

INGENIERÍA DEL CONOCIMIENTO

Juan Alberto Sigüenza Pizarro y Julia Díaz García
Escuela Técnica Superior de Informática
Universidad Autónoma de Madrid

siguenza@irenc.iic.uam.es

Ubicación de la asignatura dentro del plan de estudios

La asignatura Ingeniería del Conocimiento (IC) es una materia obligatoria ubicada en el segundo cuatrimestre del tercer curso de la titulación de Ingeniería Informática en la Universidad Autónoma de Madrid. Dicha titulación es una Ingeniería Superior distribuida en cuatro años con una duración total de 300 créditos.

Esta asignatura tiene una asignación de 75 horas (7.5 créditos), repartidos a lo largo de 15 semanas de duración. Esto da una distribución de 5 horas semanales de clases: 3 de teoría y 2 de prácticas.

Dentro del plan de estudios presenta las siguientes vinculaciones con otras asignaturas del mismo área del conocimiento:

Inteligencia Artificial (asignatura troncal): se encuentra ubicada en el primer cuatrimestre de tercer curso y por tanto precede a la IC. Es recomendado que se curse antes de la IC.

Lógica (asignatura optativa): se encuentra ubicada en el segundo cuatrimestre de tercer curso y por tanto es simultánea a la IC.

Fundamentos de Neurocomputación (asignatura optativa): se encuentra ubicada en el primer cuatrimestre de cuarto curso y por tanto sucede a la IC.

Objetivos

El objetivo general de esta asignatura es el de dotar a los alumnos de tercer curso de Ingeniería Informática de la UAM de una introducción general al mundo de la Ingeniería del Conocimiento. Por este motivo, el programa incluye una visión amplia y de profundidad suficiente, para ubicarles dentro del área.

Como grandes objetivos particulares podemos destacar:

-Integración con otras materias del mismo área del conocimiento como la Lógica, la Inteligencia Artificial y la Neurocomputación.

-Conocimiento preciso de los principales tipos de Sistemas Basados en el Conocimiento.

-Capacidad de resolver problemas utilizando sistemas de forward y backward chaining.

-Capacidad de resolver problemas relacionados con el manejo de probabilidad e incertidumbre dentro de los diferentes tipos de Sistemas Basados en el Conocimiento.

-Profundizar en el conocimiento de la programación en LISP.

Contenidos

La asignatura se encuentra dividida en dos partes íntimamente coordinadas entre si: Teoría y Prácticas con una relación de esfuerzo docente 3:2. Los contenidos de cada parte son los siguientes:

Teoría (entre paréntesis el número de horas dedicadas a cada tema):

- 1.-Bases morfológicas y fisiológicas del pensamiento y el conocimiento humano. (2)
- 2.-Pensamiento y conocimiento humano. Tipos y fuentes. (2)
- 3.-Introducción a los sistemas basados en el conocimiento.(1)
- 4.-La búsqueda como base de la inteligencia artificial. El algoritmo A*. (2)
- 5.-Estructura de los sistemas basados en el conocimiento. Puntos de vista del usuario, ingeniero del conocimiento y desarrollador. (1)
- 6.-Introducción a los sistemas basados en reglas. Tipos de razonamiento. Arquitecturas. Ventaja y desventajas. (1)
- 7.-Razonamiento hacia adelante. Redes de inferencia. Sistemas de pattern matching. Algoritmo Rete. (3)
- 8.-Razonamiento hacia atrás. Redes de inferencia. Sistemas de pattern matching. (2)
- 9.-Redes Asociativas. (2)
- 10.-Marcos. (2)
- 11.-Arquitecturas de Pizarra. (2)
- 12.-Introducción al razonamiento basado en incertidumbre. Aproximación Bayesiana. (2)
- 13.-Factores de certidumbre. (2)
- 14.-Teoría Dempster-Shafer de la Evidencia. (1)
- 15.-Lógica difusa. (2)
- 16.-Técnicas avanzadas de razonamiento: Razonamiento basado en modelos. Razonamiento cualitativo. Razonamiento cuantitativo. Razonamiento basado en casos. Razonamiento Temporal. (2)
- 17.-Introducción a la adquisición del conocimiento. (2)
- 18.-Características generales del desarrollo práctico de los sistemas basados en el conocimiento. (1)
- 19.-Introducción a las Redes neuronales. (3)

Prácticas (entre paréntesis el número de horas dedicadas a cada tema):

Se realizan 5 prácticas como las que se enumeran a continuación, que son una recopilación de las realizadas hasta ahora.

1.- Resolución de un problema, como el 8-puzzle, mediante distintos mecanismos de búsqueda (4)

2.- Aplicación de reglas y restricciones en resolución de problemas como el de las Torres de Hanoi o el movimiento de bloques (4)

3.- Implementación de pattern matching con primitivas que puedan aplicar restricciones sobre patrones (4)

4.- Representación de conocimiento utilizando técnicas hipertextuales. (4)

5.- Creación de un pequeño motor de inferencia que realice forward-chaining sobre una base de hechos y reglas, y efectúe pattern-matching de las premisas sobre los hechos. (6)

6.- Creación de un pequeño motor de inferencia que realice backward-chaining, sobre una base de hechos y reglas, y efectúe pattern-matching de las premisas sobre los hechos. (6)

7.- Aplicación de la aproximación Bayesiana al razonamiento basado en incertidumbre. (4)

Método de enseñanza

Teoría y Problemas

Existe un grupo de clases teóricas, para cada tema del programa las clases comienzan con explicaciones sobre pizarra o utilizando transparencias por parte del profesor, en la mayoría de los casos a las explicaciones teóricas se suceden planteamiento de casos prácticos y problemas a resolver en primera instancia por los alumnos y su resolución final por parte del profesor.

Prácticas

Las prácticas se realizan de forma individual o en grupos de dos, como máximo.

En las sesiones iniciales se les da una guía con las pautas a seguir para la adecuada elaboración de las mismas. Se consigue así que, desde el primer momento conozcan cuáles son los criterios de evaluación que se van a aplicar, así como que exista una uniformidad, en la medida de lo posible, en los trabajos que presenten los alumnos. Dentro de los puntos que se especifican en esta guía se encuentran los siguientes:

- Formato de entrega.

Los alumnos deben presentar las prácticas en disquete con los fuentes correspondientes. Como en cada entrega presentan más de una práctica y a veces los disquetes se comparten por distintos grupos, la estructura del disquete será la siguiente:

1. Si el disquete es compartido por varias personas, un directorio por cada grupo.
2. Un directorio por cada hoja de prácticas (\HOJA1\...)
3. Un fichero LEEME en cada directorio con los comentarios oportunos a la práctica en cuestión: problemas encontrados, respuestas a preguntas teóricas, ...
4. Un fichero de ejemplos y otro/s con los programas. *

Los ficheros de ejemplos son muy importantes sobre todo para que los alumnos confirmen la coherencia de los datos que presentan. También ayudan a la evaluación puesto que en la mayoría de los casos demuestran el grado de depuración, por parte de los alumnos, de la práctica que presentan.

De los ficheros LEEME se obtiene información acerca de la evolución del grupo, tanto en nivel de conocimiento del lenguaje como de la materia en sí. La mayoría de los mensajes que los alumnos redactan suelen ser acerca de las dificultades que han encontrado ya sea en el uso del lenguaje de programación, en la comprensión de la práctica o en la aplicación de la teoría. Por supuesto todos estos mensajes son de gran ayuda y se tienen en cuenta en el resto de las clases.

Una innovación añadida este año es que, para evitar en la medida de lo posible copias entre alumnos, la evaluación de las prácticas concluye con una entrevista oral en la que los alumnos responden por turno a preguntas acerca de la implementación de las prácticas.

- Fechas de entrega.

Se intenta que los alumnos empiecen a ser responsables, están en su tercer curso de la carrera, y se vayan organizando el trabajo. De esta forma las prácticas se van repartiendo de forma pautada, coordinándose con la teoría, y en las fechas de los exámenes parciales se van presentando las que se haya acumulado hasta ese momento. Suelen tener que presentar una media de 2 a 3 en cada parcial. Las notas son comunicadas siempre antes de la siguiente entrega junto con comentarios acerca de su código, posibles mejoras, etc.

- Documentación en el código.

Para que los programas, en código LISP, no tengan el aspecto de una ristra de paréntesis sin sentido, se establecen ciertas normas de documentación en el código para que los alumnos las sigan. La documentación hace referencia a la cabecera de los programas, de las funciones y del código.

- Criterios de evaluación.

Para que todos los alumnos dirijan sus esfuerzos en el mismo sentido y así se consiga un estilo común de trabajo, se establecen un número de criterios generales de evaluación.

Puesto que en esta asignatura se intenta que, en la medida de lo posible, los alumnos sean responsables, se establecen distintas técnicas de trabajo entre el profesor y éstos. Una de ellas consiste en que, el profesor explique la práctica y en la pizarra escriba un pseudo-código del programa.

En otros casos, el profesor da tiempo para que los alumnos lean la práctica y discute con ellos, de forma individual o por grupos, el diseño o las preguntas que tengan. Así, dependiendo del nivel del alumno, sólo discute detalles técnicos o le explica con todo detalle en qué consiste el trabajo.

Evaluación de conocimientos

Los conocimientos teóricos y prácticos se evalúan por separado, aunque la nota que se da al alumno es el resultado de la ponderación de ambos conocimientos: 75% la teoría y un 25% la práctica.

A lo largo del curso se efectúan dos exámenes intermedios para evaluar los conocimientos de teoría y problemas. Dichos exámenes no son liberatorios pero pueden influir en la nota final al tener asignados cada uno un 20% de la nota final de teoría. Dichos exámenes intermedios tienen una duración de una hora y se hacen de forma coordinada con el resto de asignaturas del curso el mismo día. En cada examen los alumnos se examinan de toda la materia vista hasta el momento.

Existe un examen final de todos los contenidos teóricos mas problemas de la asignatura. Este examen de una tres horas de duración, puede tener un peso total del 60% en la nota final, si el alumno cuenta con buenas notas en los exámenes intermedios, o del 100% si la notas en los intermedios no son buenas. Siempre se elige la opción mas beneficiosa para los alumnos.

En lo que se refiere a las prácticas, se intenta que los alumnos empiecen a ser responsables, están en su tercer curso de la carrera, y se vayan organizando el trabajo. De esta forma las prácticas se van repartiendo de forma pautada, coordinándose con la teoría, y en las fechas de los exámenes parciales se presentan las que se haya acumulado hasta ese momento. Suelen tener que presentar una media de 2 a 3 en cada parcial. Las notas son comunicadas siempre antes de la siguiente entrega junto con comentarios acerca de su código, posibles mejoras, etc.

Las prácticas son evaluadas atendiendo a los siguientes criterios:

1. Corrección y comprensión de la teoría. El programa debe hacer lo que se le pide y se deberá realizar de acuerdo a los aprendido en las clases.
2. Generalidad, modularidad. Se intentarán hacer programas y funciones lo más genéricas posible para poder ser usadas, siempre que sea posible, en otros problemas. De hecho, en algunos casos las prácticas se basan unas en otras.
3. Claridad, elegancia, concisión. La estructura de los programas habrá de ser clara, permitiendo una fácil lectura y comprensión del código que facilite el mantenimiento.
4. Complejidad. Se valorará el dominio del lenguaje de programación, así como la implementación de algoritmos en su variante más compleja.
5. Eficiencia. Se deberá cuidar el ahorro de variables y el tiempo de ejecución del programa.
6. Documentación. Los programas y funciones habrán de estar suficientemente documentados, con sus cabeceras correspondientes, instrucciones de uso, etc. Se cuidará así mismo la nomenclatura de variables y funciones.

La evaluación de las prácticas pasa por distintas fases. En primer lugar se leen todos los ficheros que ha presentado el alumno y se obtiene una primera idea acerca de la comprensión de la teoría, de la generalidad, modularidad, claridad, etc. del código, y de la documentación. A continuación se prueban con ejemplos aportados por los alumnos y con una batería elaborada por el profesor para cada práctica.

Tabla del reparto de porcentajes entre teoría y Prácticas en la calificación final

NOTA GLOBAL: 100%			
Prácticas 25%	Teoría	75%	
Prácticas	1 examen 20%	2 examen 20%	examen final 60%

Libro

Como libro de texto se utiliza:

The Engineering of Knowledge Based Systems. Theory and Practice.
A. J. González y D. D. Dankel
Prentice Hall, 1993

Papel de la programación en la asignatura

Puesto que esta asignatura se coordina con la de Inteligencia Artificial, entre otras, en la que los alumnos aprendieron a programar en Lisp, se utiliza este lenguaje de programación para que fijen sus conocimientos y porque es uno de los más adecuados para el desarrollo de programas de Inteligencia Artificial. El objetivo es que cuando utilicen una herramienta del mercado sepan qué es un motor de inferencia, cómo funciona y que problemas pueden encontrar porque ellos trataron de resolver, por supuesto en un grado de complejidad inferior, problemas similares.

Lisp aporta conocidas ventajas en la implementación de este tipo de problemas, como son:

- la recursividad como método de funcionamiento;
- el uso de expresiones simbólicas, listas y átomos, como único tipo de variables;
- el ser interpretado, con lo que los alumnos pueden estar probando sus programas de una forma muy sencilla al mismo tiempo que los implementan, etc.

Nosotros utilizamos la versión 2.0 de Allegro CL para Windows que ofrece una interfaz agradable en un entorno, Windows, conocido por nuestros alumnos.