JGOMAS

JADE Game Oriented MultiAgent System

MAS: taxonomy, services & comunication

Sistemas Inteligentes
FI, 2006

Toni Barella
tbarella@dsic.upv.es

JGOMAS

Índice

- Taxonomía
- Comunicaciones
- Servicios
- Trabajo a realizar

JGOMAS

Índice

- ► **Taxonomía**
- Comunicaciones
- Servicios
- Trabajo a realizar

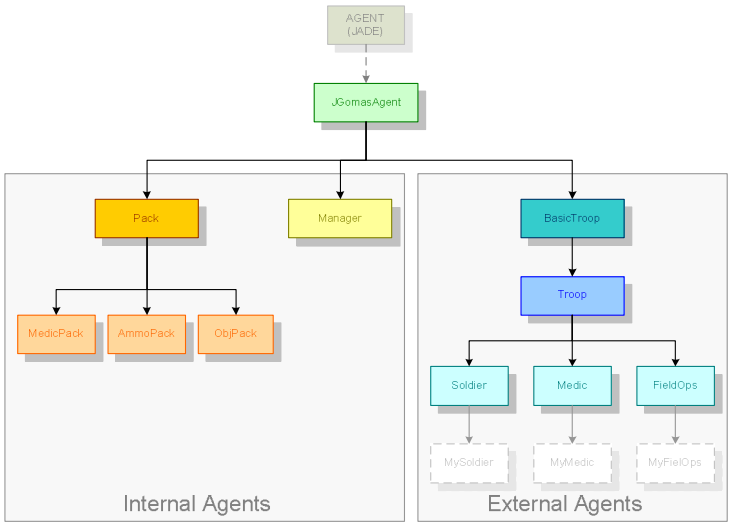
JGOMAS

Taxonomía (I)

- Hay definidos tres tipos de roles:
  - ***Soldier***: acude a dar apoyo
  - ***Medic***: acude a curar
  - ***FieldOps***: acude a dar munición
- Un agente asume un único rol durante toda la partida
- Cada rol tiene unas características y ofrece unos determinados servicios

JGOMAS

Taxonomía (II)



JGOMAS

Taxonomía (III)

- Agentes Internos:
  - ***Manager***: coordina todo el juego. Actúa de interfaz con los visualizadores gráficos.
  - ***Pack***: paquetes de medicina, munición y objetivo.
- Agentes Externos:
  - ***Troop***: agentes de usuario (*medic*, *fieldops* y *soldier*). Disponen de un comportamiento básico, que el usuario puede mejorar.

JGOMAS

Índice

- Taxonomía
- ► **Comunicaciones**
- Servicios
- Trabajo a realizar

JGOMAS

Comunicaciones

- Estándar FIPA
  - Partículas ilocutivas
- Formato de los mensajes
  - StringTokenizer
- Uso de templates y conversationID

JGOMAS Comunicaciones

Estándar FIPA

- Protocolos establecidos
  - Soportados por JADE
- Partículas ilocutivas
  - INFORM
  - REQUEST
  - AGREE
  - REFUSE
  - CANCEL

JGOMAS Comunicaciones

Formato de los mensajes

- StringTokenizer

```
StringTokenizer tokens = new StringTokenizer(sContent);

tokens.nextToken(); // Get "ID:"
Integer id = Integer.parseInt(tokens.nextToken());

tokens.nextToken(); // Get "("
double x = Double.parseDouble(tokens.nextToken());
tokens.nextToken(); // Get ","
double y = Double.parseDouble(tokens.nextToken());
tokens.nextToken(); // Get ","
double z = Double.parseDouble(tokens.nextToken());
```

I	D	:		5		(		1	7	2	.	1		,		2	0	.	5		,		4	8	.	3		)	

Templates

- Uso de templates y conversationID:

```
MessageTemplate template = MessageTemplate.and(
        MessageTemplate.MatchPerformative(ACLMessage.INFORM),
        MessageTemplate.MatchConversationId("SHOT"));

ACLMessage msg = receive(template);

if ( msg != null ) {
        .
        .
        Agent does some actions
        .
        .
}
else
        block();
```


Índice

- Taxonomía
- Comunicaciones
- ► **Servicios**
- Trabajo a realizar

Servicios

- Registro
- Solicitud
- Respuesta

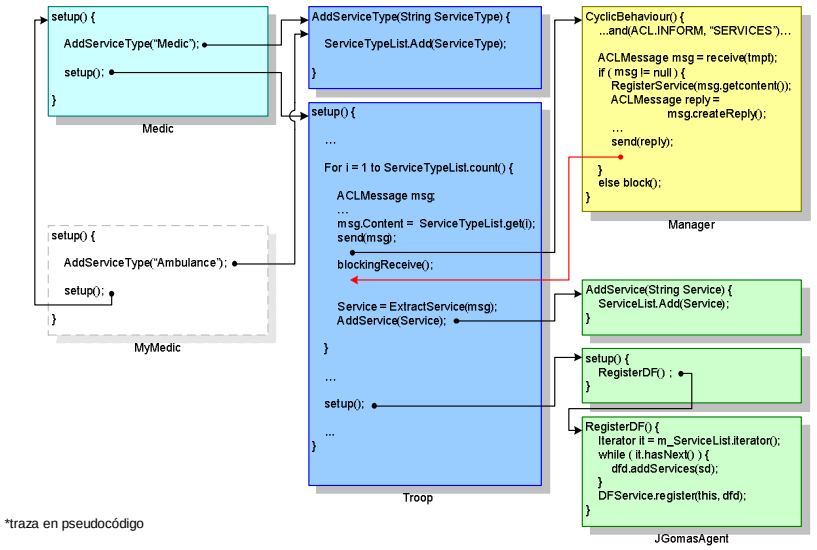
Registro (I)

- Un rol debe registrar un servicio para que el resto de roles puedan solicitarlo:

```
void setup() {
    …
    AddServiceType("Medic");
    super.setup();
    …
}
```

- Existen definidos tres tipos de servicios básicos:
  - **m_sMedicService;**
  - **m_sAmmoService;**
  - **m_sBackupService;**

Registro* (II)



*traza en pseudocódigo

Solicitud (I)

```
protected void CallForMedic() {
      try {
```

Solicitamos al DF (páginas amarillas) un listado de aquellos agentes que proporcionen el servicio "**m_sMedicService**"

Si hay alguno en la partida...

Creamos un mensaje ACL tipo REQUEST
Añadimos todos los agentes del listado recibido del DF como receptores del mensaje

ConversationID = "CFM"
Contenido = "(x , y , z) (vida)"

Enviamos el mensaje

Sino, sabemos que no hay medicos

```
      } catch (FIPAException fe)
                  fe.printStackTrace();
}
```

JGOMAS Servicios

Solicitud (II)

```
protected void CallForMedic() {
      try {
                  DFAgentDescription dfd = new DFAgentDescription();
                  ServiceDescription sd = new ServiceDescription();
                  sd.setType(m_sMedicService);
                  dfd.addServices(sd);
                  DFAgentDescription[] result = DFService.search(this, dfd);

                  if ( result.length > 0 ) {
                              m_iMedicsCount = result.length;
                              // Fill the REQUEST message
                              ACLMessage msg = new ACLMessage(ACLMessage.REQUEST);
                              for ( int i = 0; i < result.length; i++ ) {
                                          DFAgentDescription dfdMedic = result[i];
                                          AID Medic = dfdMedic.getName();
                                          if ( ! Medic.equals(getName()) )
                                                      msg.addReceiver(dfdMedic.getName());
                                          else
                                                      m_iMedicsCount--;
                              }
                              msg.setProtocol(FIPANames.InteractionProtocol.FIPA_REQUEST);
                              msg.setConversationId("CFM");
                              msg.setContent(" ( " + m_Movement.m_Position.x + " , "
                                          + m_Movement.m_Position.y + " , "
                                          + m_Movement.m_Position.z + " ) ( "
                                          + m_iHealth + " ) ");
                              send(msg);
                              System.out.println(getLocalName()+ ": Need a Medic!");
                  }
                  else
                              m_iMedicsCount = 0;
      } catch (FIPAException fe)
                  fe.printStackTrace();
}
```

JGOMAS Servicios

Respuesta (I)

```
private void Launch_CFM_ResponderBehaviour() {
      // Behaviour to handle a Call For Medic request
      addBehaviour(new CyclicBehaviour() {
                  public void action() {
                              Filtrado de Mensajes: (Tipo == REQUEST) and (ConversationID == CFM)

                              Recibimos una petición y ...

                                          si decidimos aceptar la petición
                                                crear tarea TASK_GIVE_MEDICPACKS
                                                performativa = AGREE
                                          sino
                                                performativa = REFUSE

                                          Contestamos al agente que solicita la petición

                              Bloqueamos el Comportamiento del Agente
                  }
      });
}
```

Función de Decisión

Respuesta (II)

```
private void Launch_CFM_ResponderBehaviour() {
      // Behaviour to handle a Call For Medic request
      addBehaviour(new CyclicBehaviour() {
                  public void action() {
                        MessageTemplate template = MessageTemplate.and(
                        MessageTemplate.MatchPerformative(ACLMessage.REQUEST),
                        MessageTemplate.MatchConversationId("CFM"));

                        int iPerformative;
                        ACLMessage msgCFM = receive(template);
                        if ( msgCFM != null ) {

                                    AID owner = msgCFM.getSender();
                                    String sContent = msgCFM.getContent();

                                    if ( checkMedicAction(sContent) ) {
                                                AddTask(TASK_GIVE_MEDICPAKS, owner, sContent);
                                                iPerformative = ACLMessage.AGREE;
                                    }
                                    else {
                                                iPerformative = ACLMessage.REFUSE;
                                    }
                                    ACLMessage reply = msgCFM.createReply();
                                    reply.setContent(sContent);
                                    reply.setPerformative(iPerformative);
                                    send(reply);
                        }
                        else block();
                  }
      });
}

protected boolean checkMedicAction(String _sContent) { return ( true ); }
```


Índice

- Taxonomía
- Comunicaciones
- Servicios
- ► **Trabajo a realizar**

JGOMAS

Trabajo a Realizar

- Objetivo:
  - Implementar un nuevo rol (vigía) para un equipo defensor, en el que:
    - las posiciones de patrulla estén bastante alejadas del resto del equipo
    - envíe un mensaje *LOOK_OUT* al resto del equipo cuando vea un agente enemigo

1

JGOMAS

JGOMAS

JADE Game Oriented MultiAgent System

MAS: taxonomy, services & comunication

Sistemas Inteligentes
FI, 2006

Toni Barella
tbarella@dsic.upv.es

1