

Instituto Tecnológico de Costa Rica
Sede Regional San Carlos

Escuela de Computación

Lenguajes de programación

Proyecto 3

Responsables:
Crisly González Sánchez

04 de junio del 2016

Santa Clara, San Carlos

Introducción

La programación lógica no es tan utilizada como la programación orientados e incluso la funcional, sin embargo en el desarrollo de software debemos considerar cuál paradigma es mejor en función de resolver nuestra problemática y simplificar el trabajo de los desarrolladores.

Prolog es conocido porque en él se desarrollan generalmente sistemas expertos, dicho lenguaje pertenece al paradigma lógico y cuenta con predicados y reglas que juntos asociados pueden brindar una solución.

En la programación lógica, el programador proporciona la parte lógica y el lenguaje de programación proporciona el control. La implementación de la programación lógica en las áreas en que más sentido tiene su aplicación es: inteligencia artificial, sistemas expertos, procesamiento de lenguajes, etc.

Prolog es un lenguaje de programación simple, pero poderoso. Se basa en nociones matemáticas de relaciones de inferencia. Es un lenguaje declarativo e interpretado, esto quiere decir que el lenguaje se usa para representar conocimientos sobre un determinado dominio y las relaciones entre objetos de ese dominio.

El sistema experto realizado en prolog se le llama sistema experto de detección de fallas de computadoras, en el cual mediante un sistema de preguntas se filtrará la información hasta llegar al problema final y su posterior recomendación.

Entre algunas ventajas del uso del paradigma lógico están: una ventaja desde el punto de vista del usuario es la facilidad para programar ya que se pueden escribir programas rápidamente, con pocos errores originando programas claramente legibles, aun si no se conoce muy bien el lenguaje.

Otra ventaja de este tipo de lenguajes frente a otros, como Java o C++, es que no hay que pensar demasiado en la solución del problema, ya que Prolog infiere sus respuestas basándose en las reglas declaradas dentro del programa

Análisis del problema

Se debe analizar el enunciado del problema para tomar la decisión de sus requerimientos basados en el caso, se considera que es requerido una conexión con un motor de base de datos ejemplo: MySql Server, Postgres, MySql, quedará a criterio del desarrollador cual será el que se implementará, además deberá analizarse la estructura de la base de datos ya que esto puede complicar el desarrollo del sistema experto en lenguaje lógico Prolog. Se requiere evaluar los conocimientos en el paradigma lógico por lo tanto la base datos es considera un factor mínimo en el desarrollo del software.

Es sugerido una arquitectura en la base datos relacional y que cuente con un tabla, donde los atributos serán las preguntas o secciones y la información en dicha columna será el resultado de lo requisitos que debe cumplir mínimamente un fallo para que sea problema de tarjeta madre, tarjeta Ram o tarjeta de Video. Cabe destacar que la implementación de más de una tabla en la base de datos y sus posteriores relaciones implicaría consultas más complejas para obtener un dato cada menos de 1 min, agrupado los datos en una misma tabla podemos solicitarlos a la DB (Data Base)y obtenerlos todos en una sola consulta.

Una vez que el sistema obtiene los datos de motor de base de datos, deberá almacenarse en variables que serán conocidas dentro de la lógica del sistema, posteriormente crear los hechos que requieran para diagnosticar la problemática, cabe destacar que si la tabla de la DB tiene apartado de Disco Duro, SO o Estado de la máquina, deberán existir 3 hechos con los resultados de estado de la máquina (encendida / apagada) y a partir de ahí construir las reglas necesarias.

Cabe destacar que deberá implementarse una solución simple de entender ya que Prolog trabaja por el algoritmo backtracking el cual toma decisión final basada en el resultado anterior, por cual si tenemos dos métodos y uno retorna False y el segundo se cumple con un True, el resultado será False, por lo tanto se recomienda orden.

Solución del problema

Se implementó un sistema experto con el motor de base de datos MySQL en el cual fue requerido investigar acerca de la conexión mediante a la **librería ODBC** del sistema operativo windows, una vez conectado mediante la librería y la configuración de la variable de entorno del sistema, se utilizó el IDE **SWI-Prolog** en el cual se desarrolló el sistema experto. Se evaluó las recomendaciones en el planteamiento del problema ya que tiempo una arquitectura simple era lo más recomendable. Por lo tanto la base de datos **bdpreguntas** cuenta con una tabla única llamada datos, que a su vez tiene atributos de Conexión, Alerta, SO, Mensaje, Reinicio, Descripción , Tipo y Diagnóstico.

Se estableció primeramente la conexión con la base de datos y luego una consulta mediante la sintaxis del motor Mysql y la librería ODBC, se obtuvieron los datos de según el orden en la tabla del primer row, para posteriormente ser evaluado y conocer el problema del usuario.

Con los datos retornados de la consulta se almacenan en variables que luego serían el contenido del hecho / predicado que se iba construyendo, conforme se iban creando tenían reglas asociadas que se cumplieran si y solo si la respuesta del usuario era positiva. y así hasta diagnosticar el problema que tenía el usuario.

En caso de los hechos no se cumpliera por una respuesta de usuario, la regla retornaba negativa y hacía el hecho falso, por lo tanto se devolvía y hacía el siguiente row de la tabla y así hasta hacer la búsqueda exhaustiva. El predicado know conocía todas las respuestas a hechos evaluados por lo tanto antes de evaluar la misma pregunta buscaba en su motor de conocimientos e inteligentemente la saltaba porque ya sabía su respuesta.

Cabe destacar que la librería utilizada era reconocida por el IDE SWI-Prolog, sin embargo al utilizarlo a nivel de compilador en otro editor de texto no es reconocida, porque si se desea

implementar en un proyecto fuera de el IDE SWI-Prolog se debería evaluar la metodología y librerías que permiten dichas conexiones.

Se intentó realizar la conexión de prolog con una aplicación web pero surgieron inconvenientes de arquitectura de los sistemas que no permitían su conexión fácilmente, PHP era el lenguaje que se utilizó para cubrir el requerimiento solicitado por el profesor de conexión web, sin embargo al utilizar el compilador de prolog con un editor de texto diferente a SWI-Prolog (que está capacitado para interpretar las acciones en Prolog) la conexión falló ya que no reconoció la librería ODBC.

Además al ser PHP un lenguaje del lado del servidor la interacción con el cliente (interfaz) no era posible sin utilizar alguna metodología de conexión diferente a solo conectar el archivo de javascript con PHP.

Se logra mostrar una interfaz interactiva con el cliente utilizado únicamente la consola del IDE SWI-Prolog, mostrando preguntas para analizar de manera exhaustiva hasta llegar a la solución. Dado así cubiertos los requerimientos que se solicitaron en el enunciado del proyecto.

Análisis de resultados

Tarea	Estado	Observaciones
Prolog y Base de datos		
Creación de predicados y reglas	100%	
Relación de predicados y reglas	100%	
Conexión con base de datos MySql	100%	
Consultas a la base de datos en motor MySql	100%	
Elaboración de un sistema de preguntas	100%	
Elaboración de base de datos en MySql	100%	
Elaboración de tablas de la base de datos	100%	
Análisis del problema para identificarlo y brindar una solución.	100%	
Elaboración de sistema de menú de interfaz de usuario	100%	
Implementación del sistema con preguntas, solución y diagnóstico	100%	
Conexión de Prolog / Php y interfaz de usuario.	80%	La conexión se puede realizar siempre y cuando no se utilice la librería de ODBC ya que no es reconocida por el compilador, por lo tanto su funcionalidad existe con métodos de prolog que retornen datos string.

Conclusiones

Los Sistemas Expertos incluyen componentes del sistema en sí e interfaces con individuos con varios roles. Los componentes más importantes son: Base de conocimientos, la representación declarativa de la experiencia, muchas veces en forma de reglas IF-THEN, los datos específicos al problema que se está resolviendo. y el código central del SE que deriva recomendaciones con base en la base de conocimientos y los datos específicos del problema.

El sistema experto de detección de fallas desarrollado realiza búsqueda exhaustiva mediante respuestas a las preguntas de 3 casos diferentes hasta llegar a la solución y brindar una recomendación.

Recomendaciones

Es recomendable utilizar otra librería diferente a ODBC ya que al querer implementar programación web debe utilizar editores de texto con compiladores y archivos enlazados, por lo tanto al usar SWI-Prolog se limita el desarrollo para otras tecnologías al ser un IDE dedicado a Prolog únicamente.

La documentación requerida para comprender el problema por lo tanto es recomendable crear una pequeña documentación dentro del sistema programado.

La interfaz es simple pero dinámica con el cliente, sin embargo se podría mejorar y hacer una mejor visualización de los datos.

El tiempo estimado para el proyecto no permitió un análisis más exhaustivo acerca de las arquitecturas utilizadas y cuál sería la mejor opción.

Se requiere conocimientos previos en programación lógica para que el código o los cambios puedan ser realizados con éxito.

Bibliografía

Juan María Hernández. (26 agosto 2013). ¿Que es la Programación Lógica?. 29 de mayo del 2016, de koalite Sitio web: <http://blog.koalite.com/2013/08/que-es-la-programacion-logica/>

SR. (SR). Programación lógica. 29 de mayo del 2016, de EcuRed Sitio web: http://www.ecured.cu/Programaci%C3%B3n_l%C3%B3gica

Kathryn M. Jones Pérez & Gustavo A. Yong Morales . (2007). Introducción al lenguaje de programación lógica Prolog . 29 de mayo del 2016, de Universidad de Costa Rica, escuela de Ciencias de la Computación e Informática Sitio web: <http://www.di-mare.com/adolfo/cursos/2007-2/pp-Prolog.pdf>

SR. (2011). SISTEMAS EXPERTOS. 29 de mayo del 2016, de Mpi Sitio web:
<http://www.uv.mx/aguerra/documents/2011-mpi-09.pdf>