Curso 2015-2016

|  |
| --- |
| mrodrigue212@alumno.uned.es | Manuel Rodríguez Sánchez |

|  |
| --- |
| Desarrollo en prolog de un sistema inteligente basado en reglas (sbr) |

Contenido

[1. Introducción. 2](#_Toc450252434)

[2. Descripción del conocimiento del dominio. 2](#_Toc450252435)

[3. Metodología de desarrollo 2](#_Toc450252436)

[4. Descripción de la estructura de la base de las reglas y consideraciones de eficiencia. 3](#_Toc450252437)

[5. Código comentado en PROLOG 3](#_Toc450252438)

[6. Descripción y casos de prueba. 9](#_Toc450252439)

[7. Dificultades encontradas 11](#_Toc450252440)

1. Introducción.

A lo largo de los siguientes puntos, se va a desarrollar el segundo ejercicio práctico de evaluación a distancia, consistente en la construcción en *Prolog* de un sistema basado en reglas que represente hechos y reglas de un mundo virtual.

El tema elegido para esta práctica ha sido el entorno de ejecución de un plan de formación, donde se organizan, gestionan y se imparten acciones formativas.

Este tema fue propuesto al Equipo Docente, el cual me autorizó a su realización.

1. Descripción del conocimiento del dominio.

Disponemos en este entorno de: docentes o profesores, cursos, especialidad de cada docente, curso que imparte un docente en base a su especialidad y por último, el personal que organiza los cursos y profesores, todo englobado dentro de un plan de formación.

Se establece una serie de condiciones necesarias para poder desarrollar un curso, esto es: el importe, el número de alumnos, la capacidad y especialidad de los docentes, etc. Puede haber varias ediciones de un mismo curso a lo largo del periodo de tiempo, que dura el plan de formación, cada vez que haya un grupo para la realización de un mismo curso, será una nueva edición de ese curso, identificada por un código de grupo o edición.

Cada curso tiene un coste por alumno, y en cada edición de un curso ha de haber un número de alumnos, esto nos da un presupuesto total en base a los alumnos, el curso y su precio.

1. Metodología de desarrollo

Una vez elegido el tema, se desarrolla un primer boceto en papel, estableciendo algunos aspectos como tipos de acciones formativas, número de alumnos, docentes, hechos con y sin variables, constantes, predicados, reglas, etc. siempre intentando adaptar y trasladar lo que es el día a día en mi trabajo (técnico de formación y docente en una escuela de negocios).

Con este primer borrador, se inicia el desarrollo en *Prolog*, estableciendo una base de conocimiento mediante unos hechos y reglas, y para facilitar las consultas a la base de conocimiento se desarrolla un menú de opciones.

He de indicar, que el resultado final difiere del borrador inicial, ya que se han ido ajustando y adaptando los elementos de este SBR de acuerdo con las necesidades del sistema, lenguaje Prolog, y conocimientos que poco a poco se van adquiriendo.

Muy importante indicar que los trabajos de compañeros de otros años han sido muy útiles, junto con algún que otro video tutorial, documentos aclaratorios, foros y contenidos que el Equipo Docente ha facilitado en aLF.

1. Descripción de la estructura de la base de las reglas y consideraciones de eficiencia.

De cara a que el programa tenga una ejecución lógica y eficiente, se ha creado un menú de opciones desde el que se accede a muchas de las preguntas y operaciones que se piden en el enunciado.

La base de conocimiento se ha estructurado en:

* Un primer bloque con los hechos, constantes y variables dinámicas.
* En el segundo bloque se encuentra el menú con las llamadas a cada una de las opciones, y sus correspondientes operaciones y reglas asociados a cada opción.
* En un tercer bloque, encontramos varias reglas que son llamadas desde algunas de las operaciones de las opciones de menú.

1. Código comentado en PROLOG

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Base de conocimiento de un plan de formación.

Actividad práctica II, Fundamentos de Inteligencia Artificial

Curso: 2015-2016

Autor: Manuel Rodríguez Sánchez

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

% ----------------------------BLOQUE 1 HECHOS Y PREDICADOS----------------------

% Constantes. Personal que organiza el plan de formación El

% centro tiene un director que gestiona el personal

director.

%El centro tiene un técnico que planifica acciones formativas

tecnicoFormacion.

%En un periodo de tiempo hay un plan de formación

planDeFormacion.

:-dynamic (esDocente/1).

%Docentes o profesores del centro de formación.

esDocente(juan). /\*Juan es un docente\*/

esDocente(manuel).

esDocente(sonia).

esDocente(ismael).

%Cursos del plan de formación

esCurso(excel). /\*excel es un curso\*/

esCurso(contabilidad).

esCurso(rrhh).

esCurso(comunicacion).

% docente(profesor, especialidad). Especialidad de los docentes o

% profesores

docente(manuel,informatica). /\*manuel es especialista en informática\*/

docente(ismael,economicas).

docente(sonia,derecho).

docente(juan,psicologia).

%curso(curso,profesor). Cursos que imparte cada docente.

curso(excel,manuel). /\*excel es impartido por manuel\*/

curso(contabilidad,ismael).

curso(rrhh,juan).

curso(comunicacion,sonia).

% Número de grupo. Cada acción formativa se identifica por un

% código de grupo, y esto significa que un curso puede tener varias

% ediciones, identificando estas por su código de grupo.

% numeroGrupo(codigoGrupo, curso).

numeroGrupo(1,excel). /\*el grupo 1 es una edición del curso excel\*/

numeroGrupo(2,excel).

numeroGrupo(3,excel).

numeroGrupo(4,contabilidad).

numeroGrupo(5,contabilidad).

numeroGrupo(6,rrhh). /\*el grupo 6 es una edición del curso rrhh\*/

numeroGrupo(7,rrhh).

numeroGrupo(8,rrhh).

numeroGrupo(9,comunicacion).

% Presupuesto para cada uno de los grupos.

% presupuestoCurso(codigoGrupo,totalEuros).

presupuestoCurso(1,450). /\*el grupo 1 tiene un presupuesto de 450 euros\*/

presupuestoCurso(2,360).

presupuestoCurso(3,270).

presupuestoCurso(4,640).

presupuestoCurso(5,520).

presupuestoCurso(6,125).

presupuestoCurso(7,175).

presupuestoCurso(8,275).

presupuestoCurso(9,400).

% Número de alumnos por cada grupo previsto.

% numeroAlumnos(codigoGrupo,numerodeAlumnos)

numeroAlumnos(1,15). /\*el grupo 1 tiene 15 alumnos\*/

numeroAlumnos(2,12).

numeroAlumnos(3,9).

numeroAlumnos(4,16).

numeroAlumnos(5,13).

numeroAlumnos(6,5).

numeroAlumnos(7,7).

numeroAlumnos(8,11).

numeroAlumnos(9,20).

% ------------BLOQUE 2 - MENÚ DE OPCIONES PARA HACER LAS PREGUNTAS A LA BC------------------

%Esta regla limpia la pantalla y ejecuta el menú de opciones

iniciarprograma:-limpiaPantalla,programa.

%Borrar todo lo que hay en la pantalla

limpiaPantalla:-write('\033[2J').

%Menú de opciones

programa:-write('\nMenú de opciones:\n\n'),

write(' 1-Precio por alumno de un curso previsto.\n'),

write(' 2-Verificar un curso y su especialidad\n'),

write(' 3-Verificar curso que imparte un docente\n'),

write(' 4-Agregar un nuevo docente\n'),

write(' 5-Eliminar un docente\n'),

write(' 6-Listar los grupos de cursos que han sido ejecutados\n'),

write(' 7-Comprobar la especialidad de un curso\n'),

write(' 8-Comprobar que un curso se ha ejecutado\n'),

write(' 9-Comprobar que un curso se puede desarrollar\n'),

write(' 10-Salir\n\n'),

write('Selecciona mediante su número, la opción que se desee ejecutar: '),

read(Opcion),limpiaPantalla,opcion(Opcion).

%--------------------------------------------------------------------------------------------

% En esta opción, recogemos el nombre del curso, y vemos cuanto es su

% coste por alumno. Divide el total del presupuesto del curso entre el

% número de alumnos de ese curso o edición. Cada curso (NO GRUPO) tiene

% un precio por alumno.

opcion(1):-!,write('Nombre del curso: '),

read(C),nl,

numeroGrupo(G,C),presupuestoCurso(G,P),numeroAlumnos(G,N),Resultado is P/N,write('\nEl precio del curso por alumno es: '),write(Resultado),write(' Euros'),nl,nl,programa.

% opcion(2), vamos a preguntar si un curso pertenece a una especialidad.

opcion(2):-!,write('Nombre del curso: '),

read(Curso),

write('\nEspecialidad del curso: '),

read(Especialidad),(especialidadCurso(Curso, Especialidad)->write('Correcto\n\n');write('Los datos introducidos no son correctos\n\n')),programa.

% opcion(3), preguntamos si un curso lo imparte un docente siempre y

% cuando el director organice al docente y el técnico de formación

% planifique el curso.

opcion(3):-!,write('Nombre del curso: '),

read(Curso),

write('\nIntroduce el profesor: '),

read(Docente),organiza(director,Docente),planifica(tecnicoFormacion,Curso),(imparteCurso(Curso,Docente)->write('Correcto\n\n');write('Los datos introducidos no son correctos\n\n')),programa.

% Insertamos un docente en la lista de docentes. Se llama a la regla que

% realiza el assert

opcion(4):-!,write('Nombre del docente: '),

read(Docente),insertarDocente(Docente),programa.

% Eliminamos un docente de lista de docentes. Se llama a la regla que

% realiza el retract

opcion(5):-!,write('Nombre del docente: '),

read(Docente),eliminarDocente(Docente),programa.

%Se llama a la regla que lista los 9 grupos ejecutados

opcion(6):-!,listaCursos(9),programa.

%Se verifica la especialidad a la que pertenece un curso

opcion(7):-!,write('Nombre del curso: '),

read(Curso),

write('\nIntroduce la especialidad: '),

read(Especialidad),(especialidadCurso(Curso,Especialidad)->write('Correcto\n\n');write('Los datos introducidos no son correctos\n\n')),programa.

%Se verifica si un grupo ha sido ejecutado.

opcion(8):-!,write('Numero de grupo: '),

read(Grupo),(cursoEjecutado(Grupo,\_)->write('El curso correspondiente al grupo, ha sido ejecutado\n\n');write('Los datos introducidos no son correctos\n\n')),programa.

%Se verifica si un curso se puede ejecutar o desarrollar

opcion(9):-!,write('Introduce el curso: '),

read(Curso),(desarrolloCurso(Curso)->write('El curso puede ser ejecutado\n\n');write('Los datos introducidos no son correctos\n\n')),programa.

%Si seleccionamos la opción 10, termina la ejecución del programa

opcion(10):-!,write('Programa finalizado\n').

% Si no se selecciona ninguna opción, se pedirá nuevamente una entrada

opcion(\_):-!,write('La opción no es correcta o no existen datos, inténtelo de nuevo.\n\n'),programa.

% ---------------------------- BLOQUE 3 - REGLAS ------------------------------------

% Al menos uno de los predicados se define mediante dos o mas reglas.El

% curso ha sido planificado si el tecnico de formacion lo ha gestionado,

% el curso es impartido por un docente, y el curso es realmente un

% curso.

esPlanificado(C):-planifica(tecnicoFormacion,C),imparteCurso(C,\_),esCurso(C).

%viene de la opción 4

insertarDocente(D):-(assert(esDocente(D))->write('Docente insertado correctamente\n');write('Error en datos.Operación no realizada\n')).

%viene de la opción 5

eliminarDocente(D):-(retract(esDocente(D))->write('Docente eliminado correctamente\n');write('No existe el docente\n')).

% Un curso se habrá ejecutado si ha tenido presupuesto y alumnos. Opción 8

cursoEjecutado(G,\_):-presupuestoCurso(G,\_),numeroAlumnos(G,\_).

% ¿Pertenece el curso C a la especialidad E? Llamada desde la opción 7

especialidadCurso(C,E):-curso(C,D),docente(D,E).

% ¿Qué curso es impartido por un docente? Llamada desde la opción 3

imparteCurso(C,D):-curso(C,D).

% El técnico de formación planifica un curso, siempre que el curso C sea un curso

planifica(tecnicoFormacion,C):-esCurso(C).

% El director organiza el personal docente siempre que D sea un profesor

organiza(director,D):-esDocente(D).

% Un curso se va a desarrollar o empezar si ha sido planificado por el

% técnico de formación, el curso lo imparte un docente y el director

% organiza el docente que ha de impartirlo. Llamada por la opción 9

desarrolloCurso(C):-esPlanificado(C),imparteCurso(C,D),organiza(director,D).

% Listamos los grupos de cursos ejecutados. Aquí se usa recursividad.

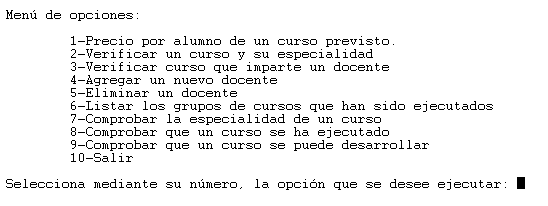
% Es llamada por la opción 6

listaCursos(G):-(G\==0->numeroGrupo(G,C),write('Grupo: '),write(G),write(' Curso: '),write(C),nl,G1 is G-1,listaCursos(G1);write('Se han listado todos los grupos')).

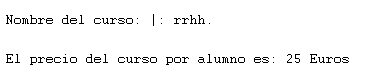
1. Descripción y casos de prueba.

Iniciamos el programa haciendo la llamada:

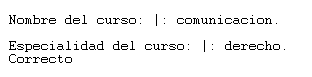
?-iniciarprograma.



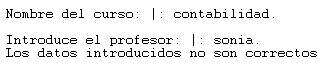
Ejecución de la opción 1:



Ejecución de la opción 2.



Ejecución de la opción 3.



Se ha introducido un profesor que no puede impartir el curso “contabilidad” por que no es su especialidad.



En este caso, la profesora Sonia puede impartir el curso “comunicación” por que es su especialidad.

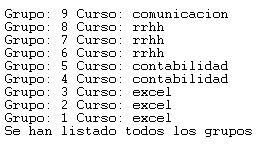
Ejecución de la opción 4.



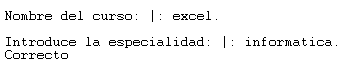
Ejecución de la opción 5.



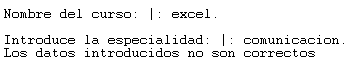
Ejecución de la opción 6.



Ejecución de la opción 7.



El curso “Excel” pertenece a la especialidad indicada.



El curso “Excel” no pertenece a la especialidad indicada.

Ejecución de la opción 8.





Ejecución de la opción 9.



El curso “rrhh” puede ser ejecutado por que cumple las condiciones.



Por alguna razón el curso “Word” no puede ser ejecutado.

1. Dificultades encontradas

Las dificultades principales con las que me he encontrado han sido, primeramente sobre que tema hacer el SBR, una vez me decidí, lo segundo que más dificultad me ha provocado es establecer los hechos, predicados y las reglas. He dado muchos palos de ciego en el diseño de la base de conocimiento, resultándome bastante complicado el desarrollo de los requisitos que se pedían en el enunciado. Con respecto al lenguaje Prolog, no me ha revestido mucha dificultad su aprendizaje, hay una extensa documentación, y si se tienen los conceptos claros, es fácil.