Magdalena Osiewicz, IO, gr 2.

Rozproszona sztuczna inteligencja.

Sprawozdanie nr 9.

Temat: Komunikacja.

W ramach zajęć zrealizowałam następujące kroki:

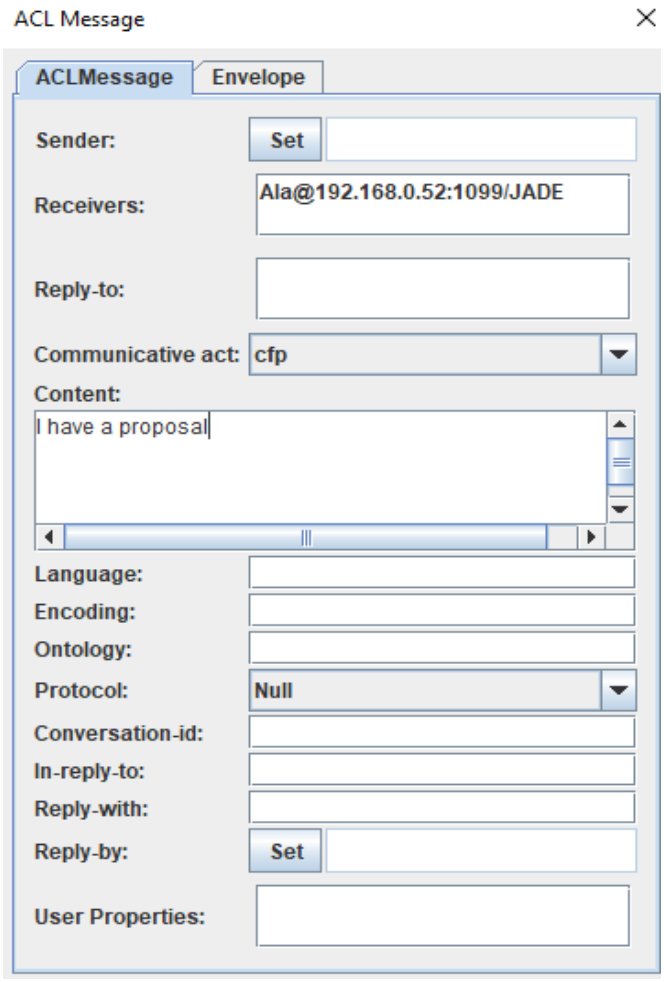
1. Utworzenie klasy agenta o nazwie Ag_odb.

Agent ten cyklicznie odbiera wiadomości ze skrzynki.

a) Jeśli wiadomość jest typu REQUEST wypisuje treść wiadomości na ekranie i w odpowiedzi wysyła do nadawcy komunikat typu INFORM z zawartością „wykonałem".
b) Jeśli wiadomość jest typu CFP wypisuje treść wiadomości na ekranie i w odpowiedzi wysyła do nadawcy komunikat typu REQUEST z zawartością „raz jeszcze".
c) Jeśli wiadomość jest innych typów wysyła komunikat typu NOT_UNDERSTOOD z dowolną zawartością.

```
if (aclMessage != null) {
    ACLMessage aclMessage1 = aclMessage.createReply();
    if (aclMessage.getPerformative() == ACLMessage.REQUEST) {
        System.out.println("The content of the message(request): "+ aclMessage.getContent());
        aclMessage1.setPerformative(ACLMessage.INFORM);
        aclMessage1.setContent("wykonalem");
    } else if(aclMessage.getPerformative() == ACLMessage.CFP){
        System.out.println("The content of the message(CFP): "+aclMessage.getContent());
        aclMessage1.setPerformative(ACLMessage.REQUEST);
        aclMessage1.setContent("raz jeszcze");
    } else {
        aclMessage1.setPerformative(ACLMessage.NOT_UNDERSTOOD);
        aclMessage1.setContent("I didn't understand.");
    }
    myAgent.send(aclMessage1);
}
```

Na załączonym zrzucie widać, w jaki sposób określono jakie akcje agent ma wykonać w zależności od typu wiadomości.



Każdy agent ma swoją „skrzynkę pocztową” (kolejkę komunikatów), w której JADE przechowuje wiadomości. Za każdym razem, gdy wiadomość jest umieszczana w kolejce, to agent odbierający jest powiadomiony.
Format wiadomości w JADE jest zgodny ze strukturą FIFA-ACL a każda z nich zawiera m.in pola takie jak: nadawca, lista odbiorców, treść wiadomości oraz typ wiadomości.

W zadaniu wykorzystano cztery z nich:
1) REQUEST – wskazuje na to, że nadawda chce, aby ten wykonał akcję

2) PROPOSE lub CFP – nadawca chce rozpocząć negocjacje

3) INFORM – przekazanie informacji

W celu pozyskania informacji na temat typu wiadomości wykorzystano w zadaniu metodę *getPerformative();*

W wyniku można zobaczyć zawartość dwóch wiadomości różnych typów: REQUEST i CFP:

```
----------------------------------------
cze 19, 2018 11:26:28 AM jade.imtp.leap.LEAPIMTPManager initialize
INFO: Listening for intra-platform commands on address:
- jicp://192.168.0.52:1099

cze 19, 2018 11:26:28 AM jade.core.BaseService init
INFO: Service jade.core.management.AgentManagement initialized
cze 19, 2018 11:26:28 AM jade.core.BaseService init
INFO: Service jade.core.messaging.Messaging initialized
cze 19, 2018 11:26:28 AM jade.core.BaseService init
INFO: Service jade.core.resource.ResourceManagement initialized
cze 19, 2018 11:26:28 AM jade.core.BaseService init
INFO: Service jade.core.mobility.AgentMobility initialized
cze 19, 2018 11:26:28 AM jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
cze 19, 2018 11:26:28 AM jade.mtp.http.HTTPServer <init>
INFO: HTTP-MTP Using XML parser com.sun.org.apache.xerces.internal.jaxp.SAXParserImpl$JAXPSAXParser
cze 19, 2018 11:26:28 AM jade.core.messaging.MessagingService boot
INFO: MTP addresses:
http://192.168.0.52:7778/acc
 :)
cze 19, 2018 11:26:28 AM jade.core.AgentContainerImpl joinPlatform
INFO: ----------------------------------------
Agent container Main-Container@192.168.0.52 is ready.
----------------------------------------
The content of the message(request): Prepare me a dinner!
The content of the message(CFP): I have a proposal
zaraz sie usune!
```

2. Utworzenie klasy agenta o nazwie ag_wys_odb.

Agent w pierwszym kroku losuje liczbę: 0 lub 1. Jeżeli wylosowano:

a) 0 – agent ten wysyła komunikat typu CFP o dowolnej treści do agenta o nazwie Ala.
b) 1 – agent ten wysyła komunikat typu REQUEST o dowolnej treści do agenta o nazwie Ala.

```
if (number == 0)
{
    ACLMessage aclMessage = new ACLMessage(ACLMessage.CFP);
    aclMessage.addReceiver(new AID( name: "Ala", AID.ISLOCALNAME));
    aclMessage.setOntology("presence");
    aclMessage.setContent("wysylam wiadomosc CFP");
    send(aclMessage);
} else if (number == 1){
    ACLMessage aclMessage = new ACLMessage(ACLMessage.REQUEST);
    aclMessage.addReceiver(new AID( name: "Ala", AID.ISLOCALNAME));
    aclMessage.setOntology("presence");
    aclMessage.setContent("wysylam wiadomosc REQUEST");
    send(aclMessage);
}
```

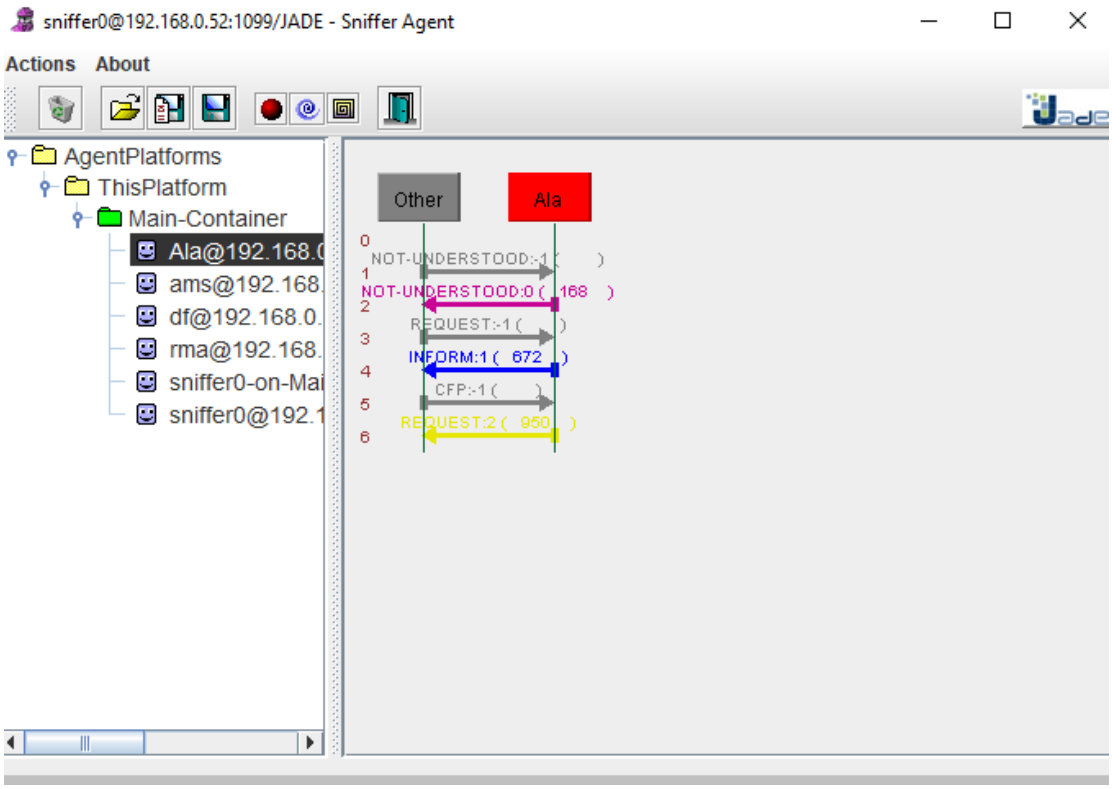
Komunikat w JADE jest zaimplementowany jako obiekt klasy jade.lang.acl.ACLMessage, która dostarcza metody get i set dla uzyskania dostępu dop wszystkich pól. W tej części zadania wykorzystano dodatkowo metodę *setOntology*, czyli wskazanie słownika symboli. Zarówno nadawca jak i odbiorca powinni przypisywać te same znaczenia tym samym symbolom, żeby komunikacja była skuteczna. Aby móc dodać odbiorcę potrzebujemy znać jego AID, w tym celu wykorzystano wyszukiwanie na podstawie nazwy lokalnej agenta „Ala”.

Po wysłaniu wiadomości agent przechodzi do drugiego kroku. W drugim kroku agent odbiera dowolną wiadomość, a następnie:
i. wypisuje ją na ekranie
ii. jeśli jest typu INFORM, agent się usuwa

```
ACLMessage msg = myAgent.receive();
if (msg != null) {
    if(msg.getPerformative()==ACLMessage.INFORM){
        System.out.println("INFORM-wiec usuwamy agenta");
        myAgent.doDelete();

    }
}
else {
    block();
}
```

W drugiej częśći zadania agent odbiera wiadamość za pomoca metody *receive*. Metoda ta zwraca pierwszą wiadomość w kolejce komunikatów powodując jej usunięcie lub null, jeśli koljek ajest pusta.

Uruchomienie platformy JADE, utworzenie agenta klasy ag_odb o nazwie Ala (z poprzedniego zadania) a następnie agenta klasy ag_odb_wys pozwoliło na zaobserowanie wymiany komunikatów z użyciem sniffera.



3. Utworzenie klasy agenta onazwie ag_pl.

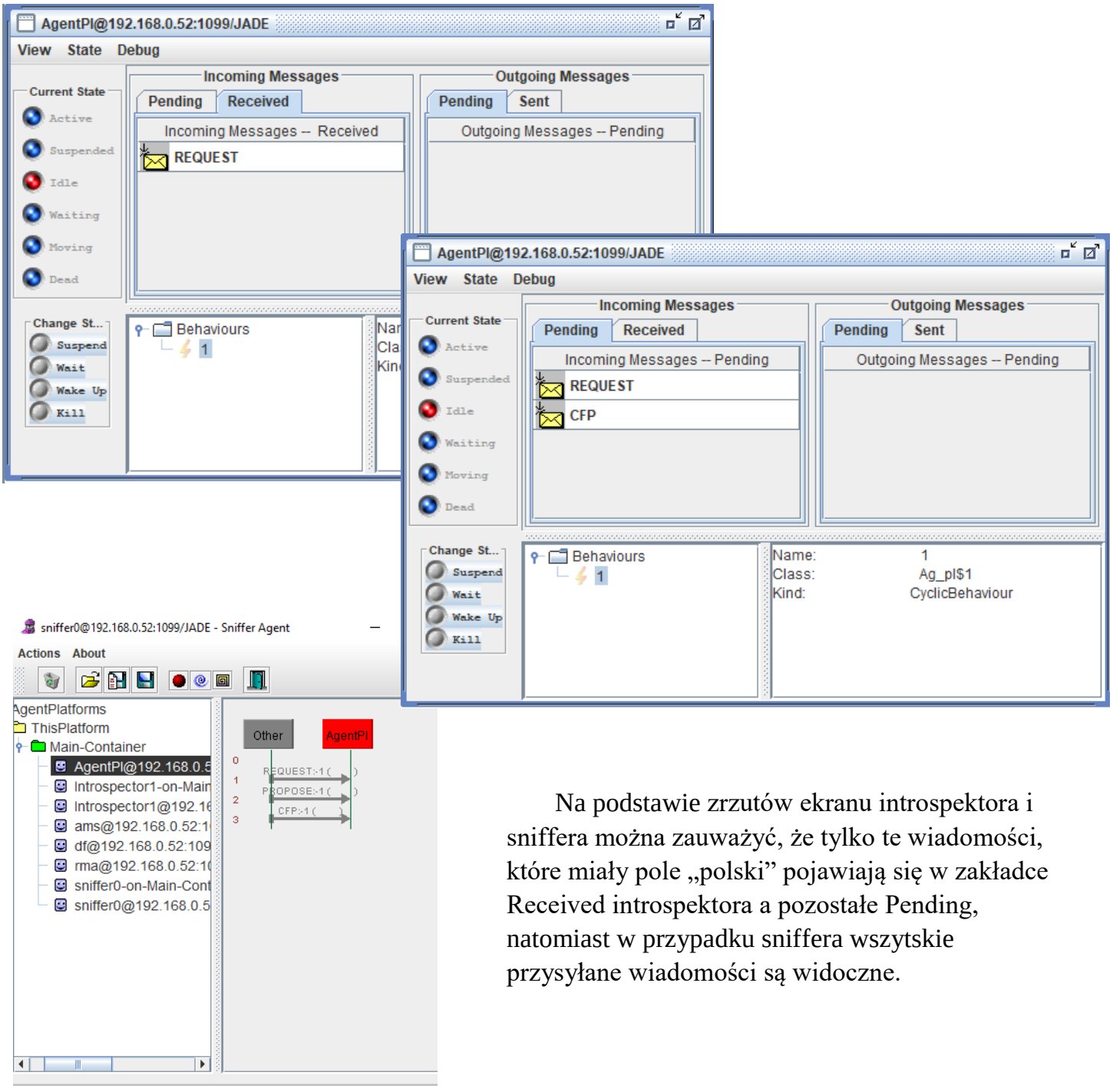
Agent ten pobiera ze skrzynki wiadomości, które w polu język mają zawartość „polski”. Wszystkie inne wiadomości zostają w skrzynce nieodebrane. Po odebraniu wiadomość jest wypisywana na ekranie.

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

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

            ACLMessage msg = myAgent.receive(template);
            if (msg != null) {
                System.out.println("The Message from agent : " + msg.getSender().getName());
                System.out.println("The zontent of the message " + msg.getContent());
            }
            else {
                block();
            }
        }
    } );
}
```

W celu odseparowania wiadomośći, które mają w polu język zawartość „polski” od pozostałych wykorzystano szablony. Są one implementowane jako instancje klasy MessageTemplate, które dostarczają wiele funkcji do tworzenia szablonów w bardzo prostu sposób.

Wyświetlono tylko wiadomości oznaczone jako „polski”.

```
INFO: Service jade.core.resource.ResourceManagement initialized
cze 19, 2018 1:15:06 PM jade.core.BaseService init
INFO: Service jade.core.mobility.AgentMobility initialized
cze 19, 2018 1:15:06 PM jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
cze 19, 2018 1:15:07 PM jade.mtp.http.HTTPServer <init>
INFO: HTTP-MTP Using XML parser com.sun.org.apache.xerces.internal.jaxp.SAXParserImpl$JAXPSAXParser
cze 19, 2018 1:15:07 PM jade.core.messaging.MessagingService boot
INFO: MTP addresses:
http://192.168.0.52:7778/acc
cze 19, 2018 1:15:07 PM jade.core.AgentContainerImpl joinPlatform
INFO: ----------------------------------------
Agent container Main-Container@192.168.0.52 is ready.
--------------------------------------------
The Message from agent : rma@192.168.0.52:1099/JADE
The zontent of the message Prepare me a dinner!
The Message from agent : rma@192.168.0.52:1099/JADE
The zontent of the message It's the eighteen of June 20018
```



Na podstawie zrzutów ekranu introspektora i sniffera można zauważyć, że tylko te wiadomości, które miały pole „polski” pojawiają się w zakładce Received introspektora a pozostałe Pending, natomiast w przypadku sniffera wszytskie przysyłane wiadomości są widoczne.