

Fabiola Dąbroś

Inżynieria Obliczeniowa gr.1

Ćwiczenia 8 – Komunikacja

Wykonanie ćwiczenia rozpoczęłam od utworzenia klasy agenta o nazwie ag odb.

Agent ten odbiera cyklicznie wiadomości ze skrzynki.

Jeśli wiadomość jest typu REQUEST wypisuje treść wiadomości na ekranie i w odpowiedzi wysyła do nadawcy komunikat typu INFORM z zawartością „wykonałem”.

Jeśli wiadomość jest typu CFP wypisuje treść wiadomości na ekranie i w odpowiedzi wysyła do nadawcy komunikat typu REQUEST z zawartością „raz jeszcze”.

Jeśli wiadomość jest innych typów wysyła komunikat typu NOT_UNDERSTOOD z dowolną zawartością.

Kod prezentuje się następująco:

```
public class ag_odb extends Agent {

    protected void setup() {

        addBehaviour(new CyclicBehaviour(this) {
            public void action() {
                ACLMessage msg = myAgent.receive();
                if (msg != null) {
                    ACLMessage reply = msg.createReply();
                    if (msg.getPerformative() == ACLMessage.REQUEST) {
                        System.out.println("Received REQUEST message from agent "+msg.getSender());
                        System.out.println("Content : "+ msg.getContent());
                        reply.setPerformative(ACLMessage.INFORM);
                        reply.setContent("Wykonałem");
                    } else if(msg.getPerformative() == ACLMessage.CFP){
                        System.out.println("Received CFP message from agent "+msg.getSender());
                        System.out.println("Content : "+msg.getContent());
                        reply.setPerformative(ACLMessage.REQUEST);
                        reply.setContent("Raz jeszcze");
                    }
                    else {
                        reply.setPerformative(ACLMessage.NOT_UNDERSTOOD);
                        reply.setContent("Nie znam");
                    }
                    myAgent.send(reply);
                }
                else {
                    block();
                }
            }
        });
    }
}
```

Należy wysłać wiadomość klikając na naszego agenta:

ACL Message

ACLMessage Envelope

Sender: Set

Receivers: aagent@192.168.21.1:1099/JADE

Reply-to:

Communicative act: request

Content: Message

Language:

Encoding:

Ontology:

Protocol: Null

Conversation-id:

In-reply-to:

Reply-with:

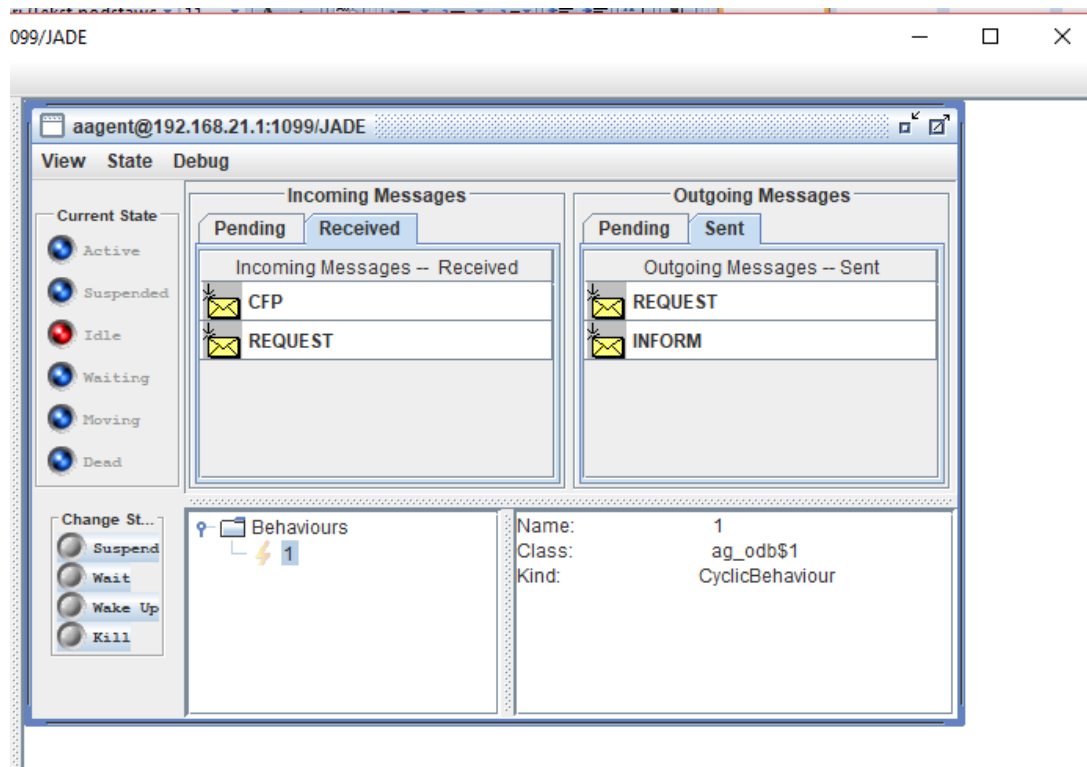
Reply-by: Set

User Properties:

OK Cancel

W wyniku czego otrzymujemy:

```
Agent connected from console@aagent@192.168.21.1:1099/JADE  
-----  
Received CFP message from agent rma@192.168.21.1:1099/JADE  
Content : Zawartosc cfp  
Received CFP message from agent rma@192.168.21.1:1099/JADE  
Content : zawartosc kolejna cfp
```



Kolejnym krokiem było utworzenie klasy agenta o nazwie ag_wys_odb. Agent ten losuje liczbę. W przypadku wylosowania 0 wysyła komunikat typu CFP o treści cfp message, a w przypadku wylosowania 1 wysyła komunikat typu Request. Kolejno po wysłaniu wiadomości agent odbiera wiadomość i wypisuje ją na ekranie, jeśli wiadomość jest typu Inform to agent się usuwa a w przeciwnym wypadku przechodzi do kroku pierwszego.

Kod prezentuje się następująco:

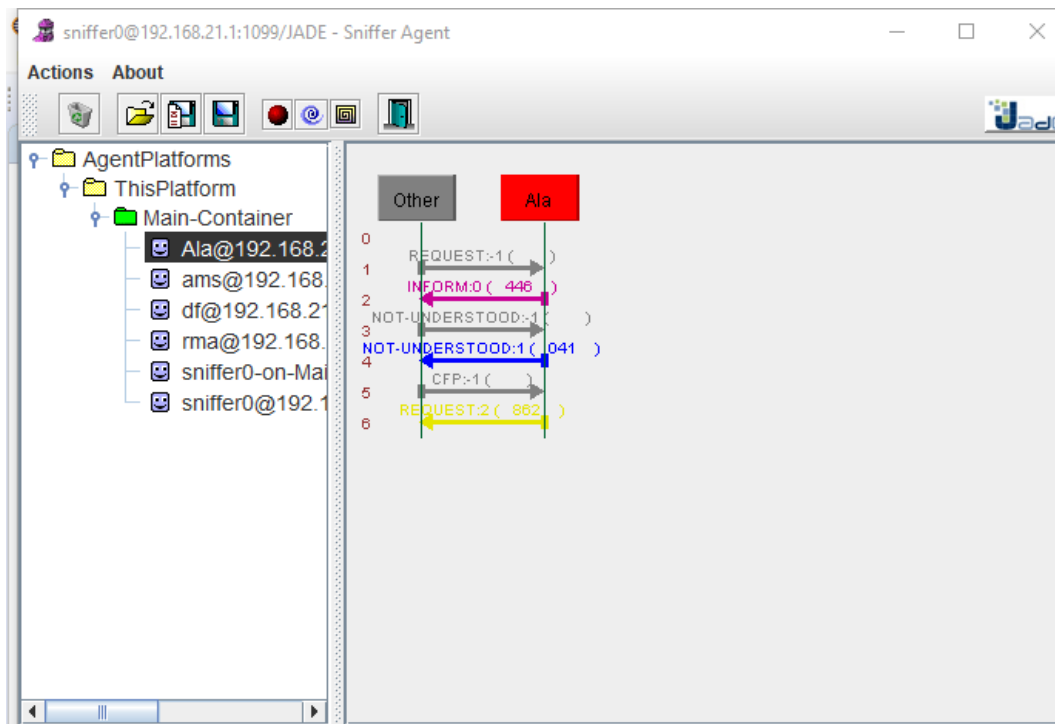
```
public class ag_wys_odb extends Agent {

    Random r;
    protected void setup() {
        System.out.println("Hello World! My name is : " +getLocalName());
        addBehaviour(new CyclicBehaviour(this) {
            public void action() {
                r = new Random();
                int state = r.nextInt(2);
                System.out.println(state);
                if (state == 0 ) {
                    ACLMessage m = new ACLMessage(ACLMessage.CFP);
                    m.addReceiver(new AID("Ala", AID.ISLOCALNAME));
                    m.setOntology("presence");
                    m.setContent("cfp message");
                    send(m);
                }else{
                    ACLMessage m = new ACLMessage(ACLMessage.REQUEST);
                    m.addReceiver(new AID("Ala", AID.ISLOCALNAME));
                    m.setOntology("presence");
                    m.setContent("request message");
                    send(m);
                }
            }

            ACLMessage msg = myAgent.receive();
            if (msg != null) {
                System.out.println("Message "+msg.getContent());
                if (msg.getPerformative() == ACLMessage.INFORM){
                    myAgent.doDelete();
                }
            }else {
                block();
            }
        }
    }
}
```

Po uruchomieniu agenta Ala z poprzedniego zadania oraz utworzenia agenta nowej klasy otrzymujemy:

```
-----
Received CFP message from agent Agent@192.168.21.1:1099/JADE
Content : cfp message
0
Message Raz jeszcze
Received CFP message from agent Agent@192.168.21.1:1099/JADE
Content : cfp message
0
Message Raz jeszcze
Received CFP message from agent Agent@192.168.21.1:1099/JADE
Content : cfp message
1
Message Raz jeszcze
0
Received REQUEST message from agent Agent@192.168.21.1:1099/JADE
Content : request message
0
Received CFP message from agent Agent@192.168.21.1:1099/JADE
Content : cfp message
Message Wykonałem
Received CFP message from agent Agent@192.168.21.1:1099/JADE
Content : cfp message
```



Kolejnym krokiem było utworzenie klasy agenta o nazwie `ag_pl`. Agent ten pobiera ze skrzynki wiadomości, które w polu język mają zawartość „polski”. Wszystkie inne wiadomości zostają w skrzynce nieodebrane. Po odebraniu wiadomość jest wypisywana na ekranie.

Kod prezentuje się następująco:

```
public class ag_pl extends Agent {

    private MessageTemplate template = MessageTemplate.MatchLanguage("Polski");
    protected void setup() {

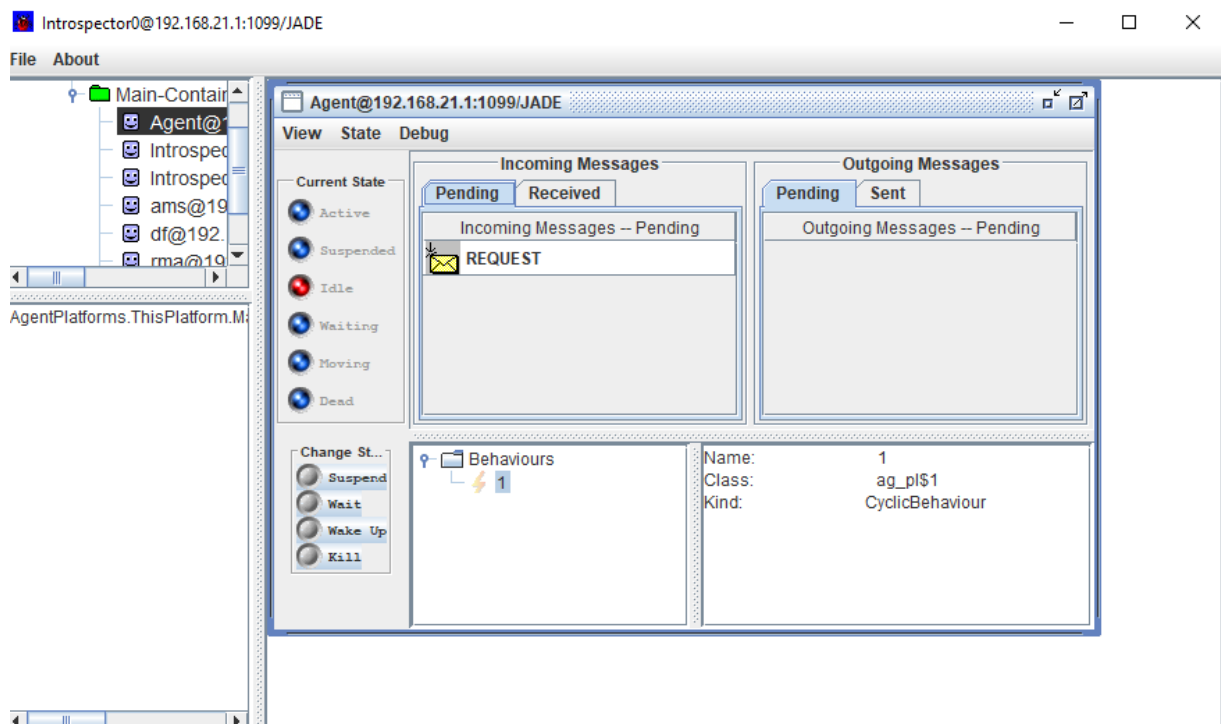
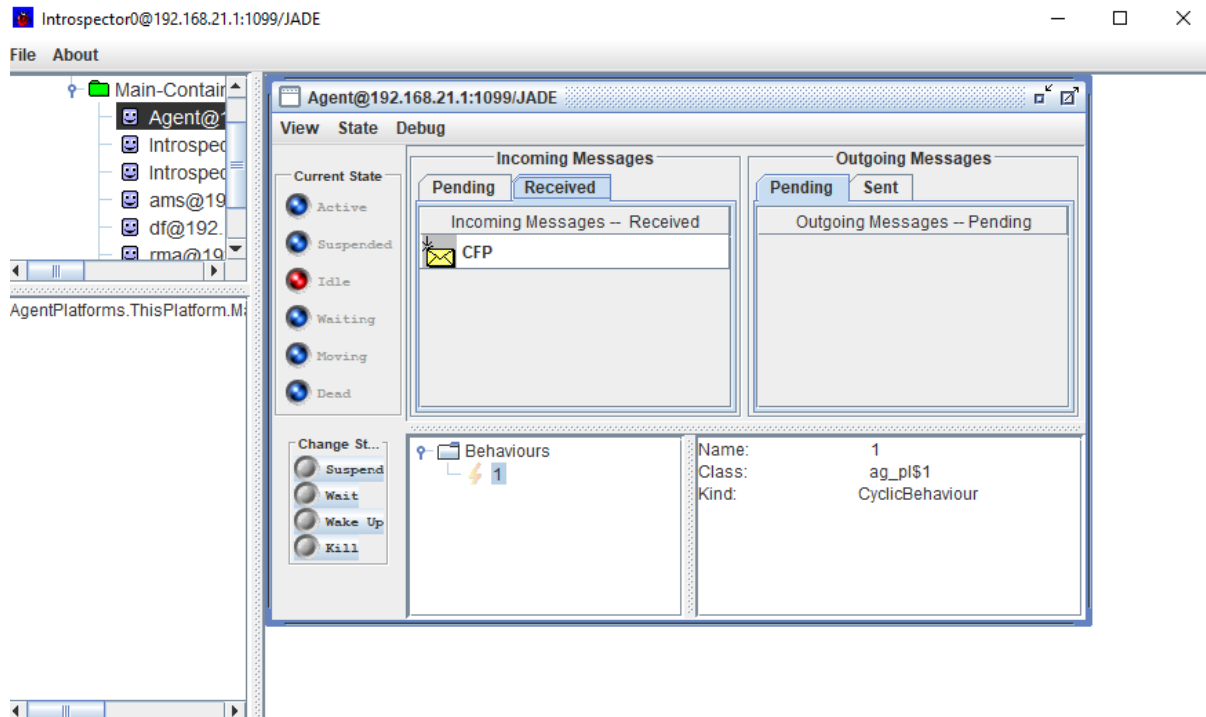
        addBehaviour(new CyclicBehaviour(this) {
            public void action() {

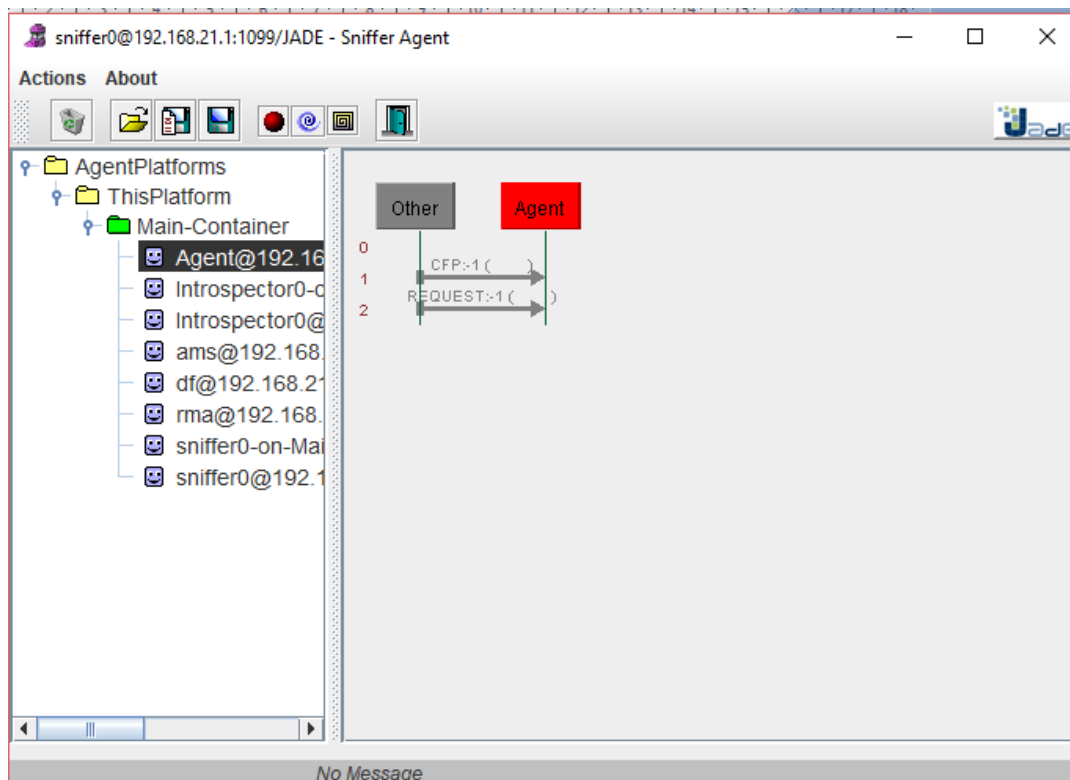
                ACLMessage msg = myAgent.receive(template);
                if (msg != null) {
                    System.out.println("Message from agent : " + msg.getSender().getNam
                    System.out.println("Content " + msg.getContent());
                }
                else {
                    block();
                }
            }
        });
    }
}
```

W wyniku czego otrzymujemy:

W sytuacji gdy wprowadzimy wiadomość po polsku, podczas analizy introspektorem zauważamy że wiadomość pojawia się w polu Received. Natomiast w przypadku wiadomości w innym języku wiadomość trafia do pola Pending.

Natomiast w przypadku analizy snifferem ruch wszystkich wiadomości jest zarejestrowany.





Message from agent : rma@192.168.21.1:1099/JADE
Content wiadomosc po polsku