

Sesión 4 - JADE.

Introducción a los Sistemas Multiagente

En esta sesión vamos a utilizar la plataforma JADE para aprender a utilizar las funcionalidades, especialmente aquellas relacionadas con la comunicación entre agentes. Utiliza las páginas <http://jade.tilab.com/documentation/tutorials-guides/> y <http://programacionjade.wikispaces.com/> (a través de TheWayBack-Machine) como referencia para buscar información de la API.

Ejercicio 1

Crea un agente que:

1. Que salude al ser creado, indicando su nombre e identificador unívoco completo (métodos *getLocalName()* y *getAID().getName()*).
2. El agente deberá aceptar como parámetro un entero que se usará como contador (método *getArguments()*).
3. El agente implementará un comportamiento cíclico (clase *CyclicBehaviour*) en la que mostrará el valor del contador y lo decrementará en cada llamada.
4. El agente se suicidará cuando el contador llegue a cero (método *doDelete()*).

Ejercicio 2

La comunicación en JADE está instrumentada según la "Teoría de Actos del Habla". Esta teoría ve el lenguaje como acciones (peticiones, sugerencias, acuerdos,...) Una comunicación supone un cambio sobre el comportamiento del que recibe la comunicación. Los agentes utilizarán un conjunto de intencionalidades o performativas (asertivas, directivas,...) fijas que se están reguladas según FIPA-ACL (Agent Communication Language).

En JADE, el intercambio de mensajes se realiza mediante objetos de la clase *jade.lang.acl.ACLMessage* y permite:

- Solicitar la realización de acciones o servicios que no son capaces de realizar.
- Coordinarse y cooperar para resolver tareas complejas.
- Solicitar información a otros agentes.
- Negociar y competir por recursos.

ACL de FIPA

Define un conjunto de performativas (22).

Define los elementos que deberían aparecer:

- Tipo de acto comunicativo (performative)
- Participante en la comunicación (sender, receiver, reply-to)
- Contenido del mensaje (content)
- Descripción del contenido (language, encoding)
- Control de conversación (protocol, conversation-id, reply-with, in-reply-to, reply-by)

Tipos de Peticiones:

- **request, request-when, request-whenever:** Petición de realizar una acción, incondicionalmente, cuando cierta condición sea verdadera o cada vez que sea verdadera
- **propose:** Proposición de realizar una acción cuando ciertas condiciones se cumplan
- **call-for-proposal:** Petición de propuestas de realización de acciones bajo unas precondiciones
- **inform-if, inform-ref, query-if, query-ref:** Pregunta al que recibe el mensaje si cree que cierta condición es verdadera o qué elemento cumple esa condición
- **subscribe:** Petición de que un agente informe cuando cierta expresión/objeto cambie su valor

Tipos de Respuestas:

- **inform:** Se informa de que cierta condición es verdadera
- **accept-proposal, reject-proposal:** Se acepta o rechaza una propuesta
- **confirm, disconfirm:** Comunicamos sobre la veracidad de un hecho a un agente que tiene incertidumbre sobre él
- **agree:** Acuerdo para realizar una acción
- **refuse:** Rechazo a hacer una acción
- **cancel:** Cancelación de una acción ya acordada
- **failure:** La acción no se pudo realizar correctamente
- **not-understood:** No te he entendido

Tareas

1. Prueba los agentes *Ej2_Envia* y *Ej2_Recibe*. Explica cómo se realiza la comunicación y qué hace cada instrucción del código relacionada con el envío y recepción de mensajes.
2. Modifica el código para que los agentes se reenvíen entre sí el mensaje inicial un número determinado de veces (se pasa como parámetro).

Ejercicio 3

Prueba los agentes *Ej3_EmisorPasajeVuelo* y *Ej3_ReceptorPasajeVuelo*. ¿Qué tipo de mensaje intercambian los dos agentes? Explica qué pasos se siguen en la comunicación entre ambos agentes.