JADEX Framework Exercises I

Integrated Master's in Informatics Engineering

Intelligent Agents

2017/2018

Synthetic Intelligence Lab

Ricardo Ramos

Filipe Gonçalves

Paulo Novais

ISLab

Useful Links

- https://sourceforge.net/projects/jadex/files/jadex/2.4/
- https://download.actoron.com/docs/releases/jadex-3.0.0-RC1/jadex-mkdocs/BDI%20V3%20Tutorial/01%20Introduction/
- https://download.actoron.com/docs/releases/jadex-3.0.0-RC80/jadex-mkdocs/getting-started/getting-started/#ide-setup
- https://download.actoron.com/docs/releases/jadex-3.0.0-RC80/jadex-mkdocs/getting-started/getting-started/#starting-the-jadex-control-center
- https://download.actoron.com/docs/releases/jadex-3.0.0-RC80/jadex-mkdocs/getting-started/getting-started/#using-intellij-idea
- https://download.actoron.com/docs/releases/jadex-3.0.43/jadex-mkdocs/tutorials/bdiv3/01%20Introduction/

Useful MicroAgent functions

```
public IFuture<Void> executeBody() { // Function called after Agent being started;

    final Future ret = new Future();

    IComponentStep step = new IComponentStep() {

        public IFuture<Void> execute(IInternalAccess ia) { // Execute step

            …

            return IFuture.DONE;

        }

}
```

Useful MicroAgent functions

```
// Function called after agent receiving a message

public void messageArrived(Map msg, MessageType mt) {

    if ((String) msg.get("performative").equals(" query-if")) {

        System.out.println("Message received");

    }

    }
```

Useful MicroAgent functions

// Sending a message

```
String convid = SUtil.createUniqueId(MyAgent.this.getAgentName());

Map msg = new HashMap();

msg.put("content", content);

msg.put("performative", "query-if");

msg.put("conversation_id", convid);

msg.put("receivers", new IComponentIdentifier[] {new ComponentIdentifier
("ReceiverAgent", getComponentIdentifier().getParent())});

MyAgent.this.sendMessage(msg, SFipa.FIPA_MESSAGE_TYPE);

MyAgent.this.waitFor(1000, this); // Timeout if needed
```

Useful MicroAgent functions

// Reply a message (inside messageArrived(Map msg, MessageType mt))

```
Map reply = createReply(msg, mt);

reply.put("content", "pong");

reply.put("performative", "inform");

reply.put("sender", getComponentIdentifier());

sendMessage(reply, mt);
```

JADEX First Exercise

1. Reuse the Java Project available at elearning Platform;

2. Create the PongAgent class that inherits from the MicroAgent class, in the package tutorial. This agent will be responsible for responding "Pong" to messages that contain "Ping" as content;

3. Now create the Ping agent. This agent will be responsible for starting the communication. For this create the PingAgent class that inherits from MicroAgent, in the package tutorial;

Note: executeBody() method runs when the agent is created. That is why in this method we send the "Ping" message to the Pong agent. This message will be sent several times, if no response is obtained the Ping agent will cease communications.

JADEX First Exercise

4. Create the PingPongScenario.application.xml file. The ADF should always have a name of the *.application.xml type of tools such as JCC can easily identify them;

5. Open JCC and run the PingPongScenario.application.xml;

Note:

- In the node componenttypes are indicated which agents available. In this case, as both are micro-agents, the path for the class files the XML file that defines the agent to be included.
- The configurations node includes the settings available for our application. In this case it must indicate that both agents must be executed.

JADEX Framework Exercises I

Integrated Master's in Informatics Engineering

Intelligent Agents

2017/2018

Synthetic Intelligence Lab

Ricardo Ramos

Filipe Gonçalves

Paulo Novais

ISLab