UNIVERSIDAD DE CASTILLA-LA MANCHA

ESCUELA SUPERIOR DE INFORMÁTICA

DEPARTAMENTO DE TECNOLOGÍAS Y SISTEMAS DE INFORMACIÓN

SISTEMAS MULTIAGENTES

GRADO EN INGENIERÍA INFORMÁTICA

TECNOLOGÍA ESPECÍFICA DE COMPUTACIÓN

COMPARADOR DE CONFIGURACIONES DE ORDENADOR POR COMPONENTES

Autor: Álvaro Ángel-Moreno Pinilla y Carlos Córdoba Ruiz

Director: Luis Rodriguez Benitez

14 Enero, 2018

Índice

1. DEFINICIÓN DEL PROBLEMA

La idea de nuestro sistema surge en las compras on-line, donde varias páginas independientes ofrecen un mismo producto con determinados precios y ofertas. En un principio, el comparador estaría orientado a ser multiproducto, en gran diversidad de páginas web. En nuestro caso, para acotar más el problema, nos centramos en los componentes de ordenador y páginas web destinadas para ello. Bajo esas condiciones, la idea principal de nuestro sistema consiste en tomar el modelo de un producto como input, para así buscarlo en las páginas web de venta de componentes que definamos. Acto seguido, comparar los precios y la disponibilidad de cada una de las ofertas para así obtener una decisión sobre qué sitio web elegir para su compra.

Las herramientas que van a ser utilizada son:

- JADE: Entorno software que simplifica la comunicación entre los agentes del sistema (middleware).
- Jsoup: Librería de Java que permite trabajar con el código HTML de las páginas web a consultar.
- regex: Librería de Java que actúa como reconocedor de patrones, utilizado para obtener las características de cualquier producto en los sitios web.

2. ANTECEDENTES

Actualmente existen numerosos servicios web que actúan mediante sistemas multiagente a modo de comparador. Entre ellos, se encuentran páginas como www.rastreator.com, un comparador de multiproducto de telefonía, viajes, finanzas y seguros. También cabe destacar www.acierto.com, comparador de seguros con los precios y coberturas de más de 30 compañías distintas.

Sin embargo, a pesar de la amplitud de productos que ambos ejemplos presentan, ninguno de ellos ofrece sus servicios en productos de ésta índole. Es por ello que podemos definir nuestra propuesta de sistema multiagente como una idea viable en la que trabajar.

3. SITIOS WEB A CONSULTAR

Existe una amplia diversidad de tiendas on-line que pueden vender un producto determinado, pero nuestro objetivo, acotado a componentes de ordenador, nos indica tomar páginas web más generalizadas dentro de la propia especificación electrónica. En dichos sitios web, la situación idónea es que se presenten una multitud de productos similares no sólo en características sino en

el propio modelo del producto. Así, la meta de comparador sería la más veraz y natural posible. Los dominios web en los que hemos elegido establecer nuestras comparaciones son:

- www.pccomponentes.com
- www.aussar.es
- www.newhomepc.net

4. AGENTES Y COMPORTAMIENTOS

El sistema final está formado por dos agentes independientes, entre los que existe una relación de comunicación para proporcionar el resultado de cada una de las búsquedas.

En el caso del primer agente, llamado 'Crawler', tiene lugar la definición del modelo a investigar y el lanzamiento de cada una de las búsquedas. Los tres comportamientos que lo forman, de tipo OneShotBehaviour, son creados como hebras dentro de una misma factoría de comportamientos de hebras. Todos ellos son declarados en el método setup() del agente. Cada uno realiza la consulta en su página correspondiente, toma la mejor oferta del producto y lo manda al agente 'AgenteFinal', para finalizar allí el proceso de comparación. Acto seguido, el comportamiento termina. Una vez finalizados todos los comportamientos, el agente termina, interrumpiendo cada una de las hebras generadas y acabando así su ejecución.

El segundo agente, 'AgenteFinal', recibe la mejor oferta de cada una de las páginas web, enviadas mediante un mensaje por cada uno de los comportamientos de 'Crawler', y las compara para así obtener la mejor. Los comportamientos que llevan a cabo esta funcionalidad son:

- fbehaviour: Define una plantilla de mensajes con el objetivo de recibir los productos desde el agente 'Crawler'. A cada uno que recibe, perteneciente a los diferentes comportamientos de búsqueda, lo añade a una lista de productos con la que tratará el siguiente comportamiento. Una vez recibida la cantidad de mensajes concretada, uno por cada comportamiento de búsqueda en 'Crawler', se añade el siguiente comportamiento 'anunciador' y termina el presente.
- anunciador: Recorre la lista donde se presentan las ofertas de cada una de las búsquedas. Compara sus precios (asumiendo el hecho de que los productos seleccionados tienen la misma disponibilidad) y muestra el producto seleccionado como el mejor, la página a la que pertenece y su precio.

5. FLUJO DE MENSAJES

El intercambio de mensajes producido en nuestro sistema consiste en la comunicación entre el agente que toma la información de Internet 'Crawler', los propios sitios web y el 'AgenteFinal', que recibe los mejores productos de cada página. La secuencia de comunicación transcurre así:

1. Dentro de 'Crawler', cada uno de los comportamientos de búsqueda consultan el producto, dado como argumento, en la página web que le corresponde a cada uno.

2. Los sitios web consultados devuelven la información del elemento más barato y disponible cuyo modelo corresponda con el consultado.

3. Cada uno de los tres comportamientos de búsqueda manda un mensaje a 'AgenteFinal' con el producto elegido. El envío de los mensajes es independiente, utilizando la performativa ACLMessage.INFORM.

6. INSTALACIÓN Y EJECUCIÓN

Como requisitos para poder ejecutar nuestro programa necesitaremos tener instaladas en nuestro computador las librerías de Jade y Jsoup. En el caso de instalar estas librerias en alguna distribución linux nos bastaría exportando en el CLASSPATH la ruta de los archivos .jar.

A la hora de la ejecución del programa, el orden en el que creamos los agentes es muy importante, ya que primero debemos inicializar el AgenteFinal y después ejecutar el agente Crawler.

Cuando ejecutemos nuestro agente Crawler, le debemos pasar como argumento el producto que deseamos comparar en cada página, es recomendable no incluir la marca del producto, simplemente el modelo, ya que el programa podría mostrarnos un producto que no desearamos de la misma marca.