

Aspectos docentes de la asignatura de “Inteligencia Artificial e Ingeniería del Conocimiento 1” del plan de estudios de Ingeniería Informática de la Universidad de Zaragoza

Jose A. Bañares Bañares
Julio Rubio García
Pedro R. Muro Medrano
Javier Zarazaga Soria

Departamento de Informática e Ingeniería de Sistemas
Centro Politécnico Superior
Universidad de Zaragoza
María de Luna 3
50015 Zaragoza

Resumen

En este trabajo se presentan los aspectos docentes de la asignatura de Inteligencia Artificial e Ingeniería del Conocimiento 1. Es una asignatura restringida a la parte de introducción, búsqueda y representación del conocimiento y con un especial hincapié en la programación, generalmente con Common Lisp.

1. Contexto de la asignatura

El presente trabajo se centra en algunos aspectos docentes relacionados con la asignatura de “Inteligencia Artificial e Ingeniería del Conocimiento 1”. Esta es una asignatura troncal del plan de estudios de Ingeniería Informática que se imparte en el Centro Politécnico Superior de la Universidad de Zaragoza.

La mencionada asignatura tiene una carga lectiva de 4,5 créditos totales de los cuales 3 son teóricos y otros 1,5 prácticos. Se encuentra situada en el séptimo cuatrimestre de los estudios de Ingeniería Informática, constituyendo la primera asignatura de las dos en que está dividida la materia troncal de Inteligencia Artificial e Ingeniería del Conocimiento. La otra asignatura, con similar contenido de créditos, se imparte en el octavo cuatrimestre.

2. Objetivos del curso

El objetivo general en esta primera asignatura de la materia Inteligencia Artificial e Ingeniería del Conocimiento es que el alumno tome el un primer contacto con este campo y se sensibilice con las posibilidades que ofrece para la resolución de problemas. De forma más concreta, el planteamiento que hemos seguido para desarrollar este curso ha estado guiado por tres pautas importantes:

1. Mostrar las bases históricas y el papel de la IA en la informática, cuáles son sus contribuciones reales y cuáles sus limitaciones. En este contexto creemos interesante que el alumno sea también consciente de que la IA es un área fundamentalmente multidisciplinar en la que la informática se ha visto beneficiada y ha beneficiado a otros campos como la psicología, la lógica, la investigación operativa, la lingüística, la robótica, la ingeniería, etc.
2. Introducir al alumno los conceptos, técnicas y sistemas que cubran los aspectos temáticos asignados al curso relacionados con la búsqueda y la representación y manipulación del conocimiento.
3. También nos interesa, y de forma muy especial, sus aspectos informáticos más prácticos. Esta vertiente práctica, fundamentada en la programación, impone su sesgo en el planteamiento de toda la asignatura. Esta vertiente práctica permite cubrir dos aspectos:
 - Por una parte, nos interesa que el alumno conozca y adquiera experiencia en unos lenguajes de programación, que si bien su utilización ha estado bastante restringida al área de la IA, tienen un enorme interés informático fundamentalmente por diferencias para el diseño de programas con respecto a lenguajes procedurales más tradicionales. Si bien los alumnos ya han recibido las bases de estos lenguajes en asignaturas anteriores, sólo en el contexto de la IA pueden llegar a conocer realmente el alcance de sus posibilidades y su gran interés como lenguajes de programación.
 - Por otra parte, nos permite observar la IA desde su punto de vista más de ingeniería en su vertiente de herramientas para la construcción de sistemas de software. Así queremos que, en la medida de lo posible, el alumno aprenda a crear programas que implementen y saquen partido de las técnicas de resolución de problemas y representación del conocimiento.

3. Contenidos

A continuación exponemos un extracto de los contenidos teóricos de la asignatura:

Introducción a la Inteligencia Artificial y la Ingeniería del Conocimiento
Programación en Common Lisp
Búsqueda en el espacio de estados
Estrategias de control ciegas
Búsqueda heurística
Estrategias de búsqueda en juegos con adversario
Sistemas basados en reglas
Representaciones estructuradas del conocimiento
 Redes semánticas
 Frames y objetos

Puede notarse la carencia de un punto especial para la parte de lógica. Hemos decidido no incluir un punto específico en la asignatura debido a la falta de tiempo y a que este tema ya está cubierto de forma parcial en varias asignaturas y de forma intensa en una optativa por lo que hemos decidido

4. Métodos de enseñanza utilizados

Podemos desglosar los métodos de enseñanza utilizados en las siguientes partes:

- Una serie de clases teóricas, soportadas por transparencias, en las que se estudian los problemas, métodos y sistemas. El alumno dispone de antemano de las copias de las transparencias y de un libro con una explicación bastante completa de las clases (actualmente el libro sólo cubre la parte de búsqueda) [RMB96].

- Las prácticas constituyen una parte fundamental del curso. La estrategia general que seguimos para las prácticas es proporcionar siempre un código ya elaborado por nosotros. La primera parte de las prácticas consiste en estudiar dicho código y utilizarlo, de esta forma los alumnos aprenden sobre programas ya hechos las formas de implementar los distintos conceptos y técnicas. Con esta Estrategia evitamos el gran trauma que supone para el alumno tener que partir de cero para la realización del trabajo. En la segunda parte se requiere una participación más creativa por parte del alumno, así se requiere que modifique o expanda el código para ampliar su funcionalidad, para ello necesita entender profundamente el código dado.
- Adicionalmente ponemos a disposición de los alumnos interesados diversas versiones de que cubre la mayor parte de los temas tratados, de esta forma tienen acceso a implementaciones de algunas partes que no se ven en prácticas y pueden comprobar asimismo distintas implementaciones de la misma técnica.

5. Papel de la programación y las herramientas dentro del curso

Como ya se clarificó en la sección de objetivos, la programación tiene un papel clave dentro de esta asignatura. Nuestra intención es que el alumno aprenda a desarrollar programas que implementen la mayor parte posible de conceptos y técnicas.

6. Lenguajes de programación utilizados en los trabajos prácticos

- Common Lisp. El lenguaje de codificación básico para la realización de los trabajos prácticos es Common Lisp. Este es un lenguaje de programación del que tenemos gran experiencia y del que estamos absolutamente convencidos de su utilidad para esta asignatura. Asimismo disponemos de un buen compilador de Common Lisp (concretamente Allegro Common Lisp) y tenemos disponible gran cantidad de material tanto para ejemplos y sistemas como para prácticas. Adicionalmente está la ventaja de que disponemos del software para trabajar en plataforma Macintosh, en PC con Windows y en HP con Unix con la ventaja de que el mismo código que desarrolla el alumno puede correr sin ningún cambio en cualquiera de los tres sistemas. De esta forma el alumno puede trabajar en casa y en cualquier sala de computadoras del centro sin ningún problema de compatibilidades.
- Para la parte de representaciones estructuradas del conocimiento utilizamos un pequeño lenguaje creado sobre Lisp y el sistema de objetos estándar de Lisp, CLOS.
- Para las prácticas de programación con sistemas basados en reglas hemos utilizado anteriormente OPS5 que corre sobre lisp pero este último año hemos cambiado a Clips.

7. Tipos de trabajos prácticos

En cuanto a trabajos prácticos el presente curso hemos realizado los siguientes:

- Práctica de “Programación en Common Lisp”. Su objetivo es por una parte, familiarizar al alumno con el entorno de programación de Allegro Common Lisp (editor emacs, manual de ayuda, depurador, uso del entorno). Por otra parte se practica con estructuras de datos, funciones de mapping y desarrollo de funciones recursivas y las herramientas de depuración propias de Lisp.
- Práctica de “Representación y búsqueda en el espacio de estados”. Esta práctica tiene dos objetivos: seguir familiarizando al alumno con el lenguaje Common Lisp (esta vez con una

utilización más complicada) y poner en práctica los conocimientos de búsqueda y representación en el espacio de estados. Para ello se les da el código de una búsqueda para el problema de los misioneros y los caníbales que tienen que utilizar y entender. Posteriormente tienen que modificar código para distintas estrategias, generalizarlo y aplicarlo al problema del 8-puzzle. Adicionalmente tienen que modificar el código para generar estadísticas sobre el árbol de búsqueda generado.

- Práctica de “Búsqueda heurística”. Es objetivo de esta práctica es experimentar con búsquedas heurísticas. Para ello se proporciona al alumno el código del A*, que el adapta para utilizarlo con el problema del 8-puzzle. Se pide asimismo que compare estadísticas con distintas heurísticas y con las estrategias ciegas.
- Práctica de “Representación del conocimiento basado en reglas”. El objetivo de la práctica es conocer y programar en un lenguaje basado en reglas. Para ello se utiliza el lenguaje Clips y se le proporciona al alumno el código de un simple problema de búsqueda (de prácticas anteriores conoce bien el funcionamiento de la búsqueda y sus problemas por lo que le resulta atractivo ver su solución con un lenguaje que exige un diseño absolutamente distinto). Una vez el alumno practica con el uso del código y el entorno del lenguaje, se le pide que programe unas mejoras.
- Práctica de “Representación estructurada del conocimiento”. El objetivo de la práctica es adquirir experiencia con representaciones estructuradas del conocimiento basadas en frames. Para ello les proporcionamos el código completo de un pequeño sistema de frames y la base de conocimiento de una aplicación inmobiliaria con la que se pueden practicar los típicos aspectos de herencia y la utilidad if-needed. Posteriormente se les pide ampliar el lenguaje de representación. Para esta práctica se requiere ya un razonable dominio de Common Lisp.

8. Estrategias para incorporar la programación en el curriculum de la asignatura

Como ya ha sido aclarado la programación constituye ya una parte fundamental de la asignatura por lo que no tiene sentido ninguna Estrategia de incorporación adicional.

9. Material didáctico utilizado

El material didáctico utilizado se puede clasificar en los siguientes apartados:

- Transparencias de clase y fotocopias que se proporcionan a los alumnos.
- Bibliografía que sirve de apoyo al curso. Para la parte de búsqueda proporcionamos al alumno un cuaderno de apuntes elaborado por los profesores [RMB96].
- Documentación y software que sirve de base a las prácticas.
- Software de dominio público en Lisp con código relacionado con la asignatura.
- Direcciones de internet con información de Lisp.

10. Referencias

A continuación ponemos la bibliografía que hemos utilizado más extensamente para preparar la asignatura:

Textos básicos:

[RMB96] J. Rubio, P.R. Muro-Medrano y J.A. Bañares. *Inteligencia Artificial e Ingeniería del Conocimiento 1. Búsqueda*, Depto. de Informática e Ingeniería de Sistemas, Universidad de Zaragoza. 126 pags. 1996.

Textos no básicos:

- [LS93] G.F. Luger, W.A. Stubblefield. *Artificial Intelligence. Structures and Strategies for Complex Problem Solving. 2nd ed.* The Benjamin/Cummings Publishing Company. 1993.
- [WH89] P.H. Winston, B.K.P. Horn. *Lisp. 3rd Edition.* Addison-Wesley Publishing Company. 1989.
- [STEE] G.L. Steele. *Lisp. The Language. Second Edition.* Digital Press. 1990.

ÍNDICE

1. <i>Contexto de la asignatura</i>	1
2. <i>Objetivos del curso</i>	1
3. <i>Contenidos</i>	2
4. <i>Métodos de enseñanza utilizados</i>	2
5. <i>Papel de la programación y las herramientas dentro del curso</i>	3
6. <i>Lenguajes de programación utilizados en los trabajos prácticos</i>	3
7. <i>Tipos de trabajos prácticos</i>	3
8. <i>Estrategias para incorporar la programación en el curriculum de la asignatura</i>	4
9. <i>Material didáctico utilizado</i>	4
10. <i>Referencias</i>	4