SISTEMAS MULTIAGENTES

ENTREGA 2:
Comparación de precios de Smartphones utilizando agentes

Realizado por Fernando Vallejo Banegas y Álvaro Guerrero del Pozo

1. Objetivos Generales

Nuestro trabajo consistirá en el desarrollo de un sistema basado en agentes capaz de recuperar información relativa a Smartphones de dos páginas web, las de PC Componentes y Amazon, para su posterior procesamiento y comparación de precios y puntuación para ayudar al usuario en la decisión de en qué tienda comprar su producto deseado de forma en la que pueda ahorrar dinero. El usuario introducirá información relativa a las especificaciones del móvil, y ambos agentes intentarán encontrar un modelo de móvil que cumpla dichas especificaciones, y se informará al usuario de en cuál de ambas tiendas es más barato, y las valoraciones del producto.

2. Manual de Usuario

Para poder compilar nuestros agentes de JADE y ejecutarlos debemos seguir los siguientes pasos:

1. Primero compilamos y añadimos a jade/classes con el siguiente comando:
   javac -classpath jade/lib/jade.jar:DIRECTORIO/jsoup-1.11.3.jar -d jade/classes/ repositorios/Multiagentes/src/dominio/Agente*.java
   Cambiando 'DIRECTORIO' por el path donde se encuentre el .jar de jsoup que se encuentra en nuestro proyecto de github en: /lib
2. De esta forma ya estamos listos para ejecutar con:
   java jade.Boot -agents ag1:dominio.AgenteAmazon o
   java jade.Boot -agents ag1:dominio.AgentePC
3. No existe comunicación aún, así que ambos agentes deben ser probados por separado.

3. Definición de agentes y comunicación

En nuestro sistema, interactuarán tres agentes, a fin de cumplir los objetivos antes propuestos, siendo dos de ellos los recuperadores de información de Internet y el último el encargado de comunicarse con ellos e interactuar con el usuario.

AgentePC: Se trata del agente encargado de recuperar información de PC Componentes. Actualmente cuenta con los comportamientos:

- GetURL: Comportamiento de tipo AchieveREResponder encargado de realizar la búsqueda de los parámetros introducidos por el usuario y obtener la URL del primer producto que los cumpla.
- GetName: Comportamiento de tipo OneShot encargado de, dado la URL de un producto, obtener el nombre de dicho producto.
- GetPrice:Comportamiento de tipo OneShot encargado de, dado la URL de un producto, obtener el precio de dicho producto.
- GetRating:Comportamiento de tipo OneShot encargado de, dado la URL de un producto, obtener la valoración de dicho producto.
- SendInfo: Comportamiento de tipo AchieveREResponder encargado de enviar la información obtenida (Precio y Valoración) al agente intermediario.

AgenteAmazon: Se trata de un agente análogo al anterior encargado de recuperar la información de Amazon. Cuenta con los comportamientos:

- GetURL: Comportamiento de tipo OneShot encargado de realizar la búsqueda de un producto, dado su nombre, y obtener la URL del primer producto que los cumpla.
- GetPrice: Comportamiento de tipo OneShot encargado de, dado la URL de un producto, obtener el precio de dicho producto.
- GetRating:Comportamiento de tipo OneShot encargado de, dado la URL de un producto, obtener la valoración de dicho producto.
- SendInfo: Comportamiento de tipo AchieveREResponder encargado de enviar la información obtenida (Precio y Valoración) al agente intermediario.

El motivo por el que este agente no obtiene el nombre del producto (de hecho, lo recibe), se detallará posteriormente, al tratar los mensajes enviados entre agentes.

Intermediario: Se trata del agente que recibirá las especificaciones del producto a buscar. También es el agente encargado de enviar información a ambos agentes recuperadores, y de recibir y mostrar la información obtenida. Contará con los siguientes comportamientos:

- SendSpecs: Comportamiento AchieveREInitiator encargado de enviar las especificaciones del Smartphone a AgentePC. Si todo transcurre con normalidad, recibirá el nombre del producto como respuesta.
- GetInfoPC: Comportamiento AchieveREInitiator encargado de interactuar con AgentePC, con el fin de recibir la información del producto (Precio y Valoración).
- GetInfoAmazon: Comportamiento AchieveREInitiator encargado de interactuar con AgenteAmazon, con el fin de recibir la información del producto (Precio y Valoración).
- MostrarUsuario: Comportamiento de tipo OneShot encargado de presentar la información obtenida por ambos agentes recuperadores de información al usuario.

4. Uso de Regex

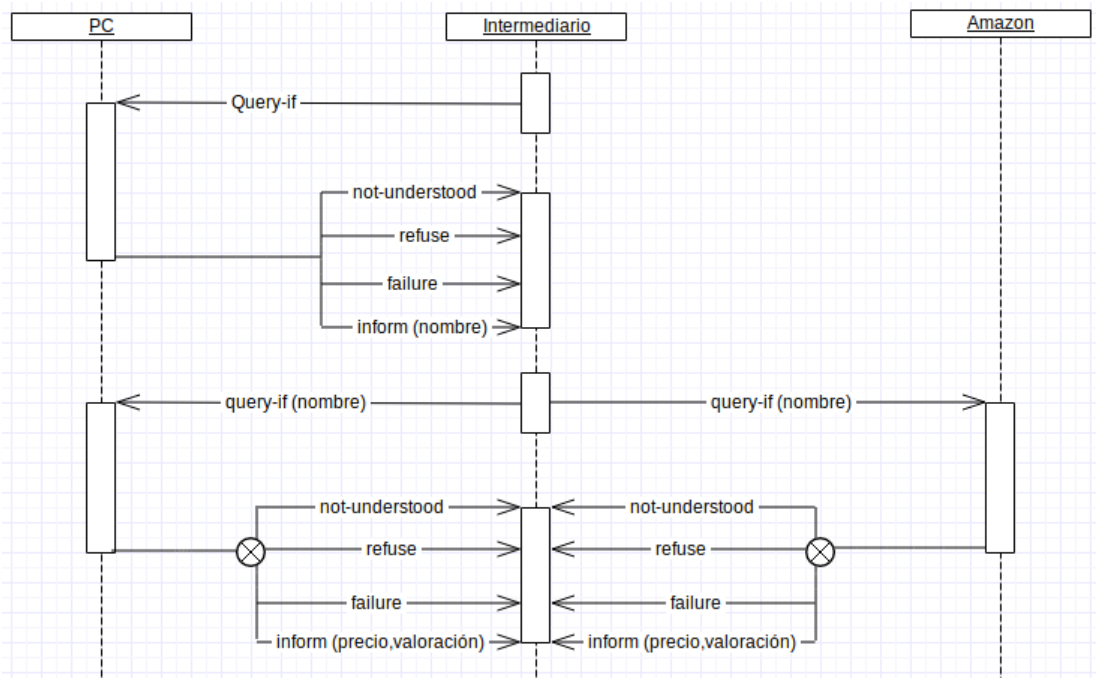
En el código actual estamos usando Regex, para realizar ciertas comprobaciones en los formatos, tanto del precio como de la valoración. Ya que queremos que todos, los de ambas páginas web, se muestren al usuario con el mismo formato de salida. Para ello, se deben hacer ciertas comprobaciones y modificaciones, como eliminar el símbolo “€” para convertir el precio en un entero. También nos servirá para detectar errores (ej: error de comunicación devuelve **null**)

5. Diagrama de comunicación

A continuación se muestra el diagrama de comunicación que implementarán nuestros agentes, el protocolo a implementar es el FIPA Query Interaction Protocol, en este protocolo al agente receptor se le pide que realice alguna clase de acción de informar. Dicha clase de acción es un Query. Así pues, hemos pensado que es el protocolo que más puede encajar con nuestros sistema, el agente Intermediario será el encargado de realizar las Query-if (existe otro tipo de Query llamadas Query-ref, que no usaremos en este caso) al resto de agentes.

El funcionamiento de nuestro sistema de comunicación es simple, el Intermediario iniciará la comunicación con el Agente de PC Componentes para pedir el primer nombre del primer elemento que encaje con la descripción del usuario (lo realizamos de esta forma porque PC Componentes nos provee de unos nombres más sencillos de buscar posteriormente en Amazon).

Posteriormente cuando el Agente de PC responde al Intermediario con el nombre, este último iniciará la comunicación con ambos agentes para que estos busquen dicho producto y recuperen los datos necesarios (Precio y Valoración), que serán devueltos al Intermediario que nos dará la mejor oferta e información relacionada con la valoración del producto.



6. Código

El código de esta práctica se encuentra en el repositorio de github https://github.com/AlvaroGdP/Multiagentes, que también incluye una rama para esta entrega (Entrega2).
Esta entrega incluye dos agentes, el de PC Componentes y el de Amazon, que actualmente funcionan de forma independiente y únicamente tienen la función de mostrar cómo serán dichos agentes, excepcionando comunicación, en la entrega final.