

Trabajo de Diploma para optar por el Título Ingeniero en Ciencias Informáticas



**Título: Modelos para organizar Bases de Casos
en los Sistemas desarrollados con la Herramienta HESEI.**



Autora: Amirka Palacio Macías.

Tutora: Dra. Natalia Sánchez Martínez.

Junio del 2010. “Año 52 de la Revolución”

Declaración de Autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

<Nombre autor>

Amirka Palacio Macia

Firma del Autor

<Nombre tutor>

Natalia Sánchez Martínez

Firma del Tutor

Agradecimientos

A mi madre, mi hermana, mi abuela, Ángel y René por su esfuerzo, dedicación y sacrificio;

A Rey, por tu amor, apoyo y optimismo ante las dificultades;

A mi tutora, por su apoyo incondicional, puedo asegurar que es la mejor;

A Alleyne, por ser buen profe, amigo y excelente persona;

A todos los miembros del tribunal por apoyarme y guiarme;

A Jogito, por significar tanto durante toda la universidad;

A Carlos Torres, por ser muy buen profesor y amigo;

A Jehisy y Yordanis por ser muy buenos amigos;

A mis amigos Franklin, Alvarito, Yasmani, Leo, y Tony.

A las niñas del apartamento Miriam, Marisleidys, Yulla, Aymelis y Dianabel.

Y a todas las personas que hicieron posible esta realidad;

<<< Gracias >>>

Dedicatoria

A Caridad por siempre estar a mi lado;

A mi madre, Pilar Macías; por ser mi universo.

A mi hermana, Mainerys Susana; por su dedicación y esmero.

A mi sobrino bello, Alejandrino; por su inocencia y ser un buen niño.

Resumen

La Inteligencia Artificial (IA) es una de las áreas de las ciencias computacionales encargadas de la creación de softwares que tengan comportamientos inteligentes, o sea basados en la experiencia y el conocimiento humano. Uno de sus estudios más exitosos son los Sistemas Basados en Casos, con grandes aplicaciones en los procesos de toma de decisiones, por su gran potencia para realizar inferencias. Actualmente se ha incrementado el uso de estos sistemas debido a su eficiencia en diversos sectores sociales. HESEI es una herramienta de autor creada en la universidad central de Las Villas que permite la creación de sistemas basados en casos para el proceso educativo.

Con vías a perfeccionar el funcionamiento de esta herramienta y ampliar su utilización en el país no sólo en el ámbito educativo sino también en otros sectores de la sociedad, se realiza esta investigación científica, con el objetivo de estudiar los diversos modelos de organización de las bases de casos existentes en la actualidad y finalmente proponer la implementación de modelos jerárquicos que faciliten la optimización y eficiencia de los sistemas de enseñanza - aprendizaje inteligentes elaborados con la herramienta HESEI.

Palabras Claves

Inteligencia Artificial (IA), Sistemas Basados en Casos (SBC), Base de Casos (BC), Modelos de Organización de las Bases de Casos.

Índice

DECLARACIÓN DE AUTORÍA.....	II
AGRADECIMIENTOS	I
DEDICATORIA	I
RESUMEN.....	II
PALABRAS CLAVES	II
ÍNDICE	3
ÍNDICE DE FIGURAS.....	5
INTRODUCCIÓN	6
CAPÍTULO 1 REFERENTES TEÓRICOS DE LA INVESTIGACIÓN.....	10
1.1. Herramienta Computacional para elaborar SEAI: HESEI	10
1.1.1. SEAI elaborados con la herramienta computacional HESEI	11
1.1.2. Aspectos sobre la filosofía de trabajo con HESEI.....	12
1.1.3. Aspectos de la implementación computacional de HESEI	13
1.2. Sistemas Basados en Casos.....	14
1.2.1. Base de Casos (BC)	15
1.2.2. Módulo de Recuperación.....	15
1.2.3. Módulo de Adaptación	16
1.2.4. Representación del Conocimiento en la Base de Casos	16
1.3. Esquemas para la Representación de la Memoria de Casos.....	17
1.4. Empleo de índices para organizar la memoria	18
1.5. Redes Neuronales Artificiales	19
1.6. Organización de los Casos en la Memoria de HESEI.....	21
CONSIDERACIONES PARCIALES.....	21

CAPÍTULO 2 TENDENCIAS Y TECNOLOGÍAS ACTUALES	22
2.1. Modelos para representar Base de Casos	22
2.1.1. Redes de características compartidas	23
2.1.2. Redes de discriminación con prioridades	24
2.1.3. Paquetes de Organización de Memoria	24
2.1.4. Redes de discriminación redundante	25
2.1.5. Modelo categorías-ejemplares	27
2.2. Sistemas Basados en Casos.....	28
2.3. Herramientas para crear sistemas que utilizan RBC	32
2.3.1. ART* Enterprise.....	32
2.3.2. CBR3	32
2.3.3. ReMind	33
CONSIDERACIONES PARCIALES.....	34
CAPÍTULO 3 PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA	35
3.1. Modelos Jerárquicos para la organización de Bases de Casos.....	35
3.2. Selección de rasgos en la definición del modelo de la base de casos.....	37
3.3. Modelo Registros con Encabezamiento	38
3.4. Modelo de Memoria Asociativa Indexada Conexionista.....	39
3.2. Validación de la Solución Propuesta	42
CONSIDERACIONES PARCIALES.....	46
CONCLUSIONES	47
RECOMENDACIONES	48
BIBLIOGRAFÍA Y REFERENCIAS.....	49
ANEXOS	53

GLOSARIO DE TÉRMINOS..... 55

Índice de Figuras

Figura 1 Diagrama de componentes de la herramienta HESEI. 14

Figura 2 Estructura General de un SBC y su principio de funcionamiento. 15

Figura 3 Memoria plana y memoria jerárquica..... 18

Figura 4 Esquema general de la construcción del modelo de la Base de Casos. 36

Introducción

En muchos dominios de la vida real, resolver un problema conduce a un proceso de identificación y selección de la acción adecuada para su solución. A este proceso se le denomina toma de decisiones, donde una decisión es la elección de una entre diversas alternativas. En la actualidad la toma de decisiones objetivamente fundamentada, se ha convertido en una tarea fundamental en la dirección de diferentes procesos, con énfasis en el empresarial. (Alonso 1994).

Los Sistemas Basados en el Conocimiento son una de las técnicas de Inteligencia Artificial (IA) más empleadas en la actualidad en los procesos de toma de decisiones, constituyen modelos computacionales que se componen de tres componentes básicos: la base de conocimientos (BC), la máquina de inferencia (MI) y la interfaz usuario (IU). (Bello 2002).

En la BC se almacena el conocimiento necesario para resolver los problemas del dominio de aplicación atendiendo a una forma de representación del conocimiento, la MI es un procedimiento basado en un esquema de razonamiento o método de solución de problemas que utiliza el conocimiento para resolver los problemas de ese dominio y la IU permite la comunicación entre el usuario y el ordenador. Los sistemas basados en casos (SBC), constituyen casos particulares de los sistemas basados en conocimientos, y paradigmas en la resolución de problemas, pues en lugar de confiar únicamente en el conocimiento general del dominio del problema, o realizar asociaciones a lo largo de relaciones entre descripciones del problema y conclusiones, este paradigma es capaz de utilizar conocimiento específico de experiencias previas, es decir, situaciones de un problema concreto. (Bello 2002).

Ante el planteamiento de un problema no abordado con anterioridad, intenta localizar un caso pasado similar y adaptar su solución a la situación del problema nuevo. De esta adaptación se obtiene una nueva experiencia a la hora de resolver problemas con ciertas similitudes, conocido como aprendizaje incremental, ya que las nuevas adaptaciones se almacenan como nuevos casos, relacionados, y disponibles para comparaciones futuras. (Carrillo 2007).

Estos sistemas apoyan sus predicciones en ejemplos que se almacenan en la fase de aprendizaje, utilizan una función de distancia o de semejanza para determinar los casos más semejantes al nuevo problema y las soluciones de los casos recuperados se adaptan para obtener una solución.

Lo esencial para el trabajo de un SBC¹ es que todos los casos relevantes al nuevo problema pueden ser recuperados eficientemente, siendo necesario organizar el conocimiento en la base de casos en uno ó más esquemas para facilitar el proceso de recuperación y la comparación entre las descripciones de los casos y el problema a resolver. (Kolodner 1993).

Una amplia variedad de esquemas para la organización de los casos se han desarrollado a lo largo de los años compartiendo todos ellos dos características. La primera de ellas es que estos esquemas pueden programarse con lenguajes de programación existentes y almacenarse en memoria. Segundo, todos ellos se diseñan de modo que el conocimiento que almacenan pueda usarse en el razonamiento, es decir, la BC contiene una estructura de datos que puede manejarse por un sistema de inferencia que usa técnicas de búsqueda y patrones de equivalencia para responder preguntas, trazar conclusiones o ejecutar una función inteligente. (Kolodner 1993).

En el grupo de Informática Educativa e Inteligencia Artificial de la Universidad Central de Las Villas se desarrolló la herramienta computacional HESEI (herramienta para la elaboración de sistemas de enseñanza inteligentes) que facilita la elaboración de SBC para el proceso de enseñanza-aprendizaje. El trabajo con HESEI se ha extendido a diferentes centros de educación e investigación desarrollándose sistemas en áreas como: Humedales en el Laboratorio de Propagación Masiva de Plantas del Instituto de Biotecnología de las Plantas, UCLV², Teoría de Grafos, Análisis y Diseño de Sistemas y Estructura de Datos (Facultad de Matemática Física y Computación, UCLV). (Martínez 2010).

HESEI utiliza el razonamiento basado en casos (RBC) para crear sistemas inteligentes, el funcionamiento de estos sistemas depende de la estructura y el contenido de su memoria de casos. La base de casos implementada en HESEI utiliza una estructura lineal, este enfoque consiste en almacenar los casos secuencialmente y analizar cada uno para resolver el nuevo problema.

Este tipo de organización hace demasiado lento el proceso de recuperación, siendo uno de los aspectos fundamentales en la representación del conocimiento, la flexibilidad en la representación y la eficiencia en la recuperación. Teniendo en cuenta que la representación y manejo de grandes bases de casos no son tareas triviales y constituyen un problema de investigación en la actualidad, pues la organización de la memoria de casos define cómo será el acceso a los casos y de ésta depende que la velocidad en la

¹ Se utiliza SBC para definir los sistemas basados en casos tanto en plural como en singular.

² Universidad Central de las Villas.

recuperación no se afecte por el volumen de la experiencia almacenada, en términos de preguntas de investigación el **problema a resolver** se formula:

¿Cómo facilitar a la herramienta HESEI un proceso de organización de la memoria de los casos de forma eficiente?

El **objeto de estudio** de la investigación está enmarcado en el proceso de organización y recuperación de los casos en los sistemas basados en casos.

Se tiene como **campo de acción** la organización de la base de casos implementada en sistemas a partir de la herramienta HESEI. El **objetivo general** del trabajo consiste en la propuesta de modelos jerárquicos para la organización de las bases de casos implementadas con la herramienta HESEI.

Para lograr el objetivo se plantean las siguientes **tareas investigativas**:

1. Hacer una revisión bibliográfica sobre los modelos de organización de las bases de casos reportados en la literatura científica.
2. Realizar un análisis del modelo de bases de casos utilizado en la herramienta HESEI.
3. Proponer modelos de organización de la memoria que permitan un acceso a los casos rápido y eficiente en aplicaciones desarrolladas a partir de HESEI.
4. Validar la propuesta de solución.

Para el desarrollo de las tareas científicas se utilizan diferentes **métodos de investigación** teóricos en la búsqueda y procesamiento de la información.

- El **método Análisis Histórico-Lógico**, utilizado para el estudio de los modelos de organización de la memoria de sistemas basados en casos desarrollados a largo de los años, así como sus aplicaciones en la actualidad.
- El **Analítico-Sintético**, se utiliza para el análisis de la herramienta HESEI, permite profundizar el estudio de la memoria de casos de esta herramienta para la búsqueda de una solución óptima al problema de investigación.

La tesis está conformada por tres capítulos. El primero describe los referentes teóricos de la investigación. Sobre la herramienta HESEI se analizan sus características, estructura, aspectos relacionados con la implementación y sus aplicaciones en la actualidad. Además se describen los elementos fundamentales de

los SBC, así como la representación del conocimiento en la BC y los esquemas de organización de la memoria existentes.

En el segundo capítulo se realiza un análisis sobre los principales modelos de organización de bases de casos utilizados actualmente. Se refiere a las características distintivas, facilidades que brindan desde el punto de vista funcionales mediante el estudio de sistemas que utilizan estas estructuras, también se hace énfasis en las herramientas que permiten la elaboración de estos sistemas, analizando las ventajas que brindan desde el punto de vista del diseño e implementación de estas aplicaciones.

En el tercer capítulo se propone la solución al problema a resolver, describiéndose los modelos de organización de la memoria de casos propuestos así como elementos relevantes a tener en cuenta para el perfeccionamiento de la organización de los casos de HESEI. Por último, se establecen conclusiones, se emiten recomendaciones, se relacionan las referencias bibliográficas y se incluye un conjunto de anexos que facilitan la comprensión de la memoria gráfica de la tesis.

Capítulo 1 Referentes Teóricos de la Investigación

Los Sistemas Basados en Casos constituyen técnicas de Inteligencia Artificial utilizadas en diversos sistemas actuales debido a su eficiencia en el diagnóstico, búsqueda y toma de decisiones. Este capítulo describe la arquitectura y principales características de HESEI, una herramienta para construir aplicaciones que utilizan razonamiento basado en casos. Se ofrece una visión general sobre la estructura, componentes de este tipo de sistemas. Además se describen conceptos fundamentales sobre la organización de la memoria y la representación de la información en las bases de casos.

1.1. Herramienta Computacional para elaborar SEAI: HESEI

HESEI es una herramienta de autor que facilita la elaboración de Sistemas de Enseñanza-Aprendizaje Inteligentes (SEAI) a usuarios no expertos en el campo informático; pero sí en dominios donde ejercen su profesión como docentes. (Martínez 2010)

Como filosofía de trabajo de este software se definen dos fases bien delimitadas: diseño del SEAI y trabajo con la herramienta computacional HESEI. La fase de diseño del SEAI se refiere al trabajo metodológico implícito o ingeniería del conocimiento en la elaboración de éste.

En esta fase se selecciona el tema, se estructura el mismo, se definen los objetivos que se persiguen, así como los estilos de aprendizaje, particularidades en las que se basa la caracterización de los estudiantes, los entrenadores o materiales adecuados a cada modelo de estudiante, entre otros. En la fase anterior es concebida la base de casos estructuralmente, sin embargo es en la fase de trabajo donde se edita la BC y por tanto quedan completados los rasgos predictores que caracterizan modelos de estudiante y el rasgo objetivo que describe el material didáctico necesario a ese modelo. Por tanto un caso, representa un modelado del estudiante sin prever otros factores. (Martínez 2010).

El esquema en forma de mapa conceptual de la figura 1 muestra la estructura general del modelo, el cual integra el paradigma del razonamiento basado en casos y los SEAI.

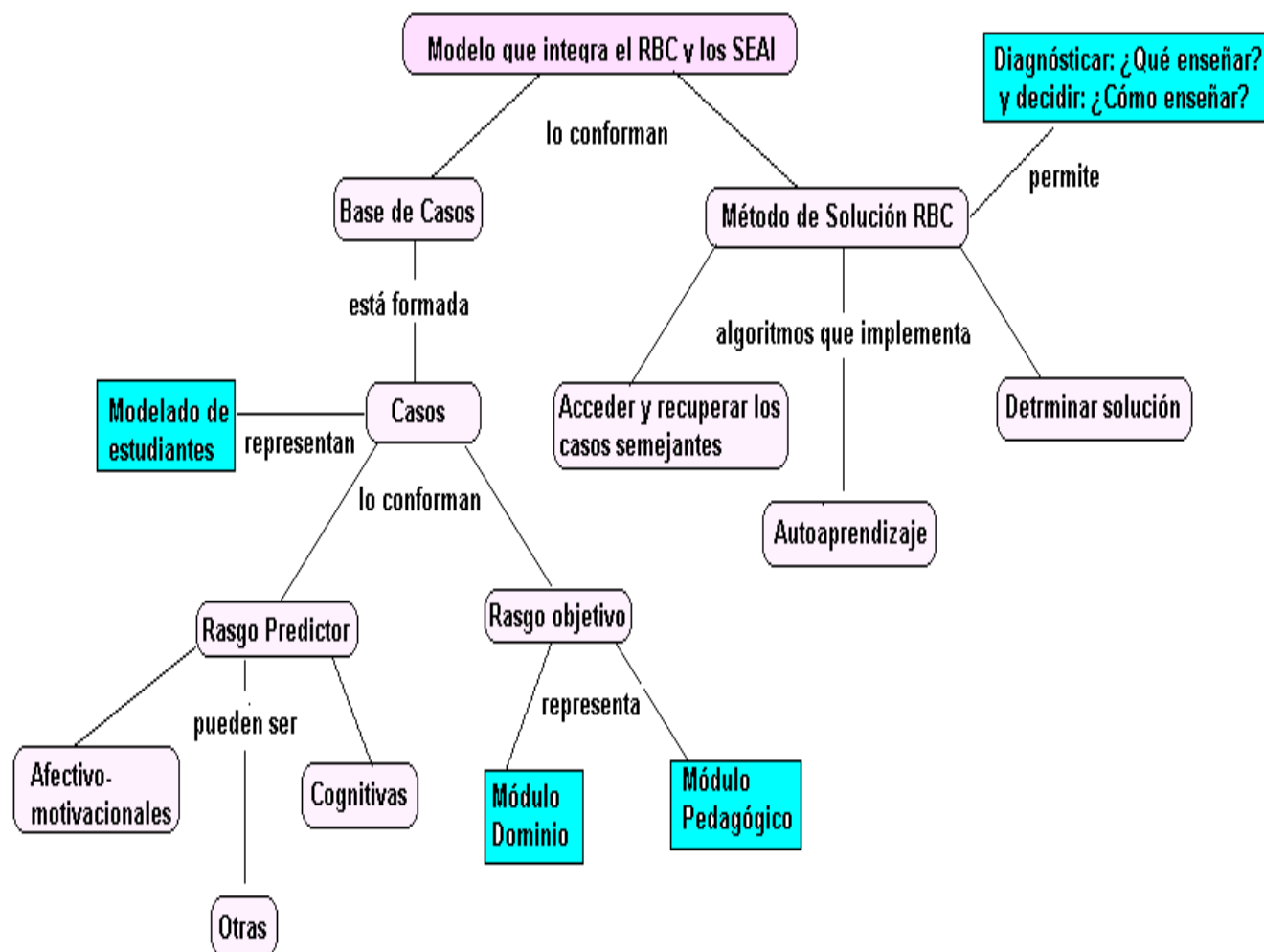


Figura 1 Estructura general del modelo de la herramienta HESEI. (Martínez 2010).

1.1.1. SEAI elaborados con la herramienta computacional HESEI

El trabajo con la herramienta computacional HESEI se ha extendido a diferentes centros de educación e investigación, desarrollándose diversos SEAI en áreas como: Humedales en el Laboratorio de Propagación Masiva de Plantas del Instituto de Biotecnología de las Plantas, de la UCLV, Teoría de Grafos, Análisis y Diseño de Sistemas y Estructura de Datos (Facultad de Matemática Física y Computación de la UCLV), Contabilidad y Finanzas (sedes universitarias municipales de Santa Clara) y Carreteras (Facultad de Construcciones de la UCLV). Las aplicaciones anteriores se encuentran en diferentes estados de desarrollo. Se describen aspectos generales de tres de las aplicaciones ya terminadas y que han sido utilizadas para el objetivo de su elaboración:

✚ SEAI para la Educación Ambiental sobre Humedades

Para el desarrollo del SEAI se trabajó con un equipo multidisciplinario conformado por especialistas del Laboratorio de Propagación Masiva de Plantas del Instituto de Biotecnología de las Plantas. (León E. 2008). El SEAI para la Educación Ambiental sobre Humedades obtiene el premio de Relevante en el evento municipal de Santa Clara, ECOJOVEN 2007 y en el evento PROECO 2007.

✚ SEAI para apoyar el estudio de la asignatura de Contabilidad Básica

El SEAI se introdujo en las sedes universitarias municipales, teniendo en cuenta el modelo semi-presencial de la enseñanza en estas instituciones y la necesidad de atender las individualidades de los alumnos en la asimilación de contenidos que en muchos casos resultan complejos. El SEAI para apoyar el estudio independiente de la Asignatura de Contabilidad Básica obtiene Destacado en el XVI Fórum de Ciencia y Técnica Municipal Santa Clara y en el XVI Fórum de Ciencia y Técnica Municipal UCLV. (Fortún F 2007).

✚ SEAI para apoyar el estudio de la Teoría de Grafos

Se desarrolla para apoyar el estudio de la teoría de grafos a estudiantes que cursan el segundo año de la carrera de Ciencias de la Computación; ya que el objeto inmediato de estudio del profesional en Computación lo constituyen las estructuras y procesos que hacen posible la solución computacional de clases arbitrarias de problemas. Como se refiere en el modelo del especialista, este deriva la descripción y modelación de estas estructuras y procesos que pueden ser alternativamente concebidos desde un punto de vista informacional, utilizando diversas teorías y estructuras matemáticas, tanto continuas como discretas. Entre las teorías de carácter discreto se distinguen, por sus grandes utilidades computacionales aquellas que suelen ser reunidas bajo el nombre de Matemática Discreta la cual incluye el desarrollo de la Teoría de Grafos.

1.1.2. Aspectos sobre la filosofía de trabajo con HESEI

La herramienta computacional HESEI tiene dos actores fundamentales: profesor y estudiante. Las funcionalidades del profesor se corresponden con la obtención de un prototipo computarizado del diseño realizado en la tercera etapa de la guía de orientación a la ingeniería del conocimiento, y las correspondientes al estudiante se fundamentan en el diagnóstico del conocimiento del estudiante implementando el RBC como método de solución de problema. Secuencia de trabajo en HESEI para el desarrollo de un Sistema de Enseñanza Aprendizaje Inteligente (SEAI):

I) Identificación del usuario existiendo ambientes de trabajo distinto para el profesor y el estudiante, con dominios diferentes si se trabaja en red o en una PC aislada.

El sistema tiene su propia política de seguridad, disponiendo de tres tipos de permisos:

Administrador: permite acceder a todas las funcionalidades, controla la autorización de accesos y administra las cuentas de los usuarios registrados en cualquiera de los dominios concebidos.

Profesor: permite las funcionalidades relacionadas con el profesor: crear, modificar y almacenar SEAI.

Estudiante: permite las funcionalidades relacionadas al estudiante: seleccionar un SEAI.

II) En dependencia del tipo de usuario: para el usuario **Profesor**:

- Crear o modificar un SEAI auxiliándose de un editor de tópicos, editor de preguntas, base de datos, entre otros elementos.
- Recuperar materiales didácticos.

El editor de tópicos permite definir: el número de preguntas que conforman el cuestionario, validar los tópicos, calcular el grado de certeza relacionado con el valor del tópico, entre otros. El editor de preguntas permite el trabajo sobre tres tipos de preguntas: verdadero o falso, marcar la correcta y relacionar columnas. Su diseño e implementación favorece la incorporación de otros tipos de preguntas para ofrecer una mayor variedad de formas para obtener los rasgos cognitivos de un estudiante. Para el usuario Estudiante:

- Seleccionar un SEAI previamente elaborado.
- Describir su modelo.
- Llenar cuestionario aplicado.
- Interactuar con un Sistema para Aplicación de Test (SAT).

1.1.3. Aspectos de la implementación computacional de HESEI

El sistema se desarrolla en Borland Delphi 7.0. Su ejecutable ocupa 400 KB y requiere, que junto con este, se encuentre los ficheros .txt elaborados con el sistema correspondientes a las bases de casos donde está almacenada la información sobre los SEAI y los ficheros asociados a los materiales didácticos que serán utilizados. Es posible ejecutar HESEI en cualquier versión de Windows. El software HESEI adicionalmente se provee de otras implementaciones: Un SAT, para la captación del estado afectivo-motivacional a través de test psicométricos usando procesamiento del lenguaje natural y un módulo para la modelación de la evaluación de preguntas utilizando la lógica difusa. (Martínez 2010)

En la figura 2 se relacionan los principales componentes de la herramienta HESEI, dígame la base de casos, el ejecutable (HESEI.exe) y los materiales didácticos asociados.

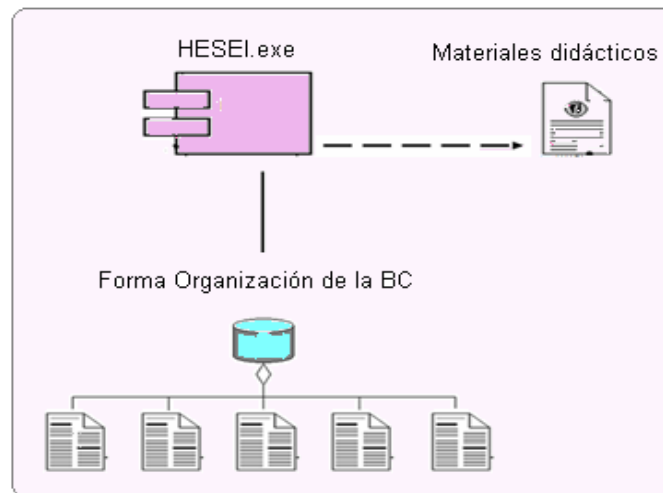


Figura 1 Diagrama de componentes de la herramienta HESEI.

La herramienta computacional ofrece la posibilidad de elegir el estilo de ventana de la aplicación a través de la componente Vclskin, disponible para Delphi, por lo que el usuario puede predeterminar la configuración que le acompañe en su interacción con el sistema. Para esta finalidad se cuenta con una colección de ficheros de configuración de estilos llamados skins, disponibles en Internet, de los cuales puede seleccionarse el deseado según el interés del software y a quién esté destinado.

Si el usuario desea incorporar algún nuevo estilo de su preferencia, solo debe copiarlo en la carpeta que contiene los demás estilos y posteriormente seleccionarlo para que la herramienta tome esa configuración en su interfaz visual. El resultado de la utilización de HESEI es un SBC con aplicaciones en el sistema educativo.

1.2. Sistemas Basados en Casos

El Razonamiento Basado en Casos (RBC), (Kolodner 1993; Bello 2002), es un enfoque que aborda nuevos problemas tomando como referencia problemas similares resueltos en el pasado. De modo que problemas similares tienen soluciones similares, donde la similitud juega un rol esencial.

Sus componentes fundamentales son la base de casos, el módulo de recuperación de casos y el módulo de adaptación de las soluciones (García 1997). Cada estructura tiene su participación en el funcionamiento del SBC, la figura 3 muestra cómo se relaciona cada componente para dar solución a un nuevo problema.

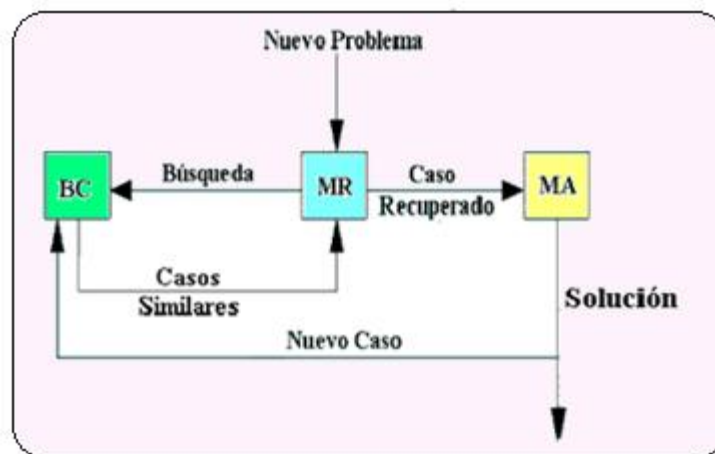


Figura 2 Estructura General de un SBC y su principio de funcionamiento.

1.2.1. Base de Casos (BC)

La BC contiene las experiencias, ejemplos o casos a partir de los cuales el sistema hace sus inferencias. Esta base puede ser generada a partir de casos o ejemplos resultantes del trabajo de expertos humanos o por un procedimiento automático o semiautomático que construye los casos desde datos existentes registrados, por ejemplo, en una base de datos. (Kolodner 1992).

1.2.2. Módulo de Recuperación

En este módulo se recuperan de la Base de Casos los casos más semejantes al problema. No existe una medida de semejanza única, general, para cualquier dominio, de ahí que la eficiencia del sistema radica en la función de semejanza que se defina. La función de semejanza más sencilla consiste en contar el número de rasgos predictores similares entre ambos, sin embargo se presenta el problema de que la importancia de los rasgos predictores varía de un contexto a otro (Ruíz 1993).

La representación más general de función de semejanza pudiera ser:

$$\beta(O_0, O_t) = \frac{\sum_{i=1}^n w_i \cdot \delta_i(x_i(O_0), x_i(O_t))}{\sum_{i=1}^n w_i}$$

donde n es el número de rasgos predictores y w_i la importancia asociada al rasgo i .

Otros ejemplos de funciones de semejanza aparecen en la literatura científica (Ruiz 1993), (Rodríguez y García 2007), (Yang 2007), entre otros.

1.2.3. Módulo de Adaptación

Después de la determinación de los casos más semejantes, las soluciones contenidas en dichos casos pueden usarse directamente como solución al nuevo problema, pero comúnmente necesitan ser modificadas. En (Kolodner 1993) y (Bonzano 1998) aparecen métodos y reglas de adaptación para realizar dicha modificación.

1.2.4. Representación del Conocimiento en la Base de Casos

Un razonador basado en casos depende de la estructura y el contenido de la BC, la flexibilidad en la representación y la eficiencia en la recuperación constituyen aspectos fundamentales en la representación del conocimiento. Los conceptos de flexibilidad y eficiencia están usualmente relacionados a los conceptos de expresividad, accesibilidad, contexto y transparencia.

La expresividad es relativa a cómo el contenido del conocimiento puede ser representado (hechos, hipótesis, heurísticas, incertidumbre etc.) y no a cómo pudiera ser codificado en algún lenguaje de representación. La expresividad está además relacionada con la capacidad del sistema de representar y manipular conocimiento incierto (Pérez 2002)

En los SBC la incertidumbre está presente en casi todos los componentes del conocimiento: en los rasgos predictores usados para describir los casos (vocabulario), en las relaciones entre casos y sus rasgos, en la forma de evaluar la semejanza, y en la manera que la base de casos se organiza y accede. La accesibilidad se define como el problema de localizar aquellos casos en la base que son más relevantes al problema actual. La misma está muy relacionada con el problema de la indexación de los casos de la base (Hinrichs 2002.).

La noción de contexto ha sido usada en los sistemas basados en casos para imponer una estructura modular de representación. En el RBC el contexto debe ser considerado para determinar la relevancia de los casos almacenados y aumentar la eficiencia del proceso de recuperación. La noción de transparencia está basada en dos aspectos fundamentales: en primer lugar, cuando el sistema se está diseñando, el ingeniero del conocimiento y el experto del dominio tienen que tener una idea clara de los tipos de conocimiento que pueden ser representados y cómo el sistema los manipula para derivar nueva información (Hinrichs 2002.)

En segundo lugar, cuando el sistema se está explotando por usuarios que no son experimentados, éstos necesitan conocer cómo representar el conocimiento, cuál es su expresividad, sus posibilidades y limitaciones. (Perez, 2002).

1.3. Esquemas para la Representación de la Memoria de Casos

Dado que los casos constituyen el elemento principal de todo SBC, la manera de almacenarlos repercutirá directamente en el rendimiento del mismo. La BC es el módulo encargado de almacenar y organizar todos los casos disponibles y su estructura es crucial para la fase de recuperación de casos similares. En la creación de la estructura que formará la BC hay que tener en cuenta aspectos fundamentales: el tamaño que se aspira que tenga la BC, los requerimientos que se imponen en el dominio específico en el que se trabaja y la inserción eficiente de nuevos casos (Pérez 2002). Tradicionalmente se han propuesto dos estructuras principales para la el almacenamiento de casos:

Memoria plana: En esta estructura se presentan los casos completos de forma secuencial utilizando algún tipo de indexación ya sea manual o automática.

Los casos se almacenan secuencialmente en una lista simple, un arreglo o un fichero. Para lograr una recuperación eficiente, se indexan los casos de la base. Los índices se eligen para representar los aspectos importantes del caso y la recuperación involucra comparar las características consultadas con cada caso de la BC.

Esta estructura tiene el principal inconveniente de que la búsqueda de casos es menos eficiente, por lo que no es recomendable utilizarla en sistemas que trabajan con una base de casos de gran tamaño y necesitan respuesta en tiempo real. Por otra parte la inserción de nuevos casos en este tipo de estructuras es muy sencilla ya que basta con incluir un nuevo registro con el nuevo caso respetando el método de indexación que se esté utilizando. (Mora 2008).

Memoria jerárquica: En una estructura con memoria jerárquica se utilizan representaciones en forma de árbol, en los que cada nodo interior representa un atributo del caso y en las hojas se almacenan las soluciones a los mismos. Cada recorrido desde la raíz hasta una de las hojas del árbol representa un caso completo. Se debe intentar que los atributos más discriminantes estén almacenados en los niveles superiores del árbol empezando desde la raíz.

La gran ventaja de este tipo de almacenamiento es la eficiencia en la búsqueda pero a cambio se sacrifica la sencillez de inserción de nuevos casos. (Mora 2008).

En la figura 4 se muestra un esquema en el que se observa las diferencias entre la estructura de una memoria plana y una memoria jerárquica.

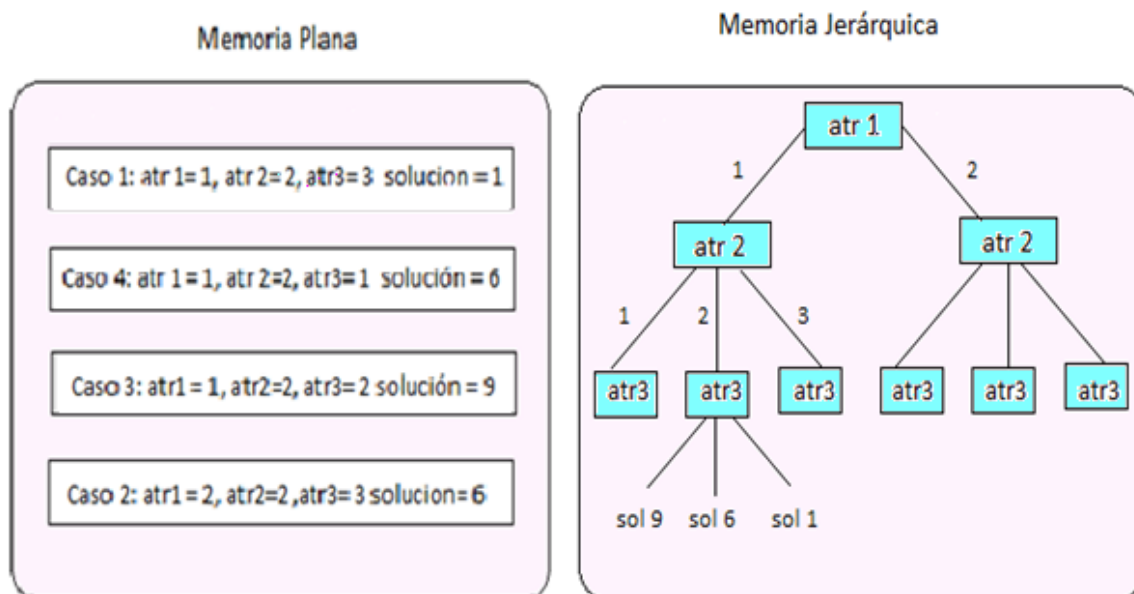


Figura 3 Memoria plana y memoria jerárquica.

Los desarrolladores de aplicaciones que utilizan RBC tienen que llegar a un compromiso entre eficiencia de recuperación y eficiencia de inserción para acertar en la elección de una de las estructuras anteriores para crear la base de casos. De la correcta elección de esta estructura dependerá en gran medida la eficiencia global del funcionamiento de todo el sistema.

1.4. Empleo de índices para organizar la memoria

Los SBC derivan su habilidad para recuperar casos relevantes de una biblioteca de casos de manera eficiente gracias al proceso de indexación de la memoria. Este mecanismo es fundamental para que la base de casos se organice de modo que el proceso de recuperación sea preciso y eficiente. A partir de la descripción del problema, los índices deben apuntar cuáles características del caso deben ser comparadas, determinando así el caso que puede ser útil para llegar a una solución.

El problema de indexar consiste en cómo etiquetar los casos en memoria y recuperarlos cuando se necesiten en una forma que evite realizar las comparaciones con todos los elementos de la base de casos. (Bonzano 1998). Se propone relacionar los casos en la memoria de acuerdo a un conjunto de etiquetas o rasgos destinados a indicar cuándo será útil el caso, y luego extraer un conjunto similar de etiquetas del nuevo problema. Los índices deben basarse en los rasgos de interés de las categorías, es decir, los rasgos

que son adecuados para ser usados en identificar un elemento de la memoria que se debe recuperar de la categoría o en describir un evento para ser añadido a una categoría.

La secuencia de trabajo vinculada con el uso de los índices propuesta para organizar la base de casos y realizar el acceso a ellos es la siguiente:

- Especificar que clases de índices son los más útiles.
- Designar un vocabulario para los índices.
- Crear algoritmos y heurísticas para automatizar la selección de índices.
- Organizar los casos basándose en estos índices.
- Buscar en la memoria usando los índices.
- Seleccionar el mejor de los casos recuperados.

Para la selección de los índices se proponen las siguientes estrategias:

- Usar todos los rasgos para realizar la clasificación.
- Usar solamente los rasgos dados en la descripción del problema a resolver. Este enfoque es simple y directo pero puede ser demasiado simple para ser útil en algunos dominios.
- Generar propiedades más abstractas de la descripción del problema y usar solamente éstas en la clasificación. (Bonzano 1998).

Cuando se añadan nuevos casos a la memoria tienen que ser indexados en las categorías de memoria apropiadas. La indexación sirve tanto para discriminar eventos individuales como para dividir una categoría en sub categorías de tamaño razonable. Además, este permite también detectar las similitudes entre eventos y hacer generalizaciones. Cuando dos eventos sean indexados en la misma forma se puede formar una sub categoría, indexando al padre por el rasgo que indexó los dos eventos.

Las similitudes entre los dos eventos se pueden extraer y almacenar como información generalizada asociada con la nueva sub categoría. Esa información generalizada se puede usar durante la comprensión, recuperación y posterior actualización de la memoria. La información generalizada no es estática, cambia con la adición de nuevos eventos.

1.5. Redes Neuronales Artificiales

Una red neuronal artificial (RNA), según Freman y Skapura (Freeman 1993), es un sistema de procesadores paralelos conectados entre sí en forma de grafo dirigido. Esquemáticamente cada elemento de procesamiento de la red (neuronas) se representa como un nodo.

Estas conexiones establecen una estructura jerárquica que tratando de emular la fisiología del cerebro busca nuevos modelos de procesamiento para solucionar problemas concretos del mundo real.

Las RNA están formadas por una gran cantidad de neuronas, denominarse nodos o unidades de salida, un nodo o neurona cuenta con una cantidad variable de entradas que provienen del exterior (X_1, X_2, \dots, X_m).

A su vez dispone de una sola salida (X_j) que transmitirá la información al exterior o hacia otras neuronas. Cada X_j o señal de salida tiene asociada una magnitud llamada peso, este se calcula en función de las entradas, por lo que cada una de ellas es afectada por un determinado peso. Los pesos corresponden a la intensidad de los enlaces sinápticos entre neuronas y varían libremente en función del tiempo y en cada una de las neuronas que forman parte de la red. (Sotolongo, 2001).

Los elementos fundamentales de una RNA son: topología, modelo de neurona y algoritmo de aprendizaje. Se define como topología de una red neuronal a la organización o arquitectura del conjunto de neuronas que la forman; esta organización comprende la distribución espacial de las mismas y los enlaces entre ellas. El modelo de la neurona define el comportamiento de la misma al recibir una entrada para producir una respuesta, a esta respuesta se le conoce como nivel de activación.

El proceso de aprendizaje consiste en hallar los pesos que codifican los conocimientos, mediante la utilización de algoritmos o reglas, haciendo variar el valor de los pesos de una red hasta que estos adoptan un valor constante, cuando esto ocurre se dice que la red ya "ha aprendido".

El procesamiento de información en una RNA comprende, generalmente, dos fases: una fase de entrenamiento y una fase de producción. Durante el entrenamiento, la red ajusta los pesos de conexión entre las distintas unidades siguiendo un determinado algoritmo de aprendizaje. El objetivo del entrenamiento es encontrar un conjunto de pesos para los que la aplicación de un conjunto de vectores de entrada genere el conjunto de salidas deseadas. Una vez que la red se estabiliza, es decir, no hay modificación de los pesos, comienza la explotación de la red. Se aplican unos valores de entrada, se propaga la señal a través de la red y se obtienen los valores de salida. (Valdivia 2006).

En el contexto de las RNA puede definirse el aprendizaje como el proceso por el que se produce el ajuste de los parámetros libres de la red a partir de un proceso de estimulación por el entorno que rodea la red. El tipo de aprendizaje está determinado por la forma en la que dichos parámetros son adaptados. En la mayor parte de las ocasiones el aprendizaje consiste simplemente en determinar un conjunto de pesos sinápticos que permita a la red realizar correctamente el tipo de procesamiento deseado. (Salas 2006).

1.6. Organización de los Casos en la Memoria de HESEI

Los casos en la BC de la herramienta HESEI representan el estado del conocimiento y comportamiento del estudiante, así como el entrenador o material didáctico más adecuado.

Cada caso es un ejemplo de modelado de estudiante, el cual se divide en modelo del estudiante (rasgos predictores), materiales didácticos más adecuados para ese modelo (rasgo objetivo).

Dado un nuevo estudiante se diagnostica usando el paradigma del RBC los entrenadores sugeridos para el mismo, adaptados a sus conocimientos y comportamientos. Los rasgos predictores reflejan el estado cognitivo, el estado afectivo y otros elementos de interés sobre el estudiante. No se limita el número de rasgos para caracterizar el estado cognitivo y afectivo del estudiante (Martínez 2010).

Cada rasgo tiene un valor asociado y una medida de certeza. El rasgo objetivo es un rasgo multievaluado, los valores del mismo se corresponden con los materiales didácticos propuestos para ese modelo de estudiante, sugiriendo un orden. Los rasgos predictores, contienen los datos de entrada, o sea la información a partir de la cual el sistema infiere el estado del estudiante, ya sea cognitivo, afectivo-motivacional u otras características que se consideren importantes a tener en cuenta en el SEAI que se desarrolla (Martínez 2010).

La estructura secuencial que implementa como forma de organización no favorece el acceso y recuperación de los casos cuando el número de casos y rasgos predictivos es grande trayendo como resultado que esta herramienta no tenga mayores aplicaciones en la creación de sistemas de enseñanza inteligentes que utilicen grandes bases de conocimientos. (Martínez 2010).

Consideraciones Parciales

HESEI es una herramienta que permite el desarrollo de sistemas que utilizan razonamiento basado en casos, estas aplicaciones constituyen paradigmas en la resolución de problemas por su gran potencia en los procesos de inferencia y toma de decisiones. Por lo que requieren flexibilidad para escoger la mejor alternativa de representación de los casos. Presentan una estructura que facilita el funcionamiento efectivo del sistema pues responden a esquemas de organización de la información que utilizan técnicas y modelos de procesamiento de la información que permiten la eficiencia y optimización de estos sistemas.

Capítulo 2 Tendencias y Tecnologías Actuales

En los últimos años, el desarrollo de sistemas basado en casos ha experimentado un rápido crecimiento desde su nacimiento en Estados Unidos. Lo que sólo parecía interesante para un área de investigación muy reducida, se ha convertido en una materia de amplio interés, multidisciplinar y de gran interés comercial. (Carrillo 2007). En este capítulo se describen los diferentes modelos para organizar las bases de casos utilizados en la actualidad, además se realiza un análisis de algunos sistemas que utilizan estas estructuras y se hace énfasis en el estudio de herramientas para la elaboración de estos sistemas.

2.1. Modelos para representar Base de Casos

Los casos pueden representar distintos tipos de conocimiento y pueden ser almacenados en diferentes formatos en dependencia del tipo de SBC. Una de las ventajas del razonamiento basado en casos es la flexibilidad que ofrece respecto a la representación. (C. Riesbck 1989)

Se puede elegir la implementación del modelo de organización adecuado dependiendo del tipo de información a representar. Hasta el presente, para resolver el problema de organizar una base de casos, un enfoque ha sido almacenar los casos de forma secuencial y analizarlos todos para resolver el nuevo problema. Este tipo de organización hace lento el proceso de recuperación.

Un método más sofisticado consiste en dividir los casos en grupos y organizarlos jerárquicamente. Esta jerarquía permite una búsqueda más eficiente ya que se sigue por un determinado camino en dependencia de los valores de los rasgos predictores del nuevo problema. Diferentes jerarquías han sido propuestas en diferentes modelos., un enfoque es el uso de árboles de decisión en los cuales las hojas contienen todos los casos y los nodos intermedios contienen particiones de la base original. Este tipo de enfoque es particularmente útil cuando las BC son grandes. Sin embargo, cuando los casos no están completamente disponibles y el dominio no está bien definido este enfoque resulta más difícil de aplicar. (Pérez 2002).

Otros autores plantean que un enfoque consiste en usar una jerarquía abstracta donde cada nodo es una abstracción de los casos representados por sus hijos. Estas jerarquías se conocen como redes de discriminación donde los nodos representan regiones de superposición de casos. Sin embargo, estos sistemas requieren mucho más memoria para almacenar la red y los procedimientos para agregar nuevos

casos son muy caros ya que la jerarquía de abstracción necesita ser reestructurada cada vez que se incorpora un nuevo caso. (Mora 2008).

Un método alternativo, quizás más aplicable a dominios débiles, consiste en almacenar sólo casos prototipos. Los modelos basados en ejemplares no necesariamente requieren todos los rasgos o todos los casos por adelantado. Sin embargo buscar un criterio para determinar qué es un buen ejemplar no resulta una tarea trivial. Independientemente del tipo de modelo de representación que se elija siempre se debe tener en cuenta que la información que almacene un caso debe ser relevante tanto para el propósito del sistema como para asegurar que siempre será elegido el caso más apropiado para solucionar un nuevo problema en un determinado contexto. (Bello 2000).

A continuación se profundizará en el estudio de diversos modelos de organización de la memoria de casos utilizados en sistemas con amplias aplicaciones en la actualidad, teniendo en cuenta la forma de almacenamiento de los casos, y el acceso a la BC.

2.1.1. Redes de características compartidas

Los casos se almacenan en un árbol o en un grafo acíclico directamente. El grafo subdivide el espacio de casos de acuerdo con los atributos que les caracterizan (clustering).

Las características comunes ocupan nodos del grafo de donde cuelgan los casos que las comparten. La mayoría de los sistemas tienen alguna especie de umbral relativo al número de casos que debe compartir el valor de una característica, de modo que se justifique su posición en un nodo. Este procedimiento es computacionalmente más caro, ya que la base requiere un mayor espacio de almacenamiento y su actualización resulta más compleja pues generalmente modifica una parte de la red. Además, el diseño y mantenimiento de la red óptima es costoso. Los casos recuperados dependen del contenido de los nodos: la jerarquía establecida debe responder a la importancia de las características. (Carrillo 2007).

El algoritmo de búsqueda del caso que mejor se ajusta al de entrada es también sencillo:

Inicializar N = nodo padre

Repetir hasta que N sea un caso:

Encontrar el nodo bajo N más similar a la entrada.

Devolver N

2.1.2. Redes de discriminación con prioridades

Los tipos de sistema anteriores no resuelven el problema de búsqueda cuando la entrada está incompleta. En las redes de discriminación con prioridades cada nodo contiene una pregunta para la cual los sub nodos correspondientes ofrecen respuestas alternativas.

Las preguntas más importantes se formulan primero, situándose más arriba en la jerarquía. Así como las redes de características compartidas, las redes de discriminación subdividen el conjunto de casos y comparten la mayoría de las ventajas y desventajas de esta técnica. La formulación de preguntas puede implementarse más eficientemente que el emparejamiento en cada sub nodo. Al mismo tiempo, separar los atributos de sus valores particulares hace más fácil identificar qué atributos han resultado más útiles para la caracterización del caso. (C. Riesbck 1989).

Las desventajas que plantean son las mismas que las de las redes de características compartidas: algunos casos significativos pueden ignorarse si la ordenación de las preguntas no es la óptima. El algoritmo de búsqueda resulta:

Inicializar N = nodo padre

Repetir hasta que N sea un caso:

Formular pregunta en N sobre la entrada

Asignar N = sub nodo conteniendo la respuesta

Devolver N

2.1.3. Paquetes de Organización de Memoria

Los paquetes de organización de memoria son unidades primarias de almacenamiento de una memoria dinámica que sirven para detectar las partes comunes de las experiencias contenidas en distintos episodios. A diferencia de los guiones, los MOP³ reúnen, mediante enlaces creados dinámicamente, las escenas y los diferentes elementos de conocimiento que son relevantes para interpretar una determinada situación. Estos enlaces cambian continuamente, en función del descubrimiento de nuevas semejanzas o diferencias que se consideran más significativas para la tarea en cuestión. Este modelo dinámico de memoria pretende integrar conocimiento a partir de episodios específicos con conocimiento generalizado

³ MOP en inglés, Model Organization Packets: Paquetes de Organización de Memoria, se utiliza tanto para el plural como el singular.

extraído de los mismos episodios, sirve al mismo tiempo de soporte para el aprendizaje de conocimiento generalizado. (Boticario 2006).

Este funcionamiento se apoya en la suposición de que no es posible realizar comprensión sin aprendizaje. Se puede considerar básico y clásico este modelo de organización pues ha servido como base para el desarrollo de otros modelos más específicos de memoria dinámica. La técnica de los MOP involucra nociones estándares de la Inteligencia Artificial tales como, frames, abstracción, herencia; pero permite que la BC cambie dinámicamente durante el aprendizaje. Un MOP se usa para representar conocimiento sobre clases de eventos especialmente eventos complejos, contiene un conjunto de normas que representan sus rasgos básicos es decir qué eventos ocurren, qué objetivos se deben realizar, qué actores participan, o sea, información similar a la representada en un script. (Boticario 2006).

2.1.4. Redes de discriminación redundante

Las redes de discriminación redundantes o memoria dinámica resuelven este problema organizando los casos mediante varias redes de discriminación, cada una con una ordenación diferente de las preguntas. La variedad más común de las redes redundantes incluye en su organización las propiedades de las redes de características compartidas para mantener su tamaño bajo control. (Pérez 2002).

El modelo de memoria dinámica fue desarrollado a partir de la teoría general del científico Schank (conocida como MOP), donde la memoria de casos es una estructura jerárquica de episodios generalizados (EG⁴). La idea básica es organizar los casos que comparten propiedades similares en una estructura más general (el episodio generalizado o MOP). Un EG contiene tres tipos de objetos: normas, casos e índices.

- Las normas, que son características comunes a todos los casos indexados.
- Los casos, que hacen referencia a la situación planteada.
- Los índices, que son las características que diferencian a los casos.

La memoria de casos es una red donde cada nodo es un EG (conteniendo las normas), o un nombre de índice, o un valor de índice, o un caso. Cada par índice-valor apunta desde un EG a otro EG o a un caso. Un valor de índice apunta a un único caso o a un único EG. El esquema de indexación es redundante,

⁴ Se utilizará EG para definir Episodios Generalizados tanto en singular como plural.

puesto que existen múltiples caminos hasta un caso concreto o un EG. Cuando un nuevo caso es dado se busca el mejor emparejamiento y el nuevo caso se coloca en la red comenzando por el nodo raíz. El procedimiento de búsqueda para el almacenamiento de casos es similar al que se utiliza en la recuperación de casos. Cuando una o más características del caso ajustan con una o más características de un EG, el caso es distinguido por el resto de sus características. Normalmente se encuentra el caso que más características tiene en común con el caso de entrada (Carrillo 2007).

Durante el almacenamiento de un nuevo caso, cuando una característica del nuevo caso se ajusta a la de un caso existente se crea un nuevo EG. Los dos casos se distinguen indexándolos con diferentes índices bajo el EG. Si durante el almacenamiento de un caso, dos casos (o dos EG) finalizan bajo el mismo índice, automáticamente se crea un nuevo EG. Por tanto, la memoria de casos es dinámica en el sentido de que las características que comparten dos casos son dinámicamente generalizadas en un EG y los casos son indexados bajo ese EG por sus características diferenciadoras.

Para la recuperación de un caso, se busca el EG que tiene más normas en común con la descripción del problema. Una vez encontrado se recorren los índices bajo el EG para encontrar el caso que contiene más características adicionales en común con el problema.

El almacenamiento de un nuevo caso se realiza del mismo modo, con el proceso adicional de la creación dinámica de EG. Puesto que la estructura de índices es una red discriminatoria, un caso (ó un puntero a un caso) es almacenado bajo un índice que le diferencia del resto de casos. Esto puede hacer crecer de forma explosiva el número de índices a medida que aumenta el número de casos, de forma que la mayoría de los sistemas que usan este esquema de indexación ponen algunos límites a la hora de seleccionar índices para casos, como por ejemplo, sólo permitir un pequeño vocabulario de índices.

El principal rol de un EG es representar una estructura indexada para el emparejamiento y recuperación de casos. Las propiedades dinámicas de esta organización de memoria podrían ser vistas como un intento de construir una memoria que integra conocimiento de episodios específicos con conocimiento generalizado de los mismos episodios.

Por tanto, esta organización del conocimiento es adecuada para el aprendizaje tanto de conocimiento generalizado como de conocimiento específico y es un modelo admisible, aunque simplificado, del razonamiento y aprendizaje humano. (Carrillo 2007).

El algoritmo de búsqueda aplicable es el siguiente:

Algoritmo de discriminación en paralelo en cada red.

Devolver el conjunto unión de los casos encontrados.

Comparar la entrada con cada elemento del conjunto para identificar los mejores casos

2.1.5. Modelo categorías-ejemplares

La filosofía básica de este método consiste en entender los casos como ejemplos del mundo real, no como simple conceptos. En el mundo real no todas las características de los objetos tienen la misma importancia, hay alguna que tienen más peso que otras en la descripción de los mismo, es en esencia lo que este modelo intenta destacar. Además estos ejemplos pueden ser categorizados, se pueden organizar según las categorías a las cuales pertenece. (Duque 2007).

La base de hechos se encuentra dentro de una red estructurada por categorías, casos y apuntadores a índices. Tal como se ha definido cada caso está asociado a una categoría determinada y los índices, teniendo la misma definición que en modelo anterior, son de tres tipos: índices que unen las características de los problemas con los casos o las categorías, también son las uniones entre diferentes casos vecinos, entendiendo por vecinos aquellos casos que solo varían en unas pocas características. El último tipo es el de aquellos que representan la unión entre las categorías y los casos asociados a esas categorías. Es bueno destacar que las características no solamente pueden estar vinculadas a los ejemplos sino que también a las propias características, siendo entonces estas características comunes para todos los ejemplos que formen parte de esa categoría. (Gómez, 2008).

Encontrar un caso en la base de conocimientos que coincida con la descripción del mismo se hace mediante la combinación de las características de la entrada y seleccionando aquellas que sean de la misma categoría, teniendo en cuenta que para seleccionar la categoría nos tendremos que basar en aquella categoría que tenga más características en común con el caso del cual partimos.

Para almacenar un nuevo caso lo que tenemos que hacer es buscar un caso que coincida con el que queremos introducir utilizando para ellos los índices. Si el caso que encontramos solamente tiene pequeñas

diferencias con el caso que ya existe no existe la necesidad de guardado o bien se pueden unir las características de los dos casos, en el caso de que no se encuentre ningún caso, el nuevo caso se introducirá dentro de la base de hechos. (Gómez, 2008). La memoria se compone de una red de estructuras y categorías, semánticas, relaciones, casos y punteros índices. Cada caso se asocia a una categoría. Un índice puede apuntar a un caso o a una categoría, pudiendo ser de tres tipos:

- enlaces de atributos, desde los descriptores de un problema a los casos o a las categorías,
- punteros de caso desde una categoría hasta sus casos atribuidos,
- punteros de diferencias desde un caso hasta los casos vecinos más próximos, con pequeñas diferencias entre ambos.

La búsqueda de casos se realiza recorriendo las características del problema hasta obtener los casos o categorías con mayor semejanza. La inserción de casos se realiza buscando la categoría o categorías donde debe insertarse. En conclusión, se asume que los casos tienen dos componentes: la especificación del problema y la solución. Normalmente, la especificación del problema suele consistir en un conjunto de atributos y valores. Los atributos de un caso deben definir el caso de forma única y deben ser suficientes para pronosticar una solución. La representación puede ser una estructura de datos plana o una compleja jerarquía de objetos. (Carrillo 2007).

2.2. Sistemas Basados en Casos

🚩 Aplicación que utiliza el modelo de Redes de Discriminación - CHEF

Es un planificador basado en casos, cuyo dominio son las recetas de cocina. CHEF crea las nuevas recetas a partir de otras ya conocidas, como respuesta a unos requisitos (sub metas) de platos con ingredientes específicos, sabores particulares, etc. (Hammond 1986.)

Este sistema tiene que construir planes que satisfagan un número dado de metas simultáneamente. Cuando se pide una nueva receta a CHEF con unas características determinadas, lo primero que se hace es buscar un plan (una de las recetas que ya conoce y dispone en su base de casos) que satisfaga el mayor número posible de características o metas que se solicitan. Una vez que ha encontrado ese plan, lo modifica para que se cumpla todas aquellas características que aun no satisface. CHEF altera estas recetas con reglas de modificación y un conjunto de "objetivos críticos". Estas reglas son específicas de la clase de plato de que se trata (frito, soufflé, etc.) y de la característica a modificar. (Hammond 1986.)

Los “objetos críticos”, son los que cambian los tiempos de cocción de los ingredientes, teniendo en cuenta los cambios que ha sufrido la receta inicial. Una vez hechas estas modificaciones, se ha conseguido una receta que debería verificar todas las condiciones que se le han impuesto.

Sin embargo, puede darse el caso de que el resultado no sea correcto. En algunos sistemas de RBC es el usuario el que indica que la respuesta es errónea. En el caso de CHEF, es el propio sistema (a través de un simulador de la ejecución del plan) quien detecta el error. Para CHEF un error en un plan significa dos cosas: la necesidad de reparar la receta, y de “reparar” la comprensión que tiene del mundo que le llevo a cometer el error. Para esas dos tareas, necesita comprender exactamente el motivo por el cual se produjo el fallo. No es suficiente que pueda describir en qué consiste el fallo, sino que debe ser capaz de explicar por qué ese fallo concreto ha ocurrido. CHEF explica el fallo a través de una descripción causal de por qué sucedió.

La explicación describe el fallo, la causa y las condiciones que deben cumplirse para que éste se presente. Ésta explicación tiene dos funciones. Primero describe el problema de la planificación en un vocabulario causal de tipo general que puede usarse para acceder a alguna estrategia general de reparación. Segundo, señala cuáles características han interactuado para causar ese fallo, y por tanto en que características se debería estar atento en un futuro. (Carrillo 2007).

Las estrategias de resolución están indexadas bajo TOP⁵. Los TOP son similares a los MOP, e indexan un tipo particular de errores de planificación que son capaces de reparar. CHEF utiliza la explicación causal para buscar el TOP que tiene las estrategias que reparará el error. Los TOP son estructuras que encierran conocimiento general del dominio. Relacionan el conocimiento acerca de cómo interactúan entre sí los objetivos del caso considerado. Son importantes porque capturan conocimiento a veces independiente del dominio de resolución de problemas.

Una vez el plan se ha corregido, además de indexarlo como solución errónea (para no volverlo a repetir), CHEF también cambia su comprensión del mundo, de forma que sea capaz, en otra ocasión similar, de evitar ese fallo. Su comprensión del mundo cambia creando nuevas abstracciones: una abstracción que describe las características que condujeron al fallo, bajo la cual pone la receta reparada para poderla utilizar en la realización de futuras recetas que no presenten esas características que conducen al fallo. CHEF aprende de los planes que genera así como de sus fallos. (Hammond, 1986).

⁵ TOP en inglés, Topics Organization Packets: Paquetes de Organización Temática

Los nuevos planes que obtiene los indexa tanto por las metas que consiguen como por los problemas que evitan. Al mismo tiempo, al indexar los fallos con las características que los provocaron, es capaz de anticiparse a estos problemas cuando se le soliciten de nuevo recetas con esas características.

✚ Sistema que implementa el modelo de Memoria dinámica - JULIA

Es un sistema de diseño basado en casos que opera en el dominio de planificación de comidas. Como en otros dominios de diseño los problemas se describen en términos de las restricciones que tienen que cumplir, las soluciones describen la estructura y cumplen la mayor cantidad de restricciones posibles. Los problemas son muy extensos y en general no pueden resolverse encontrando un caso anterior que se pueda adaptar para resolver el problema.

Habitualmente los problemas se dividen en partes que se resuelven independientemente. Obviamente se debe tener en cuenta las dependencias entre las partes en que se descompone el problema. Para ello JULIA refuerza el RBC con un proceso de propagación de restricciones, usando las restricciones como índices del diseño. Las restricciones provienen de diversas fuentes, unas del conocimiento general del artefacto a diseñar (ingredientes incompatibles, por ejemplo) y otras del nuevo problema (los ingredientes que quiere emplear el usuario). (Hinrichs 2002).

JULIA debe tener un conocimiento general del tipo de artefacto a diseñar almacenándolo en prototipos de objetos, es decir, los prototipos de objetos se corresponden con los diferentes tipos de comidas que conoce. Cada prototipo tiene una estructura específica y relaciones entre sus partes. El sistema usa los prototipos de comida para proporcionar frameworks como soluciones cuando los casos no constituyen opciones correctas. Además de posibles fallos en el proceso de diseño puede ser afectado cuando el cliente introduce nuevas restricciones, lo cual ocurre de manera habitual en la vida real (Hinrichs 2002).

✚ Sistema que utiliza Modelo categorías- ejemplares - PROTOS:

Implementa clasificación y adquisición de conocimiento basado en casos, dada una descripción de una situación u objeto la clasifica por su tipo. Cuando clasifica un elemento de forma incorrecta, su consultor experto le informa del error y del conocimiento que debe usar para clasificarlo correctamente. El dominio de PROTOS es el de los desórdenes auditivos. Dada una descripción de síntomas y los resultados de algunas pruebas de un paciente, PROTOS determina qué problema auditivo tiene el paciente. (Avesani 2003). El

consultor experto es un otorrino experto, puede emplearse para otras tareas como el reconocimiento del estado emocional de un agente.

Para hacer la clasificación de una situación u objeto, PROTOS busca el objeto o situación, ya conocido, que mejor ajusta a la nueva situación u objeto y le asigna la clasificación del que mejor ajusta al nuevo. El proceso llevado a cabo es, primeramente, determinar la categoría del nuevo problema buscando, en las categorías de problemas auditivos que conoce, aquella cuyas características importantes coinciden con las características importantes del nuevo problema. Luego verifica esta hipótesis intentando ajustar su nuevo caso a ejemplares de la categoría de hipótesis para ver si puede encontrar un buen emparejamiento. Si lo encuentra el proceso acaba. Si no, utiliza los resultados de este proceso de ajuste para seleccionar una hipótesis mejor. Este proceso se guía por su conocimiento sobre los tipos de errores de clasificación más comunes en este dominio. (Avesani 2003).

Cuando se hace una clasificación incorrecta debido a la fuerte coincidencia del caso con los casos de la categoría, PROTOS añade enlaces de diferencia desde la categoría incorrecta a la correcta para evitar cometer el mismo error en el futuro. El proceso de adquisición de conocimiento es dirigido por los fallos del proceso de clasificación. Cuando no es capaz de identificar correctamente la categoría de una entrada, establece una conversación con el consultor experto con el objetivo de añadir nuevo conocimiento y revisar las conexiones de su memoria. La comunicación entre el experto y PROTOS se realiza mediante un vocabulario fijado de relaciones causales, taxonómicas, funcionales y correlacionales. (Avesani 2003; Carrillo 2007).

CASEY

Este sistema diagnostica fallos en el corazón, a partir de una descripción de un paciente con síntomas y produce una red causal de posibles estados internos que pueden llevar a estos síntomas, este sistema es lo más completo de modelo de sistemas de diagnóstico. Almacena una gran cantidad de información en los casos. Para todos los rasgos observados guarda, la explicación causal para encontrar el diagnóstico, como fluir la lista de situaciones en el corazón, y el modelo de fallas para el cual este fue evidenciado en el paciente. Esta situación referencia como generalizar situaciones casuales con los índices primarios para el caso (Pérez 2002).

2.3. Herramientas para crear sistemas que utilizan RBC

Para la realización de este estudio se investigan herramientas para la elaboración de sistemas RBC existentes. Es difícil obtener una herramienta completa y de libre disposición, pues la mayoría de las referencias existentes a este tipo de herramientas están desactualizadas. Existe gran cantidad de información sobre la aplicación teórica del RBC a proyectos en desarrollo, pero muy pocas están disponibles para argumentar de forma objetiva sus capacidades, de forma que a continuación se presentan algunas de las herramientas relevantes.

2.3.1. ART* Enterprise

Producto de la empresa Brightware, una antigua división de Inference Corporation, Uno de los primeros desarrolladores de herramientas de inteligencia artificial. ART* Enterprise se comercializa como una aplicación de desarrollo orientada al objeto y ofrece una variedad de paradigmas de representación, incluyendo: Un lenguaje de programación procesual, Herencia múltiple, polimorfismo y encapsulación de objetos, Reglas, Casos, entre otros elementos. (Home/Ingenegros, 2010).

Incluye una interfaz gráfica de construcción, control de versiones, y posibilidad de incorporar datos de repositorios en la mayoría de formatos DBMS propietarios para desarrollar aplicaciones cliente servidor. Además, ART* Enterprise proporciona un soporte multi-plataforma para la mayoría de sistemas operativos actuales. Así como para plataformas hardware (por ejemplo, se puede desarrollar en una plataforma aplicaciones para otras).Incluso permite crear representaciones de casos jerárquicas complejas, controlar estrechamente el indexado de casos y las estrategias de recuperación. No obstante, también tiene sus inconvenientes, como por ejemplo que los atributos de los casos en general son pares atributo: valor planos y no se ofrece soporte para indexado inductivo. (Home/Ingenegros, 2010).

2.3.2. CBR3

Desarrollado por Inference Corporation, los productos de la familia CBR3 constituyen para muchos usuarios los más exitosos y mayormente trabajados. (Carrillo 2007).

Las herramientas CBR3 están formadas por:

- CBR Express, entorno de desarrollo para bases de casos.
- Case Point, motor de búsqueda en memoria eficiente para BC desarrolladas usando CBR Express.
- Generator, herramienta que automatiza la creación de BC a partir de una colección de ficheros de texto, como ASCII, Ms Word, etc.
- Tester, herramienta que provee una variedad de métricas para desarrolladores de BC
- Case Point Web Server, herramienta que soporta la funcionalidad de Case Point en Internet.

CBR3 usa una estructura de registros simple que son almacenados en una base de datos relacional. Los casos incluyen un título, una descripción, un conjunto de cuestiones con un cierto peso, y un conjunto de acciones. Los casos pueden compartirse a través de la red de una organización. Los casos pueden heredar características de otros casos. Pueden citarse un subconjunto de casos donde todos ellos cumplen ciertas propiedades. (Carrillo 2007).

Para la búsqueda en la BC se usa Case Point, que examina una entrada de texto del usuario en formato libre y se trata de ajustar con los títulos y descripciones de casos almacenados, obteniendo un conjunto de casos posibles junto con su valor de exactitud y una serie de cuestiones. Las respuestas a estas cuestiones permiten reducir el número de casos y guiar a una solución más precisa. Si no se halla una solución satisfactoria, se usa el concepto de caso sin resolver, lo que significa que se guarda la transcripción de consulta de fichero o incluso, se puede mandar un mensaje al administrador del sistema para que verifique el caso. Case Point también permite utilizar reglas que identifiquen palabras claves en el texto de consulta para discriminar la búsqueda y presentar las soluciones ordenando por utilidad o por coste. (Mora 2008).

2.3.3. ReMind

Inicialmente desarrollada con el soporte del programa DARPA de EEUU por la empresa Cognitive System. Los casos en ReMind se representan como pares atributo: valor y ofrece varias posibilidades de recuperación:

- Por el vecino más próximo, con ayuda de la importancia definido por el usuario para las características de cada caso.
- Por patrones, soportando consultas SQL simples.

- Recuperación inductiva, bien automáticamente sin interacción del usuario, o bien inductiva guiada por el conocimiento, donde el usuario crea un modelo cualitativo para guiar el algoritmo de inducción con un cierto trasfondo de conocimiento.
- Por medio de modelos cualitativos que se crean gráficamente para indicar cuales conceptos dependen de otros. Al mismo tiempo asignar un peso a las dependencias según su importancia.

Cuando el algoritmo de inducción se guíe por estos modelos, los árboles de decisión serán un mejor reflejo de la relación causal de los conceptos en los casos. Con respecto al proceso de adaptación, simplemente reseñar que se utilizan ciertas formulas de adaptación que ajustan los calores basándose en las diferencias entre el caso recuperado y el nuevo caso, utilizando para ello un método de sustitución. Además, cabe destacar la posibilidad de importar bases de casos de Bases de Datos. No obstante, un punto negativo es el tratamiento de textos en formato libre, no tan bueno como los métodos de algunas de las aplicaciones ya mencionadas. (Carrillo, 2007).

Consideraciones Parciales

Los sistemas que utilizan RBC dependen de la estructura y organización de la memoria de los casos; siendo directamente proporcional el incremento de la utilización de estos sistemas en la actualidad y el desarrollo de diferentes modelos de organización de las base de casos para facilitar la eficiencia y optimización. Estos modelos propician una estructura de los casos viable para los procesos de recuperación de las experiencias en memoria (casos). Presentan características específicas teniendo en cuenta la forma de organización que proponen, con el objetivo de perfeccionar el proceso de acceso, recuperación y almacenamiento de la BC.

Capítulo 3 Presentación de la Solución Propuesta

Los modelos de organización de las bases de casos han sido ampliamente utilizados en sistemas con aplicaciones en diferentes sectores de la sociedad, debido a las ventajas que ofrecen se trabaja en vías de su desarrollo. Actualmente se mezclan diferentes técnicas o aplicaciones para aprovechar las ventajas que cada una de estas ofrecen y obtener sistemas cada vez más eficientes. (Cataldi, 2006). En este capítulo se describen dos modelos jerárquicos de organización de la memoria de casos como propuesta de solución al tipo de organización que implementa la herramienta HESEI, con el objetivo de mejorar su eficiencia y ampliar su utilización.

3.1. Modelos Jerárquicos para la organización de Bases de Casos

El modelo de organización de la memoria que utilizan las aplicaciones desarrolladas a partir de la herramienta HESEI es eficiente en sistemas sencillos, no siendo así para aquellos sistemas que presentan base de casos con grandes volúmenes de información. En estos últimos las estructuras jerárquicas son eficientes, a pesar de que el proceso de adicionar un nuevo caso en la memoria no es tan simple como en las secuenciales. Debido a lo antes expuesto se propone adicionar dos modelos jerárquicos al módulo de organización de BC de la herramienta computacional HESEI, con el objetivo de aprovechar las ventajas que aportan ambos esquemas de organización de memoria.

Es necesario destacar que los modelos de organización jerárquicos que se proponen como solución no sustituyen el que utiliza HESEI actualmente, sino que complementan esta forma de organización, dándole al usuario la facilidad de escoger según el tipo de sistema que se desarrolla el modelo de organización de la BC factible (plano o jerárquico) que facilite el acceso y recuperación de los casos en memoria, permitiendo la optimización y eficiencia de los sistemas. Al proponer más de un modelo jerárquico el usuario puede comparar y finalmente seleccionar cuál cumple mejor sus expectativas, véase figura 5.

Se proponen los modelos jerárquicos Registro con Encabezamiento y el de Memoria Asociativa Indexada Conexiónista pues constituyen modelos jerárquicos sencillos, fiables y flexibles respecto a la representación de la información. Su implementación es viable sin complejidades, no requieren de

procedimientos costosos (reestructuración de jerarquía de abstracción, análisis léxico- lógico, entre otros).

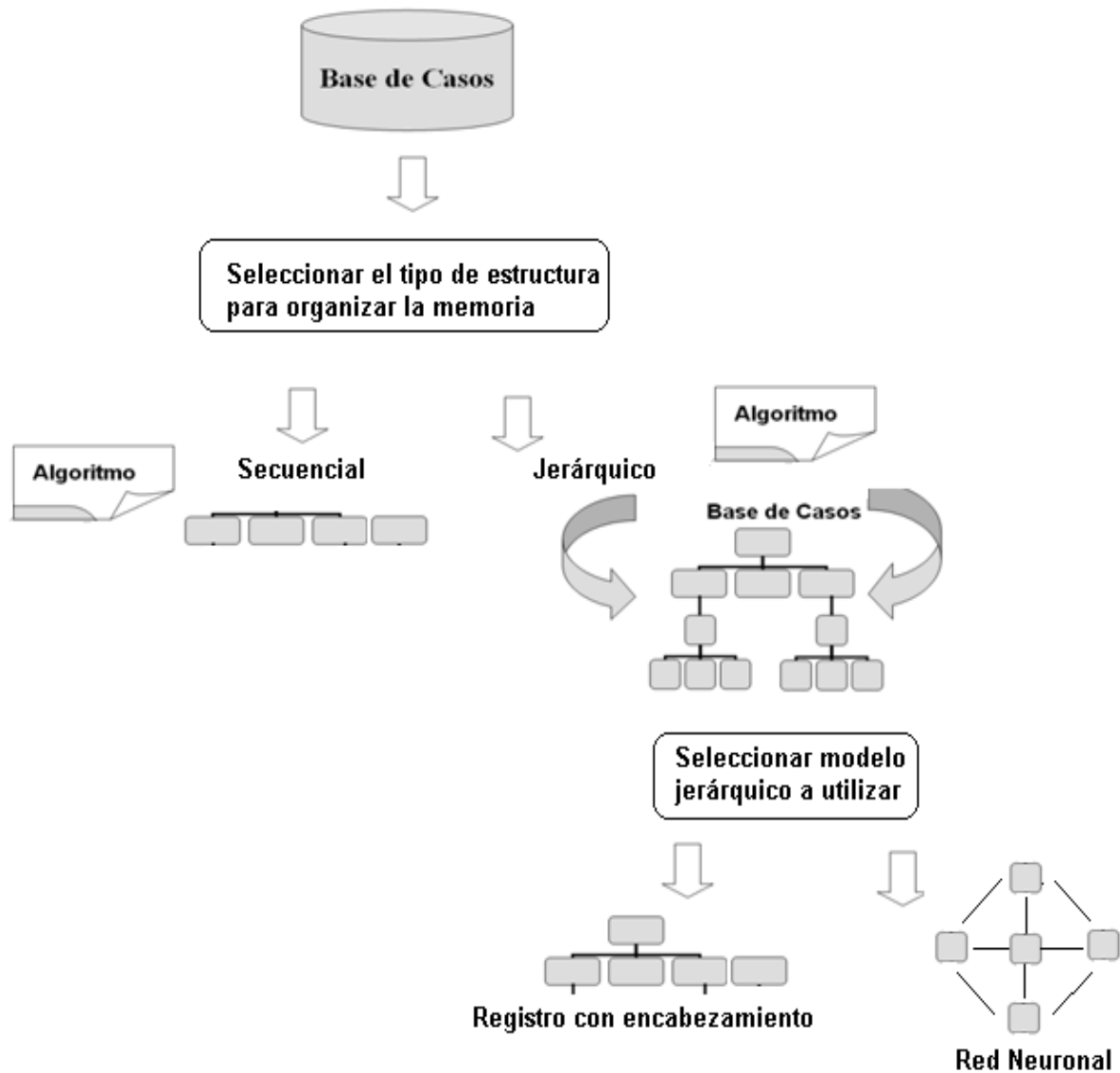


Figura 4 Esquema general de la construcción del modelo de la Base de Casos.

3.2. Selección de rasgos en la definición del modelo de la base de casos

La selección de rasgos es cuestión central tanto en la definición del modelo de la base de casos como en el modelo de recuperación de casos. Potencialmente, en el conjunto de casos podrían estar todas las propiedades que describen los objetos; pero existen rasgos inútiles que carecen de importancia de acuerdo al dominio de aplicación. El conjunto de rasgos determina qué información será almacenada en memoria para cada uno de sus elementos, la cual debe permitir la posterior recuperación de los casos semejantes dada la descripción de un nuevo problema. Los casos en la BC de la herramienta HESEI representan el estado del conocimiento y comportamiento del estudiante, así como el entrenador o material didáctico más adecuado.

Cada caso es un ejemplo de modelado de estudiante, el cual se divide en modelo del estudiante (rasgos predictores) y los materiales didácticos más adecuados para ese modelo (rasgo objetivo). Los rasgos predictores, contienen los datos de entrada, o sea la información a partir de la cual el sistema infiere el estado del estudiante, ya sea cognitivo, afectivo-motivacional u otras características que se consideren importantes a tener en cuenta en el SEAI que se desarrolla. El rasgo objetivo es un rasgo multievaluado, los valores del mismo se corresponden con los materiales didácticos propuestos para ese modelo de estudiante, sugiriendo un orden. (Martínez 2010).

Dentro del conjunto de rasgos que se seleccionan no todos tienen la misma importancia, cada rasgo tiene un valor asociado y una medida de certeza. Dentro de los criterios más comúnmente utilizados en el cálculo de la importancia de cada rasgo están: criterio de los especialistas del dominio de aplicación, dispersión de los valores del rasgo, frecuencia del valor dado al rasgo, carácter diferenciante del rasgo, fuerza predictiva del rasgo, entre otros. De forma general el problema de selección de rasgos consiste en encontrar el subconjunto de rasgos que mejor describe los objetos del dominio; usualmente este subconjunto se encuentra maximizando o minimizando una función objetivo.

Se reportan trabajos en la literatura para seleccionar rasgos relevantes con diferentes enfoques y técnicas, tales como: técnicas de optimización de colonias de hormigas (Jensen and Qiang 2003), (Bello, Nowe et al. 2005), (Bello, Nowe et al. 2005), las de computación evolutiva (Wróblewski 1995), (Sánchez, Lazo et al. 1999), (Wróblewski 2001), (Martínez, Sánchez et al. 2002), (Sánchez, Barandela et al. 2003), (Guevara 2004), las basadas en la teoría de los conjuntos aproximados (Kuo and Yajima 2003), (Mi, Wu et

al. 2003), (Korzes and Jaroszewicz 2005), (Caballero and Bello 2006) y el enfoque lógico combinatorio (Lazo, Ruiz et al. 2001), (Carrasco, Ruíz et al. 2004), (Ruiz 2005), entre otros.

Para realizar las comparaciones entre las descripciones de problemas y los rasgos predictores se utilizan las funciones de semejanzas implementadas en HESEI.

3.3. Modelo Registros con Encabezamiento

El modelo tiene como principio que en la memoria se almacenen dos clases de información, una clase sirve para realizar la recuperación del otro tipo de información; es decir, todo elemento de memoria tiene dos partes, una de ellas sirve para realizar la recuperación de la otra. Atendiendo a esta división de la información almacenada en los elementos de memoria éstos se representan en el modelo mediante dos registros, un registro cabeza y un registro de información. (Duque 2007).

La información almacenada en un registro cabeza (RC) se denomina Información Indicadora (II) y la contenida en el otro registro Información Deseada (ID). Se sitúa en el registro cabeza información indicadora sobre el problema, que constituyen rasgos seleccionados a partir del criterio de un especialista en el dominio de aplicación o a partir de la fuerza predictiva del rasgo o por el carácter diferenciante de los rasgos. El RC contiene parte de la información que describe el caso, y permiten realizar comparaciones mediante funciones de semejanza entre él y la descripción del nuevo problema.

Se tiene toda la descripción del caso contenida en el registro de memoria (RM). La información almacenada en una memoria no necesariamente sirve como II pudiendo ocurrir que la II no tenga una relación explícita con el contenido de la memoria, por ejemplo, en ocasiones la II de una ID está determinada por algún elemento que forma parte del entorno, donde ocurrió el evento descrito en la ID; elementos de información como lugares, olores y sonidos son típicamente II de esta clase, pues cuando aparecen hacen recordar algunos elementos de la memoria. Realmente las condiciones bajo las cuales un rasgo particular de una experiencia se vuelve una II es una cuestión empírica la cual es susceptible de análisis experimentales. (Duque 2007).

La definición formal del modelo está dada por las reglas siguientes:

- R1. El conocimiento se define por pares (RC, RM).
- R2. Los RC son independientes.
- R3. Los RC no son modificables.

- R4. La recuperación de un RM implica primero la comparación entre una descripción de un problema y un RC.
- R5. Un RM puede ser accedido solamente si un RC ha sido seleccionado.
- R6. La búsqueda permite la recuperación de todos los RM asociados al RC seleccionado.
- R7. Los RC no son recuperables.
- R8. Todo lo que se percibe se interpreta a partir del conocimiento existente.
- R9. Nuevos registros se crean a partir de nuevas percepciones (viejos registros o relaciones).
- R10. El sistema opera a partir de una descripción existente.

Estas reglas formalizan el hecho de que la memoria queda constituida como un conjunto discreto de registros en los cuales se divide la experiencia, y todo el proceso de búsqueda se realiza a partir de una descripción de la nueva percepción (o problema en el RBC), la cual se compara con los RC, siendo estos la única vía de acceder a los RM.

El modelo propuesto se basa en el de organización de registros con encabezamiento; pero se modifica en lo que se refiere a que cada RC tiene asociado un sólo RM, que permite el ahorro de tiempo en procesamiento pues desde la primera iteración de búsqueda se recuperan todos los RM correspondientes. Y mayor eficiencia pues el módulo responsable de determinar si se satisfacen los requerimientos del nuevo problema, ahora lo decide analizando todos los RM recuperados con un mismo RC.

3.4. Modelo de Memoria Asociativa Indexada Conexionista

En 1989 (Greene 1989) Greene propuso un método interesante de implementar un modelo de memoria que utiliza índices, y además, que permite la búsqueda asociativa conocido como Memoria Asociativa Indexada Conexionista. Es un modelo de memoria que utiliza índices, permite la búsqueda asociativa. La esencia del modelo está en mantener los elementos de memoria en diferentes listas, cada una de las cuales coleccionan elementos semejantes como si ellas representaran categorías conceptuales.

Mediante una red neuronal se determina la dirección de la lista en la cual debe insertarse un nuevo elemento de la memoria o deben buscarse los elementos semejantes al problema (dado como patrón para la búsqueda asociativa) a resolver.

El modelo de red neuronal posee dos capas, la capa de entrada con n neuronas donde:

$$n = \sum_{i=1}^N \text{Tamaño } (i)$$

En esta expresión N denota la cantidad de rasgos que caracterizan las unidades de información que se almacenan (registros). Cada rasgo se representa por un arreglo de unidades simples de información, la cantidad de unidades simples necesarias para representar los valores del dominio de un rasgo es a lo que se llama Tamaño del rasgo, en la red existe una neurona de entrada por cada unidad simple requerida para los rasgos. La segunda capa del modelo de red neuronal es la de salida, donde la cantidad de registro a almacenar se espera que sea m, int indica la parte entera superior, el número de neuronas en la capa de salida está dado por:

$$m = \text{int} (\log_2 M)$$

Se establecen conexiones de cada neurona de entrada a cada unidad de salida con pesos w_{ij} iniciales que se asignan teniendo en cuenta la frecuencia de aparición.

El mecanismo de aprendizaje es no supervisado competitivo y cooperativo, ya que la red no recibe ninguna información por parte del entorno que le indique si la salida generada en respuesta a una determinada entrada es o no correcta. Se selecciona este aprendizaje, pues el objetivo de éste es categorizar los datos que se introducen en la red.

Las clases o categorías se crean por la propia red, puesto que se trata de un aprendizaje no supervisado. Esta red neuronal, se utiliza como una función hash para almacenar y recuperar datos. Se plantea que dado un vector de entrada (x_1, \dots, x_n) las salidas (y_1, \dots, y_n) se calculan por:

$$Y_j = f \left(\sum_{i=1}^n w_{ji} * x_i + B_i \right) \quad \text{donde} \quad f(u) = \begin{cases} 0 & \text{si } u < 0; \\ 1 & \text{si } u > 0 \end{cases}$$

El vector Y constituye una representación binaria de un índice a un arreglo de cabezas de listas enlazadas. Cuando se desea almacenar la unidad de información que corresponde al vector de entrada X, el vector Y representa la dirección de la lista enlazada en la cual se debe insertar la unidad de información. Si lo que se

desea es recuperar, si el vector X está libre de ruidos el vector Y representa la dirección de la lista donde se debe buscar la información deseada (si está en memoria). Si la entrada tiene ruidos o rasgos no especificados en la red, la salida es una dirección cercana a la posición de la lista donde debe o debería estar la unidad de información que se busca.

La arquitectura de la red neuronal propuesta es de dos capas, donde el número de neuronas de la capa de entrada y de la de salida se calculan según Greene. El aprendizaje está dado por el algoritmo competitivo; sólo una unidad de salida está activa en cada momento. Todas las neuronas de la red competitiva reciben idéntica información de las unidades de entrada pero las unidades de salida compiten entre sí para ser la que se activa como respuesta a esa señal de entrada.

Con este tipo de aprendizaje, se pretende que cuando se presente a la red cierta información de entrada, sólo una de las neuronas de salida de la red, o una por cierto grupo de neuronas, se active (alcance su valor de respuesta máximo). Por tanto, las neuronas compiten por activarse, quedando finalmente una, o una por grupo, como vencedora, quedando anuladas las restantes, que son forzadas a sus valores de respuestas mínimos. La competencia entre neuronas se realiza en todas las capas de la red, existiendo en estas neuronas, conexiones recurrentes de autoexcitación y conexiones de inhibición (signo negativo) por parte de neuronas vecinas.

El objetivo de este aprendizaje es categorizar los datos que se introducen en la red, de esta forma las informaciones similares son clasificadas formando parte de la misma categoría, y por tanto deben activar la misma neurona de salida. Las clases o categorías deben ser creadas por la propia red, puesto que se trata de un aprendizaje no supervisado, a través de las correlaciones entre los datos de entrada.

Ventajas de la utilización del modelo propuesto:

1. Capacidad de adaptación y auto organización de la información.
2. Memoria de los casos distribuida
3. Procesamiento paralelo
4. Capacidad de generalización de los datos.

Los elementos antes expuestos posibilitan el perfeccionamiento no solo del proceso de recuperación de los casos en memoria sino también el proceso de adaptación. La utilización de las RNA permite además aprovechar la capacidad de aprender a partir de variables que identifican el problema, extrayendo los

datos necesarios para generar un modelo y una red capaz de resolverlo y, sobre todo, partiendo de un conocimiento mínimo de la esencia del problema se obtiene una solución óptima.

3.2. Validación de la Solución Propuesta

Resulta de interés analizar la eficiencia y optimización de los modelos jerárquicos para organizar las bases de casos que se proponen como solución, respecto al modelo lineal que implementa HESEI. Para ello se realiza un experimento analizando el comportamiento de ambos esquemas organizativos en el proceso de recuperación de los casos relevantes al problema. Para ilustrar los resultados de la validación de los modelos propuestos se toman como muestra dos BC correspondientes a dos SEAI desarrollados y dos BC experimentales que responden a características típicas de BC y formas de organización como aparecen descritas en la Tabla 1.

La BC: Estructura de Datos (ED) se obtiene de un SEAI que está en proceso de explotación, dirigido por el colectivo de profesores de la asignatura: “Estructura de Datos y Algoritmos I” de la carrera de Ciencias de la Computación, en el tema de “TDA Listas”, Universidad de Las Villas. (Martínez 2010).

La BC: Grafos es la correspondiente al SEAI que se desarrolló para apoyar el estudio de la teoría de grafos a estudiantes que cursan el segundo año de la carrera de Ciencias de la Computación y que ha sido utilizado como material complementario para el estudio independiente de este tema.

Tabla 1. Características generales de las bases de casos.

Base de Casos	No. de Casos	Rasgos Predictores	Número de Clases
1. Estructura de Datos	84	13	4 (5/4/3/2)
2. Grafos	37	7	3 (B/R/M)

En este estudio, las BC: Estructura de Datos y Grafos se organizan de forma jerárquica, para cada par de BC (organizada en una estructura plana y organizada en forma jerárquica). El análisis se realiza para 10 casos seleccionados de forma aleatoria entre los almacenados en las BC, utilizando como algoritmo de selección el cálculo de la importancia de los rasgos.

Dado un nuevo problema, se realiza la recuperación de los 10 casos más semejantes utilizando la función de semejanza que opera en la herramienta HESEI.

$$\beta(O_0, O_t) = \frac{\sum_{i=1}^n w_i \cdot \delta_i(x_i(O_0), x_i(O_t))}{\sum_{i=1}^n w_i}$$

w_i denota la importancia de los rasgos a la hora de diagnosticar el estado cognitivo - afectivo de un estudiante, donde la función de comparación de los rasgos es:

$$\delta_i(x_i(O_0), x_i(O_t)) = \begin{cases} 1 & \text{si } x_i(O_0) = x_i(O_t) \\ 0 & \text{e.o.c} \end{cases}$$

La función toma valor 1, si los rasgos son iguales y 0 en otros casos.

Se comparan estos dos conjuntos de casos, así como también las decisiones (elementos que se resaltan en la Tabla 2 y en la Tabla 3) que se proponen en el proceso de adaptación en condiciones de incertidumbre. La información relacionada con el experimento se refleja en las siguientes tablas.

Tabla 2. Casos seleccionados para la Adaptación para la BC de ED.

Problema	ESTRUCTURA DE DATOS (ED)			
	Casos Seleccionados para la Adaptación (10 vecinos)			
	ESTRUCTURA PLANA	#	ESTRUCTURA JERÁRQUICA	#
Caso 3	54,57,65, <u>34</u> ,41,46,47,19,13,23	10	<u>34</u> , 47, 23	3
Caso 11	15, <u>69</u> ,51,73,74,68,17,53,67,70	10	53,67,70,15,51, <u>69</u> ,74,73, 68,17	10
Caso 17	11,23,9,19, <u>48</u> ,12,18,67,22, 84	10	84,67,54, <u>48</u> ,18,22,11, 9,19,23	10
Caso 18	33,49,53,41,29,37,14,19,21, <u>43</u>	10	<u>43</u> ,33,41,49,53,29,37,14,19,21	10
Caso 42	26,34,70, <u>79</u> ,17,22,76,18,31,38	10	70, <u>79</u>	2
Caso 55	23,79,1, 9,5,12, <u>38</u> , 50, 51,59	10	26,12, <u>38</u>	3
Caso 59	<u>28</u> ,72,10,43,65,67, 20,6,21,22	10	<u>28</u> ,47,73	3
Caso 61	57,60, <u>65</u> ,34,41,47,73,18,34,68	10	57,60, <u>65</u> , 34,41,47,73,18,34,68	10
Caso 71	11,13,54,62,9, <u>79</u> ,81,18,19,22	10	<u>35</u> ,13,54,62,10,18,22,11,9	9
Caso 84	12,15,90, <u>33</u> ,38,43,18,12,20,35	10	20, <u>33</u> , 38, 12, 52, 58	6

Tabla 3 Casos seleccionados para la Adaptación para la BC de Grafos.

Problema	GRAFOS			
	Casos Seleccionados para la Adaptación (10 vecinos)			
	ESTRUCTURA PLANA	#	ESTRUCTURA JERÁRQUICA	#
Caso 3	4, 35,11,21,37,8,7, <u>32</u> ,9,8	10	4,35,11,21,37,8,7, <u>32</u> ,9,8	10
Caso 8	1, <u>13</u> ,23,7,18,26,19,2,15,21	10	1, <u>13</u> ,7,18,26,19,2,15	8
Caso 15	8, 5, 6, 34, 10, 28,36, <u>12</u> , 16	10	4, <u>12</u> ,16	3
Caso 17	6,18,2,3,6,7,4, <u>10</u> ,8,14	10	6,3,4, <u>10</u> ,14	5
Caso 19	12,14, <u>29</u> ,9,1,10,2,11,14,16	10	12,14,11,14, <u>29</u> ,9,10,1,2,16	10
Caso 25	6,18,2,3,6,7, <u>4</u> ,10,8,14	10	6,3, <u>4</u> ,10,14	5
Caso 27	8,4, 5, 6, 9, 10, <u>28</u> , 36,15,16	10	4, <u>28</u> ,16	3
Caso 31	5,8,14,27,22,32, <u>11</u> ,8,9,1	10	9,27, <u>11</u>	3
Caso 33	1,16, <u>21</u> ,20,34,12,16,24,15,31	10	20,1,16, <u>21</u> ,34,15,31,18	8
Caso 36	13,21, <u>33</u> ,3,7,9,14,25,19,6	10	13, <u>33</u> ,9,14,19,6	6

Las Tablas 2 y 3 muestran que un elevado por ciento del conjunto de casos potenciales de la BC organizada jerárquicamente, que se utilizan para la adaptación, están incluidos en el conjunto de casos que se usan cuando se realiza la búsqueda en la BC organizada en una estructura plana.

La cantidad de casos con los que se compara en la estructura plana es siempre 10. Con la estructura jerárquica, para cada caso se realizan como máximo 10 comparaciones. Ambas tablas muestran casos en los que este número se reduce considerablemente. La reducción en el número de comparaciones es significativa, es un resultado bueno, pero no suficiente. Debe demostrarse además que la calidad de la respuesta no se afecta.

Para probarlo, se cuantificó la cantidad de veces en los que la respuesta coincidió (19 veces) y la cantidad en las que no coincidió (una vez). Se aplicó una prueba “chi cuadrado” de bondad de ajuste y el resultado fue altamente significativo (0.000), lo que demuestra que la cantidad de veces en las que hubo coincidencia, es significativamente mayor que la cantidad de veces en los que los resultados no coincidieron. Puede concluirse que la calidad de la respuesta en la estructura jerárquica no se afectó.

Para aplicar esta prueba estadística, solo se tuvo en cuenta si los casos coincidían o no. Cabe señalar que solo cuando se presenta como nuevo problema el caso 71 (Tabla 2) la solución propuesta no coincide; al calcular el grado de semejanza entre ambas propuesta utilizando la función de semejanza se obtiene el valor 0.92, lo que muestra que no es grande la diferencia entre las soluciones dadas.

Esto significa que la estructura propuesta aumenta la eficiencia en el acceso, al disminuir el número de casos que se utilizan en el proceso de recuperación, conservando la eficiencia en la búsqueda de soluciones.

Consideraciones Parciales

Con vías a perfeccionar la herramienta HESEI y ampliar su utilización en el país no sólo en el ámbito educativo sino también en otros sectores de la sociedad cubana, se propone la implementación de los modelos jerárquicos descritos en este capítulo para la optimización y eficiencia de los sistemas de enseñanza - aprendizaje inteligentes elaborados con la herramienta HESEI.

Conclusiones

El desarrollo de la investigación científica ha permitido llegar a las siguientes conclusiones generales:

- Se realizó un estudio del estado del arte sobre los sistemas basados en casos, sus aplicaciones, desarrollo en la actualidad, técnicas y herramientas que utilizan, así como los modelos de organización de las bases de casos que implementan, que permitió realizar un análisis sobre la estructura de organización de la memoria de casos que utiliza HESEI.
- Se seleccionó el esquema jerárquico para complementar el módulo de organización de la base de casos de la herramienta computacional HESEI, con el objetivo de aprovechar las ventajas que aporta esta forma de organización de la memoria de los casos.
- Se proponen los modelos Registro con Encabezamiento y Memoria Asociativa Indexada Conexionista como modelos jerárquicos para la organización de las bases de casos en aplicaciones desarrolladas a partir de HESEI que permiten la eficiencia y optimización de estas aplicaciones.
- Finalmente se realizó la validación de la solución propuesta mediante comparaciones realizadas en la recuperación de casos en memoria, utilizando base de casos de sistemas experimentales con estructuras jerárquicas y lineales, que permitieron validar satisfactoriamente la calidad de los modelos propuestos.

Recomendaciones

Como resultado del proceso de investigación han surgido ideas que serían recomendables tener en cuenta para el perfeccionamiento de HESEI, a continuación se enuncian las mismas:

- Implementar los modelos de organización jerárquicos de la base de casos propuestos como solución al problema de investigación.
- Ampliar la utilización de la herramienta HESEI a otros sectores sociales, no solo en el ámbito educativo.

Bibliografía y Referencias

1. Alonso, C. M., Gallego, D. J. , Ed. (1994). Los Estilos de Aprendizaje: Procedimientos de Diagnóstico y Mejora., Bilbao: Ediciones Mensajero, 1994.
2. Avesani, P., Ferrari, S. y Susi, A. (2003). "Case-Based Ranking for Decision Support REC." (ICCBR 2003: 35-49.).
3. Bello, R. (2002). Aplicaciones de la Inteligencia Artificial.
4. Bello, R. (2002). "Aplicaciones de la Inteligencia Artificial." Ediciones de la Noche, Guadalajara, Jalisco, México. ISBN: 970-27-0177-5.
5. Bello, R., A. Nowe, et al. (2005). "A model based on Ant Colony System and Rough Set Theory to Feature Selection." Genetic and Evolutionary Conference (GECCO05), Washington, USA.
6. Bello, R., A. Nowe, et al. (2005). "Using ACO and Rough Set Theory to Feature Selection." 6th WSEAS Evolutionary Computing Conference (EC05), Lisbon, Portugal.
7. Bello, R. M. (2000). "Modelos para el razonamiento basado en casos en presencia de rasgos borrosos. ." Reporte de investigación terminada. Código 001.535 Bell M, CDICT UCLV.
8. Bonzano, A. (1998). "ISAC: a Case-Based Reasoning System for Aircraft Conflict Resolution." A thesis submitted to the University of Dublin, Trinity College, for the degree of Doctor in Philosophy.
9. Bonzano, A. (1998). "ISAC: a Case-Based Reasoning System for Aircraft Conflict Resolution." A thesis submitted to the University of Dublin, Trinity College, for the degree of Doctor in Philosophy."
10. Boticario, P. J. G. (2006). "Representación y uso del conocimiento." from https://adenu.ia.uned.es/web/en/members/jgb/teaching/etsi_informatica/ia_e_ingenieria_del_conocimiento/contenidos_a/contenido/representacion_y_uso_del_conocimiento.
11. Riesbeck, R. S. H., New Jersey: (1989) Inside Cased-Based Reasoning. . Lawrence Erlbaum Associates Publishers, 1989. Volume, DOI:
12. Caballero, Y. and R. Bello (2006). "Two new feature selection algorithms with Rough Sets Theory." Artificial Intelligence in Theory and Practice. M. Bramer, Springer Boston. 217: 209-216.

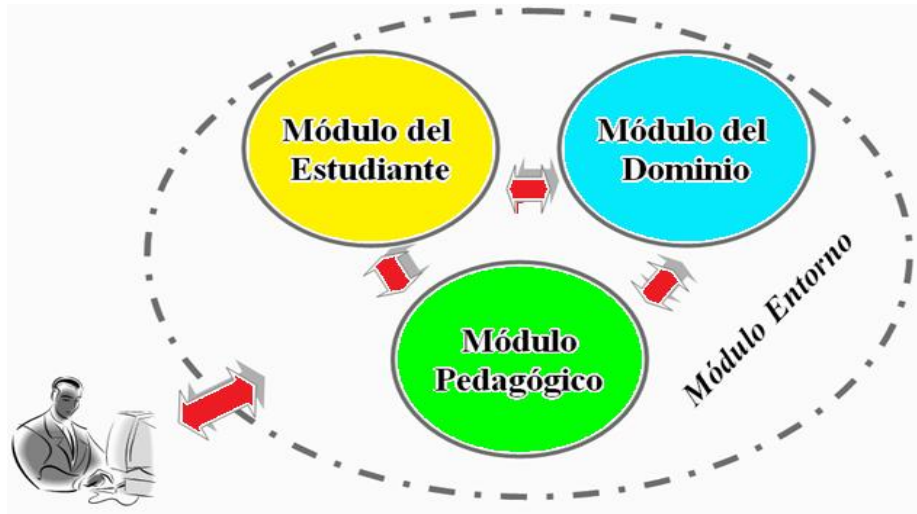
13. Carrasco, J. A., J. Ruíz, et al. (2004). "Feature Selection Using Typical e -Testors." Working on Dynamical Data. Progress in Pattern Recognition, Image Analysis and Applications, Puebla, Mexico, Springer Verlag.
14. Carrillo, J. A. (2007). "Introducción al Razonamiento Basado en Casos (CBR) ."
15. Duque, N. (2007). "Propuesta de Doctorado en Ingeniería de Sistemas". : Sede en Medellín. pp. 34. Universidad Nacional de Colombia. .
16. Fortún F, S. d. R. (2007) Metodología para la elaboración de materiales didácticos adaptativos para la Contabilidad Básica en las Sedes Universitarias Municipales. . Volume, DOI: XVI Forum de Ciencia y Técnica Municipal Santa Clara.
17. Freeman, J. A. S. (1993) Redes Neuronales. Algoritmos, aplicaciones y técnicas de propagación. Volume, DOI: México: Addison-Wesley
18. García, M. R. B. (1997). El empleo del razonamiento basado en casos en el desarrollo de Sistemas basados en el conocimiento para el diagnóstico. UCLV, Cuba.
19. Guevara, M. E. (2004). "Algoritmo Genético para la selección de variables y cálculo de su importancia informacional, usando el concepto de FS-Testore Difuso." Benemérita, Universidad Autónoma de Puebla.
20. Hammond, K. J. (1986.) CHEF: Un modelo de la planificación a la sentencia. Volume, DOI:
21. Hinrichs, T. (2002). "Uso de Razonamiento Basado en Casos para las tareas de diseño plausible."
22. Hinrichs, T. s. I. A. d. I. A. t. s. s. b. e. e. a. d. d. (2002.) Uso de Razonamiento Basado en Casos para las tareas de diseño plausible. . Volume, DOI:
23. Jensen, R. and S. Qiang (2003). "Finding rough sets reducts with Ant colony optimization." UK Workshop on Computational Intelligence.
24. Kolodner, J. L. (1992). "An Introduction to Case-Based Reasoning. Artificial Intelligence " Review 6,.
25. Kolodner, J. L. (1993). "Case-Based Reasoning,." Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1993.
26. Kolodner, J. L. M. K. P. (1993). "Case-Based Reasoning."

27. Korzes, M. and S. Jaroszewicz (2005). "Finding reducts without building the discernibility matrix. ." 5th International Conference on Intelligent Systems Design and Applications (ISDA'05), Wroclaw, Poland, IEEE Computer Society.
28. Kuo, T.-F. y Y. Yajima (2003). "Approximate Reducts of an Information System. Rough Sets, Fuzzy Sets,." Data Mining and Granular Computing. 9th International Conference, RSFDGRC2003, Chongqing, China.
29. Lazo, M., J. Ruiz, et al. (2001). "An overview of the evolution of the concept of testor." Pattern Recognition (34): 753-762.
30. León E., B. R. (2008). Sistema de Enseñanza-Aprendizaje Inteligente Sobre Humedales Para La Educación Ambiental. Conferencia UNIVERSIDAD 2008 Provincial.
31. Maria Victoria Sotolongo, G. G. (2001) Aplicaciones de las redes neuronales. El caso de la Bibliometría. El caso de la bibliometría. Ciencias de la Información. (preprint) 2001; 32(1):27-34 Volume, DOI:
32. Martínez, F., G. Sánchez, et al. (2002). ""GeneticAlgorithm to Compute FS-Testors."" WESEAS Transaction on Systems 1(2): 267-272.
33. Martínez, N. S. (2010). "Herramienta Computacional para Elaborar Sistemas de Enseñanza Aprendizaje Inteligentes."
34. Mi, J.-S., W.-Z. Wu, et al. (2003). "Approaches to Approximation Reducts in Inconsistent Decision Tables." Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing 9th International Conference, RSFDGRC2003, Chongqing, China.
35. Mora, A. M. (2008). "Modelos de Organización de Memoria."
36. Pérez, A. (2002). Memoria Organizacional Basada en Casos. Perez, Alonso. 2002. Memoria Organizacional Basada en Casos. Recife, Brasil. : Revista de Ciencia e, 2002.
37. Ruíz, J. (1993). "Modelos Matemáticos para el Reconocimiento de Patrones." Edit. UCLV.
38. Ruiz, S. R. (2005). "Selección de Atributos mediante proyecciones." Memoria del proyecto de Tesis Doctoral para optar al grado de Doctor en Informática por la Universidad de Sevilla.

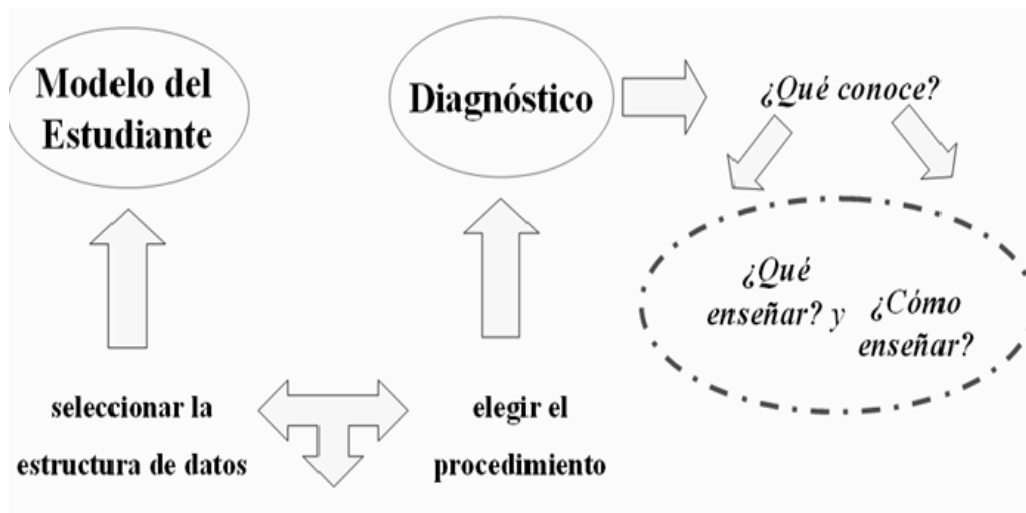
39. Salas, R. (2006) Redes Neuronales Artificiales. Universidad de Valparaíso. : s.n., 2006 Volume, DOI:
40. Sánchez, G., M. Lazo, et al. (1999). ""Algoritmo genético para calcular testores típicos de costo mínimo." "Memorias del IV Simposio Iberoamericano de Reconocimiento de Patrones SIARP'99, Ciudad Habana: 207-213.
41. Sánchez, J. S., R. Barandela, et al. (2003). ""Analysis of new techniques to obtain quality training sets ""." Pattern Recognition Letters 24(7): 1015-1022.
42. Valdivia, M. G. V. L. A. U. L. M. T. M. (2006). "Aprendizaje competitivo LVQ para la desambiguación léxica." Universidad de Jaén. Madrid. España.
43. Wróblewski, J. (1995). "Finding Minimal Reducts using Genetic Algorithm." Proc. of the Second Annual Join Conference on Information Sciences. Wrightsville Beach, NC. Also in: ICS Research report 16/95, Warsaw University of Technology.
44. Wróblewski, J. (2001). "Ensembles of classifiers based on approximate reducts." Proc. of CSP 2000 Workshop, Informatik-Bericht Nr. 140. Humboldt-Universitat zu Berlin (2000). Revised and extended version in: Fundamenta Informaticae 47(3,4), IOS Press (2001).

Anexos

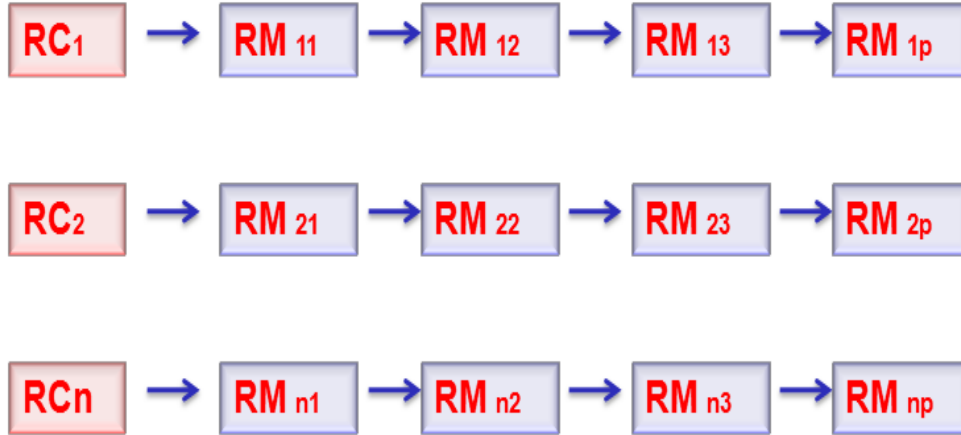
Anexo 1. Arquitectura General de los SEAI elaborados con la herramienta HESEI.



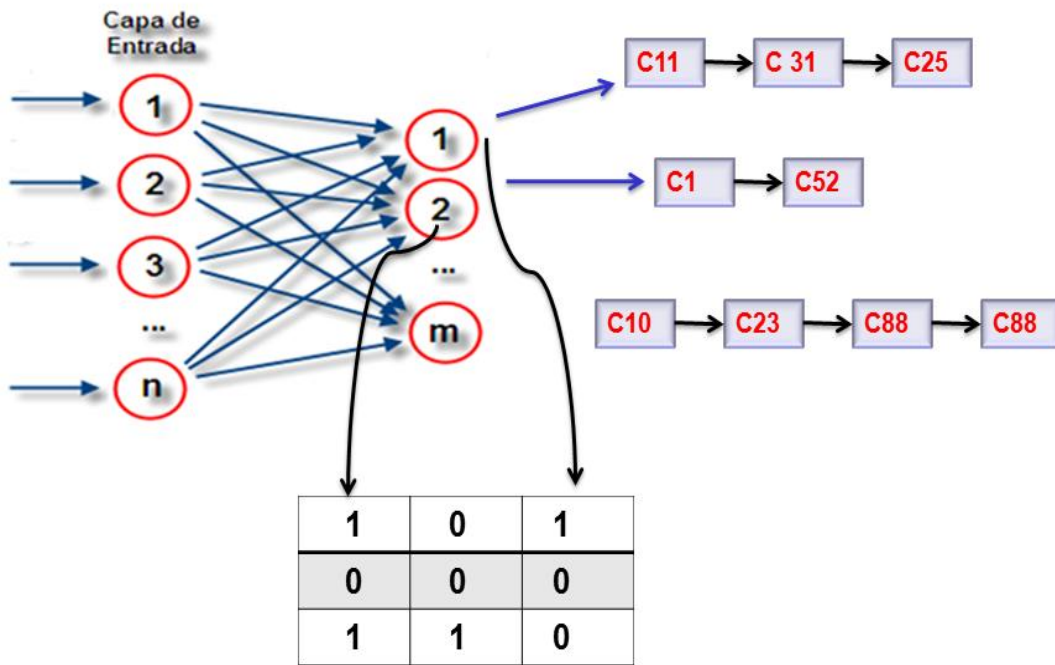
Anexo 2. Esquema que representa el Modelado del Estudiante en HESEI.



Anexo 3. Modelo Registro con Encabezamiento



Anexo 4. Modelo de Memoria Asociativa Indexada Conexionista



Glosario de Términos

BC: Base de Conocimientos

EG: Episodio Generalizado

HESEI: Herramienta para la Elaboración de Sistemas de Enseñanza Inteligentes

IA: Inteligencia Artificial

ID: Información Deseada

II: Información Indicadora

IU: Interfaz de Usuario

MA: Módulo de Adaptación

MI: Máquina de Inferencia

MOP: Paquetes de Organización de Memoria

MR: Módulo de Recuperación

RBC: Razonamiento Basado en Casos

RC: Registro Cabeza

RM: Registro de Memoria

RNA: Redes Neuronales Artificiales

SAT: Sistemas de Aplicación de Test

SBC: Sistemas Basados en Casos

SEAI: Sistemas de Enseñanza-Aprendizaje Inteligentes

SUM: Sedes Universitarias Municipales

TOP: Paquetes de Organización Temática

UCLV: Universidad de Central de Las Villas