

ANÁLISIS CRÍTICO DEL RAZONAMIENTO BASADO EN CASOS COMO UN MÉTODO DE SOLUCIÓN DE PROBLEMAS DE DIAGNÓSTICO

Lic. María M. García Lorenzo
Dr. Rafael Bello Pérez
Lic. Hipólito Iroel Pérez García*

Resumen

La solución de tareas de diagnóstico es una de las áreas donde mayor aplicación han tenido los sistemas basados en el conocimiento, siendo particularmente los sistemas basados en reglas los más populares. Sin embargo, la aparición del razonamiento basado en casos ha motivado el desarrollo de numerosos sistemas basados en casos para tareas de diagnóstico.

Con este trabajo presentamos el razonamiento basado en casos como un método de solución de problemas y su aplicabilidad en tareas de diagnóstico y en tal sentido analizamos la factibilidad del Sistema Inteligente de Selección de Información (shell que implementa el razonamiento basado en casos) implementado por nosotros en tareas de diagnóstico.

1. Introducción

Los sistemas basados en el conocimiento son un modelo computacional de más alto nivel que el paradigma de la programación procedimental en el cual los sistemas están

formados por tres componentes básicas: la base de conocimientos, la máquina de inferencia y la interfase. En la base de conocimientos se almacena el conocimiento necesario para resolver los problemas del dominio de aplicación atendiendo a alguna forma de representación del conocimiento (FRC), y la máquina de inferencia es un procedimiento basado en un esquema de razonamiento o método de solución de problemas (MSP) que utiliza el conocimiento para resolver los problemas de ese dominio. Uno de los campos donde mayor empleo han tenido los sistemas basados en el conocimiento son los problemas de diagnóstico [1].

El tipo de sistema basado en el conocimiento más popular es el Sistema Basado en Reglas [8], en el cual la FRC usada es la regla de producción y como MSP se emplean la búsqueda dirigida por datos o por objetivos. En los sistemas basados en reglas es necesario desarrollar un complejo y prolongado proceso de extracción del conocimiento desde diferentes fuentes de conocimientos (expertos humanos, casos resueltos, publicaciones, etc.) y de formalización de éste en forma de reglas de producción.

* Departamento de Ciencia de la Computación. Universidad Central de Las Villas. Santa Clara, Villa Clara, CUBA

En busca de simplificar la complejidad de este proceso de adquisición del conocimiento se han desarrollado otros tipos de sistemas basados en el conocimiento, siendo uno de ellos los Sistemas Basados en Casos [3 y 7]. En este tipo de sistema la base de conocimientos es esencialmente una colección de casos del dominio de aplicación mientras que como procedimiento de solución de problemas se utiliza el Razonamiento Basado en Casos (RBC).

Como se plantea en [2] el RBC es un mecanismo de razonamiento utilizado para resolver problemas por medio de analogías. La analogía es una potente herramienta de inferencia. Según Carbonell [12], la solución de problemas por analogía consiste en transferir el conocimiento sobre la solución de problemas pasados a los nuevos que comparten aspectos significativos con las experiencias pasadas, y usar el conocimiento transferido para construir las soluciones de éstos. La solución de problemas por analogías y el RBC son dos lados de la misma moneda [12]. El RBC se distingue por trabajar dentro de un solo dominio, recuperar casos específicos útiles, y tratar en lo posible de buscar una correspondencia lo más exacta posible entre el problema a resolver y el caso.

En este trabajo se analiza el uso del RBC como un método de solución de problemas de diagnóstico, relacionándose varios de los sistemas basados en casos desarrollados en esta área.

2. El razonamiento basado en casos como un método de solución de problemas.

El término razonamiento se refiere a diferentes clases de actividad, entre ellas extraer conclusiones desde un conjunto de hechos, diagnosticar posibles causas para

una situación, etc. Razonar incluye realizar inferencias pequeñas organizadas en función de un objetivo, aunque ambos términos se emplean con frecuencia indistintamente. Los esquemas de razonamiento se clasifican atendiendo a su precisión, nivel y generalidad.

Un sistema de razonamiento tiene varias componentes básicas, entre ellas el esquema o forma de representación del conocimiento, un conjunto de reglas de inferencia y algunos medios para controlar la forma en que el sistema aplica las reglas de inferencia para buscar una solución. El RBC representa una nueva técnica de razonamiento pues en éste, el formalismo que se utiliza sirve para representar no un conocimiento explícito, sino ejemplos de problemas del dominio de aplicación, estando en este sentido la base de casos más cerca de las bases de datos que de las bases de conocimiento. Las reglas de inferencia se emplean para construir la solución del nuevo problema a partir de las soluciones dadas a problemas resueltos previamente y los mecanismos de control están orientados a mejorar la búsqueda y recuperación de casos semejantes al problema por resolver.

Entre las características del RBC como un esquema de razonamiento están:

- a) En el RBC el sistema razona a nivel del dominio del problema.
- b) El grado de generalidad de los sistemas con RBC se determina por el modelo de memoria (modelo de la base de casos) y el criterio de recuperación de los casos. La generalidad está limitada porque los casos de dominios diferentes no sólo tienen rasgos diferentes sino porque las características de los rasgos son también diferentes. Por otra parte, generalmente

es necesario construir funciones de semejanza específicas al problema para poder alcanzar una buena medida.

- c) El RBC permite definir relaciones causales entre las propiedades de los objetos. Sin embargo, esta relación se limita al marco de un caso y no es posible establecer secuencias causales entre casos.
- d) Desde el punto de vista del grado de precisión con que se obtienen los resultados, el RBC ofrece una medida numérica del grado de certidumbre de la veracidad de la solución hallada. Este nivel de certidumbre es la medida de semejanza calculada entre la descripción del problema a resolver y el caso recuperado.

Otro aspecto importante para valorar al RBC como un esquema de razonamiento es que si atendemos a los criterios expresados por Stephen Slade [11] y otros autores respecto a que la capacidad para resolver problemas descansa en los episodios pasados de modo que la experiencia guía el razonamiento, y que en la mente del hombre no puede aparecer un pensamiento cuyos elementos no se encuentren registrados en la memoria, el RBC es el esquema de razonamiento que más próximo está de este comportamiento.

La estructura de los sistemas que emplean el RBC se corresponde con la visión que se tiene del razonamiento humano. La base de casos sería el equivalente de la memoria del hombre; ésta es predominantemente episódica siendo su contenido primario la experiencia, está ricamente indexada y es dinámica, lo cual se corresponde con los casos, el modelo de organización de la base de casos y las posibilidades de reordenamiento de la base de casos cuando esta crece (por ejemplo cuando se utilizan estructuras del tipo MOP [8]).

El mecanismo de recuperación de casos semejantes es el homólogo de la capacidad de búsqueda asociativa del hombre, y se reconoce que esta es una de las cualidades principales de la inteligencia humana. Aquellas personas que son consideradas «muy inteligentes» lo que hacen es procesar la información muy profundamente, es decir, poseen múltiples rutas en el cerebro que le permiten hacer la recolección de datos mucho más accesible y relacionar información que para otros parecen no asociadas [4].

3. Los problemas de diagnóstico y el razonamiento basado en casos.

El diagnóstico es la tarea de identificar las causas del mal funcionamiento de algún dispositivo, y se puede formular como «dado un conjunto de síntomas y la descripción de un dispositivo, encontrar una explicación a esos síntomas».

Existen diferentes enfoques para resolver problemas de diagnóstico usando distintos tipos de conocimiento y mecanismos de inferencia. En el enfoque asociativo se usa un conocimiento que enlaza los síntomas con fallas en una manera directa. En el enfoque basado en modelos se explota un modelo del equipo bajo consideración a partir del cual se pueden derivar las expectativas. En los sistemas basados en reglas éstas relacionan de modo causal síntomas y explicaciones. Cuando la cantidad de explicaciones posibles es pequeña se puede tratar el diagnóstico como un problema de clasificación, y en este caso se pueden emplear modelos de clasificación como los usados en reconocimiento de patrones o redes neuronales artificiales; pero si el conjunto de explicaciones no puede ser fácilmente enumerado se puede ver el diagnóstico como el problema de crear una explicación.

En [1] al analizar la problemática del diagnóstico se ve esta tarea como consistente de tres subtareas: detección de los síntomas, generación de hipótesis y discriminación de las hipótesis. Los síntomas se identifican a partir de las observaciones y las expectativas del modelo existente, ellos dirigen la generación de las hipótesis, y finalmente éstas son discriminadas considerando información adicional; lo anterior da como resultado un conjunto de hipótesis consistente (con respecto a las observaciones), el cual se puede aceptar como un diagnóstico o puede ser refinado aún más.

Generar un diagnóstico desde un punto inicial es una tarea que consume tiempo. Sin embargo, en casi todos los dominios en el que se usa el diagnóstico existe la regularidad suficiente como para emplear un enfoque basado en casos para generar un diagnóstico de probada eficiencia. Por supuesto, que éste no puede asumirse como la respuesta definitiva, debe ser validada, pero la validación es mucho más fácil que la generación del diagnóstico a partir del planteo del problema.

En el diagnóstico basado en casos, se usan los casos para proponer las explicaciones a los síntomas o alertar sobre explicaciones que han sido inapropiadas en el pasado. Los casos previos permiten generar una hipótesis sobre el diagnóstico del nuevo problema. Esta hipótesis ofrece un razonamiento rápido y permite evitar errores anteriores, además, orienta la atención sobre rasgos del problema que podrían no haberse considerado si se utiliza otro tipo de razonamiento. Los sistemas de diagnóstico basados en casos, descansan en la hipótesis de que si los síntomas de dos problemas son similares, su diagnóstico será semejante.

Para las tareas de diagnóstico son aplicables tanto los sistemas basados en casos interpretativos como los solucionadores de problema, aunque según nuestro criterio éste es uno de los tipos de tareas donde los primeros resultan más adecuados. En el modo interpretativo los síntomas o rasgos que indican el mal funcionamiento observado sugieren, a priori o por algún mecanismo de inferencia simple, una causa de la falla, con cuya información se realiza la recuperación de los casos semejantes que aportaría los argumentos o la justificación para afirmar o rechazar la causa supuesta. La consideración a priori de dicha causa es similar a lo que sucede en los sistemas basados en reglas cuando se utiliza una búsqueda dirigida por objetivo, pues en este caso la causa supuesta es el objetivo a probar y el sistema tratará de encontrar la secuencia de reglas que relaciona los síntomas con el objetivo establecido.

Se han desarrollado numerosos sistemas de diagnóstico basados en casos. Ejemplos de ellos son los relacionados en [3 y 7]:

SHRINK (Kolodner, 1987), diseñado para el diagnóstico en psiquiatría.

CASEY (Koton, 1988), aplicado al diagnóstico de problemas del corazón.

PROTOS (Bareiss, 1989), sistema para el diagnóstico de problemas auditivos, en esta área los problemas que se presentan tienen rasgos muy similares y solamente sutiles diferencias permiten diferenciarlos.

A continuación se presentan otros sistemas basados en casos dirigidos a resolver problemas de diagnóstico.

El sistema BOLERO [5], es uno en el cual se combina la planificación y el diagnóstico, en

donde sus autores ven la tarea de diagnóstico como un proceso incremental en el cual paso a paso se obtienen nuevas evidencias que guían la búsqueda en los pasos siguientes. Aparecen integrados un razonador basado en casos y un sistema basado en reglas; el razonador basado en casos actúa como el planificador del sistema basado en reglas, el cual posee un escaso conocimiento sobre la estrategia de búsqueda a seguir.

El sistema basado en casos construye los planes de acuerdo con el estado de la memoria de trabajo del sistema basado en reglas mientras la máquina de inferencia de este último ejecuta esos planes.

Este modelo es muy adecuado al diagnóstico médico, pues se caracteriza por utilizar información incompleta e imprecisa, por lo que se hace necesario poder alterar dinámicamente el plan de búsqueda trazado. La base de casos del sistema es la memoria de planes.

El sistema FRANK [9], es un sistema híbrido que integra casos, reglas y procedimientos de planificación para generar un diagnóstico médico.

El sistema basado en casos CASCADE [10] está orientado a facilitar la respuesta a los clientes que reportan fallas en el sistema operativo VMS de la Digital Equipment Corporation. En este sistema cada caso contiene rasgos superficiales que ofrecen información sobre la falla y el ambiente en que ocurrió la misma, rasgos derivados relevantes y punteros a las funciones que permitieron derivarlos y la acción de reparación que resuelve el problema y se utiliza un algoritmo de recuperación bidireccional el cual a partir de los síntomas o rasgos generales de la falla selecciona un

conjunto de casos y basado en el contenido de estos casos se obtienen nuevos rasgos que permiten recuperar los casos más apropiados; sólo cuando todos los rasgos superficiales y derivados se corresponden con los del problema se selecciona el caso.

Otro ejemplo de sistema basado en casos para problemas de diagnóstico en sistemas de computación es el Case-Based Help Desk System [6]. El desarrollo del sistema partió del estudio de varios cientos de casos recopilados por el grupo responsable de dar asistencia a las PC de una región. Cada caso consistía de un número para la identificación del problema, una descripción inicial del problema y el análisis y recomendación del operador. El sistema cuenta con un procesador de textos el cual identifica la información que resulta de interés para la búsqueda y un sistema híbrido de recuperación de casos que realiza la selección de los casos en dos etapas; en la primera se utiliza un árbol de decisión que selecciona un conjunto de casos y luego al aplicar una función de semejanza se recuperaran los casos mejores.

4. Sistema Inteligente de Selección de Información (SISI)

Este sistema es el resultado de nuestro trabajo, con el mismo se logra implementar el razonamiento basado en casos orientado a tareas de diagnóstico. Los casos en **SISI** se toman de artículos de ficheros de bases de datos relacionales. Para realizar el razonamiento se utiliza una red neuronal la cual se construye a partir del conocimiento de la base de casos y se emplea para agilizar este proceso. Los pesos calculados por el algoritmo de aprendizaje de la red se utilizan en la función de semejanza para la recuperación de los casos más parecidos al problema planteado al sistema. Toda esta

estructura ha sido programada utilizando el paradigma orientado a objetos sobre el lenguaje de programación C++.

4.1. Fundamentos teóricos del sistema.

En **SISI** un caso está descrito por un conjunto de rasgos que es común para todos los casos. Un rasgo puede tener uno o más valores por ejemplo: el rasgo SÍNTOMAS de un paciente en ocasiones tiene valor "Dolor de cabeza", pero casi siempre los SÍNTOMAS de un paciente que asiste a una consulta son más de uno. Por otra parte existen rasgos que por su naturaleza no pueden ser multievaluados, por ejemplo la EDAD, nadie tiene a la vez 21 y 22 años.

En **SISI** un caso es un artículo de un fichero de bases de datos (BD) y el usuario del sistema debe determinar cuáles de los campos de la BD constituirán rasgos de interés en el trabajo.

Para ilustrar el razonamiento basado en casos en **SISI** veamos el siguiente ejemplo. Supóngase que se tiene una BD relacional que contiene información acerca de los pacientes que han asistido a una consulta médica. De cada paciente se conocen los siguientes rasgos: NOMBRE, N° IDENTIDAD, EDAD, SEXO, SÍNTOMAS, ENFERMEDAD y TRATAMIENTO correspondiéndose éstos con los campos de la base de datos. Sea ésta nuestra base de casos. Cuando llega a la consulta un nuevo paciente del cual se conocen los cuatro primeros rasgos (rasgos predictores) se desea conocer qué ENFERMEDAD presenta y el TRATAMIENTO a seguir (rasgos objetivos), entonces a partir de los casos de la BC, debemos valorar los rasgos que quedaron vacíos en la solicitud reciente. Este proceso consiste en completar dichos rasgos a través de una búsqueda de los casos en la BC completando la solicitud

al encontrar el caso (o los casos) más semejantes al caso planteado.

El proceso de búsqueda para integrar la solicitud hecha por el usuario del sistema debe determinar los casos más semejantes a la solicitud.

Una forma de implementar el módulo de búsqueda y recuperación es utilizando una función que defina la semejanza entre el caso a solucionar y los casos de la BC.

El algoritmo de completamiento del caso debe inferir valores para los rasgos objetivos a partir de los valores asignados a los rasgos predictores. Para realizar esta inferencia es necesario utilizar la información almacenada en la BC lo cual implicaría una revisión doble de la misma para cada demanda. Una revisión para recuperar la información necesaria e incorporar en la función de semejanza criterios sobre la importancia de los rasgos y la equivalencia de valores, y otra para calcular la medida semejanza entre el caso a resolver y los casos de la BC .

Es obvio que para BC con muchos casos, lo cual es muy probable, resultaría ineficiente la doble pasada. Sería importante realizar la inferencia de los valores de los rasgos objetivos sin tener que recorrer la base de datos para cada demanda, es decir que se ha obtenido un algoritmo que permite extraer el conocimiento necesario para el completamiento del caso a partir de la BC y almacenarlo de modo que cada vez que se desee completar un caso se resuelva y solamente con una pasada se encuentran los casos que justifiquen la solución.

Para resolver este problema se proponen dos procedimientos principales: el primero extrae y almacena el conocimiento desde la BC, este procedimiento quizás sea un poco

lento, pero debe ser ejecutado sólo una vez en el sistema mientras el usuario no cambie la BC, el segundo es el que se encarga de completar el caso para cada demanda sin usar el conocimiento de la BC sino el conocimiento extraído en el paso anterior.

El conocimiento se almacena en una red neuronal para luego completar casos con éste. El primer procedimiento define la topología de la red y a la vez la entrena mientras que el segundo la usa para inferir los rasgos objetivos.

La red que se utiliza es un modelo similar al de Activación Interactiva y Competencia (AIC). Este modelo forma varios grupos de neuronas conectadas por enlaces pesados con las neuronas de otros grupos, la neuronas de un mismo grupo tiene peso cero en su enlace. Para cada uno de los elementos del dominio de cada rasgo se sitúa una neurona. En la matriz W se sitúan los pesos de todas la conexiones.

Además de completar el caso se debe brindar una explicación como justificación de la solución a partir del hecho de que **SISI** es un SBC interpretativo. La argumentación de la solución en **SISI** consiste en presentar los N casos más parecidos al caso planteado al sistema.

El sistema para argumentar la explicación le informa al usuario cuáles de los rasgos

Menú # 1:

File Help New About Open Index Exit

Menú # 2:

File Expert Help New Trait Selection About Open Domain restriction
Index Exit Alias Learn

Menú # 3:

File Define Expert Result Help New Target Case Inter Case Case
About Open Similar Cases Count Find Rel. Trait Relevant Traits Index Save
Find Sim. Cases Similar Cases Save as Exit

valorados fueron los que más influencia tuvieron en la solución obtenida por el sistema.

4.2 El uso del sistema

SISI es un programa que se ejecuta sobre el sistema operativo Microsoft **Windows**. El trabajo con este programa no se diferencia del trabajo con otras aplicaciones comunes de **Windows**. La forma de ejecutarlo es presionando sobre el icono del sistema o utilizando la opción "Run" del menú principal de **Windows** escribiendo **SISI**, que es el nombre con que se activa dicho programa. El fichero ejecutable "sisi.exe" consta de 222 kbytes y se puede almacenar en un disco normal sin necesidad de ningún otro fichero para su ejecución.

La aplicación consta de un menú principal donde aparecen las opciones que en cada momento el usuario puede seleccionar. Para lograr una interacción más amena el sistema cambia el conjunto de opciones del menú en lugar de abigarrarnos con un montón de opciones que se activan y se desactivan en dependencia del estado del sistema. En esta versión de **SISI** se tienen tres estados del menú principal que se presentan dependiendo de lo analizado anteriormente:

Descripción del Menú # 1 del sistema

Submenú File:

New: Esta opción se utiliza para crear un nuevo experto o sea una nueva sección de trabajo. El sistema muestra un diálogo en cual se capta el título que tendrá el nuevo experto y el fichero de bases de datos que servirá como base de casos. No está permitido salir con "Ok" hasta tanto no se escriban los nombres del experto y el de la base de datos. Para escribir el nombre de la base de datos más fácilmente seleccione el botón "Browse" (Listar) y se le presenta el diálogo estándar de abrir un fichero. Seleccione en éste el fichero .DBF donde se almacena su base de datos.

Open: Esta opción es para cargar una sección de trabajo hecha con anterioridad . Para ello el sistema le muestra el diálogo estándar de apertura de fichero de **Windows**. Seleccione el fichero .SEF de la sección de trabajo que usted desea cargar. En caso de que el fichero esté dañado o sea que no guarde la información válida de una sección de trabajo, usted será amonestado con un mensaje de error. De concluir con éxito se activa el tercer estado del menú principal.

Exit: Esta opción aborta la ejecución del programa, pero antes le pregunta si desea salvar la sección de trabajo que con que usted está trabajando.

Submenú Help:

Este submenú es común para todos los menús del sistema.

About: Muestra los créditos del sistema.

Index: Muestra la ayuda del sistema. Esta opción no se puede utilizar puesto que el fichero de ayuda no se ha escrito.

Descripción del menú # 2 del sistema

Submenú File: Este submenú se mantiene igual al del menú anterior.

Submenú Expert: Este submenú brinda las opciones necesarias para construir el experto basado en casos que opera sobre la base de datos que usted indicó en la opción "New".

Traits Selection: Esta opción le muestra al usuario un diálogo en el cual usando el mouse o la barra espaciadora usted debe seleccionar cuáles de los campos de la base de datos serán los rasgos de la base de casos. Seleccione con el mouse sobre el botón "Select all" si está interesado en seleccionar todos los campos de la base de datos. Si usted conoce los rasgos a desechar y éstos son pocos, usted puede marcarlos y luego seleccione el botón "Invert" así se quedará con la selección de los campos que le interesan.

Domain restriction: Esta opción muestra un diálogo con los rasgos numéricos dentro de los rasgos seleccionados en la opción anterior, si nunca se entró a ésta, toma los rasgos numéricos de los campos de toda la base de datos. El botón "Restrict" indica un diálogo en el que aparecen todos los valores del rasgo que estaba seleccionado en la lista. Usted debe marcar los valores que desea eliminar del dominio del rasgo seleccionado. Concluya esta opción seleccionado el botón "Close".

Alias: En esta opción se presenta un diálogo para seleccionar el rasgo que se le desea cambiar el nombre y luego seleccione el botón "Alias", entonces aparecerá la caja de diálogo donde se edita el nuevo nombre.

Learn: Esta opción es la encargada de construir un experto basado en casos con la

base de casos que usted indicó tomando los rasgo que usted seleccionó y los dominios que usted modificó. Tanto la selección de rasgos como la selección de dominios son tareas opcionales para usted. Esta opción cambia el menú actual por el menú # 3.

Descripción del menú # 3 del sistema

Submenú File: Este submenú se mantiene igual al de los menús anteriores pero agrega la opciones siguiente:

Save: Esta opción presenta el diálogo estándar de **Windows** para la salva de ficheros. Allí se entra el nombre y el camino del fichero donde salvará su sección de trabajo.

Save As: Esta opción se utiliza para salvar con un nuevo nombre la sección de trabajo.

Submenú Define: Este submenú define el caso a completar por el experto así como la cantidad de casos similares que se desean buscar.

Target Case: Esta opción sirve para definir el nuevo caso a resolver por el experto. El

diálogo de la siguiente figura muestra cómo se capta dicho caso.

Para asignarle valores a un rasgo posicione la barra de la lista sobre el rasgo y luego seleccione el botón "Edit" e inmediatamente se le muestra la lista de los posibles valores de los cuales se deben marcar los valores a asignar al rasgo en edición. Una vez editado el valor correspondiente al rasgo, si desea hacer lo contrario o sea limpiar un rasgo que tiene valores, utilice el botón "Clear" y compruebe que a la derecha del signo igual del rasgo aparece un signo de interrogación indicando que este rasgo está vacío. Todos los rasgos que queden vacíos serán completados por el experto cuando se le indique.

Similar Cases Count: Esta opción le permite al experto conocer la cantidad de casos más parecidos al caso planteado que él debe buscar.

Submenú Expert: Este submenú brinda las principales tareas que se pueden pedir a un experto basado en casos en este sistema.

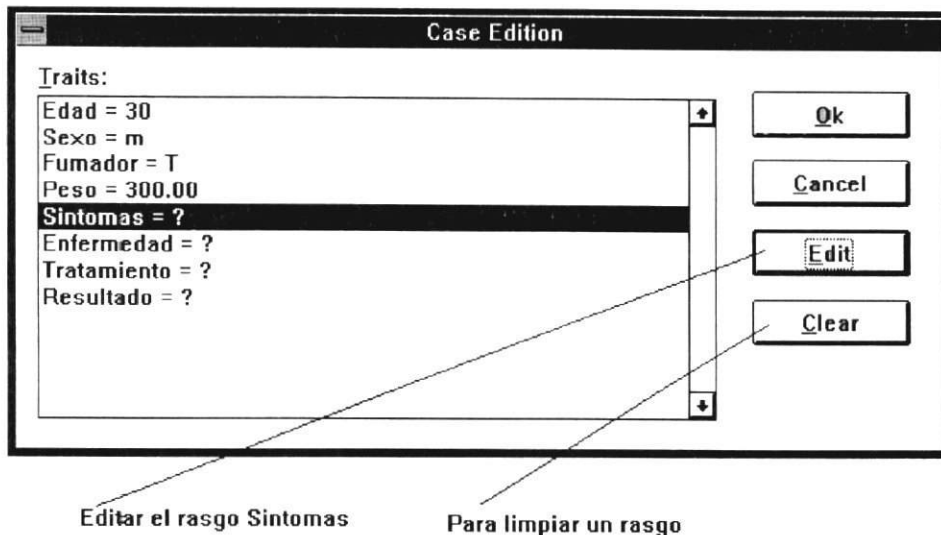


figura 1

Infer Case: Mediante esta opción se le indica al experto que infiera los rasgos objetivos del caso editado. Esta opción está inhabilitada hasta tanto no se edite un caso. El experto sólo completa el caso si existen algunos rasgos vacíos y otros rasgos valorados, en cuyo caso el sistema le reporta error.

Find Rel. Traits: El experto busca cuál es el rasgo que más influencia tuvo en la inferencia de los rasgos que usted dejó como objetivos. Esta opción no se activa hasta tanto no sea completado un caso por el experto.

Find Sim Cases: Esta opción indica al experto que busque los casos más semejantes al caso inferido tomando la cantidad indicada en la opción "Similar Cases Count" del menú # 3. Esta opción no se activa hasta tanto no se complete un caso.

Submenú Result: En este submenú se muestran los resultados obtenidos por el experto.

Case: Esta opción muestra el caso inferido en un diálogo como el de la figura de la opción "Target Case".

Relevant. Traits: Esta opción muestra el o los rasgos más relevantes extraídos por el experto.

Similar Cases: Esta opción muestra en un diálogo con la lista de los casos más parecidos al caso planteado al sistema, seleccionando el botón "View" se le mostrará el caso el cual está seleccionado en ese momento.

CONCLUSIONES

La solución de tareas de diagnóstico es una de las áreas donde mayor aplicación han tenido los sistemas basados en el conocimiento, y son particularmente los sistemas basados en reglas los más populares. Sin embargo, la aparición del razonamiento basado en casos ha motivado el desarrollo de numerosos sistemas basados en casos para tareas de diagnóstico.

En este trabajo hemos presentado una evaluación del razonamiento basado en casos como un método de solución de problemas, valorado el esquema de razonamiento para resolver problemas de diagnóstico y relacionando varios ejemplos de sistemas que ejecutan el diagnóstico basado en casos ya sea en forma pura o híbrida, integrándolo a otros métodos.

REFERENCIAS

- [1] BENJAMINS R. et. al. *Structuring diagnostic reasoning*, Department of Social Science Informatics, University of Amsterdam.
- [2] GUARDATI S. *Razonamiento basado en casos, Soluciones Avanzadas*, Año 2, N° 13, sept. 1994.
- [3] JURISICA, I. *Representation and management issues for case-based reasoning systems*, Department of Computer Science, University of Toronto, Canada, Sept. 1993.
- [4] *La sombra del olvido*. Revista Semana, Colombia, mayo 2, 1995.
- [5] LÓPEZ B. y PLAZA E. *Case-based planning for medical diagnosis*, in Proceeding of International Symposium on A.I., Ed. Cantú-Ortiz, F., LIMUSA, 1991.
- [6] KRIEGSMAN M. y BARLETTA R. *Building a Case-based help desk application*, IEEE Expert, dec. 1993, pp. 18-26.
- [7] KOLODNER, J.L. *An introduction to case-based reasoning*, Artificial Intelligence Review 6, 3-34, 1992.
- [8] RICH E. *Artificial Intelligence*, McGraw-Hill, 1993.
- [9] RISSLAND E.L. et al. *Case-based diagnostic analysis in a blackboard architecture*, in Proceeding of AAAI-93.
- [10] SIMOUDIS E. *Using case-based retrieval for customer technical support*, IEEE Expert, oct. 1992, pp. 7-11.
- [11] SLADE S. *Case-based reasoning: a research paradigm*, AI Magazine, Spring 1991, pp. 42-55.
- [12] SRIRAM D. *Computer-aided engineering: the knowledge frontier* (vol. I), Intelligent engineering systems laboratory, MIT, 1992.