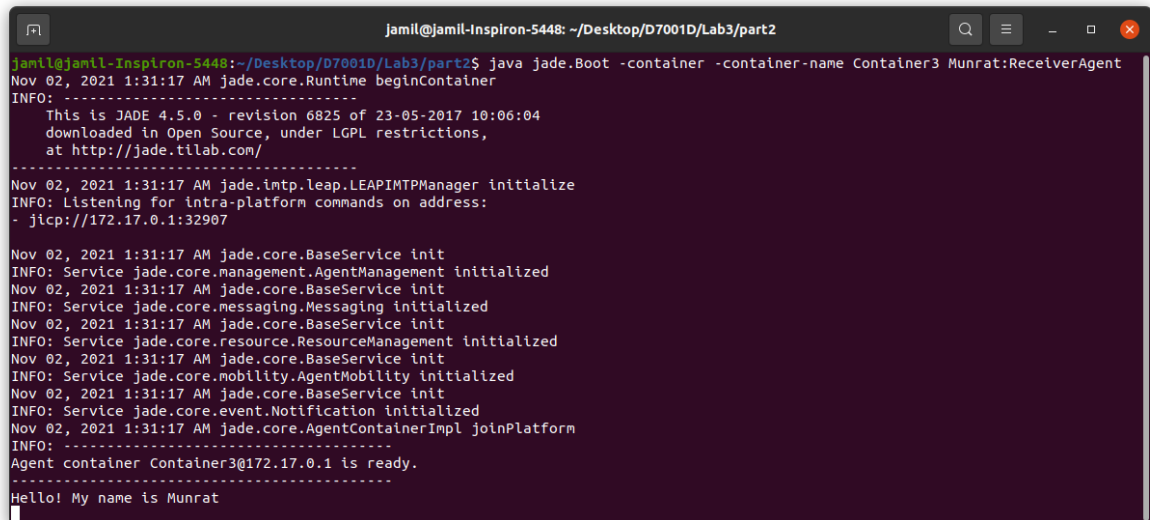


that the sender sent to the recipient, will be shown in the “Sent Messages” text area, while the response from the ReceiverAgent to SenderAgent will be shown in the “Received Messages” TextArea, as illustrated below.

The screenshot shows a window titled "Agent Smith" with a dark header bar. Inside the window, the interface is organized into several sections:

- Message Performative:** A dropdown menu currently displaying "INFORM".
- Receiver:** A dropdown menu displaying "Fatima" and a "Refresh" button to its right.
- Content:** A text input field containing "Hello Fatima." Below this field are two buttons: "Send" and "Cancel".
- Sent Messages:** A text area containing the text "Hello Fatima.".
- Received Messages:** A text area containing the text: "Reply from the receiver (Fatima) --> Successfully received a message from the sender where, the Content of the Message is --> Hello Fatima. and message Performative is --> INFORM".

Now, suppose another ReceiverAgent, named Munrat, has been created on Container3, as shown below.



```

jamil@jamil-Inspiron-5448: ~/Desktop/D7001D/Lab3/part2
jamil@jamil-Inspiron-5448:~/Desktop/D7001D/Lab3/part2$ java jade.Boot -container -container-name Container3 Munrat:ReceiverAgent
Nov 02, 2021 1:31:17 AM jade.core.Runtime beginContainer
INFO: -----
This is JADE 4.5.0 - revision 6825 of 23-05-2017 10:06:04
downloaded in Open Source, under LGPL restrictions,
at http://jade.tilab.com/
-----
Nov 02, 2021 1:31:17 AM jade.imtp.leap.LEAPIMTPManager initialize
INFO: Listening for intra-platform commands on address:
- jicp://172.17.0.1:32907

Nov 02, 2021 1:31:17 AM jade.core.BaseService init
INFO: Service jade.core.management.AgentManagement initialized
Nov 02, 2021 1:31:17 AM jade.core.BaseService init
INFO: Service jade.core.messaging.Messaging initialized
Nov 02, 2021 1:31:17 AM jade.core.BaseService init
INFO: Service jade.core.resource.ResourceManagement initialized
Nov 02, 2021 1:31:17 AM jade.core.BaseService init
INFO: Service jade.core.mobility.AgentMobility initialized
Nov 02, 2021 1:31:17 AM jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
Nov 02, 2021 1:31:17 AM jade.core.AgentContainerImpl joinPlatform
INFO: -----
Agent container Container3@172.17.0.1 is ready.
-----
Hello! My name is Munrat

```

Now, in the GUI, if we click on “Refresh” button, Munrat will also appear in the “Receiver” drop-down list. Suppose, we want to send a message to Munrat. So first, we select the message performative from the drop-down list. Suppose we select “Request”. After selecting the message performative, we select the recipient ‘Munrat’ from the “Receiver” drop-down list. Then we can write the desired content. After that, we can click on the “Send” button. After clicking the “Send” button, the content that the sender sent, will be shown in the “Send Message” text area, while the response from the ReceiverAgent to SenderAgent will be shown in the “Received Message” text area.

**Agent Smith**

**Message Performative:**

REQUEST

**Receiver:**

Munrat

Refresh

**Content:**

Hello Munrat. How are you?

Send

Cancel

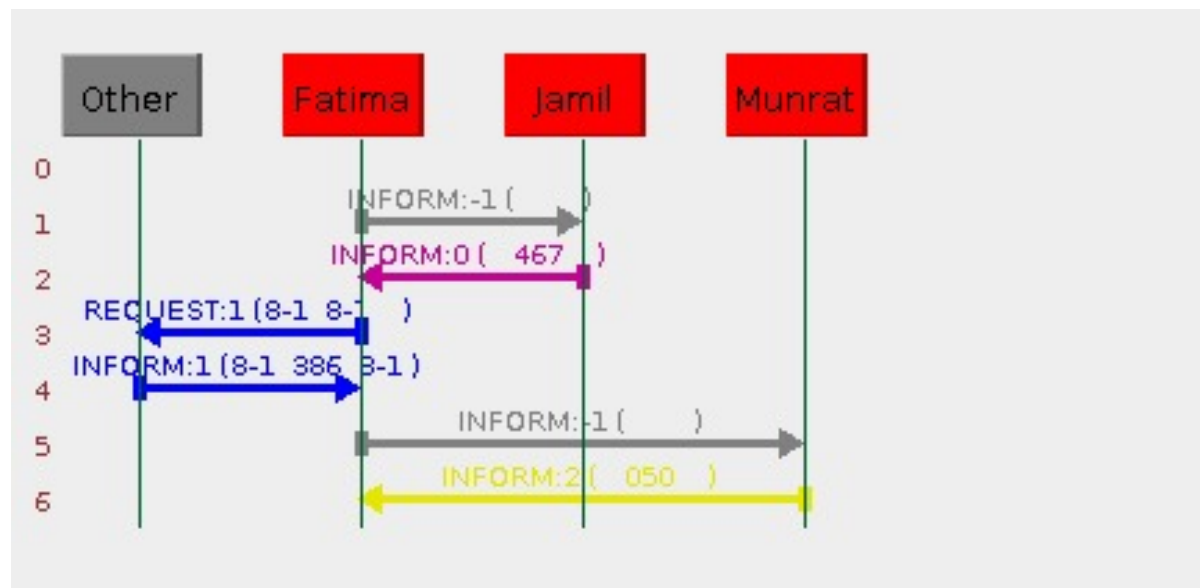
**Sent Messages**

Hello Munrat. How are you?

**Received Messages**

Reply from the receiver (Munrat) -->  
Successfully received a message from the sender where,  
the Content of the Message is --> Hello Munrat. How are you?  
and message Performative is --> REQUEST

To check the communication, sniffer agents were used:



# Bibliography

Vaucher, J., & Ncho, A. (2003). *Using jade behaviours*. Retrieved November 1, 2021, from <https://www.iro.umontreal.ca/~vaucher/Agents/Jade/primer6.html>