RISI 7(2), 2010 (43-48)
Revista de Investigación de Sistemas e Informática
Facultad de Ingeniería de Sistemas e Informática
Universidad Nacional Mayor de San Marcos

ISSN 1815-0268 (versión impresa)
ISSN 1816-3823 (versión electrónica)

Sistemas de razonamiento basado en casos aplicado a sistemas de líneas de productos software

System case based reasogning applied to system software product lines

Augusto Cortez Vásquez, Carlos Navarro Depaz, Jaime Pariona Quispe

Universidad Nacional Mayor de San Marcos
Facultad de Ingeniería de Sistemas e Informática

cortez_augusto@yahoo.fr, cnavarrod@unmsm.edu.pe, jparionaq@unmsm.edu.pe

RESUMEN

La construcción de sistemas inteligentes simula de algún modo la manera en que los seres humanos resuelven problemas. Los Sistemas de razonamiento basado en casos (SRBC) parten de un problema ya resuelto (caso) alojado en una librería de casos. Estas tareas son las que usualmente realiza en la vida cotidiana un profesor, un servidor público, etc. El problema es recuperar un caso almacenado parecido al problema que se intenta resolver, y para ello se presenta una metodología. La reutilización de software es una de las aproximaciones más efectivas de los SRBC, y de ello trata el presente artículo. Se trata de aprender de ejemplos, casos o datos conocidos de tal forma que se tome decisión sobre nuevos casos. Desde el enfoque de minería de datos, en tareas de clasificación, podemos asignar una clase a un nuevo caso observando las clases similares. Igualmente, en tareas de agrupamiento, asignaremos un nuevo ejemplo al grupo donde estén los individuos mas similares. En el caso de regresión, el valor predicho para un nuevo ejemplo no puede distar mucho de los valores obtenidos para ejemplos similares. Las líneas de producto software generalmente surgen a partir de aplicaciones existentes. Una organización tiene una librería de aplicaciones (casos) y cuando se requiere una nueva aplicación, se utiliza la primera como base para la nueva aplicación. Se propone la utilización de Sistemas de razonamiento basados en casos para desarrollo de líneas de software.

Palabras clave: ingeniería de conocimiento, sistemas inteligente, sistema de razonamiento basado en casos, reutilización de software

ABSTRACT

Building intelligent systems, somehow simulate the way humans solve problems. Systems based reasoning (SRBC), part of a problem already solved (case) housed in a case library. These tasks are usually performed in everyday life a teacher, public servant, etc.. The problem is to retrieve a stored case similar to the problem they are trying to solve, and it presents a methodology. Software reuse is one of the most effective approaches SRBC, and it is this article. It's about learning from examples, cases or known data so as to take decision on new cases . From the data mining approach in classification tasks, we can assign a class to a new case, noting similar classes. Similarly, grouping tasks, assign a new example to the group where individuals are more similar. In the case of regression, the predicted value for a new example can not be far from the values obtained for similar examples. Software product lines generally arise from existing applications. An organization has a library of applications (cases) and when it requires a new application, the former is used as the basis for the new application. It is proposed razonamientobasados Systems utiliacion cases for software development lines

Keywords: knowledge engineering, intelligent systems, system of case-based reasoning, reuse of software

1. INTRODUCCIÓN

Dentro de Inteligencia artificial existe una disciplina denominada Ingeniería de conocimiento (IC) que proporciona los métodos y técnicas para construir sistemas computacionales denominados Sistemas Basados en Conocimiento (SBC), estos sistemas se diferencian de otros en su manejo de grandes volúmenes de conocimiento de un dominio. Un Sistema de razonamiento basado en casos SRBC es básicamente un modelo de razonamiento que permite resolver problema, entender situaciones y aprender. Un médico que diagnostica a un paciente porque recordó que otro paciente presentaba los mismos síntomas, está usando razonamiento basado en casos. Un abogado que apela a precedentes legales para defender alguna causa está usando razonamiento basado en casos. También un programador que reutiliza librerías de software está tratando a esta como una "base de datos de soluciones". El razonamiento basado en casos es una manera de razonar haciendo analogías [1]. Por otro lado, desde hace ya varias décadas, la gente del mundo de la ingeniería del software comenzó a imaginar un futuro donde solo se avanzaría. No habría que escribir programas para problemas ya resueltos.

2. FUNDAMENTACIÓN TEÓRICA

Un SRBC es básicamente un tipo de sistema experto; por tanto, debemos definir qué es un sistema experto.

Un sistema experto, desde el punto de la inteligencia artificial, es un sistema que intenta imitar el comportamiento de un ser humano experto en alguna temática, es decir, imitan las actividades de un ser humano para intentar resolver los problemas de distinta índole.

Existen básicamente tres tipos de sistemas expertos:

Sistemas de razonamiento basados en reglas SRBR: son sistemas expertos que utilizan para el proceso de inferencia un conjunto de reglas que constituyen la base de conocimiento del experto. Este conjunto de reglas pueden ser activadas a medida que las condiciones son evaluadas positivamente y su utilización implica la creación de nuevos hechos. Este proceso permitirá, a partir de unos hechos iniciales, desarrollar un proceso deductivo que concluirá en el momento en que no quede ninguna otra regla por utilizar.

Sistemas de razonamientos bayesianos SRB: N sistemas que basan su funcionamiento como su nombre lo indica, en las redes bayesianas. Así, pues, se trata de un modelo probabilístico que relaciona un conjunto de variables aleatorias mediante un grafo dirigido. El motor de inferencia que utiliza para procesar las evidencias se basa en la teoría de probabilidades y más concretamente con el Teorema de Bayes. Este método es especialmente una herramienta extremadamente útil en la estimación de probabilidades ante nuevas evidencias [2].

Sistemas de razonamientos basado en casos SRBC: Modelo de razonamiento que permite resolver problemas, entender situaciones y aprender utilizando mecanismos de memorización, problemas superpuestos y criterios de optimalidad.

Los SRBC se sustentan en tres principios básicos:

a. **Solución de problemas superpuestos:** se aplica en casos que utiliza casos resueltos menores.

b. **Principio de optimalidad de Bellman:** memoriza la mejor solución, luego de un proceso de selección.

c. **Memorización:** memoriza las soluciones obtenidas en la librería de casos para uso posterior.

Los SRBC presenta algunas ventajas frente a los sistemas tradicionales [3]:

Adquisición de conocimiento: La adquisición del conocimiento se realiza a partir de la experiencia previas almacenada en la librería de casos (LC). Esto evita el proceso de extracción de reglas a partir de un experto en dominio, lo cual ya constituía de por si en un "cuello de botella" y además no se garantizaba la validez de las reglas.

Mantenimiento del conocimiento: Los SRBC permite incrementar nuevos casos a la LC sin la intervención del experto, haciendo innecesario el proceso de mantenimiento de la base de conocimiento en SRBR que resulta costoso.

Eficiencia en la resolución de problemas: la reutilización es un principio básico de la informática. Los SRBC permite que se puedan resolver casos similares sin tener que rehacer la base de conocimientos.

Calidad de la solución: Al aplicar el principio de optimalidad, se garantiza memorizar la mejor solución o lo que ha sucedido en un contexto determinado.

Aceptación del usuario: Utilizar soluciones basados en casos que ya han sido utilizados y probados da confianza y aceptación al usuario, lo que no ocurre en soluciones como las redes neuronales y los SRBR, ya que pueden resultar incomprensibles para los usuarios.

Justificación de los SRBC

Los SRBC se utilizan en situaciones en las que se cumple:

1. Dominios difíciles de estructurar, por tanto es difícil de establecer un modelo del dominio.

2. Situaciones en donde resulta difícil explicar la solución a un problema.

Restricciones de los SRBC

1. El dominio de aplicación de los casos debe ser regular, es decir, no debe ser cambiante. Lo que es cierto hoy, debe serlo también mañana.

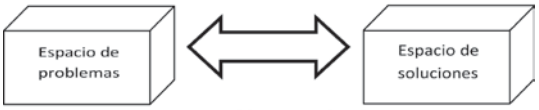
2. Los problemas deben ser recurrentes, es decir deben ocurrir regularmente, de no ser así, no será necesario memorizar un caso.

Los métodos SRBC deben procesar el conjunto de ejemplos existentes para poder hacer comparaciones con los nuevos casos y los casos pasados. De hecho existen tres maneras para procesar estos ejemplos:

- Memorización de todos los ejemplos (causa retardo en los métodos.

- Memorización de parte (seleccionando ejemplos significativos).

- Creación de prototipos (representantes ficticios de conjuntos de datos).



Caso = Problema x Solución

3. MATERIALES Y MÉTODOS

La solución de un SRBC requiere los siguientes pasos:

- Se crea una estructura para almacenar los problemas.
- Se crea una estructura para almacenar las soluciones.
- Se diseña un solución para resolver el caso. La solución consta de dos fases:
  - Recuperación de casos
  - Selección del caso mas similar al problema

El ciclo del razonamiento basado en casos

En el proceso de dividir el razonamiento basado en casos en diferentes subprocesos nos encontramos el ciclo que lo conforman puede ser dividido en 4 procesos claramente diferenciados:

Descripción de casos

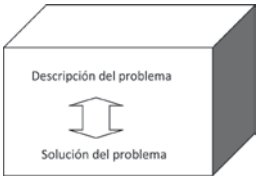
El caso es el elemento principal de un SRBC

Librería de casos (LC). Es una estructura que permite organizar los casos de situaciones en forma estructurada. La organización deberá permitir: primero la recuperación de un subconjunto de casos que puedan ser aplicados al problema planteado, y luego aplicar medidas de similitud para seleccionar de entre conjunto de casos, el que esté más próximo al problema planteado.

La realización más sencilla de la LC es mediante una memoria plana (Lista o arreglo), aunque puede implementarse también mediante memoria jerárquica (grafo o árboles).

Caso 1	At^{11}, At^{12},… A $t^{1}n$
Caso 2	At^{21}, At^{22},… A $t^{2}n$
Caso n	Atn^{1}, Atn^{2},… A tnn

Un caso debe contener:



4. RESULTADOS

Un problema está compuesto por un conjunto de atributos que determinan los valores de las diferentes características que definen el problema. Los problemas que describen los casos están compuestos por un conjunto finito de atributos, quedando descrito un problema por

los valores que toman dichos atributos, así:

Problema = (At1, At2,… A tn)

At_i puede ser cuantitativo o cualitativo

At_i: número de créditos

At_i: domicilio del alumno

Caso alumno = problema alumno x solución alumno

Problema alumno = (código, nombre, número repitencias)

Solución alumno = (condición alumno)

Caso 1 : = ((92M123, Juan Pérez, 0),(alumno regular))

Caso 2 : = ((92M123, Juan Pérez, 5),(alumno observado))

Algoritmo propuesto para un SRBC

Acción SRBC()

Inicio

RECORDAR los casos similares al que analizamos.

REUTILIZAR la información y el conocimiento que tenemos en este caso para resolver el problema.

REVISAR la solución propuesta.

RETENER las partes de esta experiencia que nos puedan ser útiles para la resolución de futuros problemas.

Fin

El primer paso que tenemos que realizar al enfrentarnos a un problema es recordar los casos relevantes que pueden solucionarlo. Este es quizás en principal problema, ya que se tendrá que seleccionar los casos relevantes en la base de conocimiento y recuperarlos. Luego de recuperados, el conjunto de casos que guardan una serie de similitudes con el caso para el cual tenemos que proponer una solución lo que tenemos que hacer es adaptar la solución de todos esos problemas, en su globalidad o solamente en alguna de sus partes que nos interese para transformar el contexto de esos problemas en el problema que tenemos actualmente.

Con todo esto tendríamos una primera versión de la solución. Sin embargo, es necesario probar la solución en el mundo real o en una simulación y si es necesario revisarla. Con esto lo que se quiere decir es que este es un proceso circular en el que reutiliza diversos casos de la base de conocimiento, se revisa la solución y si no es satisfactoria se vuelve a modificar con la inclusión o la eliminación de los casos que fuesen incorrectos o añadiendo aquellos que faltasen para perfeccionar la solución. Finalmente, el último paso es la retención. Después de que la solución haya sido adaptada satisfactoriamente para solucionar el problema dado, almacenaríamos la experiencia resultante como un nuevo caso en la LC.

Medidas de distancia

Es importante formalizar el concepto de similitud es a través de métricas o medidas de distancia. Si se quiere saber la similitud entre dos objetos, es necesario definir una función de distancia y calcular con ella la distancia entre los dos objetos.

Funciones de distancia

Distancia Euclidea: es la distancia clásica, como la longitud de la recta que une dos puntos en el espacio euclideo

$$D(x,y) = \text{Raíz}\left(\sum_{i=1}^{n} (xi-yi)^2\right)$$

Las medidas de distancia más tradicionales son:

- Distancia de Manhattan
- Distancia ded Chebychev
- Distancia del Coseno
- Distancia de Mahalanobis

Cuando se trata de buscar casos similares surgen dos nociones muy importantes: que se entiende por similitud o distancia, y segundo, cuando se va a explotar

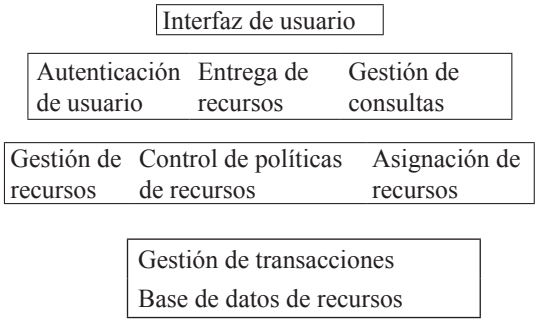


Figura N.° 1. Arquitectura de un sistema de asignación de recursos.

Interfaz de usuario	interfaz de comunicaciones del sistema

Autenticación del operador	planificación de mapas y rutas	generador de informes	Gestor de consultas

Gestor de estados de vehículos	registro de incidencias	despachador de vehículos	Gestor de equipamiento	Localizador de vehículos

Base de datos De equipos	Gestión de transacciones Base de datos de vehículos	Base de datos mapas

Figura N.° 2. Arquitectura de líneas de productos de un sistema de gestión de vehículos

Figura N.° 3. Desarrollo de una instancia de un producto.

dicha similitud. En muchos casos conviene la no creación de un modelo general, sino que en el camino se predice el modelo en función de cada caso concreto. Es por eso que muchos de estos modelos reciben nombres diferentes según el contexto: aprendizaje basado en instancias o en casos, métodos retardados o perezosos, métodos basados en similitud, etc.

Caso práctico

Las líneas de productos software son casos especializados de las arquitecturas de aplicaciones para un tipo concreto de aplicación. Consideremos un caso práctico: Un sistema de líneas de productos diseñados para gestionar el uso de un vehículo para servicios de emergencia de algún rubro. Los operadores de este sistema a menudo recogen llamadas sobre incidentes ocurridos, encuentran el vehículo adecuado para responder al incidente y envían el vehículo al lugar del incidente. Los desarrolladores de un sistema pueden, a partir de una arquitectura base, adaptar versiones de dicho sistema para servicios de ambulancia, de bomberos y de policía, servicios de pedidos.

El sistema de gestión de vehículos es un ejemplo de un sistema de gestión de recursos cuya arquitectura de las aplicaciones se muestra en la Figura 1

La Figura 3 grafica los pasos comprendidos en la adaptación de una familia de aplicaciones para crear una nueva aplicación.

A partir de los nuevos requerimientos, se busca en la librería de casos aquel que más se adecue, luego de seleccionado el más similar, se adapta o modifica para satisfacer los requerimientos, y cuando esté finalizado se entrega la solución y se almacena como un nuevo miembro de la familia

Beneficios de la reutilización

Puede decirse que la reutilización es la única aproximación realista para llegar a los índices de productividad y calidad que la industria del software necesita.

- Mejora de la productividad:
- Disminución tiempo de desarrollo:
  - Mejor adaptación requisitos cambiantes
  - Los requisitos no son estables
- Disminución de costos
- Mejora de la calidad del software
- Mayor fiabilidad
- Mayor eficiencia

Dificultades de la reutilización

- En muchas empresas no existe plan de reutilización (no se considera prioritario)
- Escasa formación
- Resistencia del personal

- Pobre soporte metodológico
- Uso de métodos que no promueven la reutilización (estructurados)

5. CONCLUSIONES

Los SRBC se basan en los principios de problemas superpuestos, memorización y el principio de optimalidad. Sin embargo, se justifica su utilización solo en casos en que el problema sea recurrente. Los SRBC son de fácil compresión por parte del experto, ya que el sistema no maneja conceptos abstractos, sino situaciones concretas (casos) del dominio conocido por el experto. Por otro lado, los sistemas incrementan su efectividad con el tiempo, por cuanto se nutre de nuevos casos. La ventaja de la reutilización de software son la reducción de costos, rapidez en el desarrollo de software y reducción de riesgos.

Este trabajo puede ser un punto de partida para introducir el SRBC como ayuda a los desarrolladores de sistemas y contribuir así a lograr software con mayor calidad aprovechando la experiencia acumulada, de manera que se puedan construir soluciones cada vez más complejas y con la rapidez que exige el creciente ritmo de la tecnología de la información.

Los patrones de diseño son abstracciones de alto nivel que documentan soluciones de diseño exitosas. No es de extrañar que las líneas de productos software surgen a partir de aplicaciones existentes. Es decir, una organización desarrolla una aplicación y, cuando se requiere una nueva aplicación, se utiliza la primera como base para la nueva aplicación. En adelante, las demandas de nuevas aplicaciones hacen que el proceso continua. Sin embargo, debido a que los cambios tienden a degradar la estructura de la aplicación, en algún momento debe tomarse alguna decisión específica para diseñar una línea de productos genérica. El diseño debe basarse en la reutilización de conocimiento adquirido a partir del desarrollo del conjunto inicial de aplicaciones

6. REFERENCIAS BIBLIOGRÁFICAS

[1] Palma Méndez José. Inteligencia Artificial. McGraw-Hill, España 2008.

[2] Anderson James. “Redes neuronales” Edit. Alfaomega, México 2007.

[3] Hernández Orallo, José. Introducción a la minería de datos. Edit. Prentice Hall, España, 2004.

[4] Del Brio Bonifacio. Redes neuronales y sistemas borrosos. Edit. Alfaomega, México, 2007.

[5] Delgado Dapena Martha Dunia. Utilización del razonamiento basado en casos en las revisiones de la definición del modelo de negocio.