

Organización Docente de la Asignatura Ingeniería de los Sistemas Basados en el Conocimiento

Jose A. Bañares Bañares
Pedro R. Muro Medrano
Javier Zarazaga Soria
Santiago Comella Dorda

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

E-mail: {Banares,prmuro,javy}@posta.unizar.es, scomella@ebro.cps.unizar.es

Resumen

En este trabajo se presentan los objetivos, los contenidos y los métodos de enseñanza de la asignatura de Ingeniería de los Sistemas Basados en el Conocimiento (ISBC). La asignatura es una optativa que complementa las asignaturas troncales de Inteligencia Artificial e Ingeniería del Conocimiento. En la asignatura se hace especial hincapié en los temas de representación del conocimiento y en el empleo de entornos que integran distintos lenguajes de representación. El enfoque de la asignatura es eminentemente práctico, planteándose las clases teóricas alrededor de ejemplos de diseño e implementación para la presentación de los conceptos. En cuanto a los trabajos de la asignatura, además de las prácticas, se ha dado opción a la participación en proyectos en los que se ven las aplicaciones de la ISBC como un valor añadido.

1 Contexto de la asignatura

El presente trabajo se centra en algunos aspectos docentes relacionados con la asignatura de "Ingeniería de los Sistemas Basados en el Conocimiento (ISBC). Esta es una asignatura optativa 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 6 créditos totales de los cuales 4 son teóricos y otros 2 prácticos. Es una de las dos optativas, junto con "Visión por Computador" que completa la optatividad asociada a las dos asignaturas en que está dividida la materia troncal de Inteligencia Artificial e Ingeniería del Conocimiento (IAIC). La optativa de ISBC tiene como requisito haber cursado la primera de las asignaturas troncales.

Cada asignatura troncal de IAIC tiene una carga lectiva de 4,5 créditos totales, de los cuales 3 son teóricos y 1,5 son prácticos. El contenido de la primera troncal se restringe a la parte de introducción a la IA, búsqueda y representación del conocimiento, y el contenido de la segunda se centra en las técnicas de aprendizaje y de percepción del entorno.

2 Objetivos del curso

Este curso cubre los fundamentos teóricos de los sistemas basados en el conocimiento (SBC) como una rama de la inteligencia artificial, así como los aspectos prácticos del desarrollo de SBC. Se persiguen dos objetivos: 1) Que el estudiante comprenda la naturaleza, limitaciones, y aplicaciones viables de los sistemas basados en el conocimiento; y 2) Que el estudiante utilice, de manera efectiva, entornos de desarrollo de siste-

mas basados en el conocimiento.

De forma más concreta, el planteamiento que hemos seguido para desarrollar este curso ha estado guiado por tres pautas importantes:

1. Mostrar la ingeniería del conocimiento como una rama de la inteligencia artificial que persigue soluciones heurísticas, al modo de un experto en un dominio de aplicación, frente a soluciones algorítmicas. La IC se presenta como 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. Debido a la limitación a un cuatrimestre de la primera asignatura troncal, esta se centra en los temas de búsqueda y trata los aspectos de representación del conocimiento de forma escueta. La experiencia con las asignaturas troncales ha mostrado que el alumno se interesa más por la materia y retiene los conceptos cuando se muestran dando una relevancia especial a sus aspectos informáticos más prácticos. Esta línea, seguida en los temas de búsqueda de la primera troncal, es retomado en ISBC. Se profundiza en los conceptos de representación del conocimiento siguiendo una aproximación práctica en la exposición de los conceptos.
3. El carácter multidisciplinar y la necesidad de integrar paradigmas diferentes de programación permiten al alumno experimentar con lenguajes, que si bien su utilización ha estado bastante restringida al área de la IA, tienen un enorme interés informático. A pesar de que los alumnos ya han recibido las bases de estos lenguajes en asignaturas anteriores, pronto queda patente su incapacidad para diseñar e implementar utilizando lenguajes diferentes de los lenguajes procedurales más tradicionales. Una carencia especialmente grave detectada es la falta de experiencia en programación orientada a objetos. Siguiendo el sesgo práctico apuntado anteriormente, se persigue que en la medida de lo posible, el alumno aprenda a crear programas que implementen y saquen

partido de las técnicas de representación del conocimiento y de resolución de problemas.

3 Contenidos

A continuación se muestran los contenidos teóricos de la asignatura:

- **Sistemas basados en el conocimiento.** Definición. Funcionalidad. Estructura. Desarrollo. Ventajas y Desventajas.
- **Modelos Básicos de Representación del conocimiento.**

Sistemas basados en reglas. Arquitectura. Memoria de Trabajo y memoria de producción. Proceso de reconocimiento: Red de inferencia y sistemas de reconocimiento de patrones. Proceso de razonamiento: Encadenamiento progresivo y regresivo. Estrategias de control. Programación de sistemas de producción tipo OPS5 (CLIPS). Desarrollo y control. Eficiencia en el proceso de reconocimiento de patrones: RETE, TREAT. Ventajas y Desventajas.

Sistemas de representación estructurada. Frames: Implementación de un lenguaje basado en frames (IERL). Objetos: Common Lisp Object System (CLOS).

Integración de los distintos esquemas de representación. Programación con objetos y reglas en CLIPS. Entorno de desarrollo de SBC: KEE.

- **Modelos para representación del razonamiento complejo.**

Control de Alto nivel: Arquitecturas de pizarra.

Tratamiento de la incertidumbre en los SE. Razonamiento no monótono. Sistemas de mantenimiento del razonamiento. Razonamiento con incertidumbre: Modelos Bayesianos, Factores de certeza, Teoría de la evidencia, Cuantificación borrosa de la incertidumbre.

- **Aspectos metodológicos** Adquisición del conocimiento. Metodos estructurados de resolución de problemas.

El temario ha sido impartido por primera vez durante el curso 97/98. Hay que destacar el hecho de que ante la falta de experiencia en la programación y el diseño orientado a objetos se decidió cubrir estas deficiencias dedicando más tiempo del previsto a la representación estructurada del conocimiento. Esto ha hecho que haya sido imposible cubrir los últimos temas relacionados con aspectos metodológicos. Es de esperar sin embargo que estas deficiencias detectadas sean resueltas por otras asignaturas y se pueda ir más rápido en los temas de representación del conocimiento para dar cabida a aspectos metodológicos.

4 Métodos de enseñanza utilizados

Podemos hacer un desglose de los métodos de enseñanza utilizados en las clases teóricas, prácticas:

Una serie de clases teóricas, soportadas por transparencias, en las que se estudian los problemas, métodos y sistemas. Las clases teóricas incluyen siempre que es posible la presentación de algún ejemplo de implementación de los conceptos propuestos. El alumno dispone de antemano de las copias de las transparencias. No existe un libro que se siga a lo largo de todo el curso, pero para cada tema se da bibliografía recomendada. En cuanto a texto básicos de la asignatura [7] y [6] cubren los principales temas del curso, destacando de ambos la sencillez en la presentación de los temas. Una mayor profundización en los lenguajes basados en reglas se puede encontrar en [2] y [5]. Las referencias más didácticas de CLOS son [9], y [1] que presenta los aspectos de diseño e implementación en CLOS de una arquitectura de pizarra. En cuanto a los temas de razonamiento bajo incertidumbre, éstos son tratados con profusión en [5] y [3].

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 el software de dominio público relacionado con 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 Lenguajes de programación utilizados en los trabajos prácticos

Para la presentación de conceptos y realización de prácticas de programación con sistemas basados en reglas hemos utilizado CLIPS. CLIPS está disponible para cualquier plataforma, de manera que el alumno tiene fácil acceso a la herramienta. Ha sido útil para cubrir muchos de los temas de la asignatura: reglas, objetos, demonios, mantenimiento de la verdad.

Common Lisp. El alumno ya tiene experiencia de programar en Common Lisp los trabajos prácticos de la asignatura IAIC1. Este es un lenguaje de programación del que tenemos gran experiencia y del que estamos absolutamente convencidos de su utilidad para las asignaturas de Inteligencia Artificial. 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. En el curso de ISBC se profundiza en aspectos más avanzados de Lisp como macros, cierres léxicos y especialmente en la programación con Common Lisp Object Sys-

tem (CLOS).

Para el tema de representación estructurada del conocimiento utilizamos primero un pequeño lenguaje creado sobre Lisp que nos permite presentar los conceptos y mostrar como implementarlos creando un lenguaje basado en frames, sobre el que se proponen al alumno mejoras del lenguaje. El lenguaje es una adaptación del XRL (experimental representation language), presentado en [4]. Posteriormente se presenta CLOS, cuyo modelo de objeto es similar al de CLIPS, y permite pasar fácilmente a presentar ejemplos de integración de representaciones estructuradas y basadas en reglas en CLIPS.

Por último se presenta KEE [8], un entorno de desarrollo de SBC, que permite el encadenamiento adelante y atrás de las reglas, tiene un sistema de mantenimiento de la verdad, integra el lenguaje de frames y reglas, y ofrece un conjunto de imágenes activas que permiten la construcción rápida de prototipos con interfaz gráficos. Se muestran las ventajas de un entorno de este tipo para crear rápidamente prototipos, y las desventajas por su baja eficiencia y dificultad de integración.

6 Tipos de trabajos prácticos

En cuanto a trabajos prácticos, en el presente curso los alumnos han realizado los siguientes:

1. **Representación del conocimiento basada en reglas.** El objetivo de ésta práctica es coger experiencia en el desarrollo de programas con el lenguaje de reglas de CLIPS, tanto escribiendo código como depurando programas. Para ello se debe escribir un programa que resuelva el problema del mono y las bananas. Se deben derivar las reglas de forma sistemática a partir de los objetivos en que se clasificó el problema. Además se deben escribir las reglas de comprobación rigurosa de cada objetivo.
2. **Representación del conocimiento estructurada.** Esta práctica tiene por objetivo aprender los aspectos más relevantes de CLOS en aplicaciones de representación del conocimiento. Se pide la implementación de un paquete que permite definir circuitos. El programa ilustra como incorporar el conocimiento procedural mediante dos paradig-

mas de programación complementarios que facilitan la reusabilidad y extensibilidad del modelo a través de la encapsulación y la herencia: La programación orientada a objeto (técnica imperativa), y la programación dirigida por los datos o basada en demonios (técnica declarativa).

3. **Razonamiento Monótono.** El objetivo de la práctica es mostrar la utilidad de los sistemas de mantenimiento de la verdad. Se pide reproducir el caso del asesino ABC (Cap.7 [10]) en CLIPS.
4. **Entorno de desarrollo de sistemas basados en el conocimiento.** El objetivo de la práctica es presentar un entorno de desarrollo que permite crear prototipos rápidamente y que integra los esquemas de representación presentados. Se presentan las características de KEE a través de un tutorial y de un ejemplo que muestra como utilizar en un mismo problema el encadenamiento hacia adelante y hacia atrás de reglas.

Además de las prácticas se debe realizar un trabajo de asignatura. Los trabajos pueden hacerse en grupos de hasta tres alumnos. Entre los trabajos propuestos destacan:

- Diseño e implementación de una arquitectura de pizarra para descifrar mensajes cifrados en los que se ha hecho una simple sustitución de una letra por otra.
- Venta de billetes de autobús en las que hay involucradas diferentes concesiones.
- Ajuste a rutas de posiciones obtenidas con GPS.

En el primer trabajo se ha permitido el intercambio de información entre los grupos, siempre que se deje claro las aportaciones externas al grupo. El trabajo presenta problemáticas relacionadas con gran parte de los temas de la asignatura, aunque su interés es puramente académico.

Los dos últimos trabajos están integrados dentro de un proyecto de desarrollo de un sistema de localización de móviles. El sistema denominado OODISMAL (*Object Oriente Distributed Information System for Automatic Movil Location*,

<http://diana.cps.unizar.es/iaaa>), es una arquitectura que consiste en un conjunto de componentes distribuidos en una red de área local. Los componentes interactúan entre sí utilizando el estándar CORBA [11] y tienen como propósito la adquisición, almacenamiento y procesamiento de datos de localización provenientes de móviles que incorporen un dispositivo GPS. Sobre esta arquitectura se están desarrollando aplicaciones de monitorización y gestión de autobuses interurbanos. Los trabajos propuestos son respectivamente parte de la componente de localización y del sistema de billeteaje. Este tipo de trabajos permiten al alumno apreciar las aplicaciones de ISBC como un valor añadido más a otras aplicaciones, y no como aplicaciones aisladas. En la medida que sea posible todos los trabajos de la asignatura se orientarán en el futuro a proyectos desarrollados por nuestro grupo.

En este punto hay que destacar que sería interesante contar con CORBA para CLOS, de forma que pudiéramos contar con las ventajas de este lenguaje, adecuado para el estilo de desarrollo iterativo e incremental propio de los sistemas software de cierta complejidad, y pudiésemos además integrar fácilmente componentes desarrolladas en este lenguajes con el resto del sistema. Sin embargo hasta el momento no hemos encontrado ningún CORBA para CLOS disponible.

Referencias

- [1] G. Booch. *Object Oriented Design with Applications*. The Benjamin/Cummings Publishing Company, Inc., 1991.
- [2] L. Brownston, F. Robert, E. Kant, and N. Martin. *Programming Expert Systems. Second Edition*. Addison Wesley, 1989.
- [3] E. Castillo, J.M. Gutierrez, and A.S. Hadi. *Sistemas Expertos y Modelos de redes probabilísticas*. Monografías de la academia de Ingeniería, 1996.
- [4] E. Charniak, C.K. Riesbeck, D.V. McDermonntt, and J.R. Meehan. *Artificial Intelligence Programming*. Lawrence Erlbaum Associates, Publishers, 1987.
- [5] Joseph Giarratano and Gary Riley. *Expert Systems. Principles and Programming. Second Edition*. PWS Publishing Company, 1994.
- [6] A. Gómez, N. Juristo, C. Montes, and J. Pazos. *Ingeniería del Conocimiento*. Editorial Centro de Estudios Ramón Areces, S.A., 1997.
- [7] Avelino J. Gonzalez and Douglas D. Dankell. *The Engineering of Knowledge Based Systems*. Prentice Hall, 1993.
- [8] Intellicorp. *KEE User Guide*. Intellicorp, 1989.
- [9] S.E. Keene. *Object-Oriented Programming in Common Lisp. A programmer guide to CLOS*. Addison-Wesley, 1989.
- [10] E. Rich and K. Knight. *Inteligencia Artificial. Segunda Edición*. Prentice Hall, 1994.
- [11] S. Vinoski. Corba: Integrating diverse applications within distributed heterogeneous environments. *IEEE Communications Magazine*, pages 46–55, February 1997.