

Sistemas Basados en el Conocimiento y de Recomendación

Construcción de Sistemas Expertos

LISTA de desarrollos de SEs famosos, de Esqueletos y de lenguajes para el desarrollo de SEs

Posted by Sergio Marcellin
Category: No category

SEs “famosos” desarrollados:

Dendral (Dendral, 1965, Stanford University) y Meta dendral (Bruce Buchanan, Joshua Lederberg and Edward A. Feigenbaum)

MYCIN desarrollado por Edward H. Shortliffe, a mediados de los 70's en Stanford Medical School. E-MYCIN es la máquina de inferencias de MYCIN utilizada para desarrollar otros sistemas expertos como Puff y Caduceus

Prospector

Prospector desarrollado por SRI International con el fin de localizar yacimientos de minerales y/o depósitos de carbón..

XCON/R1 desarrollado por DEC para configurar computadoras VAX.

Bacon (Carnegie- Mellon University).

Otros desarrollos:

KAS (Knowledge Acquisition System) and Teiresias help knowledge engineers build expert systems.

- ONCOCIN assists physicians in managing complex drug regimens for treating cancer patients.
- Molgen helps molecular biologists in planning DNA experiments.
- Guidon is an education expert that teaches students by correcting answers to technical questions.
- Genesis assists scientists in planning cloning experiments.
- TATR is used by the Air Force in planning attacks on enemy air bases.

Para otras herramientas ver:

Ejemplos de desarrollo de Sistemas Expertos “famosos” por sus resultados.

DENDRAL [Lindsay, Robert K., Bruce G. Buchanan, Edward A. Feigenbaum, and Joshua Lederberg. Applications of Artificial Intelligence for Organic Chemistry: The Dendral Project. McGraw-Hill Book Company, 1980].

DENDRAL (Dendritic Algorithm) es un sistema experto (SE) en el área de química orgánica considerado como el primer SE desarrollado en la Universidad de Stanford durante los años sesentas, capaz de generar posibles estructuras químicas correspondientes a los datos de espectrometrías de masa.

Buchanan diseñó "Meta-Dendral" a partir de Dendral, haciendo que se volviera un herramienta general para el desarrollo de sistemas expertos en otras áreas diferentes a la química orgánica.

Muchas sistemas expertos exitosos, fueron derivados de Dendral; como ejemplos tenemos a: MYCIN, PROSPECTOR, XCON así como STEAMER; MOLGEN Y MACSYMA.

MYCIN

MYCIN es un sistema experto desarrollado a principios de los años 70 por Edgar Shortliffe, en la Universidad de Stanford. Fue escrito en Lisp, e inicialmente estaba inspirado en Dendral, otro sistema experto que tuvo cierto éxito a finales de los años 60. Su principal función consistía en el diagnóstico de enfermedades infecciosas de la sangre; además, Mycin era capaz de "razonar" el proceso seguido para llegar a estos diagnósticos, y de recetar medicaciones personalizadas a cada paciente (según su estatura, peso, etc.).

Método

El funcionamiento de Mycin se basaba principalmente en un sencillo motor de inferencia, que manejaba una base de conocimiento de aproximadamente unas 500 reglas. El programa capturaba las entradas a partir de una serie de preguntas (como por ejemplo, ¿Tiene el paciente molestias en el pecho?, o ¿Ha sido operado el paciente anteriormente?), que usualmente respondía el médico del paciente. Tras este proceso, Mycin mostraba la salida por pantalla, que consistía en una serie de posibles enfermedades (ordenadas por su probabilidad asociada), la explicación del por qué de cada uno de estos diagnósticos, y una serie de recomendaciones sobre el tratamiento a seguir por el paciente. Para calcular la probabilidad de cada uno de los resultados, los autores desarrollaron una técnica empírica basada en factores de certeza. Estos factores de certeza se calculaban de tal manera que en función de unas evidencias se asigna a la hipótesis un factor de certeza.

Resultados

Las investigaciones realizadas por la Stanford Medical School, desvelaron que Mycin tuvo una tasa de aciertos de aproximadamente el 65%, lo cual mejoraba las estadísticas de la mayoría de los médicos no especializados en el diagnóstico de infecciones bacterianas (dominio en el que Mycin estaba especializado), que ejercían la profesión en aquellos años. Los médicos que trabajaban exclusivamente en este campo conseguían una tasa del 80%.

Poco a poco Mycin fue cayendo en desuso, debido principalmente a alguna de las debilidades que el programa presentaba, y también, por cuestiones éticas y legales que surgían al pasar la responsabilidad de la salud de una persona a una máquina (por ejemplo, si Mycin se equivocaba en algún diagnóstico, ¿quién asumía la culpa, el programador o el médico?).

XCON [for eXpert CONfigurer], desarrollado en 1978 por DEC para asistir a los vendedores a generar las ordenes de compra de los computadores VAX; no requiere el manejo de incertidumbre porque en esta "nube", siempre es posible especificar exactamente lo que es correcto hacer para cada conjunto particular de circunstancias.

PROSPECTOR [Ro Duda, 1978] sistema que evalúa el potencial mineral de una región o sitio geológico.

Utiliza reglas de producción y redes semánticas para estructurar organizar el dominio; inferencias y búsquedas por encadenamiento hacia atrás (“backward Chiang”); para el manejo de incertidumbre, utiliza una teoría subjetiva de probabilidades basada en una forma del teorema de Bayes, llamada “odds-likelihood”, junto con factores de certeza de las inferencias de las reglas (igual que en MYCIN [Shortliffe, 1976]), el cual combina varias fuentes de información inconclusa para formar una conclusión que puede ser casi siempre cierta.

En PROSPECTOR, las probabilidades se dan subjetivamente por parte de los expertos humanos. Las evidencias son independientes de la hipótesis; las creencias y no creencias corresponden a los conceptos intuitivos de confirmación y de no-confirmación. PROSPECTOR realiza consultas para determinar:

Qué modelo es el que más se apega a los datos recolectados.

Donde se encuentran los sitios más favorables para la perforación.

Qué datos adicionales son los requeridos para obtener conclusiones más contundentes.

Cuál es la base de las conclusiones y/o recomendaciones que emite.

PROSPECTOR compara observaciones con modelos computacionales de depósitos de minerales (minas).

PROSPECTOR registra similitudes, diferencias e información faltante.

PROSPECTOR pregunta por información adicional si es necesario.

PROSPECTOR realiza la COMUNICACIÓN CON LOS USUARIOS (Expertos y Geólogos) bajo los siguientes lineamientos:

Escoge una hipótesis sobre la existencia de una mina y le realiza preguntas al usuario con el fin de probar o rechazar las hipótesis.

Vuelve a formular sus preguntas cuando el usuario le dice que no las entendió.

Le puede decir al usuario porque requiere la respuesta a una pregunta específica.

Presenta sus conclusiones acompañadas de un resumen de las razones por las cuales llevo a ellas.

Permite interactuar con el usuario en términos que le son familiares a los expertos (les habla en su lenguaje).

Le permite que al usuario agregar información sobre la región en cualquier momento durante una consulta.

Cambia las hipótesis (desecha unas y toma nuevas) en base a las respuestas dadas por los usuarios.

Busca inconsistencias en las respuestas proporcionadas por el usuario, las señalas y permite que se modifiquen las respuestas (nuevas y viejas).

El sistema PROSPECTOR contiene reglas que ligan una evidencia observada ‘E’ referente a un descubrimiento específico particular (geológico) con la hipótesis ‘H’ implicada por dicha evidencia:

If E then H (to degree) LS, LN;

LS y LN son valores previamente almacenados (con rangos de +5 a -5) y que no cambian durante la ejecución del programa. Existen también, para cada par de evidencia (E1, E2, E3, ...) e hipótesis (H1, H2,...) una probabilidad asignada (P1, P2, ...) y que si puede cambiar durante la ejecución del programa, en base al teorema de Bayes.

Sistema TMS, ATMS

Truth Maintenance System (TMS) o sistema de mantenimiento de veracidad es un método de representación del conocimiento utilizado para representar tanto la certidumbre como de lo que esta depende. En sistemas de un solo contexto, la consistencia es mantenida entre todos los hechos registrados en memoria (base de datos). Los sistemas multi contexto permiten que la consistencia sea relevante a un

subconjunto de hechos etiquetando cada hecho ocurrido en memoria junto con su historia lógica.

Los TMS o Sistemas de mantenimiento de veracidad multi-agente realizan un mantenimiento de la veracidad por medio de múltiples memorias, frecuentemente localizadas en diferentes maquinas. El ATMS de “de Kleer” (1986) fue utilizado en sistemas basados en KEE en la maquina de Lisp. El primer TMS multi-agente fue creado por Mason y Johnson. Eran un sistema multi-contexto.

Sistema JTP [Gleb Frank (OO)]

La arquitectura JTP es una arquitectura modular orientada a objetos (OO) para razonamiento híbrido desarrollada por Gleb Frank.

La arquitectura JTP esta organizada en dos niveles de abstracción. En la capa más abstracta, se hacen muy pocos compromisos para con el lenguaje de representación y la naturaleza del contenido con el que el sistema esta trabajando. La idea es dar al implementador máxima flexibilidad al crear sistemas razonadores con esta arquitectura al mismo tiempo de proveerle de una estructura y de un vocabulario común para facilitar la integración y la interacción de módulos heterogéneos.

En el más alto nivel de abstracción, la arquitectura JTP es independiente de la representación del lenguaje. Para el propósito de este texto, se asume que la representación del lenguaje es un lenguaje de símbolos lógicos y que el sistema de razonamiento JTP Implementado utiliza un lenguaje de representación lógica de primer orden (First Order Logic FOL).

HERRAMIENTAS DE USO LIBRE:

HAY que checar las ligas.....

CLIPS

CLIPS 6.0 (C Language Integrated Production System) is an OPS-like forward chaining production system written in ANSI C by NASA. The CLIPS inference engine includes truth maintenance, dynamic rule addition, and customizable conflict resolution strategies. CLIPS, including the runtime version, is easily embeddable in other applications. CLIPS includes an object-oriented language called COOL (CLIPS Object-Oriented Language) which is directly integrated with the inference engine.

The Software Technology Branch's home page is

http://www.jsc.nasa.gov/stb/STB_homepage.html

NASA Information Services's home page is

http://hypatia.gsfc.nasa.gov/NASA_homepage.html

and the CLIPS home page is <http://www.jsc.nasa.gov/~clips/CLIPS.html>

[Note: Folks who obtain CLIPS from COSMIC can do anything they wish with it, including redistribute it. Folks who obtain CLIPS directly from NASA are restricted to using it for US government purposes only.]

The CLIPS FAQ list and bug fixes are available from hubble.jsc.nasa.gov/pub/clips/ and are maintained by Gary Riley <riley@gothamcity.jsc.nasa.gov>.

Joseph Giarratano and Gary Riley's book, "Expert Systems Principles and Practice", comes with an MS-DOS CLIPS 6.0 interpreter (see [1-3] above).

Basic Programming Guide CLIPS 6.04:

FuzzyClips (en cap 5 que se llama Fuzzy Expert Systems, leer para hacer un resumen).

FuzzyCLIPS 6.02 is a version of the CLIPS rule-based expert system shell with extensions for representing and manipulating fuzzy facts and rules. In addition to the CLIPS functionality, FuzzyCLIPS can deal with exact, fuzzy (or inexact), and combined reasoning, allowing fuzzy and normal terms to be freely mixed in the rules and facts of an expert system. The system uses two basic inexact concepts, fuzziness and uncertainty. Versions are available for UNIX systems, Macintosh systems and PC systems. There is no cost for the software, but please read the terms for use in the FuzzyCLIPS documentation. FuzzyCLIPS is available via WWW (World Wide Web).

Start a version of FuzzyCLIPS.

Load the shower example (load .shwrNOUI.clp. from the fuzzy examples directory)

Run an example [(reset) and (run)] The program will ask for the values of parameters for temperature, pressure and valve positions. It stops when it reaches a flow between 11 and 13 L/min and water temperature between 34 and 38.C. The values of certain parameters will be printed after each set of fuzzy rules and defuzzification has taken place.

You will be prompted to enter further values or to quit.

BABYLON

BABYLON is a development environment for expert systems. It includes frames, constraints, a prolog-like logic formalism, and a description language for diagnostic applications. It is implemented in Common Lisp and has been ported to a wide range of hardware platforms.

OPS5 (Carnegie Mellon University)

De pagina:

KEE

KEE, ProKappa, and Kappa are expert system development packages that run on personal computers, workstations, and Lisp machines. Features include ATM, rule-based reasoning, and OOP support.

EXYS Professional

EXSYS Professional runs under MS-DOS, MS-Windows, Macintosh, SunOS, Solaris, Unix and Vax. It supports backward and forward chaining, linear programming, fuzzy logic, neural networks, and has a SQL interface.

For more information please visit their web page at

<http://www.exsysinfo.com/> ;

For examples of EXSYS applications running over the web try their Web Runtime Engine demonstrations at

<http://www.exsysinfo.com/Wren/wren.html>

NEXPERT OBJECT runs on over 30 platforms supported including personal computers (\$5000), Macintosh (\$5000), workstations (\$12,000), minicomputers, and mainframes. Nexpert Object is written in C, and includes a graphical user interface, knowledge acquisition tools, and forms system. For more information, write to Neuron Data, 156

University Avenue, Palo Alto, CA 94301, call 800-876-4900

XpertRule

XpertRule for Windows represents knowledge as decision trees, tables of decision examples, exception trees and sets of pattern rules. It can produce C code. Fuzzy Logic and Genetic Algorithm optimization are included. For more information, write to Attar Software USA, PO Box 68, Harvard, MA 01451-0068, call 800-456-3966, or fax 508-456-8383. European customers should write to Attar Software Limited, Newlands

OPS86

Ver ref: Libro muy viejo.....

<http://books.google.com.mx/books?id=I6SUWg0DAgsC&pg=PA251&lpg=PA251&dq=ops86&source=...>

Comments

Sergio Marcellin:

Más información sobre herramientas para SBKs:

Una herramienta/lenguaje especial:

<http://www.visual-prolog.com>

En efecto con Drools se pueden definir y manejar casos basados en reglas tipo IF THEN, así como manejar eventos y resolver problemas de optimización,.....
Los elementos que lo conforman son:

JBoss BRMS is made up from a number of community

projects - Drools Expert, Fusion, jBPM and Guvnor.

ver esta liga:

<https://www.jboss.org/drools/>

<http://docs.jboss.org/drools/release/pamphlet.pdf>

Este producto (herramienta o plataforma si es de generación actual, no como clips) y además integrada con JBoss de Redhat.

Yo en lo personal no la he utilizado, pero suena muy bien para integrar problemas como el de las redes de comunicación.

Tampoco he utilizado gensym G2, pero se ve bien para aplicaciones con necesidades robustas y en tiempo real (para los jugadores de fut ?). En el caso de la planta química no describen nada sólo que les ayudo a optimizar procesos.....

Lo bueno es que estamos enriqueciendo la BDD de los esqueletos para hacer SBKs.

Con G2: Fix Faults Faster

Drastically reduce the time it takes to respond to faults in your network

Smart Service

Diagnose the root cause of any problem to ensure better service right away.

[2020](#)

[Jboss](#)

<https://www.optaplanner.org/>

<http://www.gensym.com/en/telecommunications>

Todo lo relacionado con IA: Reglas, procesos, eventos, agentes, planeación, ontologías y más:

<http://blog.athico.com/>

Redes bayesianas TUTORIALES:

I've been working through this excellent tutorial:

<http://www.mathcs.emory.edu/~whalen/Papers/BNs/Intros/BayesianNetworksTutorial.pdf>

And this tutorial too:

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.135.7921&rep=rep1&type=pdf>

Ejemplos del universo de posibilidades de los ejemplos de vendedor, nube y asignación de salones para clases.

Caso del uso del map reduce para resolver problemas de optimización. Malisimo!!!

G2 es mencionado en libro:

http://wps.prenhall.com/wps/media/objects/3778/3869053/Turban_Online_Chapter_W18.pdf