Workshop 9 in AOM & MAS

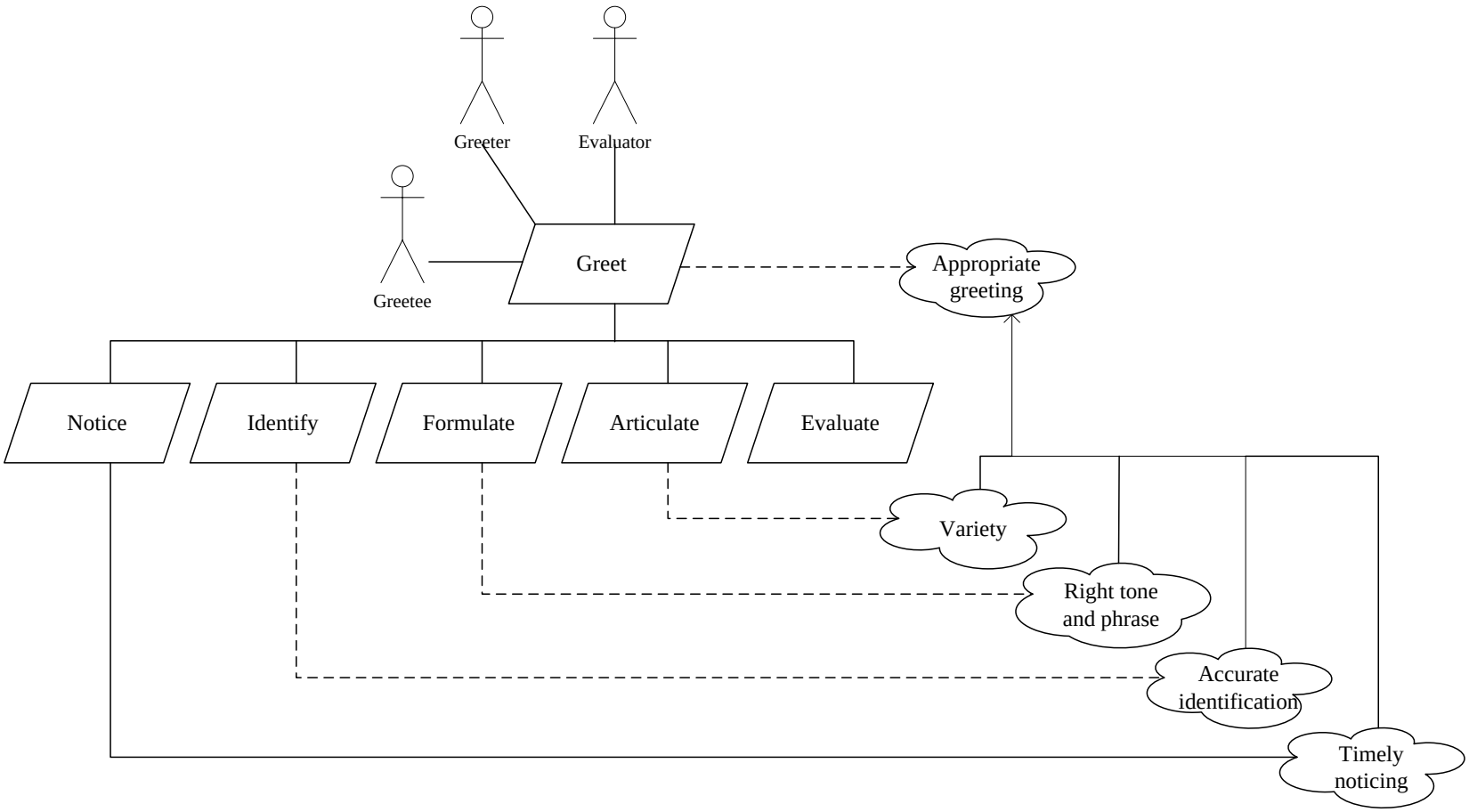
Prof Kuldar Taveter,

Tallinn University of Technology

Have you decided the team and topic of your miniproject?

- The teams and topics of your miniprojects should have been decided by ***24 March***
- If you have not done so yet, please mail the names of your team members and the title of your topic to Prof Taveter

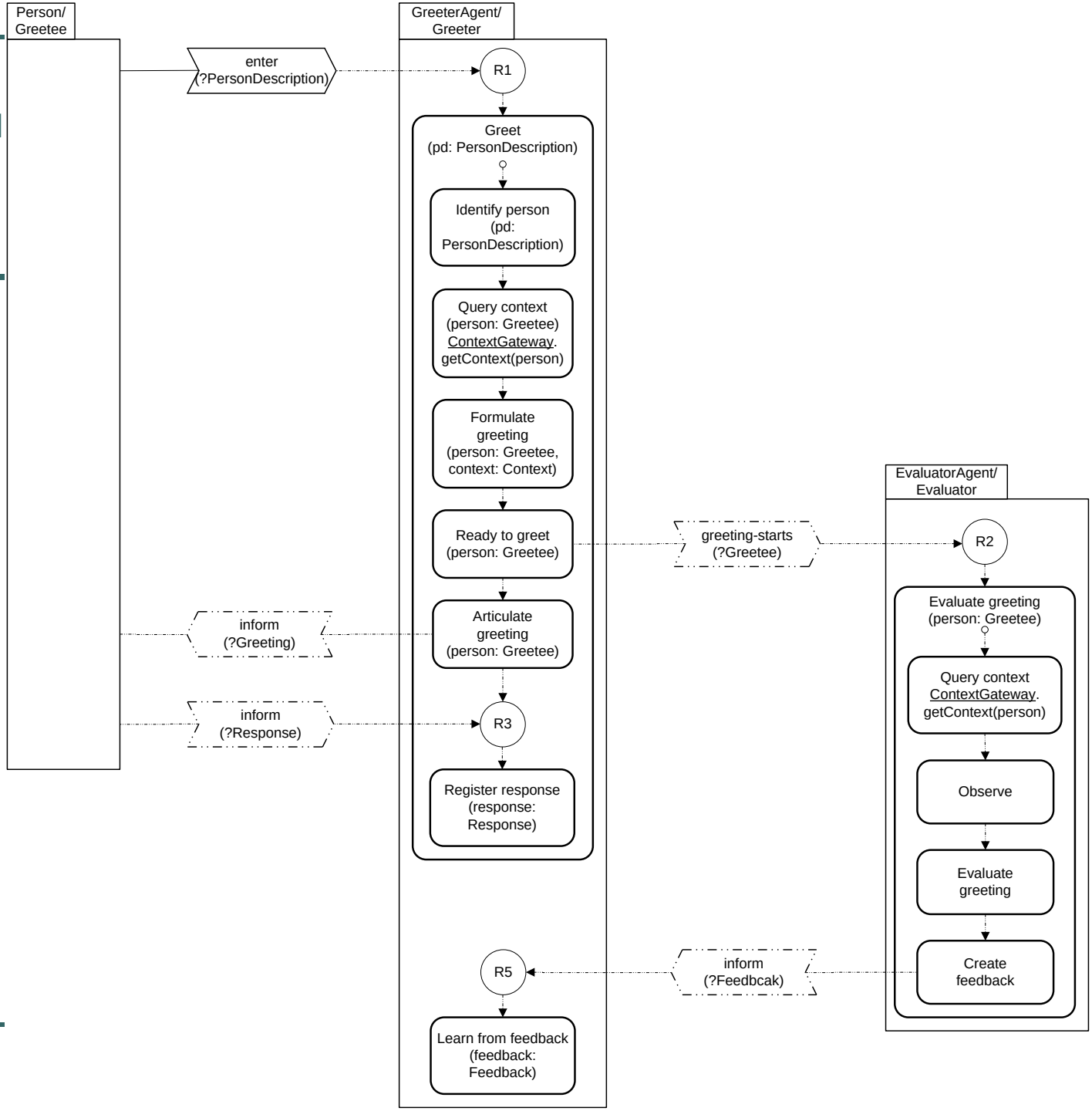
Greeting Goal Model



Greeting Role Model

- Greetee:
  - *To be greeted by greeter*
  - Responsibilities:
    - To be noticed by greeter; To perceive greeting
  - Constraints: None
- Greeter:
  - *To greet another agent coming within environment*
  - Responsibilities:
    - To notice greetee; To formulate greeting; To articulate greeting
  - Constraints: Articulation within 10 seconds of noticing; Formulation must be appropriate to greetee + environment
- Evaluator:
  - *To evaluate the greeting*
  - Responsibilities:
    - To observe greeting; To evaluate greeting; To publish report
  - Constraints: timeliness

Combined behaviour and interaction model for greeting



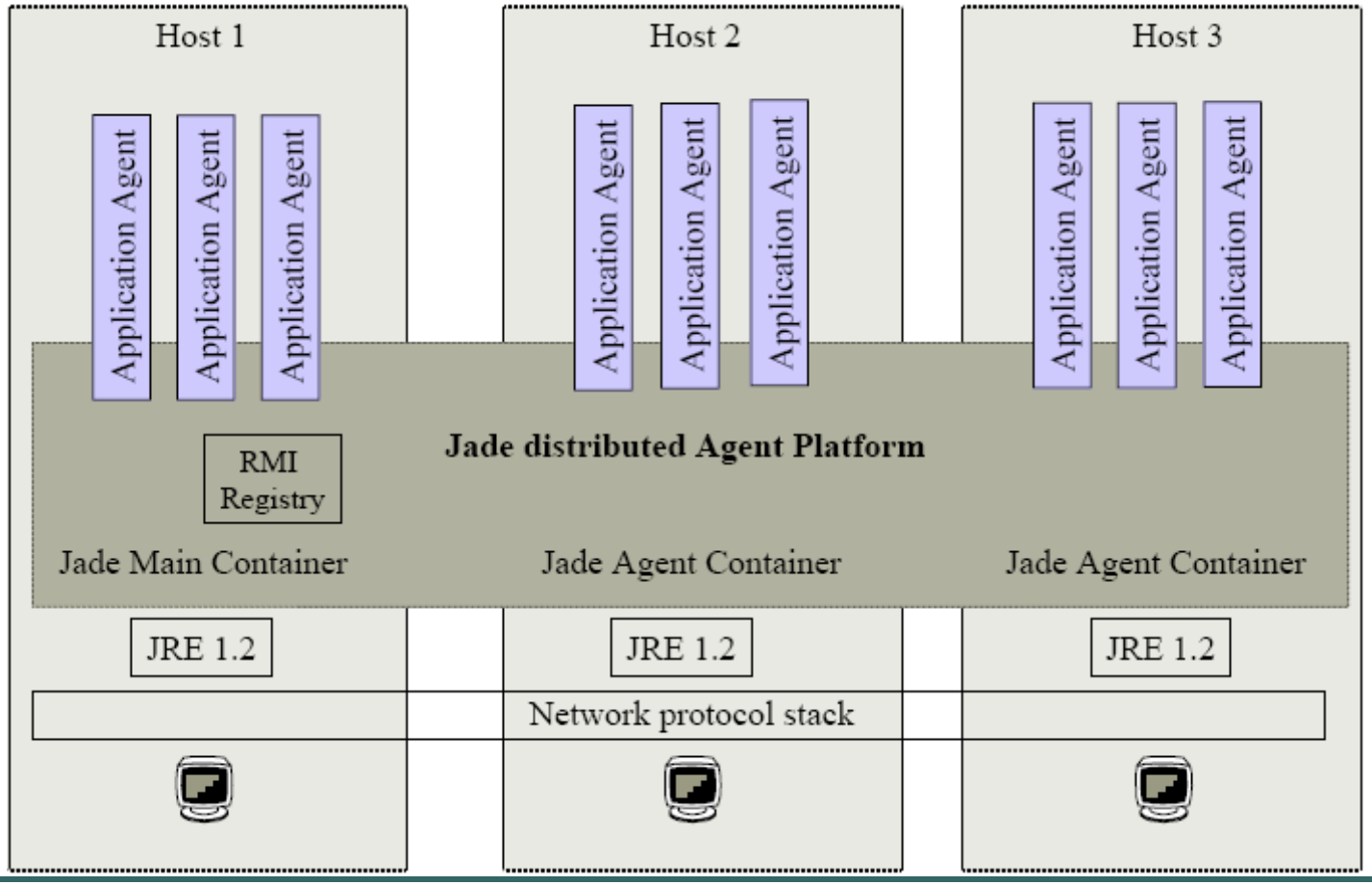
Exercises

- Create two JADE agents that greet each other. Follow the Greeting Goal Model and Greeting Role Model.
- Continue with the design for your miniproject either manually or using a suitable tool.

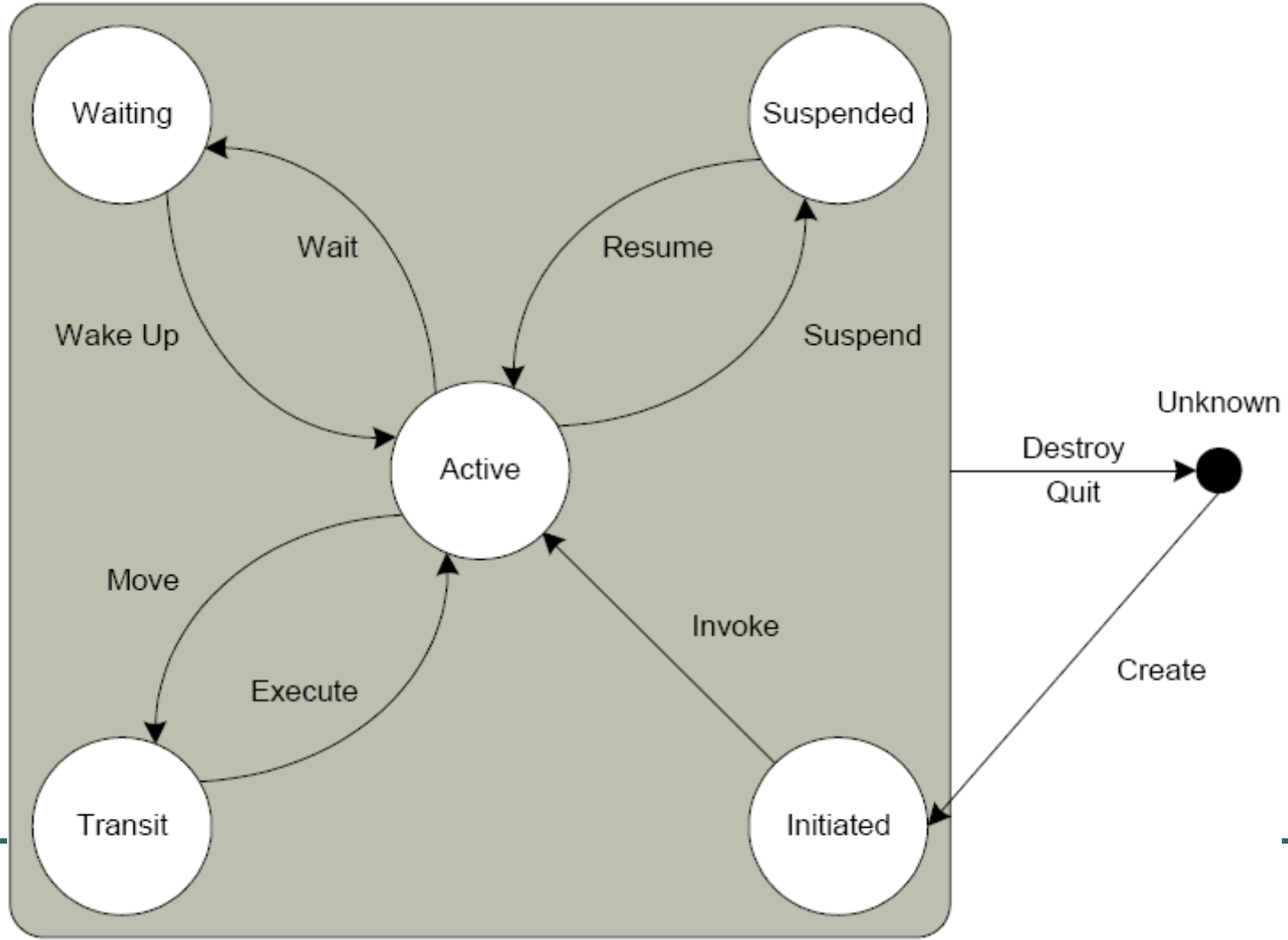
JADE (Java Agent Development Environment)

- Distributed agent platform which can be split among several hosts
- Java Application Programmer's Interface.
- Graphical User Interface to manage several agents from the same Remote Management Agent
- Library of FIPA interaction protocols, such as Contract Net
- Available at http://jade.cselt.it/

JADE Agent Platform



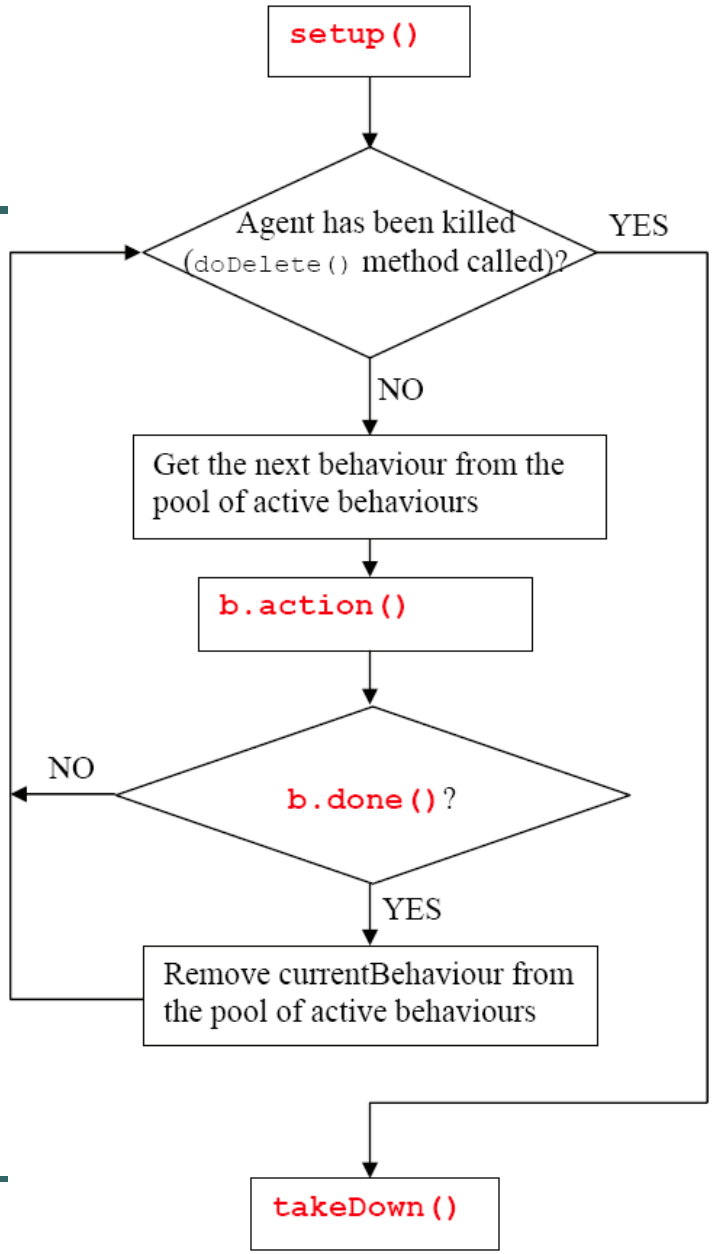
Agent life cycle



Concurrent tasks

- An agent must be able to carry out several concurrent tasks in response to different external events
- Every JADE agent is composed of a single execution thread
- Concurrent tasks are modelled and can be implemented as instances of jade.core.behaviours.Behaviour

Agent thread



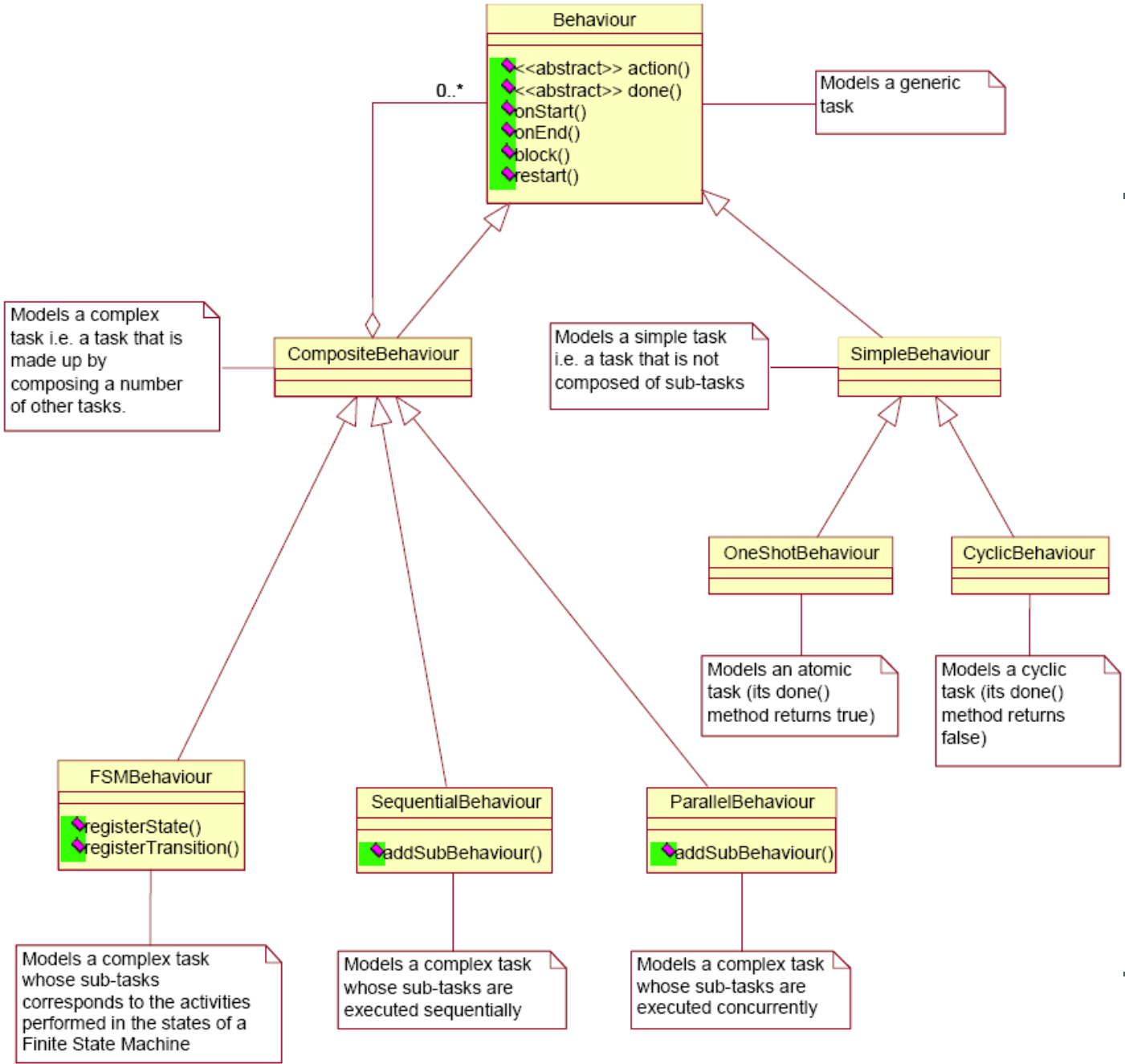
- Initializations
- Addition of initial behaviours

Highlighted in red the methods that programmers have to implement

- Agent "life" (execution of behaviours)

- Clean-up operations

Hierarchy of behaviours



Defining JADE agents

```
package DigitalPet;
import jade.core.*;

public class Tamagotchi extends Agent {

    // Put agent initializations here
    protected void setup() {
        // Adding behaviours
        addBehaviour(new MessageHandler (this));
        …
    }

    // If needed, put agent clean-up operations here
    protected void takeDown() {
        System.out.println("Tamagotchi "+getAID().getName()+" terminating.");
        …
    }
}
```

Defining behaviours

```
package DigitalPet;

import jade.core.*;
import jade.core.behaviours.*;
import jade.lang.acl.*;

public class MyOneShotBehaviour extends OneShotBehaviour {
        public void action() {
            // perform operation X
        }
}
public class MyCyclicBehaviour extends CyclicBehaviour {
        public void action() {
            // perform operation Y
        }
}
```

Sending messages

```
ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
msg.addReceiver(new AID(“tama1”, false);
msg.setLanguage(“English”);
msg.setOntology(“Weather-forecast-ontology”);
msg.setContent(“Today it’s raining”);
myAgent.send(msg);

// Message carrying a request for offer
ACLMessage cfp = new ACLMessage(ACLMessage.CFP);
for (int i = 0; i < sellerAgents.lenght; ++i) {
    cfp.addReceiver(sellerAgents[i]);
}
cfp.setContent(targetBookTitle);
myAgent.send(cfp);
```

Receiving messages

```
public void action() {
    ACLMessage msg = myAgent.receive();
    if (msg != null) {
        // Message received. Process it
        ...
    }
    else {
        block();
    }
}
```

Setting classpath

- Please include in the classpath the following library files:
  - …\jade\lib\jade.jar
  - …\jade\lib\jadeTools.jar
  - …\jade\lib\http.jar
  - …\jade\lib\iiop.jar
- Please include in the classpath the location(s) of your Java class files

Compiling and running JADE agents

javac Tamagotchi.java Behaviours.java

…

java jade.Boot –gui -platform

java jade.Boot –container tama1:DigitalPet.Tamagotchi

Please consult API!

- http://jade.tilab.com/doc/api/index.html

Passing arguments to an agent

```
public class BookBuyerAgent extends Agent {
    private String targetBookTitle;
    // The list of known seller agents
    private AID[] sellerAgents = {new AID("seller1", AID.ISLOCALNAME),
                                  new AID("seller2", AID.ISLOCALNAME)};
    // Put agent initializations here
    protected void setup() {
        // Printout a welcome message
        System.out.println("Hello! Buyer-agent" +getAID().getName()+
                " is ready.");
        // Get the title of the book to buy as a start-up argument
        Object[] args = getArguments();
        if (args != null && args.length > 0) {
            targetBookTitle = (String) args[0];
            System.out.println("Trying to buy" + targetBookTitle);
        }
        else {
            // Make the agent terminate immediately
            System.out.println("No book title specified");
            doDelete();
        }
    }
…
}
```

Running an agent with arguments

java jade.Boot –container buyer:BookBuyerAgent (The-Lord-of-the-rings)