INTELLIGENT SYSTEM DESIGN BY USING CASE BASED REASONING

DISEÑO DE UN SISTEMA INTELIGENTE UTILIZANDO RAZONAMIENTO BASADO EN CASOS

MSc. Carolina González[1,2] , Ph.D. Juan Carlos Burguillo[1], Ph.D. Martín Llamas[1]

[1] Universidad de Vigo
E.T.S.E de Telecomunicaciones, Departamento de Telemática, Vigo, España
E-mail:{cgonzals, jrial, martin}@ det.uvigo.es
[2] Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Sistemas, Popayán, Cauca
E-mail: cgonzals@ ucauca.edu.co

Abstract: Intelligent Tutoring Systems (ITS) are educational systems that use artificial intelligence techniques for representing knowledge. ITS design is often criticized for being a complex and challenging process. Motivated by this, this paper focus on the use of Case-Based Reasoning (CBR) as a methodology for building ITS. In the approach, a Multi-Agent System (MAS) is used to automate the reasoning cycle of a CBR system. MAS-CBR facilitates the identification and retrieval of reusable solutions in the design process, and allows the creation of tutoring systems able to personalize the teaching process in different domains.

Resumen: Los Sistemas de Tutoría Inteligente (STI) son sistemas educativos que usan técnicas de Inteligencia Artificial (IA) para representar el conocimiento. El diseño de un sistema tutor es con frecuencia criticado por ser un proceso complejo y altamente desafiante. Por esta razón, nuestro trabajo se enfoca en la utilización de la metodología de Razonamiento Basado en Casos (RBC) para la construcción de STI. La aproximación se presenta como un Sistema Multi-agente (SMA) capaz de automatizar el ciclo de razonamiento. La integración de agentes y razonamiento basado en casos permitirá la creación de sistemas tutores con capacidades reales de adaptabilidad según las características de los estudiantes.

Keywords: Intelligent Tutoring Systems, Health Education, Adaptability, Artificial Intelligence, Multi-agent Systems.

1. INTRODUCCIÓN

En la última década se ha intensificado el interés por introducir las tecnologías de la información en el dominio de la educación. Ello obedece a una mayor preocupación tanto de las instituciones como de los investigadores a la hora de mejorar las prestaciones de docentes y alumnos en los procesos de aprendizaje. En este sentido, las técnicas y paradigmas de la Inteligencia Artificial están generando una especial atención como soluciones a los problemas que surgen al intentar introducir los computadores para apoyar las diferentes estrategias de aprendizaje.

La aplicación de la Inteligencia Artificial en la educación tiene como propósito, la construcción de sistemas inteligentes que puedan funcionar como tutores personalizados para tratar de enseñar al alumno información precisa, adaptando el proceso de aprendizaje según las características del estudiante, y apoyándolo en la búsqueda, tratamiento y organización de la información de una forma eficiente.

Nuestra investigación se centra principalmente en el desarrollo de Sistemas Tutores Inteligentes como sistemas flexibles, interactivos y adaptativos que utilizan técnicas de Inteligencia Artificial como: Sistemas Multi-agente y Razonamiento Basado en Casos y para mejorar los procesos de enseñanza y aprendizaje asistidos por computador.

Este artículo está organizado de la siguiente manera: en la siguiente sección se hace una breve descripción de los Sistemas Tutores Inteligentes. En la sección 3 se introduce el paradigma de Razonamiento Basado en Casos. Posteriormente, en la sección 4, se describen los fundamentos teóricos de los sistemas multi-agente. En la sección 5 se explica el proceso de diseño de sistemas tutores usando RBC. La sección 6, describe los detalles de implementación del STIM-Tutor (Sistema Tutor Inteligente para educación médica). Finalmente, se presentan las conclusiones del trabajo desarrollado.

2. SISTEMAS DE TUTORÍA INTELIGENTE

Aunque no existe una definición precisa de lo que es un STI, hemos considerado la definición propuesta por (VanLehn, 1988):

"Un Sistema Tutor Inteligente, es un sistema software que utiliza técnicas de Inteligencia Artificial para representar el conocimiento e interactúa con los estudiantes para enseñárselo."

Un STI debe tener conocimiento del dominio, es decir, el tema o currículo a enseñar; el conocimiento del alumno, que hace referencia a su conocimiento como usuario del STI; y las estrategias, que hacen referencia a los métodos de enseñanza y a como el material deberá ser presentado al alumno. En la Figura 1 se muestra la arquitectura básica de un STI, y está compuesta de cuatro módulos:

- *Módulo experto o del dominio*: que define el dominio del conocimiento (conocimiento sobre qué enseñar) que satisface dos propósitos diferentes: (a) presentar el material de la forma adecuada para que el alumno adquiera las habilidades y conceptos; (b) resolver los problemas generados y corregir las soluciones presentadas, pudiendo explicar sus razonamientos en un lenguaje comprensible para el alumno.

- *Módulo del estudiante*: que es capaz de definir el conocimiento del estudiante en cada punto durante la sesión de trabajo (conocimiento sobre a quién enseñar). El modelo del estudiante es una representación cualitativa del conocimiento del alumno sobre cierto dominio, y el proceso que manipula esta estructura se denomina diagnóstico.

- *Módulo del tutor*: que genera las interacciones de aprendizaje basada en las discrepancias entre el experto y el estudiante (conocimiento sobre cómo enseñar). En este módulo se definen las estrategias instructoras que afectan la ordenación en la presentación de los contenidos, y las decisiones sobre cuando y cómo intervenir para proporcionar ayuda, explicaciones, preguntas o correcciones al estudiante.

- *Módulo de la interfase con el usuario:* que permite la interacción del estudiante con un STI de una manera eficiente (conocimiento sobre cómo presentar el material).

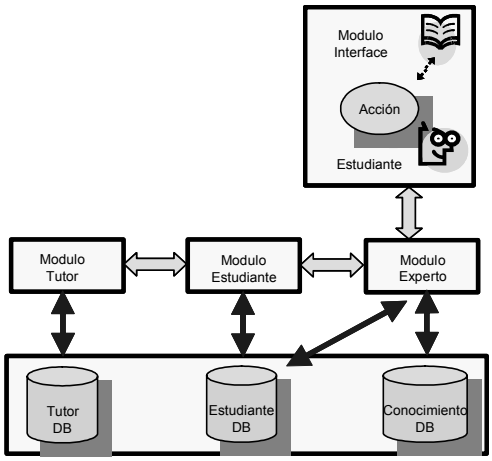


Fig. 1: Arquitectura básica de un STI

El desarrollo de un STI es una tarea compleja debido a que las necesidades individuales de los estudiantes cambian. Por esta razón, un STI debe tener la capacidad de adaptarse dinámicamente a cada alumno y monitorizar su desempeño.

La simple presentación de información no se califica como instrucción (Brusilovsky, 1997). Un STI debería incorporar varias estrategias de aprendizaje. La selección de una estrategia depende de varios factores: (a) el nivel de conocimiento del alumno; (b) el dominio; (c) la motivación; y (d) las características afectivas.

Los sistemas interactivos del tipo STI, en general no proveen de un modo de aprendizaje adaptable (Burns, et al., 1991) de acuerdo a los conocimientos previos y a la capacidad de evolución de cada estudiante. Cada estudiante debería poder elegir las características del método aplicado por el tutor según sus preferencias, y si lo desea debería poder cambiarlo de acuerdo a sus propios requerimientos.

3. RAZONAMIENTO BASADO EN CASOS

En Inteligencia Artificial se han desarrollado varias teorías para razonamiento aproximado que han venido siendo aplicadas en el desarrollo de diferentes STI con el objetivo de mejorar la adaptabilidad de los sistemas. Entre estas aproximaciones están los sistemas basados en reglas, los cuales presentan muchas restricciones durante su desarrollo. Una de ellas es la dificultad en el proceso de adquisición del conocimiento del dominio y del estudiante durante el proceso de aprendizaje, impidiendo el desarrollo de sistemas de grandes capacidades (Dorca, et al., 2003). Algunos de los sistemas encontrados en la literatura del dominio que usan este modelo son NEOMYCIN (Littman, et al., 1988) y GUIDON (Clancey, 1987). Las redes bayesianas (Cataldi, 2000) han sido otro de los principales enfoques utilizados en el proceso de razonamiento del estudiante. Sin embargo a pesar de tener una fuerte solidez teórica presentan algunas desventajas como: (a) esfuerzo en la especificación de los modelos, y estimación de las probabilidades; (b) complejidad computacional de los algoritmos de propagación; y (c) dificultad de implementación. Algunos sistemas como OLAE (VanLehn, 1998), y ANDES (Conato, et al., 1997), consideran la utilización de redes bayesianas en el proceso de modelado del alumno.

Un sistema de razonamiento basado en casos utiliza como base el modelo de razonamiento humano. Este modelo se fundamenta en el hecho de que los humanos utilizan lo aprendido en experiencias previas para resolver nuevos problemas (Plaza, 1993; Schank, 1989). RBC puede ser considerado una buena aproximación en el diseño de sistemas tutores. Por medio de su memoria permanente y los métodos de búsqueda utilizados puede poner a disposición del diseñador un conjunto de experiencias previas y casos resueltos, permitiéndole construir la solución deseada. En la Figura 2 se detallan cada una de las fases dentro de la metodología.

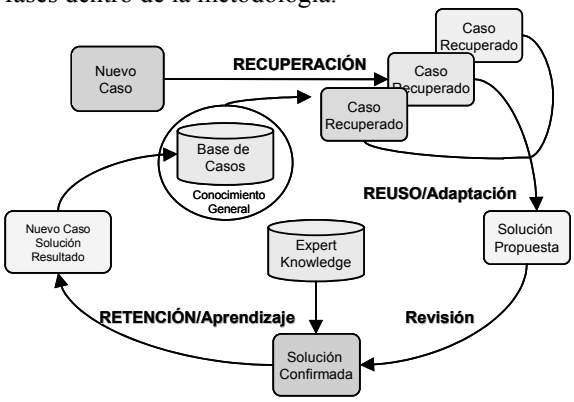


Fig. 2: Fases de RBC (Aamodt and Plaza, 1994)

- *Recuperación de Casos:* El objetivo de esta fase es extraer de la memoria del sistema los casos más similares al problema dado. Los sistemas CYRUS (Kolodner, 1983), ARC (Plaza, 1990) y PATDEX-1 (Rithcer, et al., 1991), utilizan mecanismos de recuperación de casos basados en las similitudes sintácticas existentes entre las características que describen un problema. Por el contrario PROTOS (Barey, 1989), CASEY (Koton, 1989), CREEK (Aamodt, 2004) y MMA (Plaza, 1993), recuperan casos basándose en rasgos con fuertes connotaciones semánticas.

- *Reutilización de Casos*: El objetivo de esta fase es el estudio de las diferencias entre los casos seleccionados en la etapa anterior y el problema presente. En esta etapa también se tienen en cuenta cuáles son las características de los casos recuperados, que pueden trasladarse al presente problema. En problemas simples de clasificación, la diferencia entre las características de los casos se oculta y sólo se tienen en consideración las similitudes. En estas situaciones el presente problema adopta la solución del caso recuperado. Sin embargo, en la mayoría de los sistemas donde las diferencias son importantes, es necesario realizar una adaptación en las soluciones. Algunos métodos evolutivos han sido usados para reuso de casos en (Mántaras, et al., 2005).

- *Revisión de Casos:* En esta fase, la solución propuesta por la etapa anterior se examina minuciosamente. Si la solución es aceptada, será la definitiva y se tendrá en cuenta durante la etapa de aprendizaje, tal como se mostrará en el próximo sub-apartado. Pero si la solución no es satisfactoria, se puede modificar o reparar utilizando conocimiento específico acerca del problema en cuestión. Normalmente la evaluación de una solución se realiza aplicando dicha solución a un problema real, y son sus características las que determinan cuándo se puede llevar a cabo esta evaluación, siendo en ocasiones necesarias el uso de simuladores. Si la solución tiene que modificarse, es necesario identificar y poder explicar los errores cometidos durante su generación, utilizando técnicas basadas en explicaciones. Una vez que los errores han sido identificados, la solución se repara utilizando la explicación del error, lo que garantiza que no volverá a producirse.

- *Retención o aprendizaje de Casos:* Durante esta etapa, se identifican los aspectos que se pueden aprender y, por tanto, incluir en la memoria de un sistema RBC. Los algoritmos de aprendizaje deben de tener en cuenta los resultados de la fase anterior. Para establecer los algoritmos de aprendizaje, es necesario definir si las fuentes de dicho aprendizaje son: (a) las características que describen un problema; y (b) la solución del problema o el resultado obtenido tras la puesta en práctica de la solución.

Actualmente no existe ningún sistema tutor inteligente implementado como un sistema de razonamiento basado en casos. Algunos trabajos que integran el ciclo de RBC como un paradigma de aprendizaje pueden ser revisados en (Shiri, 1998; Skran, 2006).

4. SISTEMAS MULTI-AGENTE

En un sistema multi-agente diversos agentes autónomos trabajan juntos para resolver problemas (Wooldrige, et al., 1995). En este tipo de sistemas cada agente tiene una información o capacidad incompleta para solucionar un problema, no hay un sistema global de control, los datos están descentralizados y la computación es asíncrona. Además de las características básicas que califican un agente (Wooldrige, 2002), mencionamos las principales que caracterizan a un agente inteligente dentro del ambiente RBC.

- *Autonomía:* un agente es completamente autónomo si es capaz de actuar basándose en su experiencia. El agente es capaz de adaptarse aunque el entorno cambie severamente.
- *Adaptatividad:* está relacionado con el aprendizaje que un agente es capaz de realizar, y si puede cambiar su comportamiento basándose en ese aprendizaje.
- *Racionalidad:* el agente siempre realiza «lo correcto» a partir de los datos que percibe del entorno.
- *Reactividad:* un agente actúa como resultado de cambios en su entorno. En este caso, un agente percibe el entorno y esos cambios dirigen el comportamiento del agente.
- *Benevolencia:* asunción de que un agente está dispuesto a ayudar a otros agentes, si esto no entra en conflicto con sus propios objetivos.
- *Sociabilidad:* este atributo permite a un agente comunicarse con otros agentes o incluso con otras entidades.

Algunas aproximaciones que integran RBC y agentes han sido propuestas en areas como: (a) comercio móbil (Kwon, 2004); (b) agentes adaptativos (Montazemi, 2004); (c) y RBC activo (Li, et al., 2004). Estas aproximaciones se enfocan en mecanismos de recuperación. Sin embargo, presentan grandes dificultades en las fases de adaptación y evaluación de la solución propuesta.

5. DISEÑO DE UN SISTEMA TUTOR INTELIGENTE USANDO RAZONAMIENTO BASADO EN CASOS

La relación entre Sistemas de Tutoría Inteligente y Razonamiento Basado en Casos se establece, mediante la asociación de modelos de estudiantes como casos. La ventaja de esta aproximación es que un problema puede conceptualizarse fácilmente en términos de agentes, y después ser implementado como un sistema RBC.

Nuestra propuesta se presenta como una solución que integra un conjunto de componentes, que corresponden con los módulos de la arquitectura tradicional de un STI. Estos módulos se distribuyen y se dividen en piezas más pequeñas llamadas agentes. Estos agentes implementan cada una de las fases de RBC, trabajan como entidades autónomas, y actúan racionalmente de acuerdo con sus percepciones del entorno y estado del conocimiento. Además, los agentes intercambian

información entre sí, proporcionando modularidad y cooperación. Los cuatro componentes en los que se divide la arquitectura son: (a) base de conocimiento; (b) modelo del estudiante; (c) sistema multi-agente; e (d) interfaz de usuario.

a. Base de Conocimiento

Almacena los elementos que representan todo el conocimiento que será aprendido por el estudiante en un determinado tutorial. Estos elementos son introducidos en el sistema por un experto y son organizados en una estructura llamada currículo. A continuación se describen los elementos que componen esta estructura:

- *Tutorial:* Se encuentra en un nivel superior. Incluye todo el conocimiento de un tutorial específico.
- *Tema:* Cada tutorial está dividido en temas. Cada tema contiene un número determinado de conceptos. Estos conceptos están asociados con la información a enseñar y los exámenes que deben superar los estudiantes.
- *Conceptos:* Los conceptos corresponden a las unidades de conocimiento que deben ser aprendidas por el estudiante a través del tutorial.
- *Preguntas de selección:* Tienen como objetivo evaluar el conocimiento obtenido por el estudiante en un concepto. Son insertadas entre los contenidos textuales de cada concepto.
- *Preguntas de examen:* Preguntas insertadas dentro de los temas de los exámenes. Son ejercicios que el estudiante debe resolver y que serán evaluadas y corregidas por el profesor experto en el dominio.

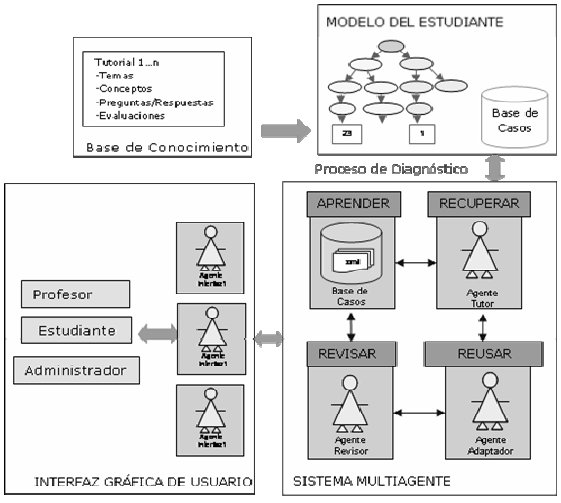


Fig. 3: Arquitectura del Sistema Tutor

b. Modelo del Estudiante

Este modelo almacena el estado de conocimiento del estudiante en el dominio. Tres aspectos fundamentales se tienen en cuenta durante la construcción de este modelo:

- *Selección de la estructura:* En STI-RBC la información de los estudiantes es almacenada en forma de casos. Un caso consiste de tres partes: (a) descripción del problema; (b) solución; y (c) relación. Se escoge el lenguaje de marcado (CaseML) (Huajun, et al., 2003), como un lenguaje estándar para representación de casos. Además, dentro del modelo del estudiante se consideran los siguientes atributos:
  - nivel de conocimiento: incluye tutorial, temas y conceptos;
  - habilidades: problemas resueltos con respuestas correctas;
  - limitaciones: ejercicios resueltos por el estudiante con respuestas equivocadas;
  - actitudes: ejercicios resueltos utilizando diferentes ayudas.
  - ruta de aprendizaje: conceptos vistos, temas explorados, durante el proceso de aprendizaje.
- *Inicialización del modelo:* La información inicial acerca de un nuevo estudiante es adquirida por medio de una entrevista y test-preliminares. La entrevista es presentada cada vez que un estudiante interactúa por primera vez con el sistema. Esta contiene preguntas relacionadas a datos personales del estudiante (e.g. nombre, edad, etc.), y otra información adicional independiente del dominio, y que puede influenciar el proceso de aprendizaje. Un test preliminar se usa para evaluar el nivel inicial de conocimiento del alumno acerca del dominio a ser enseñado. Teniendo en cuenta el desempeño del estudiante en el test, el sistema le asigna un estereotipo de acuerdo a su nivel de conocimiento (e.g. principiante, intermedio, avanzado, etc.). La información adquirida en la entrevista y el test es almacenada en el caso.
- *Proceso de Diagnóstico:* Una vez que el modelo del alumno se ha inicializado comienza la interacción con el sistema. El proceso de diagnóstico utiliza RBC para actualizar el modelo del alumno inicial tras sus interacciones con el sistema, utilizando dos fuentes de información principalmente: (a) el modelo del alumno actual; y (b) su comportamiento en el proceso interactivo de enseñanza, comportamiento que puede medirse en función de distintas variables que es preciso definir previamente: (a) soluciones a

problemas; (b) respuestas a preguntas; (c) tiempo empleado de lectura de pantallas, etc.

c. Sistema Multi-agente: El propósito del sistema multi-agente es soportar el proceso de diagnóstico del estudiante. Tiene tres funciones principales que son: (a) monitorear el comportamiento del estudiante; (b) evaluar y calcular el nivel de conocimiento; y (c) aconsejar al estudiante en las decisiones que debe tomar a lo largo del proceso de aprendizaje. El sistema está compuesto de los siguientes agentes:

- *Agente de Interfaz:* Este agente está a cargo de la interacción con el usuario y puede entender sus peticiones y traducirlas a los otros agentes. Es el punto de referencia principal para el usuario y tiene diversas tareas que son cruciales para la correcta operación del sistema global: (a) asistir al estudiante en la ejecución de peticiones y la determinación de su perfil; (b) deducir la información necesaria para comunicarse con eficacia con el estudiante y observar su comportamiento; (c) traducir las peticiones del estudiante y seleccionar el agente (o agentes) capaz de solucionar su problema (o problemas); y (d) presentar y almacenar los datos recuperados.
- *Agente Tutor:* Este agente lleva a cabo el proceso de recuperación de casos. Su función es recuperar los casos más similares al nuevo problema (nuevo estudiante). El agente lleva a cabo las siguientes tareas: (a) recibir casos candidatos de la librería de casos; (b) mezclar los casos candidatos; y (c) escoger el mejor de los casos.
- *Agente Adaptador:* Combina el nuevo caso con el caso o casos similares recuperados. Cuando se confirma la estrategia, este agente organiza los recursos para exhibir al estudiante. La estrategia usada se puede modificar según el avance del alumno.
- *Agente Revisor:* Revisa cada paso que el estudiante haga en la clase particular. El agente utiliza un sistema de evaluación, que valora el conocimiento del estudiante para comprobar si se está aconsejando el alumno adecuadamente.

d. Interfaz de Usuario

La interfaz permite acceso a cualquier usuario que esté interesado en aprender alguno de los tutoriales ofrecidos. Consiste de dos vistas, teniendo en cuenta dos tipos de usuarios: estudiante y profesor. La estructura de la interfaz del estudiante varía en estructura y distribución dependiendo de la tarea y el tutorial que el estudiante esté realizando. La interfaz del profesor o experto es similar a la del estudiante, solo que esta carece de ciertas interacciones con el sistema multi-agente dado que el profesor no necesita ser evaluado por el sistema.

5.1. Comunicación entre Agentes

La comunicación de los agentes se basa en la teoría de actos comunicativos. Los agentes utilizan el sistema de performativas para comunicar sus intenciones. La semántica de la performativa permitirá que el receptor interprete el contenido fácilmente. En este trabajo, se ha utilizado el lenguaje de comunicación de agentes (FIPA-ACL). Este lenguaje define un conjunto limitado de performativas con la semántica expresada en lógica de creencias. La Figura 4 muestra la comunicación entre los agentes del sistema.

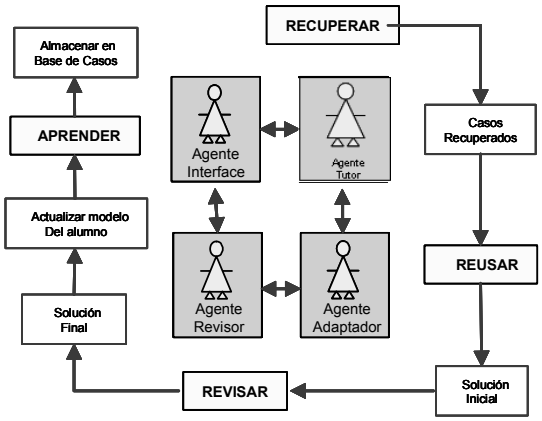


Fig. 4: Comunicación entre agentes del sistema

5.2. Recuperación de Casos

Este proceso consiste de las siguientes sub-tareas: (a) identificación de atributos clave; (b) establecimiento de similaridad; y (c) búsqueda y selección. Una vez los casos son representados y organizados de forma eficiente, el agente tutor lleva a cabo el proceso de búsqueda de casos similares. La técnica del k-vecino más cercano (Jousellin, 1987) fue seleccionada para determinar el grado de similaridad entre casos.

La Figura 5 describe el algoritmo para chequear la correspondencia y determinar el grado de similaridad entre los casos de la librería de casos.

1. Chequear la correspondencia:
   a. Chequear las correspondientes características con los casos almacenados en la librería de casos usando las siguientes reglas:
      - Valores que incluyan el mismo atributo serán seleccionados
      - Valores que incluyan el mismo rol en dos clausulas serán seleccionados

2. Determinar la similaridad:
   a. Determinar el grado de similaridad de los atributos correspondientes mediante:
      - Cálculo de la distancia sobre una escala cuantitativa y cualitativa.

3. Determinar el grado de correspondencia:
   a. Determinar el grado de correspondencia utilizando la siguiente función:

$$\frac{\sum_{i=1}^{n} w_i (sim\ (f_i^{\ l}, f_i^{\ r})}{\sum_{i-1}^{n} w_i}$$

   b. Rankear los casos de alto nivel a bajo nivel.

Fig. 5: Algoritmo para chequear similaridad

Donde,
w = el grado de importancia del índice i,
sim = función de similaridad,
f_i = valores para las características f_i en cada nuevo problema.

Es posible que mas que un caso tenga el mismo coeficiente de similaridad. Para seleccionar el caso más adecuado, el sistema a través del agente tutor sugerirá el mejor caso a escoger. Un posible problema en la fase de recuperación es que la librería de casos incrementará linealmente con el tiempo, lo que podría llegar a hacerla inmanejable. Sin embargo, en nuestro trabajo no se observa este problema dado que hemos definido para la recuperación atributos restrictivos (e.g. nivel de conocimiento). De esta forma se reduce el número de casos a ser analizados y se puede obtener mejor desempeño del sistema.

5.3. Adaptación de Casos

Una vez los casos más similares son recuperados, deben ser reutilizados o adaptados. El proceso de adaptación de casos puede llevarse a cabo si los casos similares y el problema objetivo son muy similares en términos de atributos clave; o si los casos son muy similares en algunos elementos. El proceso de adaptación es altamente dependiente del experto en el dominio. El uso de un agente adaptador permitirá automatizar esta tarea. Teniendo en cuenta la naturaleza del problema dos tipos de adaptación se pueden llevar a cabo:

- *Sustitución:* tipo de adaptación simple, en la que algunas partes de la solución recuperada son reinstanciadas y;
- *Transformación:* tipo de adaptación que altera la estructura de la solución.

5.4. Revisión y Aprendizaje de Casos

El agente revisor hace uso de un sistema de evaluación para llevar a cabo esta tarea. Cuando la solución generada en la etapa de adaptación es equivocada, el agente revisor es responsable de modificar la solución teniendo en cuenta el conocimiento disponible acerca del problema. Este agente lleva cabo dos tareas:

- *Evaluar la solución generada en la etapa de adaptación:* En esta parte, si el agente detecta condiciones irregulares reacciona para evaluarlas y tomar las acciones apropiadas. En caso contrario, si la solución es adecuada, el caso es almacenado en la librería de casos.
- *Reparar la solución generada utilizando conocimiento específico del dominio:* Esta tarea involucra detectar errores en la solución inicial, y recuperar o generar explicaciones para ellos. El agente utiliza las fallas obtenidas para llevar a cabo la explicación de forma que garantice no vuelvan a ocurrir. Una vez el agente asegura la exactitud de la solución el caso puede ser almacenado en la librería.

6. STIM-Tutor. SISTEMA INTELIGENTE PARA EDUCACIÓN MÉDICA

El STIM se desarrolla bajo un enfoque multi-agente compatible con los estándares de FIPA. En el desarrollo, se utilizan Java, Java Script y XML. Los módulos de STIM se distribuyen y se dividen en piezas más pequeñas llamadas agentes. Estos agentes trabajan como entidades autónomas, y actúan racionalmente de acuerdo con sus percepciones del ambiente y su estado del conocimiento. Cada agente implementa una fase de RBC automatizando el proceso completo. El sistema STIM direcciona tres de los principales aspectos de educación en salud:

- Ayudar a los estudiantes a encontrar ejercicios apropiados según sus habilidades y nivel de conocimiento.
- Combinar el aprendizaje de textos con aprendizaje a través de casos prácticos.
- Asistir al estudiante durante el proceso de resolución de problemas y toma de decisiones.

El primer componente creado en el STIM fue el modulo del estudiante teniendo en cuenta que es el componente que provee al sistema de inteligencia. En el proceso de modelado del estudiante la información de los alumnos fue recolectada y almacenada como casos. La Figura 6 presenta un ejemplo de representación de un caso en CaseML.

El modelo del estudiante se describe en términos de conceptos, atributos y tipos. Un estudiante es el concepto en nuestro dominio. Las características del estudiante se representan como atributos y su rango de valores válido se representa por medio de tipos. El conjunto de clases en CaseML son:

- *Caso:* Un caso tiene la descripción de un problema y su solución;
- *Problema:* Un problema tiene uno o más atributos y es una subclase de Caso.
- *Atributo:* Contiene los pares atributo-valor;
- *Solución:* Contiene un conjunto de características (e.g. diagnóstico, estrategia, etc.).
- *Medida de Similaridad:* Esta clase encapsula el detalle de cómo los casos contenidos en su base de casos deberían ser evaluados.

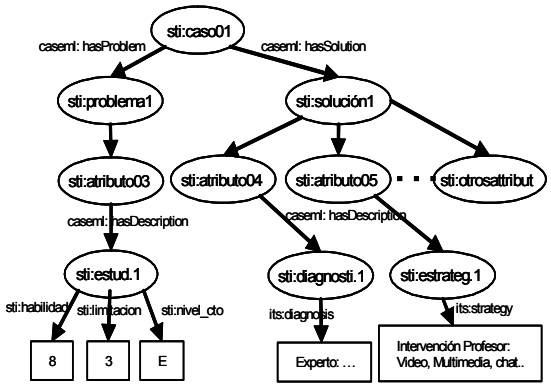
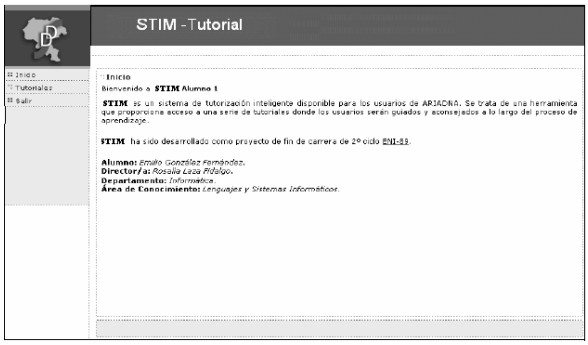


Fig. 6: Representación de Casos en CaseML

Una vez se construye el modelo del estudiante, el sistema selecciona un problema y compara su solución con la entregada por el estudiante, de esta forma obtiene un diagnostico del conocimiento del alumno basado en las diferencias. El sistema proporciona retroalimentación, y actualiza el modelo del estudiante cada vez que este interactúa con el STI. Además de evaluar lo que sabe el estudiante, el sistema también considera lo que el alumno debe saber, es decir, que parte del currículo debe ser enseñada y cual es la forma adecuada de presentación del material. En la Figura 7 se presenta, la interfaz principal del STIM-Tutor.



Fig, 7: Interfaz Inicial de STIM-Tutor

Los enlaces del menú principal tienen las siguientes funciones:

- *Inicio:* Cada vez que el alumno selecciona este enlace, se le mostrará la página de inicio de la Figura 7.
- *Tutoriales:* Mediante este enlace, el alumno accede a la página donde se le mostrarán los tutoriales disponibles para el estudiante.
- *Salir:* Cuando un alumno desea finalizar la sesión con el sistema.

En el momento en que el estudiante selecciona el listado de tutoriales disponibles, se muestran los elementos de la interfaz de estudio de un tutorial.

- *Índice de Contenidos:* Se muestra la jerarquía de contenidos en forma de árbol, con la jerarquía de los elementos que componen el tutorial al que ha accedido el alumno. El índice de contenidos está señalado con el número 1 en la Figura 8.
- *Área de Contenidos:* Se encuentra en la zona central de la página, señalada con el número 2 en la Figura 8. Allí se le muestran al alumno los contenidos pertenecientes al elemento que selecciona en el índice.
- *Barra de Navegación:* Dispone de una serie de botones que proporcionan al alumno diversas funcionalidades a lo largo de su proceso de estudio en el tutorial. Se indica con el número 3 en la Figura 8.

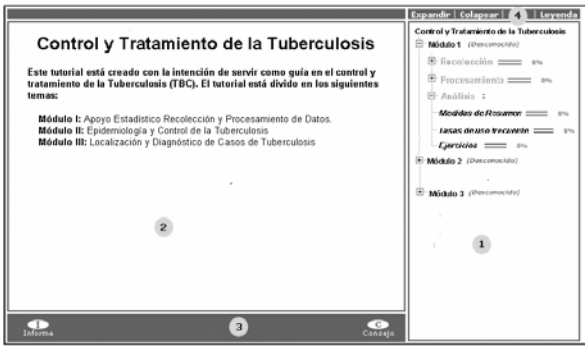


Fig. 8: Interfaz para estudio de tutorial

Cuando el profesor selección el índice de contenidos se le mostrará una descripción de los contenidos que va a encontrar en el tema seleccionado, y los exámenes disponibles para cada tema. Por ejemplo, en la Figura 9, se presenta el tutorial para capacitar en Control y Tratamiento de Tuberculosis como enfermedad de alta prevalencia.

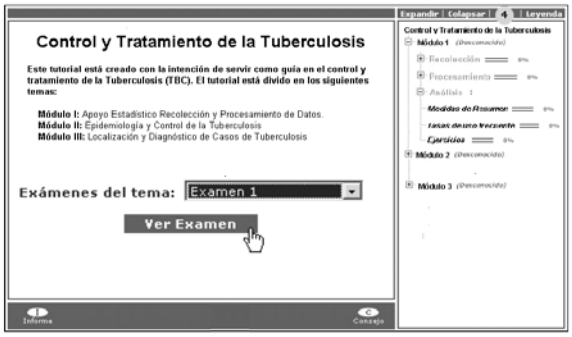


Fig. 9: Tutorial de Tuberculosis II

7. CONCLUSIONES

El modelado de los sistemas tutores inteligentes es una tarea compleja. Es por esto que el Razonamiento Basado en Casos, puede ofrecer el camino hacia la elaboración de un nuevo diseño. En lugar de resolver los problemas descomponiéndolos en sus partes, un caso sugiere una solución completa, y las partes que no se adecuen a la nueva situación se adaptan. Resolver un problema adaptando una solución vieja evita que el que resuelve problemas tenga que tratar con muchas restricciones que se cumplen en la nueva situación. Aunque sea necesario hacer un trabajo de adaptación considerable, esta tecnología es casi siempre preferible a generar una solución desde estadios muy tempranos del diseño. En general RBC resulta útil para el diseño de un STI ya que, ejecutar un diseño requiere de experiencia previa, y frecuentemente es difícil representar el conocimiento del experto a través de reglas. RBC es una tecnología adecuada pues mediante la memoria permanente y los métodos de búsqueda se puede poner a disposición del diseñador un gran volumen de experiencia previa a través de casos resueltos y, a partir de esta información él mismo puede construir la solución deseada. La aplicación de los resultados obtenidos en el dominio de salud permitió, minimizar los riesgos de los pacientes al permitir a los profesionales adquirir el conocimiento, y las habilidades prácticas que sirvieron de base en el proceso de adecuada toma de decisiones.

8. AGRADECIMIENTOS

Queremos expresar nuestro agradecimiento a la Universidad del Cauca, comisión de estudios otorgada bajo resolución 137 del 28 de Octubre de 2003.

REFERENCIAS

Aamodt and E. Plaza (1994). *Case-based reasoning: Foundational issues, methodological variations, and system approaches*. AICommunications, 7(1). pp. 39–59.

Aamodt, A (2004). *Knowledge-Intensive Case-Based Reasoning in CREEK*. ECCBR. pp. 1-4.

Bareiss, Ray (1989). *Protos: a unified approach to concept representation, classification and learning.* Technical report ai88-83, University of Texas at Austin, Dep. of Computer Science.

Brusilovsky (1977). *Intelligent tutoring systems for the world-wide web.* The Third International World-Wide Web Conference.

Burns, J. Parlett, and C (1991). Redfield. *Intelligent tutoring systems: Evolutions in design*. Lawrence Erlbaum Associates, Hillsdale, NJ.

Cataldi, Z., & Lage F., & Deanis J. M (2000). *The Scripts of University Students and Experts in the Preparation of the Examinations: A Study Process.* Frontiers in Education Conference, Kansas City Missouri, Octubre 18-21.

Clancey, W.J (1987). Knowledge based tutoring: the GUIDON Program. Cambridge, MA:MIT Press.

Conati, C., Gertner, A., VanLehn, K & Drudzel, M (1997). *On-line student modelling for coached problem solving usign Bayesian Networks.* Proceedings of the 6th Internacional Conference on User Modelling UM'97. pp. 231-242. Springer Verlag.

F. Dorca, C. Lopes, and M. Fernández (2003). *A multiagent architecture for distance education systems.* The 3rd IEEE International Conference on Advanced Learning Technologies. pp. 368–369.

Huajun Chen and Zhaohui Wu (2003). *On Case-Based Knowledge Sharing in Semantic Web.* 15th IEEE International Conference on Tools with Artificial Intelligence. 3 (2). pp. 200.

Joussellin, A and Dubuisson, B (1987). *A link between k-nearest neighbord rules and knowledge based systems by sequence análisis.* Pattern Recognitizion Letters. pp. 287-295.

Kolodner, J. L (1983). *Reconstructive Memory: A Computer Model.* Cognitive Science, 7 (4). pp. 281.

Koton, P (1989). *Using experience in learning and problem solving*. Massachusetts Institute of Technology, Laboratory of Computer Science (Ph.D. diss, October 1988). MIT/LCS/TR-441.

Kwon, O.B and Sadeh, N (2004). *Applying case-based reasoning and multi-agent intelligent system to context-aware comparative shopping.* Decision Support Systems. 37 (2). pp. 199-213.

Li, S and Yang, Q (2004). *Active CBR: An agent system that integrates case-based reasoning and active databases.* Knowledge and Information Systems. 3(2). 2004. pp. 225-251.

Littman D. & Soloway E (1988). *Evaluating ITSs: The cognitive science perspective.* In: Polson M. C. & Richardson J. J. (eds.) Foundations of Intelligent Tutoring Systems, New Jersey, 1988: Lawrence Erlbaum Associates, pp. 209-242.

López de Mantarás, McSherry, D, et. al (2005). *Retrieval, Reuse, revise and retention in case based reasoning.* The Knowledge Engienerring Review, Vol 1 (2).

Montazemi,A.R and Gupta, K.M (2004). *An adaptive agent for case description in diagnostic CBR systems.* Computers in Industry. 29 (3). pp. 209-224.

Plaza, E & López de Mántaras, R (1990). *A Case-Based Apprentice that Learns from Fuzzy Examples.* (Z. Ras et al., Eds.) Methodologies for Intelligent Systems-5 North-Holland, pp. 420-427.

Plaza, E. and Arcos J. L (1993). *Reflection and Analogy in Memory-based Learning*. Proceedings Multistrategy Learning Workshop, George Mason University. p.p 42-49.

Plaza, E. and Arcos J. L (1993). *Reflection and Analogy in Memory-based Learning.* Proceedings Multistrategy Learning Workshop, George Mason University. p.p. 42-49.

Richter, M and Wess, S (1991). *Similarity, uncertainty and case-based reasoning in PATDEX.* In R S Boyer, editor, Automated Reasoning. Essays in Honour of Woody Bledsoe, pp. 249-265. Kluwer Ed.

Schank, R., & Riesbeck, C & Kass (1989). *A. Inside Case-Based Explanation*.

Shiri, Aimeur E and Frasson, C (1998). *Student modelling by case-based reasoning.* Fourth international conference on intelligent tutoring systems. ITS'98. 1998. 182-195.

Skran, Hasan (2006). Software Cost Estimation Model Based on Integration of Multi-agent and Case-based Reasoning. Journal of Computer Science. 2 (3). pp. 276-282.

VahLehn, K (1988). *Student modelling*. Foundations of Intelligent Tutoring Systems, Lawrence Erlabum Associates Publishers. pp. 55–78.

VanLehn, K, Niu, Z., Siler, S., & Gertner, A. S. (1998). *Student modeling from convencional test data: A Bayesian approach without priors.* Lecture Notes in Computer Science: Vol 1452. Intelligent Tutoring Systems. Proceedings of 4th Internacional Conference ITS'98. Springer Verlag.

Wooldridge, Michael (2002). *Introduction to MultiAgent Systems.* John Wiley and Sons.

Wooldridge, Michael and Jennings Nicholas (1995). *Intelligent Agents. Theory and Practice.* Knowledge Engineering Review.