

PRÁCTICAS DE SISTEMAS DE AGENTES SOFTWARE

Módulo de Web Semántica, Servicios de Datos en Entornos Móviles

A continuación se plantean una serie de ejercicios incrementales, basados en el contexto de un casino de juegos, a construir utilizando agentes software implementados con la plataforma Jade, versión 3.4. Estas prácticas se realizarán en grupos de dos alumnos.

La resolución de los ejercicios puede basarse en los ejemplos que se distribuyen con la plataforma, y con el tutorial “*Jade Tutorial and Primer*” (disponible en <http://www.iro.umontreal.ca/~vaucher/Agents/Jade/JadePrimer.html>)

Se plantea:

Ejercicio 1. Implementar un agente crupier.

Supongamos que uno de los juegos del casino está basado en algo tan sencillo como adivinar cuál será el resultado cuando el crupier lance un dado. Este primer ejercicio consistirá en implementar un agente crupier que desarrolle este juego y gestione sus apuestas.

En esta versión el agente interactuará con el usuario humano, mostrando mensajes por pantalla y solicitándole que introduzca la información relativa a la apuesta que desea realizar.

El agente crupier comenzará con una cierta cantidad de euros, por ejemplo 20 euros o una cantidad que se le indique como parámetro al crear el agente crupier. Esta cantidad de dinero le permitirá al crupier aceptar apuestas del usuario hasta que se quede a 0 euros, en cuyo caso el agente finalizará su ejecución.

Una vez inicie su funcionamiento, el crupier anunciará por pantalla que va a lanzar un dado, y preguntará al usuario qué cantidad de euros desea apostar (número entero). El agente leerá la cantidad apostada por el usuario, y aceptará la apuesta si esta cantidad es inferior o igual a la cantidad de dinero de la que dispone el crupier. Si no acepta la apuesta, informará de ello por pantalla, y volverá a preguntar al usuario que cantidad de euros desea apostar. Si el crupier acepta la apuesta, informará de ello por pantalla, le preguntará al usuario qué número apuesta que saldrá al lanzar el dado, y lo leerá. Transcurridos 3 segundos (tiempo que tarda en tomar nota de la apuesta, hacer malabarismos, etc.), lanzará el dado e informará por pantalla del número obtenido al tirar el dado (valor entre 1 y 6), del resultado de la apuesta del usuario, y del saldo que le queda al crupier. El resultado de la apuesta se determinará de la siguiente forma:

- Si el usuario acertó el número resultado, gana la cantidad apostada y el crupier se descuenta esa cantidad del dinero que le quedaba.
- Si el usuario no acertó el número, pero al menos el obtenido coincide en valor par/impar con el número por el cual apostaba, entonces quedan en tablas y ninguno pierde su dinero.
- Si el usuario ni acertó el número, ni si iba a ser par o impar, entonces pierde su dinero y el agente crupier incrementa su saldo con dicha cantidad.

Ejercicio 2. Implementar un agente jugador y su interacción con el agente crupier.

Implementar un agente jugador, que desempeñe el papel que en el ejercicio anterior tenía el usuario humano. El agente jugador se iniciará con una cantidad de dinero X y apostará un máximo de N veces, ambos serán parámetros pasados en la creación del agente jugador. Se modificará el agente crupier para que ambos agentes puedan desarrollar el juego y las apuestas, de forma similar a la descrita anteriormente.

El agente jugador apostará siempre una cantidad de dinero menor o igual que la que quede, y abandonará (terminará) una vez que se quede con saldo cero o haya jugado las N partidas (apuestas). Se podrá implementar como se desee la negociación-acuerdo entre el agente jugador y el crupier para decidir la cantidad a apostar. Ninguno de ellos debe apostar más dinero del que le queda, y deben seguir jugando hasta que uno se arruine o se retire. Durante el desarrollo del juego y las apuestas, ambos agentes deberán informar al usuario humano por pantalla, indicando que acciones toman, cantidades y números por los que se apuesta, etc. Justo antes de finalizar, el agente jugador informará del saldo con el que acaba.

Para resolver este ejercicio, la comunicación entre los agentes podrá implementarse de tres formas diferentes (ver punto 7 del *Jade Tutorial and Primer*, y el manual de *Application-defined Content Languages and Ontologies* de Jade, junto con los ejemplos que los acompañan como la tienda de música o el Banco):

1. mediante intercambio de mensajes cuyo contenido sea siempre un string y contenga la información relevante en cada caso (se puede partir del ejemplo de “Book Trading” desarrollado en el manual *Jade Programming for Beginners*),
2. mediante el intercambio de objetos java serializables, como contenido de los mensajes,
3. definiendo una ontología para el vocabulario y las acciones, a utilizar en el intercambio de mensajes entre los agentes.

Ejercicio 3. Reimplementar el ejercicio 2 de forma que un agente crupier pueda interactuar con múltiples jugadores a la vez.

Para resolver este ejercicio se sugiere consultar la el *Jade Tutorial and Primer*, punto 6 cómo implementar conversaciones en paralelo, y el manual de *Jade Programmer’s Guide*.

Ejercicio 4. Reimplementar el ejercicio 3, o el 2, de forma que el agente jugador sea móvil.

Para este ejercicio se asumirá que cada mesa del casino está representada por un *container* en la plataforma Jade, con un único agente crupier en cada mesa o *container*. Los agentes jugadores se moverán entre las mesas (*containers*) de forma que hagan cada apuesta en un *container* diferente cada vez, hasta que acaben sus apuestas o recorran todas las mesas o *containers* (en cuyo caso volverán a empezar su recorrido por los *containers*).

Entrega y evaluación:

Como mínimo se entregarán las soluciones a los ejercicios 1 y 2. Quién lo desee podrá entregar varias implementaciones para el ejercicio 2, según las diferentes alternativas para implementar la comunicación entre los agentes. El desarrollo de interfaces gráficas para los agentes es totalmente opcional.

Para cada ejercicio, se entregará un archivo *.zip* con el código fuente desarrollado, y un fichero con documentación, donde se explique brevemente la implementación realizada, la forma en que interactúan los agentes, y las instrucciones para probar la implementación del ejercicio. El código fuente deberá estar comentado, explicado convenientemente, y con la información de su autor o autores.

PRIMER PLAZO DE ENTREGA: 10 DE JULIO DE 2006