Práctica 5
Representación de conocimiento mediante reglas. Recomendador de apps
IA – Curso 2016-2017

El propósito de esta práctica es la construcción del recomendador de apps propuesto en la práctica 4. Con el desarrollo de esta práctica se pretenden alcanzar los siguientes objetivos:

- Modelar el dominio de la recomendación de apps para dispositivos móviles, adquiriendo el conocimiento necesario, estructurándolo y analizando el proceso de recomendación.
- Practicar con la representación de conocimiento basada en reglas, analizando su adecuación al conocimiento a representar.
- Profundizar en el conocimiento de Jess y su aplicación al desarrollo de sistemas basados en reglas.
- Añadir una interfaz de usuario.

La entrega de la práctica 5 se realizará a través del Campus Virtual en un fichero zip que contendrá el código Jess desarrollado (sólo el archivo .clp), el código Java desarrollado (proyecto completo con librerías), y la memoria de la práctica (que incluirá las partes Jess y Java). Tanto en el comienzo del programa Jess como en la portada de la memoria figurará un comentario con el número de grupo y los nombres completos de sus integrantes. Además el nombre del zip será (NumGrupoApellidoAlumno1ApellidoAlumno2.zip).

Recomendador en Jess

En esta primera parte se implementará en Jess el recomendador de apps analizado en la práctica anterior. Se trata de un prototipo básico que no necesita operaciones de entrada/salida de datos. Es aconsejable que el prototipo tenga varios módulos correspondientes a distintas etapas de la recomendación.

Una posibilidad podría ser que el prototipo tuviera 3 módulos. En un primer módulo se analizarían las características y gustos del usuario para determinar de qué tipo de usuario se trata y qué tipo de apps necesita. Se podría clasificar las apps con respecto a distintos criterios en otro módulo. En un tercer módulo se generarían las recomendaciones concretas para el usuario que sean adecuadas a sus necesidades.

En este primer prototipo, los datos correspondientes a las respuestas del usuario sobre sus gustos se establecerán directamente en un `deffacts`, sin necesidad de realizar operaciones de lectura. Se asumirá que el usuario no sabe qué apps elegir.

Los resultados obtenidos por cada uno de los módulos se asertarán en la memoria de trabajo, pudiendo visualizarse mediante el uso de (facts *).

Se recomienda empezar por un prototipo sin módulos y sin multislots. Cuando funcione se podrá incrementar su complejidad.

Metodología a utilizar

Se utilizará una metodología simple de desarrollo incremental de sistemas basados en conocimiento mediante prototipado rápido, que consta de las siguientes etapas:

1. *Identificación de requisitos*.
2. *Diseño del prototipo*. En esta etapa se realizará una conceptualización y formalización del conocimiento a utilizar, descomponiendo el problema en subproblemas.
3. *Implementación del prototipo en Jess*.
4. *Prueba*. Validación del sistema respecto a los requisitos iniciales. Obtención de nuevos requisitos para el siguiente prototipo, volviendo de nuevo a la etapa 1.

Una primera versión de las 2 primeras etapas se debería haber realizado, al menos parcialmente, en la práctica 4.

Recomendador con entrada/salida en Java

En este prototipo se añadirán operaciones de entrada/salida en Java que permitan obtener los datos necesarios del usuario y generar los resultados adecuadamente formateados. Para ello se empotrará el prototipo anterior en un programa Java, según lo descrito en JessConJava.pdf y más específicamente en el ejemplo ESJessJava.

La entrada/salida debe ser a prueba de fallos, comprobando que los datos introducidos por el usuario son correctos, tanto en tipo como en rango, volviendo a solicitarlos, en caso necesario, hasta que sean correctos.

La aplicación funcionará en modo continuo, repitiendo la recomendación de apps para distintos usuarios hasta que se decida salir de la aplicación.

Requisitos mínimos del segundo prototipo

Se intentará que el prototipo sea fácilmente mantenible ante los cambios más habituales (por ejemplo, nuevas apps, cambio de alguna característica en alguna de ellas, etc.) y ante la aparición de nuevos perfiles de usuarios.

El prototipo imprimirá las recomendaciones correspondientes a cada usuario, seleccionando las mejores según un criterio fijo o elegido por el usuario (por ejemplo, por precio, por grado de adecuación a las necesidades del usuario, etc.). Las recomendaciones aparecerán ordenadas, por orden decreciente, de acuerdo a dicho criterio.

Si el número de recomendaciones es alto, se mostrarán sólo las N mejores. Si no hay recomendaciones para un usuario, o son muy escasas, se podrá dar la posibilidad de relajar algunas de las restricciones para aumentar su número.

No se utilizarán *ifs* en la parte Jess, ni otras construcciones ajenas a los sistemas de producción, salvo de forma muy puntual en alguna *deffunction*. Tampoco se utilizarán las prioridades ni se eliminarán reglas para simular un programa secuencial. Se podrán utilizar prioridades, módulos, hechos centinela o valores por defecto para garantizar que se completa una etapa del programa antes de iniciar la siguiente.

Sólo se utilizará *test* cuando se requiera hacer comparaciones de menor que, mayor que, negaciones o funciones *or*. Las comparaciones de igualdad se realizarán mediante encaje de patrones con los hechos disponibles en la memoria de trabajo.

Memoria

La memoria correspondiente al primer prototipo debe incluir:

- Descripción del funcionamiento del recomendador de apps, conocimiento utilizado y estructura modular.
    - En la memoria de esta práctica se describirán y justificarán los cambios realizados con respecto a la práctica 4.
    - Se describirá la estructura de las partes Jess y Java.
- Ejecución del recomendador para 2 usuarios diferentes. Cada ejecución incluirá los hechos iniciales que se han utilizado y los resultados que se han obtenido.
- URL's u otras fuentes de conocimiento utilizadas para la construcción del recomendador.