



**UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA**

**SISTEMAS MULTIAGENTES
GRADO EN INGENIERÍA INFORMÁTICA**

Sistema de recomendación de películas

Autores: Diego Pedregal Hidalgo y Alberto Velasco Mata

Director: Luis Rodríguez Benítez

Noviembre, 2019

Índice

1. Introducción	2
2. Objetivos	2
3. Tecnologías usadas	3
4. Arquitectura del Sistema	4
5. Comunicación	5
5.1. Protocolos	5
5.2. Mensajes	5
6. Arquitectura de los Agentes	7
6.1. Agentes <i>Collector</i>	7
6.2. Agente <i>Integrator</i>	7
6.3. Agente <i>Interface</i>	8

1. Introducción

Este trabajo consiste en el desarrollo de Agentes de Información haciendo uso del marco de desarrollo JADE. El código resultado de este trabajo se encuentra en el siguiente repositorio de GitHub:

https://github.com/thmasker/SMA_VelascoPedregal

2. Objetivos

El principal objetivo de este trabajo es realizar un sistema multiagente que recomiende películas al usuario. A su vez, se busca también cumplir varios subobjetivos de distinta índole:

- **Recolección de información en la web.** La información sobre las películas debe ser obtenida a partir de servicios disponibles en Internet. Con el fin de conseguir la información más variada posible, se ha decidido utilizar como proveedores las páginas *IMDB*, *TheMovieDB* y *FilmAffinity*, todas ellas conocidas por sus calificaciones asignadas a los distintos títulos filmográficos. Esta recopilación de valoraciones será llevada a cabo por algunos de los agentes de nuestro sistema.
- **Uso de expresiones regulares.** La búsqueda y/o recopilación de información deberá incorporar el uso de expresiones regulares en alguna parte del proceso.
- **Interacción entre agentes.** Los distintos agentes creados interactuarán entre ellos para llevar a cabo el objetivo principal del trabajo, esto es, recomendar a los distintos usuarios las mejores películas de un género determinado en función de la información obtenida de diversas fuentes.

3. Tecnologías usadas

Para la obtención de los títulos y sus respectivas calificaciones a partir de las fuentes mencionadas previamente se ha hecho uso de las siguientes tecnologías:

- **JSoup.** Librería de Java que permite la extracción y manipulación de información a partir de código HTML. Se ha empleado tanto para la obtención del código HTML de las fuentes como para el procesamiento y la extracción de los nombres y valoraciones de cada película.
- **Expresiones Regulares.** Han sido utilizadas para la comprobación del formato de las valoraciones, de manera que todas estén formadas por un número decimal que utilice el punto (.) o la coma (,) como separador.
- **Random.org API.** Servicio que permite la generación de números aleatorios reales basados en ruido atmosférico. Se ha utilizado para extraer un subconjunto de películas entre todas las que han sido recopiladas.

4. Arquitectura del Sistema

En primer lugar, es importante señalar que la implementación de estos agentes se va a llevar a cabo usando el paradigma ofrecido por JADE [1]. La disposición de los agentes ha sido definida de la siguiente manera:

- a. **Agentes recolectores de información (Collector).** En nuestro caso, tenemos tres agentes, que obtendrán la información deseada de IMDb [2], The Movie Database [3] y FilmAffinity [4], respectivamente. La cantidad de películas obtenidas dependerá del usuario, al que se le preguntará dicha cantidad.
- b. **Agente integrador de la información (Integrator).** Este agente es el encargado de recoger la información obtenida por los agentes recolectores; y juntarla, eliminando elementos redundantes y calculando la media de las calificaciones para las películas que se encuentran repetidas.
- c. **Agente mostrador (Interface).** Encargado de la interacción con el usuario, será el iniciador de toda la secuencia de obtención de información, y mostrará al usuario la información procesada. Además, pedirá al usuario el género de películas que este quiere obtener, así como el número de las mismas.

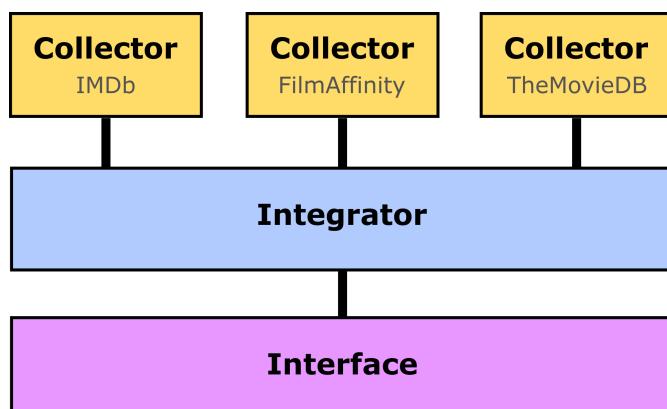


Figura 1: Estructura de agentes.

Además, se ha creado una clase adicional, *Film*, que define cada película como un objeto con un título y una valoración.

Una vez han sido definidos todos los agentes necesarios del sistema, es importante diseñar su arquitectura interna, algo que se tratará después de estudiar la interacción que tiene que haber entre cada uno de ellos.

5. Comunicación

El diseño del flujo de comunicación entre los agentes es una de las claves para el buen funcionamiento del sistema. Así pues, la comunicación se llevará a cabo por medio de los protocolos definidos por el estándar FIPA [5].

5.1. Protocolos

Este sistema, al ser poco complejo en cuanto a la arquitectura de agentes, facilita bastante la tarea de comunicación. Como se ha explicado anteriormente, hay dos flujos claros de comunicación:

a. *Interface* ⇔ *Integrator*. Esta comunicación es la iniciadora de toda la acción del sistema.

Para ello, es necesario hacer uso de un protocolo *Request* de FIPA, en el que el *Interface* solicitará al *Integrator* la obtención del número deseado por el usuario de películas del género seleccionado. Al recibir esta petición, el *Integrator* esperará a que los *Collector* le pasen los datos capturados para realizar las operaciones pertinentes sobre ellos (ver b.), y enviárselos al *Interface*, para que este pueda mostrarlos al usuario.

b. *Integrator* ⇔ *Collector*. Esta segunda comunicación hará uso igualmente del protocolo FIPA *Request*. La iniciará el *Integrator* al recibir la petición del *Interface*, e irá dirigida a cada uno de los *Collector*. Así pues, será necesario el uso de tres FIPA *Request*, uno para comunicarse con cada *Collector*. Cada uno de ellos responderá con los datos obtenidos, finalizando así el protocolo.

5.2. Mensajes

Los mensajes con los que se comunicarán los agentes están determinados ya por los protocolos. Lo único que varía es el contenido que se va a enviar de un agente a otro, dependiendo del flujo que hemos definido antes:

a. *Interface* ⇔ *Integrator*. Para este flujo, el único mensaje que envía contenido de relevancia es el *Inform*. Una vez el *Integrator* tiene la información que debe devolver (una lista con las películas que el *Interface* enseñará al usuario), la introducirá como contenido de dicho mensaje, de manera serializada.

b. *Integrator* \Leftrightarrow *Collector*. De la misma manera, cada *Collector* añadirá como contenido del mensaje *Inform* la lista serializada de películas que ha obtenido, y que deberá recibir el *Integrator*.

El resto de mensajes que requiere el protocolo FIPA *Request* serán mensajes simples, sin contenido de relevancia para el funcionamiento del sistema.

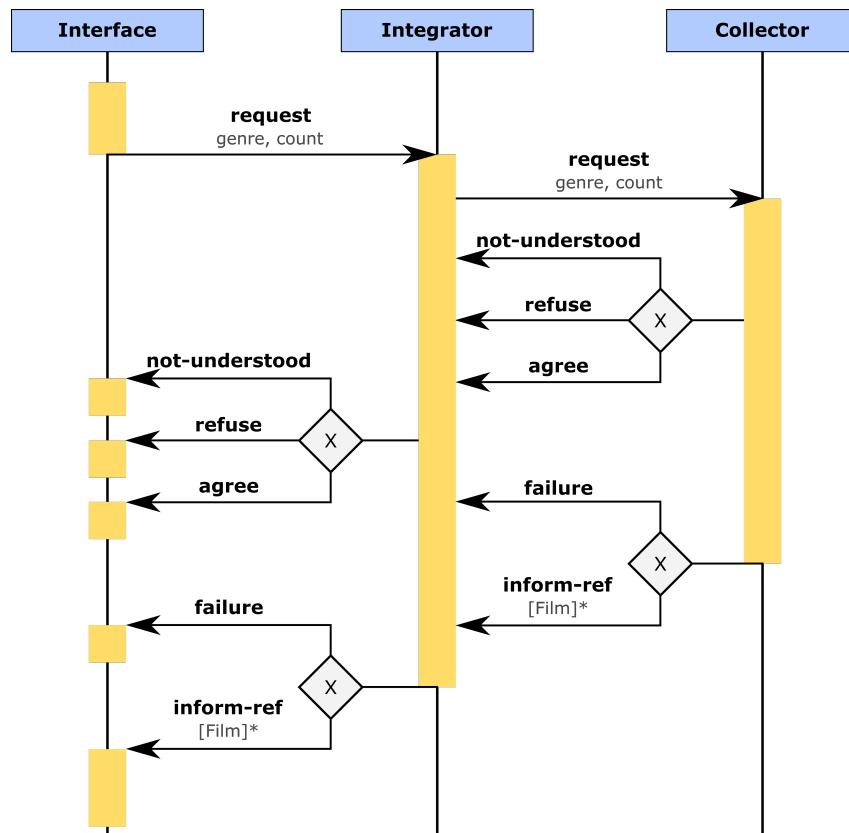


Figura 2: Diagrama de Protocolo de la comunicación entre los agentes.

6. Arquitectura de los Agentes

6.1. Agentes *Collector*

En este caso, hay tres agentes *Collector* que, al realizar la misma función, van a compartir la misma estructura interna. La única diferencia será la referida al parseo de los datos obtenidos, adaptándose a las distintas fuentes de donde se obtienen estos datos. De esta manera, se definen dos comportamientos para estos agentes:

- a. **AchieveREResponder.** Este comportamiento será el encargado de la comunicación con el agente *Integrator*. Al recibir una petición de este, activará el comportamiento siguiente.
- b. **OneShotBehaviour.** Este comportamiento es el que obtiene realmente la información sobre las películas. Creará un objeto de la clase *FilmScrapper*, obteniendo las películas de la base de datos que le corresponda.

6.2. Agente *Integrator*

Este agente actúa de intermediario entre el presentador y el colector de la información. Por tanto, tendrá tres comportamientos:

- a. **AchieveREResponder.** Este comportamiento se activará al recibir una petición del *Interface*, y estará en espera hasta haber procesado la información proveniente de los agentes *Collector*, tarea tras la cual enviará de vuelta dichos datos procesados.
- b. **AchieveREInitiator.** Al recibir la petición del *Interface*, este agente debe iniciar un protocolo FIPA *Request* para comunicarse con los *Collector*, tarea de la que se encargará este comportamiento. Una vez los *Collectores* hayan recuperado la información solicitada, la enviarán de vuelta al *Integrator*, encargándose este comportamiento también de su recepción.
- c. **OneShotBehaviour.** Cuando el *AchieveREInitiator* termina, los datos recuperados por los colectores han llegado al *Integrator*. Así, da comienzo la fase de tratamiento de los mismos. Para ello, este comportamiento se activa, y es el encargado de eliminar las películas redundantes, dando lugar a un listado único de películas, que se encargará el *AchieveREResponder* de enviar de vuelta al *Interface*.

6.3. Agente *Interface*

Por último, hay que diseñar el funcionamiento de este agente. Su función es interactuar con el usuario, hacer la petición de los datos al *Integrator*, y mostrar por pantalla la información obtenida al usuario.

Para esta tarea, es necesario un comportamiento:

- a. **AchieveREInitiator.** Es el desencadenante de toda la acción. Realiza la petición al *Integrator*, y espera su respuesta, para posteriormente mostrarla al usuario.

Referencias

- [1] JADE API v4.5.0. <https://jade.tilab.com/doc/api/index.html>.
- [2] IMDb. <https://www.imdb.com/>.
- [3] The Movie Database. <https://www.themoviedb.org/>.
- [4] FilmAffinity. <https://www.filmaffinity.com/es/main.html>.
- [5] The Foundation for Intelligent Physical Agents. <http://www.fipa.org/>.