

SISTEMA PARA LA GESTIÓN Y CONTROL DE SOLICITUDES EN UN TALLER DE ELECTRÓNICA MEDIANTE INTELIGENCIA ARTIFICIAL

Manuel Castro
Estudiante de Ingeniería de Sistemas, UBA
alejocastro2@gmail.com

Resumen

El objetivo de esta investigación fue diseñar un sistema experto basado en reglas que utiliza el método de encadenamiento hacia delante de la inferencia para servir de apoyo en las tareas de diagnóstico de fallas en electrodomésticos en talleres de electrónica. El aspecto más llamativo de este modelo es que el motor de inferencia se basa en una representación simple de reglas y hechos en tablas de bases de datos relacionales. Las reglas se descomponen y se representan en tablas a dos niveles, lo que permite desarrollo de sistemas expertos en cualquier lenguaje de programación que soporte SQL. Entre las conclusiones destaca la ventaja de este modelo en centrar el diseño de los sistemas expertos basados en reglas hacia la representación del conocimiento en una base de datos relacional, reduciendo el esfuerzo y las dificultades de programación.

Palabras clave: sistema experto basado en reglas, encadenamiento hacia delante, motor de inferencia.

SYSTEM FOR THE MANAGEMENT AND CONTROL OF REQUESTS IN AN ELECTRONICS WORKSHOP THROUGH ARTIFICIAL INTELLIGENCE

Summary

The objective of this research was to design a rule-based expert system that uses the forward chaining method of inference to support troubleshooting tasks in electrical appliances in electronic workshops. The most striking aspect of this model is that the inference engine is based on a simple representation of rules and facts in relational database tables. The rules are decomposed and represented in tables at two levels, which allows the development of expert systems in any programming language that supports SQL. Among the conclusions, it stands out the advantage of this model in focusing the design of expert systems based on rules towards the representation of knowledge in a relational database, reducing the effort and programming difficulties.

Keywords: rule-based expert system, forward chaining, inference engine.

Introducción

En los últimos años se evidencia un vertiginoso crecimiento continuo en el uso de las tecnologías de la información y las comunicaciones, particularmente con Internet como medio de publicidad y de gestión de trámites administrativos. Gracias a esa alta demanda, las empresas procuran mantenerse a la vanguardia de la tecnología asegurando su espacio dentro de la web, para promover sus productos y servicios frente a millones de internautas a cualquier hora y en cualquier parte del mundo, sin importar el tamaño de la organización. De allí que, uno de los retos que afrontan en la actualidad, es la necesidad de modernizar las gestiones para brindar mejores servicios y por consiguiente incrementar sus ingresos y mantenerse a la par de la competencia.

El objetivo de este artículo de investigación es exponer el desarrollo de un modelo para diseño de sistemas expertos basados en reglas que utilizan el método de encadenamiento hacia adelante de la inferencia. El aspecto más importante de este modelo es que el motor de inferencia se basa en una simple representación de reglas y hechos en tablas de la base de datos. Las reglas se descomponen y se representan en tablas a dos niveles, lo que da la posibilidad de desarrollar sistemas expertos en cualquier lenguaje de programación que soporte SQL.

Sistemas expertos

Un sistema experto es un sistema de información basado en el conocimiento que utiliza ese mismo conocimiento sobre un área de aplicación específica y compleja para actuar como consultor experto para los usuarios finales (O'Brien, 2001). El razonamiento simbólico de un sistema experto le permite sacar conclusiones a partir de premisas y proporcionar explicaciones. La tecnología de los sistemas se basa en el conocimiento del problema que se aborda y para el dominio del problema, define los objetos, las propiedades

tareas y eventos en los que trabaja un experto humano y la heurística que los profesionales experimentados han aprendido a utilizar para obtener mejores resultados.

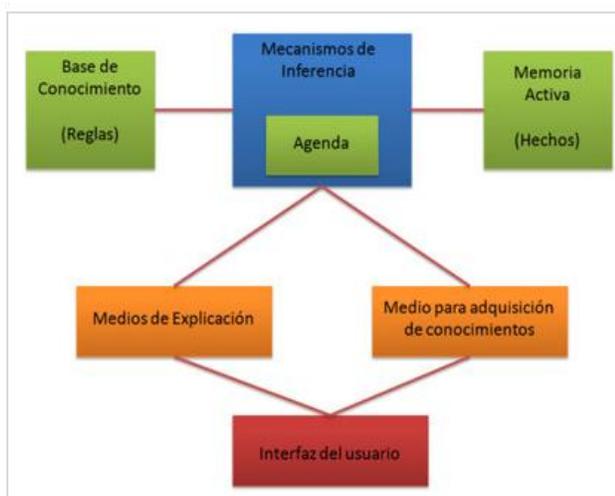
Ahora bien, los componentes de un sistema experto son: base de conocimientos y motor de inferencia. Además, puede haber una interfaz de usuario y una función de explicación (O'Brien, op.cit.). La base de conocimientos contiene datos sobre un tema específico y heurísticos que expresan los procedimientos de razonamiento de un experto. El motor de inferencia procesa las reglas y hechos relacionados con un problema específico. La interfaz de usuario permite al usuario interactuar con los sistemas expertos. La función de explicación permite al sistema explicar su razonamiento al usuario.

Existen dos (2) métodos para realizar inferencias: (a) encadenamiento hacia delante (llegar a una conclusión aplicando reglas a los hechos) y (b) encadenamiento hacia atrás (justificar una conclusión propuesta determinando si resulta de la aplicación de las reglas a los hechos). Los sistemas expertos se han construido tradicionalmente utilizando lenguajes de programación como LISP PROLOG, C, C++ y Java, o con herramientas de desarrollo como los *shells* de sistemas expertos, como *Clips* (Giarratano y Riley, 2004). Con esas herramientas de software, la mayoría de los sistemas expertos se han desarrollado teniendo en cuenta el conjunto de instrucciones y soporte de estructuras de datos.

Metodología

La programación basada en reglas es una de las técnicas más utilizadas para el desarrollo de sistemas expertos. El Sistema Experto Basado en Reglas consiste en un conjunto de pautas que pueden aplicarse repetidamente a una colección de hechos (ver figura 1):

Figura 1: Sistema Experto Basado en Reglas



Fuente: Universidad de Sevilla. Dpto. Ciencias de la Computación e Inteligencia Artificial. Disponible: <http://www.cs.us.es/~fsancho/?e=103>

Los siguientes conceptos son esenciales para los sistemas basados en reglas (Clancey, 1981): (a) los hechos representan circunstancias que describen una situación en el mundo real y (b) las reglas representan la heurística que define un conjunto de acciones que deben ejecutarse en una situación determinada.

Las reglas de producción tienen el patrón si <condición> entonces <acción>. Consisten en producir consecuencias prácticas a partir de ciertas condiciones. Las reglas se componen de una parte "si" y una parte "entonces". La parte "si" de una regla, el lado izquierdo (LHS), se llama predicado o premisas. El LHS consiste en una expresión que puede ser una sola expresión (un hecho individual hecho que debe ser verdadero para aplicar la regla) o una = serie de expresiones (expresión compuesta).

En la literatura de los lenguajes basados en reglas, una sola expresión se suele denominar patrón. Una expresión compuesta consiste en varias expresiones simples conectadas entre sí mediante el uso de los elementos condicionales elementos condicionales "y, o, no" para crear reglas complejas.

Los motores de inferencia consisten en todos los procedimientos que manejan la base de conocimientos para llegar a una conclusión (Nilsson, 1998). Los motores de inferencia suelen estar codificados en lenguajes como LISP, PROLOG, C, C++, Java.

Los principales elementos de un sistema basado en reglas son los hechos, las reglas y el motor que actúa sobre ellos. El núcleo de la arquitectura que se muestra en la figura 1, consiste en los tres (3) siguientes elementos:

1. Memoria de trabajo: contiene hechos que son la información más pequeña que soporta el motor de motor de reglas.

2. Base de reglas: contiene reglas en forma de sentencias *if then*, que representan el conocimiento proporcionado por el usuario y/o un experto del dominio del problema.

3. Motor de inferencia: compara los hechos de la memoria de trabajo con las reglas de la base de reglas y determina qué reglas son aplicables según el método de razonamiento adoptado por el motor.

Principios del sistema experto

En un sistema experto basado en reglas, las reglas se ejecutan con lógica de programación y datos de la tabla regla + datos de las otras tablas.

1. Una prueba es exitosa si detecta la falla. En ese caso se envía un mensaje y se finaliza el proceso.

2. Una prueba es fallida si no detecta falla. En ese caso se envía un mensaje, se pregunta al usuario si desea continuar la prueba. Finaliza, si el usuario responde: "No". Si responde: "Si" y hay más pruebas no ejecutadas para este equipo y avería, se continua, de lo contrario se finaliza.

Lógica básica

1. El usuario selecciona un equipo.
2. El sistema muestra la lista de averías.

3. El sistema muestra la lista de prueba para la avería seleccionada.
Pregunta: "ejecutar pruebas o salir".

4. Si el usuario selecciona "ejecutar pruebas", el sistema:

4.1. Muestra la prueba actual y el usuario debe escoger: "prueba exitosa" o "prueba fallida".

4.2. El sistema busca en la tabla reglas la acción a ejecuta. En el campo "solución exitosa" se encuentra el id de la solución cuando la prueba es exitosa, y en "solución fallida", cuando la prueba ha fallado.

Reglas de producción

El conocimiento en un sistema experto puede ser representado con un sistema de producción (Lewis 2003), un enfoque sencillo (Sadik, 2008). Un sistema de producción consta de tres (3) elementos: conjunto de reglas de producción, memoria de trabajo y un intérprete. La estructura de las reglas de producción puede ser formalmente como sigue: si <condición> entonces <conclusión> o si <condición> entonces <acción> Las producciones tienen dos partes Precondición sensorial (parte "SI") Acción (parte "THEN") Cuando el estado del "mundo" coincide con la parte "IF", la producción se dispara, lo que significa que la acción se ejecuta. Así pues, las reglas de producción vinculan los hechos (partes "IF", también llamadas antecedentes) a conclusiones (partes "THEN", también llamadas consecuentes).

Resultados

Los sistemas expertos basados en reglas, utilizan el encadenamiento hacia delante o hacia atrás como método de inferencia para llegar a una conclusión. Un modelo sencillo de encadenamiento hacia adelante es el siguiente:

Si A y B o C, entonces X
Si D o E, entonces Y
Si X e Y, entonces Z

El modelo propuesto (*Gabing*) representa las reglas a las que se aplicará el método de encadenamiento hacia adelante, según los aspectos de representación y de descomposición. Las reglas se representan según el siguiente esquema:

1. Cada regla está formada por elementos relacionados por conectores lógicos (y, o).
2. Cada regla está representada por una fila en una tabla relacional.
3. Una regla formada por *n* elementos conectados por 'o' está representada por *n* filas en una tabla relacional, cada fila contiene un solo elemento.
4. Una regla formada por *n* elementos conectados por 'y' se presenta mediante *n* filas en una tabla relacional cada una de las cuales contiene un único elemento y un campo de conexión con un valor 'AND'.
5. Los elementos de una regla se seleccionan de una tabla relacional de "atributos".
6. Cada fila de la tabla de atributos representa un atributo asociado a una regla.
7. Las reglas se aplican a los sujetos. El sujeto que cumple al menos una regla, es seleccionado e incluido en las conclusiones.
8. Un sujeto representa objetos reales o abstractos.
9. Los atributos asociados a un sujeto se representados en una tabla.
10. Los tipos de datos utilizados en la regla son: *varchar* y *float*.

Descomposición

La descomposición es una técnica mediante la cual se puede dividir una regla en trozos simples. El conjunto de piezas es equivalente a la regla original. El procedimiento de descomposición es el siguiente:

1. Una regla se representa por un conjunto de piezas de reglas.

2. Una regla y el conjunto equivalente de piezas de regla tienen la misma identificación, es decir, un campo que identifica la regla.

3. Cada pieza de regla contiene sólo y sólo una expresión lógica, más un conector lógico.

4. Los valores permitidos del conector lógico conector lógico son: 'AND' y 'NULL'.

5. Un conector lógico "AND" en la regla original se asigna a "AND" en la pieza de la regla.

6. Un conector lógico "OR" en la regla original se asigna a una nueva pieza de regla.

Discusión

Con el fin de ilustrar la aplicación del Modelo de Gabing, a continuación, se presenta un caso hipotético donde puede aplicarse; se trata de un taller de reparaciones electrónicas que necesita un mecanismo de apoyo para poder ayudar a los administrativos a diagnosticar fallas en los artefactos en ausencia de algún técnico. El objetivo es diagnosticar posibles fallas de equipos en el taller. Seguidamente se detalla la estructura generada con la aplicación del Modelo de Gabing (tablas 1 a 6):

Tabla 1

Equipo

Campo	Tipo	Longitud	Comentario
equipoid	int	8	Clave primaria. Autonumérico
equiponombre	varchar	150	Nombre del equipo

Fuente: Castro (2021)

Tabla 2

Problema

Campo	Tipo	Longitud	Comentario
problemaid	int	8	Clave primaria. Auto numérico
equipoid	int	8	Id del equipo con problema
problemanombre	varchar	120	Nombre del problema

Fuente: Castro (2021)

Tabla 3**Averia**

Campo	Tipo	Longitud	Comentario
averiaId	int	8	Clave primaria. Auto numérico
problemaId	int	8	Id del problema relacionado
averiaNombre	varchar	120	Nombre de la averia

Fuente: Castro (2021)

Tabla 4**Prueba**

Campo	Tipo	Longitud	Comentario
pruebaId	int	10	Clave primaria. Auto numérico
averiaId	int	8	Id de la averia relacionada
pruebaNombre	varchar	100	Nombre de la prueba

Fuente: Castro (2021)

Tabla 5**Solucion**

Campo	Tipo	Longitud	Comentario
solucionId	int	10	Clave primaria. Auto numérico
solucionNombre	varchar	150	Nombre de la solución
solucionEncontrada	int	1	0= solución no encontrada 1= solución encontrada
finalizar	int	1	0: No 1: Si

Fuente: Castro (2021)

Tabla 6**Regla**

Campo	Tipo	Longitud	Comentario
reglaId	int	10	Clave primaria. Auto numérico
reglaNombre	varchar	150	Nombre de la regla
pruebaId	int	10	Id de la prueba relacionada
solucionExitosa	int	10	solucionId si prueba es exitosa
solucionFallida	int	10	solucionId si prueba es fallida

Fuente: Castro (2021)

Como ejemplo de datos de la aplicación del Modelo de Gabing, con el método de encadenamiento hacia adelante, según los aspectos de representación y de descomposición aplicado para detectar fallas de equipos en un taller de reparaciones electrónicas, se exponen en forma sucesiva, en las tablas 7 a 12:

Tabla 7
Equipo

equipoid	Equipo Nombre
1	Plancha eléctrica
2	Horno de microondas
3	Lavadora automática
4	Lavadora semi-automática

Fuente: Castro (2021)

Tabla 8
Problema

problemaid	equipoid	Problema Nombre
1	1	La plancha no caliente

Fuente: Castro (2021)

Tabla 9
Averia

averiaId	problemaid	averiaNombre
1	1	Falla del Cable
2	1	Problema térmico
3	1	Falla de la resistencia
4	1	Falla del switch selector

Fuente: Castro (2021)

Tabla 10
Prueba

pruebaId	averiaId	pruebaNombre
1	1	Medir continuidad
2	2	Medir continuidad
3	3	Medir continuidad
4	4	Medir continuidad

Fuente: Castro (2021)

Tabla 11
Solucion

solucionId	solucionNombre	solucionEncontrada	finalizar
1	Reemplazar o reparar	1	1
2	Prueba del cable exitosa. No se detecta falla	0	0
3	Reemplazar	1	1
4	Prueba térmica exitosa. No se detecta falla	0	0
5	Reemplazar	1	1
6	Prueba de resistencia exitosa. No se detecta falla	0	0
7	Mantenimiento o reemplazar	1	1
8	Prueba del switch selector exitosa. No se detecta falla	0	0

Fuente: Castro (2021)

Tabla 12**Regla**

reglald	reglaNombre	pruebald	solucionexitosa	solucionfallida
1	Prueba averia cable: no mide corriente	1	1	0
2	Prueba averia cable: si mide corriente	1	0	2
3	Prueba problema térmico: no mide corriente	2	3	0
4	Prueba problema térmico: si mide corriente	2	0	4
5	Prueba problema de resistencia: no mide corriente	3	5	0
6	Prueba problema de resistencia: si mide corriente	3	0	6
7	Prueba problema de switch selector: no mide corriente	4	7	0
8	Prueba problema de switch: si mide corriente	4	-	08

Fuente: Castro (2021)

Conclusiones

Los sistemas basados en reglas emulan la experiencia humana en dominios de problemas bien definidos, utilizando una base de conocimientos expresada en términos de reglas. En este trabajo se demostró la aplicación del modelo Gabling para diseñar sistemas expertos basados en reglas, con un enfoque que representa las reglas en tablas relacionales, con descomposición en dos niveles. Este enfoque desplaza la carga del algoritmo a la base de conocimiento, con el objetivo de facilitar el diseño y la programación de sistemas expertos.

Referencias

Carrillo, G. (2010). **A Model for Designing Rule-Based Expert Systems.**

Disponible en:

<http://revistas.unellez.edu.ve/index.php/apuroquia/article/view/79/74>

Castro, M. (2022). **Arquitectura General de un Sistema Experto Basado en Reglas.** Disponible en:

<https://sites.google.com/site/sistemasexpertosunah/home/sistemas-expertos-basados-en-reglas>

- Clancey, W. (1981). **The Epistemology of a Rule-based Expert System: A Framework for Explanation.** Disponible: <ftp://reports.stanford.edu/pub/cstr/reports/cs/tr/81/896/CS-TR-81-896.pdf>.
- Giarratano, J. y Riley, G. (2004). **Expert Systems Principles and Programming.** Boston: Fourth, Course Technology.
- Lewis, P. (2003). **Knowledge Representation. Production Rules for Knowledge Representation.** Disponible en: <http://users.ecs.soton.ac.uk/phl/ctit/ho1/node2.html>.
- Nilsson, N. (1998). **Artificial Intelligence: A New Synthesis.** San Francisco: Morgan Kaufmann Publishers
- O'Brien, J. (2001). **Sistemas de Información Gerencial.** México: MacGraw-Hill.
- Sadik, A. (2008). International journal of soft computing. **Medwell Journal.** Disponible: <https://www.medwelljournals.com/>
- Segura, R. (2010). **Reparación Básica de Electrodomésticos.** Disponible en: <https://oei.int/>