

Práctica 2: Sistema de Reglas de Producción

IAIC – Curso 2008-2009

Asesor Laboral colaborativo con diferentes enfoques

Queremos hacer un sistema Asesor Laboral que, a partir de unas características introducidas por el usuario en una GUI, el sistema le ofrezca una serie de consejos para la búsqueda de trabajo, que se ajusten mejor a su situación actual. El Asesor Laboral tiene varios enfoques (técnico, jurídico, afectivo-coaching, y el que se te ocurra), el usuario puede seleccionar uno o varios para recibir consejos. Los usuarios serán personas entre 21 y 50 años, que están estudiando una carrera o acaban de terminar.

Requisitos:

- La práctica debe realizarse por **los mismos grupos que la primera práctica**.
- Organizar el problema usando varias bases o bloques de reglas, donde cada uno se refiere a un aspecto del problema.
- Debe estar organizado para que se puedan añadir, quitar y modificar reglas fácilmente.
- Enfoques de Asesor Laboral:
 - o Técnico: qué información buscar, donde encontrarla, cómo preparar un curriculum,
 - o Jurídico: qué pasos dar legalmente (papeleos), cómo estar informado de convocatorias públicas, cómo buscar y leer los boletines BOE, BOCAM.
 - o Afectivo /Coaching: Cómo animarse en los fracasos, qué metas intermedias fijarse, cómo medir el avance, cómo hacer seguimiento de las actuaciones, estrategias para entrevistas, qué imagen crear según el tipo de puesto, cómo orientarse en la búsqueda, etc...
 - o Otro enfoque que propongas (debes consultarlo antes con el profesor y que te lo acepte).
- La fase de Adquisición del Conocimiento es muy importante y consume muchos recursos. Aunque no se pide que seas tú un experto en el tema, sí que es importante que consigas conocimiento realista de alguna fuente. Para ello vamos a hacer un proceso colaborativo dividido en dos fases con *dos fechas de entrega*:
 1. Por Grupo: Crear bases de reglas organizadas en módulos según aspectos incluidos en uno de los enfoques del sistema (ver la fecha de entrega, el contenido y el modo en la sección de *entrega*).
 2. Utilizando las Bases de reglas de los demás grupos crear un Asesor Laboral organizado al menos por los tres enfoques citados. Si en uno de ellos no hay nadie que haya hecho reglas, debes crear algunas básicas. En el proceso tienes que homogeneizar los términos utilizados para que en todas las bases tengan el mismo sentido y estén representados con el mismo término.

Se recomienda:

1. Antes de escribir reglas establecer claramente qué preguntas responderá el sistema y que datos contiene la respuesta. Mantener presentes los *requisitos* indicados.
2. Probar, usando JESS, pequeños prototipos de la Base de Reglas con pocas reglas, para comprender el software y su comunicación con JAVA.
3. Determinar la GUI adecuada (sin ser demasiado sofisticada), que permita manejar el sistema adecuadamente.
4. A la vez que se crean las reglas, establecer las pruebas para validar el sistema completo.

Entrega de la práctica

- **Fecha y forma de entrega: 1º.- Entrega de tus bases de reglas:** *antes del 30-4-09* se debe subir al *Campus Virtual* y enviar al foro general una versión de tus bases y una breve descripción de los objetivos, ejemplos de todos los tipos de consultas y resultados que se pueden obtener.
- **2º.- Entrega de la práctica:** *antes del día 31 de Mayo del 2009*. El fichero empaquetado se entregará a través del *Campus Virtual* de tu grupo; o, si los alumnos son de varios grupos (mañana o tarde), se entrega en el grupo que más miembros haya.
- **Formato de la memoria:** en formato HTML con la siguiente estructura,
 - Título y autores del trabajo
 - Especificaciones del sistema desde vuestro punto de vista (no vale copiar el enunciado)
 - **IMPORTANTE:** Explicar los problemas encontrados en la Fase 1 y en la Fase2.
 - Enumerar las fuentes de conocimiento (p.e: URLs de los datos) y las decisiones de diseño: qué estructura tienen las bases de reglas, qué representan las reglas, si hay meta-conocimiento (p.e: si se modela una máquina de estados), elementos auxiliares creados para resolver los diferentes aspectos propuestos, etc...
 - Descripción a alto nivel de la implementación realizada: módulos y clases principales, se valorará el uso de diagramas UML de clases y de estado (¡sin incluir listado de código!)
 - Breve y clara descripción de cómo utilizar dicha implementación: cómo compilar el código fuente, cómo ejecutar la aplicación, cómo reproducir las pruebas más significativas que habéis hecho, además de enumerar las pruebas realizadas, etc...
 - Si se ha *reutilizado* código, se debe indicar clara y explícitamente la fuente y el tipo de reutilización llevada a cabo (uso de bibliotecas de algoritmos de búsqueda, reutilización de la interfaz gráfica...)
 - Conclusiones: opinión sobre la utilidad del sistema (ventajas, limitaciones), dificultades encontradas, mejoras futuras, etc...
- **Formato de la implementación:**
 - **Lenguaje:** Java para la aplicación y JESS para representación del conocimiento y motor de inferencia.
 - El código **no** debe depender de su instalación en directorios concretos, *la ejecución* no debe depender de ningún entorno de desarrollo, como Eclipse, Jbuilder, etc.
- **Formato de entrega:**
 - Fichero comprimido *Prac2_Apellido1Alumno1_Apellido1Alumno2_Apellido1Alumno3.zip* que contendrá, al menos, los siguientes directorios:
 - *memoria:* ficheros .html de la memoria siendo el principal index.html
 - *fuentes:* ficheros fuente de vuestra aplicación
 - *pruebas:* ficheros de prueba que hayáis usado
 - *aplicación:* un .jar para probar vuestra aplicación

Evaluación

En la evaluación de la práctica se tendrán en cuenta los siguientes criterios:

1. **Adquisición del conocimiento:** Calidad del conocimiento y de la variedad de sus bases de casos.
2. **Diseño:** estructura de las reglas y las bases de reglas, calidad de las decisiones para solucionar las características descritas en el enunciado, qué aspecto resuelve cada base, qué preguntas puede contestar el sistema.
3. **Implementación:** se valorará la claridad de la implementación, la organización del código, su arquitectura, el diseño orientado a objetos, y la facilidad de uso del sistema (posibilidad de diferentes consultas, cómo se añaden y quitan reglas, *manual de instrucciones para el usuario*, etc...).
4. **Experimentación y análisis de los resultados:** se valorará la cantidad de pruebas realizadas (*incluir en la memoria trazas de 5 consultas diferentes*). Se valora la claridad del tipo de pruebas (significativas, cobertura total o parcial de las BC,...) y la descripción para reproducir las pruebas. Deberían dispararse todas (o la mayoría) las reglas alguna vez en las pruebas.
5. **Memoria:** se valorará la claridad de la memoria: bases de conocimiento, diseño, arquitectura, pruebas y manual del usuario y del sistema. El uso de UML.