



**I
N
A
O
E**

**HERRAMIENTA DE AUTORÍA PARA TUTORES
INTELIGENTES BASADOS EN MODELOS
PROBABILISTAS RELACIONALES**

Por

Mario Alberto Romero Inzunza

Tesis

Sometida como requisito parcial
para obtener el grado de

**MAESTRO EN CIENCIAS EN EL ÁREA DE
CIENCIAS
COMPUTACIONALES**

en el

**Instituto Nacional de Astrofísica, Óptica y
Electrónica**

Tonantzintla, Puebla

Supervisada por:

Dr. Luis Enrique Sucar Succar
Dra. María del Pilar Gómez Gil

©INAOE 2009

Derechos reservados

El autor otorga al INAOE el permiso de
reproducir y distribuir copias de esta tesis
en su totalidad o en partes.



RESUMEN

El uso de sistemas tutores inteligentes (STI) para mejorar el proceso enseñanza-aprendizaje ha sido uno de los éxitos de la inteligencia artificial. Por otro lado, en los últimos años se ha impulsado el desarrollo de ambientes virtuales de enseñanza y laboratorios virtuales. El principal inconveniente es el largo tiempo de desarrollo para este tipo de sistemas y la necesidad de personas expertas tanto en la enseñanza de los temas como en la construcción de sistemas inteligentes y laboratorios virtuales.

En esta investigación se presenta una herramienta de autoría capaz de generar automáticamente un STI a partir de la descripción temática de un curso y la definición de variables a evaluar de experimentos que un estudiante realizará mediante un laboratorio virtual. La herramienta de autoría utiliza modelos gráficos probabilistas para representar el modelo del estudiante, así como para definir las relaciones de la base de conocimiento; lo anterior es transparente para los usuarios de la herramienta.

El aspecto central de la herramienta es la construcción automática de un modelo del estudiante basado en modelos probabilistas relacionales. Posteriormente el modelo es utilizado para dar tutoría personalizada a cada estudiante. La herramienta de autoría genera el modelo del estudiante basado en redes bayesianas, sin necesidad de que el docente tenga conocimiento sobre las mismas. La generación del modelo de relacional probabilista se realiza con base en la información del curso, que el docente capture; dicha información es capturada en niveles jerárquicos de temas, subtemas, conceptos y variables de experimentos.

En este trabajo se presenta la estrategia de construcción automática del modelo del estudiante y de sus actualizaciones resultantes de la ejecución de experimentos en un laboratorio virtual.

La herramienta se probó generando un pequeño STI de redes neuronales artificiales para la enseñanza del uso de perceptrones de varios niveles. La herramienta genera adecuadamente la representación del conocimiento del estudiante y fue evaluada en cuanto a su usabilidad con resultados satisfactorios.

ABSTRACT

Intelligent Tutoring Systems (ITS) can provide personalized instruction which helps to improve the education-learning process; on the other hand, in recent years the development of virtual education environments and virtual laboratories has increased. The educational effectiveness of virtual laboratories can be increased by incorporating an ITS. However, the main problem is the long time of development required for ITS, and the need of experts in the area, as well as in the construction of intelligent systems and virtual laboratories.

This research presents a novel authoring tool able to automatically generate an ITS from the thematic description of a course, and the definition of the variables to evaluate student's performance in the experiments in the virtual laboratory. The authoring tool uses probabilistic graphical models to represent the student model and to determine the knowledge base relationships, this is transparent to the users.

The central aspect of the tool is the automatic construction of a student model based on probabilistic relational models. This model is used to provide each student personalized tutoring. The generation of the probabilistic relational model is made based on information of the course, which is captured by the teacher. These information is provided in a hierarchy of topics, subtopics, concepts and experimental variables. This theses presents the method for the automatic construction of the student model, and the ITS in general; which is later used to provide assistance to a student interacting with a virtual lab.

The authoring tool was tested by generating an ITS for artificial neural networks, for teaching the use of multilayer Perceptrons. The tool was evaluated on its usability and correct inference, with satisfactory results.

AGRADECIMIENTOS

Agradezco a los Drs. Luis Enrique Sucar Succar y a la Dra. María del Pilar Gómez Gil por el apoyo y la paciencia que me han tenido en el transcurso de la realización de esta investigación.

Al Consejo Nacional de Ciencia y Tecnología por el apoyo brindado para poder mantener mis estudios de posgrado, al INAOE y a los investigadores que me han impulsado para poder realizar esta tesis con trabajo constante y dedicación.

Agradezco a la Universidad Autónoma de Sinaloa, por apoyarme para seguirme preparando, al Ing. Arnoldo Valle Leyva y al MC. Héctor Melesio Cuen Ojeda por todos los apoyos que me han brindado así como sus consejos de seguir adelante siempre en los estudios.

Un agradecimiento a mis amigos de la Facultad de Informática Culiacán donde siempre tengo las puertas abiertas y por su apoyo para evaluar la herramienta que se presenta en esta investigación; así como a la Dra. Julieta Noguez por dedicar parte de su valioso tiempo en evaluar la herramienta desarrollada.

Por último, a mis padres que me han apoyado de manera incondicional y siguen haciéndolo, mis hermanos que me brinda su cariño, a mis compañeros de maestría por los desvelos que pasamos y a Adelina por la paciencia y cariño que me tiene..

TABLA DE CONTENIDO

| | |
|---|-----------|
| RESUMEN..... | I |
| ABSTRACT | III |
| AGRADECIMIENTOS | V |
| TABLA DE CONTENIDO | VII |
| ÍNDICE DE FIGURAS..... | X |
| ÍNDICE DE TABLAS | XII |
| 1 INTRODUCCIÓN..... | 1 |
| 1.1 MOTIVACIÓN..... | 1 |
| 1.2 DESCRIPCIÓN DEL PROBLEMA | 2 |
| 1.3 OBJETIVOS..... | 3 |
| 1.4 METODOLOGÍA | 4 |
| 1.5 APORTACIONES | 5 |
| 1.6 ORGANIZACIÓN DEL DOCUMENTO | 6 |
| 2 SISTEMAS TUTORES INTELIGENTES | 7 |
| 2.1 SISTEMAS TUTORES INTELIGENTES | 7 |
| 2.2 SHELLS Y HERRAMIENTAS DE AUTORÍA PARA TUTORES INTELIGENTES..... | 12 |
| 2.2.1 Herramientas de Autoría | 14 |
| 2.2.2 Clasificación de las Herramientas de autoría..... | 15 |
| 2.3 ESTRATEGIAS DE DESARROLLO | 18 |
| 2.4 RESUMEN DEL CAPÍTULO..... | 19 |
| 3 MODELOS PROBABILISTAS | 21 |
| 3.1 REDES BAYESIANAS..... | 21 |
| 3.2 REDES BAYESIANAS DINÁMICAS..... | 23 |
| 3.3 MODELO PROBABILISTA RELACIONAL | 25 |
| 3.4 TUTORES INTELIGENTES BASADOS EN MODELOS PROBABILISTAS | 28 |
| 3.5 MODELO PROBABILISTA RELACIONAL DEL ESTUDIANTE | 30 |
| 3.6 ELVIRA | 32 |
| 3.7 RESUMEN DEL CAPÍTULO | 32 |
| 4 LABORATORIOS VIRTUALES | 35 |
| 4.1 DEFINICIÓN DE LABORATORIO VIRTUAL..... | 35 |
| 4.2 CLASIFICACIÓN DE LABORATORIOS VIRTUALES..... | 36 |
| 4.3 VENTAJAS DE LA UTILIZACIÓN DE LABORATORIOS VIRTUALES | 37 |

| | |
|--|------------|
| 4.4 RESUMEN DEL CAPÍTULO | 41 |
| 5 DESARROLLO DE LA HERRAMIENTA DE AUTORÍA | 43 |
| 5.1 DESCRIPCIÓN GENERAL | 43 |
| 5.2 DESCRIPCIÓN DE LOS MÓDULOS..... | 48 |
| 5.2.1 Interfaz con el Laboratorio Virtual | 48 |
| 5.2.2 Base de conocimientos | 50 |
| 5.2.3 Modelo del Estudiante..... | 53 |
| 5.3 GENERACIÓN DEL MODELO DEL ESTUDIANTE..... | 55 |
| 5.3.1 Ejemplo de generación de una red bayesiana desde el temario..... | 57 |
| 5.4 PROCESO DE INFERENCIA..... | 63 |
| 5.5 MÓDULO TUTOR | 64 |
| 5.5.1 Proceso de asignación y ejecución de experimentos..... | 65 |
| 5.6 UTILIZACIÓN DEL STI GENERADO..... | 67 |
| 5.7 DESCRIPCIÓN DEL MÉTODO DESARROLLADO | 67 |
| 5.8 RESUMEN DEL CAPÍTULO | 70 |
| 6 EVALUACIÓN DE LA HERRAMIENTA..... | 71 |
| 6.1 CASO DE ESTUDIO | 72 |
| 6.2 EVALUACIÓN DE USABILIDAD DE LA HERRAMIENTA..... | 72 |
| 6.3 EVALUACIÓN DEL MODELO DEL ESTUDIANTE | 75 |
| 6.4 EVALUACIÓN DEL TUTOR INTELIGENTE..... | 77 |
| 6.5 PRUEBAS DE DESEMPEÑO | 77 |
| 6.6 RESUMEN DEL CAPÍTULO..... | 79 |
| 7 CONCLUSIONES Y TRABAJO FUTURO | 81 |
| 7.1 CONCLUSIONES..... | 81 |
| 7.2 APORTACIONES | 83 |
| 7.3 TRABAJO FUTURO..... | 83 |
| REFERENCIAS | 85 |
| APÉNDICES | 89 |
| A.-GUIAS PARA EL DISEÑO DEL LABORATORIO..... | 89 |
| B.-CASO DE ESTUDIO..... | 93 |
| C.- MANUAL DE USUARIO DE HEDEA..... | 97 |
| D.-ARCHIVOS DE CONFIGURACIÓN..... | 109 |
| E.- ALGORITMO DE ELIMINACIÓN DE VARIABLE..... | 111 |
| F.- DISEÑO DE SOFTWARE | 113 |
| LISTA DE ACRONIMOS | 129 |

Índice de Figuras

| | |
|---|-----|
| Figura 1-1 Esquema general de la herramienta generada..... | 5 |
| Figura 2-1 Arquitectura básica de un sistema tutor inteligente. | 8 |
| Figura 3-1 Ejemplo de una red bayesiana simple.. | 22 |
| Figura 3-2 Tabla de probabilidad condicional para el nodo enfermedad de la figura 3-1..... | 22 |
| Figura 3-3 Ejemplo de una red bayesiana dinámica | 24 |
| Figura 3-4 Ejemplo de un modelo probabilista relacional. | 27 |
| Figura 3-5 Arquitectura de interacción entre un tutor inteligente y un laboratorio virtual. | 29 |
| Figura 3-6 Esqueleto propuesto por Noguez para el ámbito de robótica | 31 |
| Figura 5-1 Generación de un nuevo STI..... | 44 |
| Figura 5-2 Esquema de trabajo del STI con el laboratorio virtual. | 44 |
| Figura 5-3 Secuencia de interacciones entre los actores que participan en el proceso de un STI. | 46 |
| Figura 5-4 Esquema general de trabajo..... | 48 |
| Figura 5-5 Pantalla de captura de un Laboratorio Virtual..... | 49 |
| Figura 5-6 Ejemplo de mapa conceptual capturado con pesos. | 50 |
| Figura 5-7 Captura de Temario..... | 51 |
| Figura 5-8 Ejemplo de mapa jerárquico en el ámbito de redes neuronales .. | 52 |
| Figura 5-9 Enlace del laboratorio virtual con el temario. | 53 |
| Figura 5-10 Captura del modelo del estudiante. | 54 |
| Figura 5-11 Ejemplo de temario capturado..... | 57 |
| Figura 5-12 Ejemplo de grafo generado a partir del temario capturado..... | 58 |
| Figura 5-13 Ejemplo de una tabla de Probabilidad Condicional | 59 |
| Figura 5-14 Red bayesiana dinámica que considera el conocimiento previo | 60 |
| Figura 5-15 Ejemplo de cómo se mira la red base (parcial solo 2 niveles) ... | 61 |
| Figura 5-16 Ejemplo de red de transición tomando | 62 |
| Figura 5-17 Caso Genérico para comenzar experimento.. | 66 |
| Figura 5-18 Esquema general del proceso de generación de un STI..... | 69 |
| Figura C-8-1 Proceso General de trabajo | 98 |
| Figura C-8-2 Proceso General en el STI..... | 99 |
| Figura C-8-3 Pantalla inicial..... | 100 |
| Figura C-8-4 Pantalla de configuración del laboratorio | 101 |
| Figura C-8-5 Pantalla de selección de laboratorios existentes | 102 |
| Figura C-8-6 Pantalla para generar la base de conocimiento | 103 |
| Figura C-8-7 Captura de temario | 103 |
| Figura C-8-8 Captura el porcentaje de conocimiento hacia el nodo padre . | 104 |
| Figura C-8-9 Pregunta sobre lección | 104 |
| Figura C-8-10 Modelo del estudiante..... | 106 |

Índice de Tablas

| | |
|---|----|
| Tabla 2-1 Tipos de Herramienta de autoría y tipo de STI que generan | 15 |
| Tabla 4-1 Ejemplos de laboratorios virtuales que utilizan facilidades de los STI | 39 |
| Tabla 5-1 Ejemplo de probabilidades del Temario de la figura 5-11 | 58 |
| Tabla 6-1 Cuestionario de usabilidad de la herramienta de autoría..... | 73 |
| Tabla 6-2 Resultado del cuestionario de usabilidad aplicado | 74 |
| Tabla 6-3 Resultados de las variables de RNA de un caso básico..... | 75 |
| Tabla 6-4 Evaluación de estudiante que comete errores en un concepto | 76 |
| Tabla 6-5 Tiempos de Respuesta de los procesos principales de la herramienta..... | 78 |

Capítulo 1

Introducción

1.1 Motivación

Se conoce como Sistema Tutor Inteligente (STI) a un programa de computadora que instruye a un estudiante de manera que dicha enseñanza se adapte a las necesidades de cada alumno, utilizando técnicas de inteligencia artificial (VanLehn 1988). Los STI han alcanzado una fuerte popularidad en los últimos años. Sin embargo, el tiempo de desarrollo de un sistema tutor inteligente va de 200 a 300 horas para una hora de de trabajo con el STI (Alevan, et al. 2006), lo cual limita su desarrollo y aplicación. Por estas razones, se han creado diferentes herramientas que reducen los tiempos de desarrollo (Ferrero, et al. 2001, Alevan, et al. 2006, Murray 1999), conocidas como herramientas de autoría. Sin embargo, estas herramientas están restringidas a ciertas clases de tutores, y en particular no consideran en forma explícita la incertidumbre en el modelo del estudiante, como sucede en el caso de los tutores orientados a laboratorios virtuales.

1.2 Descripción del problema

El diseño de un STI requiere, entre otras cosas, una representación fidedigna del conocimiento que posee cada estudiante que lo utiliza, conocida como modelado del estudiante. La definición de este modelo es otra tarea que sigue en constante investigación; Murray (1999) presenta diversos enfoques que se han abordado para el modelado del estudiante. La implementación correcta del modelo del estudiante radica en la necesidad de obtener una estructura que permita una representación flexible y que produzca inferencias rápidas y correctas acerca de las acciones a tomar por el STI.

Un laboratorio virtual es un ambiente heterogéneo donde un estudiante interactúa para resolver ejercicios propuestos por algún docente y la retroalimentación que recibe la evalúa el propio estudiante cuando analiza los resultados que obtuvo de algún experimento. En algunos casos un estudiante no puede identificar fallas en sus resultados; por tal motivo es necesario incorporar un software que permita dar esta retroalimentación y guíe al estudiante durante el proceso enseñanza-aprendizaje. Este software es un STI que permita monitorear los resultados del experimento y pueda administrar lecciones según sea necesario.

Otro aspecto importante sobre los STI para laboratorios virtuales es el manejo de la incertidumbre, debido a que el STI sólo interactúa con el estudiante mediante los experimentos que ha resuelto. Es decir, se tiene que inferir lo que sabe el estudiante a partir de su interacción con el laboratorio virtual. Para esto se requiere un modelo que permita el manejo de incertidumbre como lo son las redes bayesianas y los modelos probabilistas relacionales. La construcción de STIs con estas características es una labor

compleja debido a la falta de una herramienta de autoría para este tipo de modelos.

Los problemas descritos anteriormente son afrontados en esta investigación a través del desarrollo de una herramienta de autoría basada en Modelos Probabilistas Relacionales (PRMs por sus siglas en inglés) (Getoor, et al. 2007). Dicha herramienta facilita la construcción automática del STI, de forma que un docente no tiene que ser experto en tutores inteligentes ni en modelos probabilistas. Se utiliza en el diseño de la herramienta una arquitectura orientada a objetos a fin de independizar cada parte de las tareas, y definiendo las interfaces requeridas para el mejor aprovechamiento de las relaciones entre los objetos que componen el STI, la herramienta y el laboratorio virtual que apoya. La herramienta se enlaza con el laboratorio mediante el uso de las variables que el laboratorio puede monitorear; las cuales son utilizadas para evaluar el resultado de experimentos que un estudiante realice, y actualizar el modelo del estudiante.

1.3 Objetivos

El objetivo general de este trabajo de investigación es desarrollar una herramienta de autoría para tutores inteligentes en laboratorios virtuales, basada en modelos probabilistas relacionales.

Para esto se establecieron los siguientes objetivos específicos:

- Definir las estructuras del modelo relacionales probabilistas que permitan una buena representación del dominio del problema.

- Diseñar un conjunto de esquemas de representación generalizada del conocimiento de un estudiante basados en modelos probabilistas, a partir de la definición temática de un curso.
- Definir una arquitectura de software para el desarrollo de la herramienta de autoría e implementar la herramienta.
- Evaluar el uso de la herramienta de autoría generada.

1.4 Metodología

En el presente trabajo se desarrolló una herramienta de autoría capaz de construir STI basados en modelos probabilistas para ser utilizados en laboratorios virtuales. La herramienta proporciona una interfaz para capturar el contenido de un curso y dar un peso a los conceptos que se van a enseñar. De manera automática se genera un modelo del estudiante que utiliza una red bayesiana dinámica, que unido a las reglas de tutoría permite al STI tomar decisiones sobre las acciones pedagógicas que se tomarán. Para poder utilizar esta herramienta es necesario que el docente cuente con un laboratorio virtual donde realizar los experimentos. La Figura 1-1 muestra de manera general la construcción de un tutor inteligente con la herramienta de autoría.

La herramienta se desarrolló utilizando un enfoque orientado a objetos y se programó en JAVA 2.0. Su funcionalidad se evaluó utilizando un pequeño laboratorio virtual de redes neuronales artificiales, este laboratorio virtual fue construido para apoyar un taller de enseñanza de redes neuronales (Gómez-Gil 2009). La herramienta permite generar el STI y es consistente en cuanto a las inferencias realizadas para escenarios de prueba. La usabilidad se evaluó por el docente a través de un cuestionario.

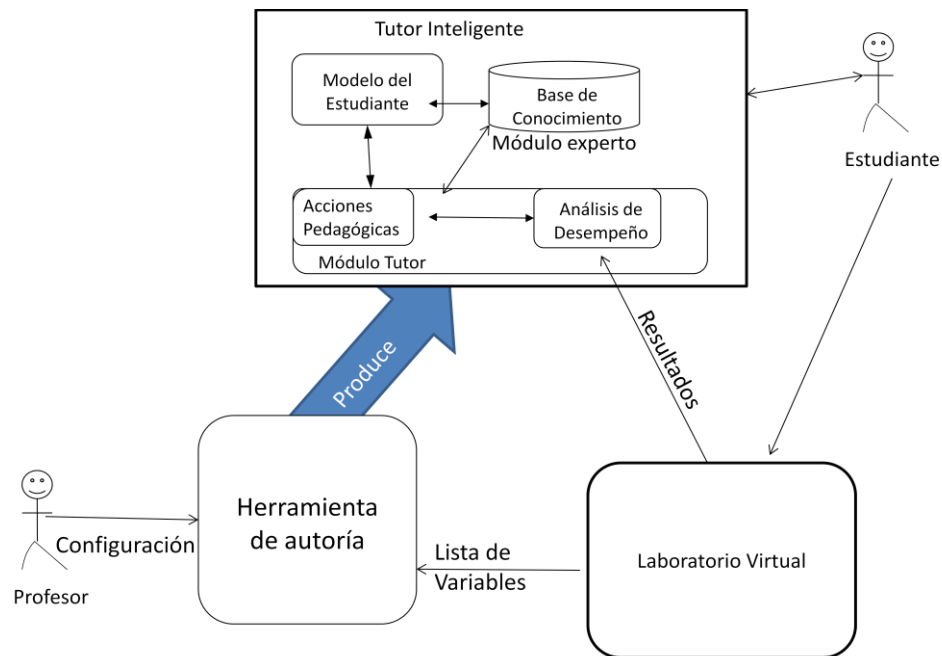


Figura 1-1 Esquema general de la herramienta generada. Un profesor utiliza la herramienta de autoría para generar un tutor inteligente, posteriormente el estudiante interactúa con el laboratorio virtual y el sistema tutor generado.

1.5 Aportaciones

Las aportaciones principales de la presente investigación consisten en:

- El desarrollo de una herramienta de autoría para sistemas tutores inteligentes basados en MPRs y orientada a laboratorios virtuales.
- Una representación novedosa del modelo del estudiante mediante redes bayesianas que facilita su construcción a partir de la especificación del docente.

1.6 Organización del documento

El presente documento consta de 7 capítulos divididos de la siguiente manera. En los capítulos 2, 3, 4, se hace una recopilación y análisis del estado del arte de los principales componentes de la investigación. El capítulo 5 describe el proceso de desarrollo de la herramienta de autoría y la metodología utilizada para el desarrollo. El capítulo 6 muestran los resultados de la evaluación realizada sobre la herramienta y el capítulo 7 muestran las conclusiones de la presente investigación así como el trabajo futuro a realizar. Los apéndices ayudan a explicar de una mejor manera los componentes desarrollados en la investigación.

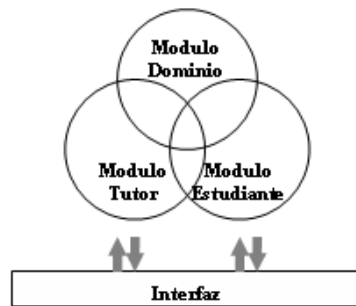
Capítulo 2

Sistemas Tutores Inteligentes

2.1 Sistemas Tutores inteligentes

Un tutor inteligente “*es un sistema de software que utiliza técnicas de inteligencia artificial (IA) para representar el conocimiento e interactúa con los estudiantes para enseñárselo*” (Freedman, Ali and McRoy 2000). Este tipo de sistemas permiten monitorear la interacción de un individuo en un ambiente educativo, evalúan al estudiante y le proporcionan retroalimentación basándose en la manera de responder a las actividades. La incorporación de los tutores inteligentes a las plataformas de enseñanza asistida por computadora y su aplicación en laboratorios virtuales brinda flexibilidad y adaptación. Además, los tutores inteligentes permiten una metodología de enseñanza personalizada para cada estudiante, basada en los estilos de aprendizaje y las necesidades de los individuos que utilizan los sistemas tutores inteligentes.

Un STI consta de tres módulos principales (ver Figura 2-0-1): módulo tutor, módulo estudiante y módulo dominio (también conocido como base de conocimiento o módulo experto); estos tres módulos están interconectados entre sí para poder mantener comunicación constante y mediante una interfaz se comunican con el usuario (Freedman, Ali and McRoy 2000), (T. Murray 1999). De esta manera se integra un STI genérico.



**Figura 2-0-1 Arquitectura básica de un sistema tutor inteligente.
(Salgueiro, et al. 2005)**

Enseguida se describe brevemente cada componente:

- Módulo tutor. Proporciona un modelo del proceso de enseñanza. Por ejemplo, este modelo contiene información acerca de cuándo hacer un repaso, cuando presentar un nuevo tema y que tema presentar, así como que actividad es más conveniente a realizarse.
- Módulo estudiante. Se usa como entrada del módulo tutor, con lo cual se permite tomar mejores decisiones sobre que técnica implementar. Este módulo proporciona al módulo tutor el nivel de conocimiento del estudiante, para que el módulo tutor asigne la siguiente actividad basado en el conocimiento del estudiante.

- Módulo dominio. Contiene la información que se le está enseñando a los estudiantes, incluyendo el material de las prácticas, exposiciones, repasos, cuestionarios, etc.

Además de los 3 componentes mencionados, puede identificarse un módulo interfaz, el cual se encarga de administrar el proceso en que el estudiante se comunica con el STI. Este módulo se visualiza cuando se utiliza cualquier sistema, incluyendo pantallas, cuadros de diálogos, menús, etc. El módulo interfaz es el mecanismo de comunicación entre un estudiante y un STI.

Los STI pueden clasificarse en 2 tipos principales: basados en modelos pedagógicos y sistemas orientados al desempeño. Cada uno de ellos puede contener diferentes variaciones, considerando las características del modelo del estudiante y del módulo tutor que los componen (VanLenh 1988) (T. Murray 1999), entre los cuales destacan:

- Tutores afectivos
- Tutores cognitivos
- Tutores basados en redes neuronales
- Tutores basados en reglas
- Tutores basados en redes bayesianas
- Basado en problemas
- Etc.

Existe una vasta investigación alrededor de los STI's. Freedman, Ali y McRoy (2000) describen las características de un sistema tutor inteligente, e identifican características del área que las difiere de otras de inteligencia artificial, señalando que los STI tienen una gran expectativa de futuras aplicaciones. En 2003, la IEEE desarrolló un estándar para los sistemas de

aprendizaje, en el cual muestra los lineamientos de desarrollo de este tipo de sistemas (IEEE 2003). Sánchez y Lama (2007) hacen una recopilación de las técnicas de inteligencia artificial aplicadas en la educación, donde destacan el uso de los sistemas tutores inteligentes en la enseñanza asistida por computadora.

Existen diversos enfoques para el modelado de un sistema tutor inteligente. Algunos utilizan un modelo cognitivo que busca determinar cuál es el modelo pedagógico más adecuado a implementar en cada usuario (Salgueiro, et al. 2005). Otros autores utilizan para el modulo tutor un enfoque de fuzzy con el cual determina qué tipo de inteligencia tiene el estudiante, basado en la teoría de inteligencias múltiples. (Zatarain Cabada, et al. 2008).

Para el modelado del estudiante algunos autores se basan en redes bayesianas, permitiendo modelar la incertidumbre y el estado de conocimiento del estudiante (López Puga y García García 2008),(Conejo, et al. 2001). Un enfoque fuzzy utilizado para el modelo del estudiante lo aplica (Nkambou 1999). El desarrollo teórico en estas áreas siguen diversas estrategias tanto pedagógicas como de inteligencia artificial y estan en constante evolución.

Cada uno de los componentes de un STI ha sido investigado a fin de mejorar su rendimiento. Salgueido y colaboradores (2005), investigan el desarrollo del módulo tutor buscando incorporar diferentes estrategias pedagógicas según el estilo de aprendizaje del estudiante. Forin y colaboradores, (2008) investigan el diseño de interfaces buscando un método estándar de desarrollo para tutores inteligentes. La importancia de la

interfaz estriba en que ésta es el único método de comunicación entre los usuarios y el STI.

Sobre el modelado del estudiante se ha trabajado en diversos ámbitos utilizando diversas estrategias. Salgueiro y colaboradores (2005) utilizan redes neuronales para el modelado del estudiante. Conati y colaboradores (1997) utilizan redes bayesianas para generar un modelo de estudiante en línea. Noguez (2005) propone una arquitectura basada en modelos probabilistas relacionales (MPR) para laboratorios virtuales de robótica móvil donde el modelado del estudiante está dado por un MPR. El concepto de MPR está descrito en (Getoor, et al. 2007) y (Friedman, et al. 1999), donde presentan de manera formal los algoritmos de inferencia y de representación. En López y García (2008) se presenta una recopilación y análisis de los casos de éxito de la utilización de redes bayesianas para el modelado del estudiante en algunos STI's. Hämäläinen y Vinni (2006), hacen una comparación entre las diversas técnicas de aprendizaje automático utilizado en el desarrollo de sistemas tutores inteligentes.

Problema del costo/tiempo de desarrollo

Los sistemas tutores inteligentes han mostrado buenos resultados durante su implementación pero tienen el problema de que son muy costosos en tiempo de desarrollo, llevando en promedio de 200 a 300 horas para una sesión de una hora de estudio (Aleven, et al. 2006). Además es necesario que personal experto en el área tanto de inteligencia artificial, ingeniería del conocimiento y de pedagogía trabaje juntos para el desarrollo de un STI. Por lo anterior, muchos de los profesores no utilizan este tipo de software debido a que no cuentan con los recursos necesarios para su desarrollo e implementación. Para atacar este problema se hacen esfuerzos para

disminuir estos tiempos lo cual se consigue mediante el desarrollo de *Shells* y herramientas de autoría; este tema se analizará en la sección siguiente.

2.2 SHELLS Y HERRAMIENTAS DE AUTORÍA PARA TUTORES INTELIGENTES

Los *Shells* y las herramientas de autoría son las dos estrategias más utilizadas para disminuir los tiempos y costos en el desarrollo de sistemas tutores inteligentes. Un *Shell* es una plataforma generalizada para la construcción de sistemas tutores inteligentes mientras que una herramienta de autoría se compone de un *Shell* con una interfaz para que un usuario no experto en programación de STI, pueda formalizar y desarrollar un sistema tutor inteligente (T. Murray 1999)

Sobre estos dos enfoques han trabajado diversos autores. Murray (1999) realizó una recopilación del estado del arte sobre las herramientas de autoría para sistemas tutores inteligentes indicando algunas debilidades sobre las herramientas. Ferrero y colaboradores (2001) desarrollan una herramienta de autoría para diagnóstico clínico. Alevén y colaboradores (2006) desarrollan una herramienta de autoría para tutores cognitivos.

Otro enfoque descrito por Moundridou y Virvou (2001), es la realización de tutores inteligentes funcionales sobre la WEB, enfocados a temas de álgebra y utilizando hipertextos. En este sentido Matsuda y Cohen (2005) presentan una evaluación de una de las herramientas de autoría construidas en la universidad Carnegie Mellon. López Puga y García García (2008) describen las bondades de utilizar redes bayesianas, para la construcción de STI, presentando una arquitectura con características de un *Shell*. Salgueiro y colaboradores (2005) presentan la elaboración de un *Shell*

para tutores inteligentes basado en redes neuronales y la selección de diversos modelos pedagógicos previamente identificados. Arruete (1998) presenta la construcción de un *Shell* para STI que es aplicado para elaborar la herramienta de autoría conocida como Iris-D (Ferrero, et al. 2001).

Kodaganallur y colaboradores (2006) utilizan dos enfoques para desarrollar tutores con una herramienta de autoría propuesta por ellos, un enfoque basado en restricciones y el otro basado en el objetivo. Mitrovic y colaboradores (2007) abordan el enfoque basado en restricciones como posible herramienta de autoría genérica. La característica que resalta en el enfoque basado en restricciones es que trabaja sobre una base de datos, aplicando las restricciones mediante el uso de SQL. Los autores aseguran que este enfoque muestra buenos resultados al seguir estrategias propuestas por los docentes. Huapaya y Arona (2002) proponen el diseño de un sistema de autoría especializado en STI's matemáticos. Ellos describen su metodología de desarrollo para dicho sistema, buscando desarrollar de manera independiente cada módulo que compone al STI. Zatarain Cabada y colaboradores (2008) desarrollan una herramienta de autoría para STI en ambientes móviles PDA y celulares, con ello permiten mayor movilidad durante el uso de los STI.

Los trabajos realizados en esta área muestran esfuerzos para disminuir los tiempos de desarrollo de sistemas tutores inteligentes para cierta clase de tutores, en particular tutores cognitivos. Y no existe una herramienta de autoría para el desarrollo de tutores basados en modelos probabilistas relacionales, los cuales son indispensables para ambientes con incertidumbre como son los laboratorios virtuales (Noguez y Sucar 2005).

Debido a esto es necesario desarrollar una nueva herramienta de autoría para STI de tal manera que el profesor de un curso pueda implementar un STI para un laboratorio virtual, sin necesidad de ser experto en STI ni en modelos probabilistas.

2.2.1 Herramientas de Autoría

Las herramientas de autoría son plataformas de desarrollo que incorporan la opción de estructurar la organización de un curso, colocar materiales de estudio, planear las estrategias de evaluación y la generación de sistemas tutores inteligentes (Murray y VanLehn 2000). Las herramientas de autoría son utilizadas para que personal no experto en el área de diseño de sistemas tutores inteligentes, pero con conocimiento del dominio donde será aplicado puedan desarrollar STIs de forma sencilla. Las herramientas de autoría permiten construir STI basados en las características que desean evaluar y con las restricciones que la herramienta requiera (Murray 1999).

El diseño de herramienta de autoría está enfocado al desarrollo de STI en dos vertientes. El primero, se enfoca en sistemas basados en modelos pedagógicos; es decir, trata de enseñar el contenido de un plan de estudios de alguna asignatura o curso. El segundo enfoque, trata de sistemas orientados al desempeño; en este enfoque las personas que interactúan con el STI necesitan recibir retroalimentación al instante sobre las decisiones tomadas, este es el caso de dispositivos simulados. La herramienta desarrollada en esta investigación sigue un enfoque híbrido. Primero, permite desarrollar un STI basado en un plan de estudios. Después, se enlaza el STI a un laboratorio virtual donde el STI dará retroalimentación a los estudiantes basándose en el desempeño obtenido en los diversos experimentos.

2.2.2 Clasificación de las Herramientas de autoría

La Tabla 2-1 muestra una clasificación definida por Murray (1999) de los tipos de herramienta de autoría existentes. Enseguida se explican cada uno de estos tipos.

Tabla 2-1 Tipos de Herramienta de autoría y tipo de STI que generan (T. Murray 1999)

| Tipo de Herramienta | Tipo de STI que desarrolla |
|---|-----------------------------------|
| Planeación y secuencia curricular. | basados en modelos pedagógicos |
| Estrategia de tutoría | basados en modelos pedagógicos |
| Dispositivos de simulación y equipos de entrenamiento | sistemas orientados al desempeño |
| Sistemas experto en el dominio | sistemas orientados al desempeño |
| Múltiples tipos de conocimiento | basados en modelos pedagógicos |
| Propósito especial | sistemas orientados al desempeño |
| Hipermedia inteligente y adaptiva | basados en modelos pedagógicos |

La tabla 2-1 se enfoca a las técnicas utilizadas por las herramientas para el desarrollo de STI, actualmente la evolución del INTERNET y los dispositivos móviles podrían añadir nuevas estrategias de desarrollo e instrucción aplicando estrategias de aprendizaje colaborativo y distribuido. Dichas estrategias deben ser utilizadas en herramientas de autoría para STI que trabajen en dichos ambientes.

A continuación se describe brevemente cada tipo de herramienta de la clasificación propuesta por Murray (1999) en donde se explican las características de los diversos tipos de herramientas así como las clases de STI que desarrolla cada herramienta.

Descripción de los diferentes tipos de herramientas según Murray:

- **Planeación y secuencia curricular:** Este tipo de herramientas de autoría se basa en la organización de los materiales educativos (unidades de enseñanza o “elementos curriculares”) en una jerarquía de cursos, módulos, lecciones, presentaciones, etc.; las cuales están asociadas con prerrequisitos y otras relaciones. El encadenamiento inteligente de los elementos curriculares (contenidos o temas) es el núcleo de este tipo de herramientas, con la cual se lleva al alumno a realizar todas las tareas de un plan de trabajo.
- **Estrategias de tutoría:** Las herramientas de autoría de esta categoría sobresalen en la flexibilidad de representar diversas estrategias de enseñanza. Funcionan de manera similar que las de planeación y secuencia curricular sólo que añaden la posibilidad de filtrar más la información basada en estrategias propuestas por los profesores. Una característica de los tutores desarrollados con este tipo de herramientas es el de tener almacenadas diversas estrategias, escogidas de acuerdo a algún resultado en los experimentos.
- **Dispositivos de simulación y equipos de entrenamiento:** Los STI contruidos con este tipo de herramienta tienen el propósito de que el estudiante identifique un dispositivo y entienda como utilizarlo. Este tipo de herramientas son utilizadas menos para fines pedagógicos y más para fines de entrenamiento. A diferencia de las categorías anteriores donde el estudiante aprende a través de lecciones, el estudiante adquiere el conocimiento realizando actividades en un simulador.

- **Sistemas expertos y tutores cognitivos:** Un tipo de tutores inteligentes que incluye reglas basadas en modelos cognitivos del dominio son elaborados con este tipo de herramientas. La idea principal se centra en generar sistemas expertos para la resolución de un problema, después plantear un problema similar al estudiante y monitorear que tanto se aleja de la ruta crítica de solución propuesta por el experto. La principal desventaja de este enfoque es que está muy limitado en su campo de acción, debido a la dificultad de modelar las rutas críticas para resolver los problemas.
- **Conocimientos Múltiples:** Las teorías pedagógicas clasifican el conocimiento y las tareas en categorías discretas, además dan métodos de enseñanza para cada una de ellas. En este tipo de herramientas los conocimientos que se pueden enseñar están limitados a las categorías que hayan realizado los autores. El esquema de trabajo de estas herramientas es identificar qué tipo de conocimiento se va a enseñar y seguir un plan de trabajo preestablecido para ese conocimiento. El hecho de que las estrategias ya estén definidas para cada tipo de conocimiento que se quiera enseñar es una fortaleza, pero a la vez una debilidad de este tipo de herramienta de autoría, debido a que el docente no puede agregar nuevos modelos o reglas de enseñanza propias.
- **Sistemas de propósito especial:** Este tipo de herramientas están muy especializadas para un solo campo de trabajo. Por ejemplo, herramientas para tutores inteligentes en la enseñanza de álgebra en nivel bachillerato o el caso de la herramientas de autoría utilizada para el desarrollo del sistema de enseñanza y diagnóstico Iris-D, el cual se especializa en diagnóstico médico. (Ferrero, et al. 2001)

- **Hipermedia inteligente y adaptiva:** Las herramientas de este tipo incorporan las estrategias de las categorías anteriores para trabajar en entornos Web. La mayor parte de las estrategias consisten en que, basado en un plan curricular, presentan hipertextos que guíen al estudiante a los temas. La mayoría las herramientas de esta categoría no generan una interfaz gráfica para un STI, sino que funcionan como un motor de búsqueda que es llamado desde una página web.

La herramienta desarrollada en esta investigación es clasificada en las categorías de “planeación y secuencia curricular” y “Estrategias de tutoría”. Además, aunque la herramienta no permite el desarrollo de dispositivos simulados, los STIs desarrollados por la herramienta son capaces de incorporar dispositivos simulados mediante el uso de laboratorios virtuales. Por tal razón la herramienta que proponemos es considerada para un híbrido entre STIs basados en modelos pedagógicos y sistemas enfocados al desempeño.

2.3 Estrategias de Desarrollo

Diversas estrategias para la construcción de cada módulo del sistema tutor inteligente han sido desarrolladas. Para nuestra herramienta utilizamos las propuestas por Jonassen, Reeves, Hong, Harvey & Peters (1997) donde usan una representación tipo mapa conceptual para el módulo dominio. Para el modelo del estudiante se utiliza la representación a través de modelos probabilistas relacionales propuestos por Noguez (2005). Para el modelo del tutor se utiliza la propuesta por Bloom, Engelhart, Murst, Hill & Drathwohl (1956) para operar las estrategias pedagógicas, la cual ha sido utilizada por Huapaya y Arona (2002) para la producción de reglas del tipo “Sí Resulta== a parámetro Entonces Acción Pedagógica a tomar”.

2.4 Resumen del Capítulo

En este capítulo se han descrito las características principales de los sistemas tutores inteligentes, al igual que los esfuerzos de investigación para desarrollar este tipo de sistemas de forma cada vez más rápida. Se habló de diversas estrategias para modelar, evaluar y generar a un sistema tutor inteligente, así como herramientas de autoría diseñadas para algunos tipos de STIs. Aunque existen diversos esfuerzos para el desarrollo de herramientas de autoría, no se considera el caso de STIs basados en modelos probabilistas y orientados a laboratorios virtuales.

En el siguiente capítulo se habla sobre los modelos gráficos probabilistas, ya que ésta es la estrategia que se utiliza para el desarrollo de la herramienta de autoría que se propone en esta investigación.

Capítulo 3

Modelos Probabilistas

3.1 Redes Bayesianas.

Las redes bayesianas son una representación gráfica de dependencias para razonamiento probabilista (Pearl 1998). Una red bayesiana es un modelo gráfico probabilístico multivariado, que relaciona un conjunto de variables aleatorias mediante un grafo acíclico dirigido que muestra relaciones de dependencia e independencia condicional. Gracias a los algoritmos de propagación de probabilidades (Dechter 1996) basados en el teorema de Bayes, las redes bayesianas son eficaces en la estimación de probabilidades dada nueva evidencia. Los nodos de una red bayesiana representan variables y los arcos representan relaciones de dependencia entre las variables. La Figura 3-1 muestra un ejemplo simple de una red bayesiana.

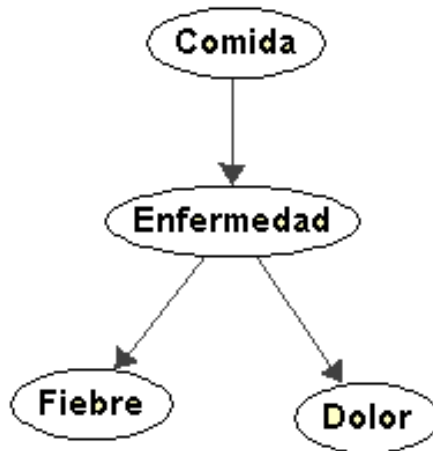


Figura 3-1 Ejemplo de una red bayesiana simple. Representa que la aparición de fiebre o dolor depende de si se tiene o no una enfermedad, la cual puede ser adquirida por consumir comida insana.

En una red bayesiana un nodo, X , es independiente de todos los nodos que no sean sus hijos dados sus nodos padres $Pa(X)$. Por ejemplo en la red de la figura 3-1, “comida” es condicionalmente independiente de “fiebre” y “dolor” dado “enfermedad”; es decir, si se conoce el valor de enfermedad, el valor de “comida” no afecta las probabilidades de “fiebre” y “dolor”. Una red bayesiana representa en forma compacta todas las independencias condicionales entre las variables del modelo. Cada nodo es una variable que tiene asociado un número determinado de valores y una tabla de probabilidad condicional donde se muestran los valores del nodo dados sus padres. La figura 3-2 muestra la tabla de probabilidad condicional para el nodo enfermedad.

| Comida | Buena | Contaminada |
|----------|-------|-------------|
| presente | 0.1 | 0.8 |
| ausente | 0.9 | 0.2 |

Figura 3-2 Tabla de probabilidad condicional para el nodo enfermedad de la figura 3-1. Por ejemplo, si se ingirió comida contaminada existe una probabilidad de .8 de adquirir una enfermedad.

La probabilidad conjunta de todas las variables en el modelo, está dada por el producto de las probabilidades de cada variable dados sus padres la cual se resume en la ecuación 3.1:

$$P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i | Pa(X_i)) \quad (3.1)$$

La propagación de probabilidades consiste en dada cierta evidencia, (variables con valores conocidos), calcular las probabilidades de las otras variables utilizando algoritmos de propagación (Dechter 1996).

3.2 Redes Bayesianas Dinámicas

Las redes bayesianas dinámicas representan procesos dinámicos; consisten en una representación de los estados del proceso en un tiempo (red base) y las relaciones temporales entre dichos procesos (red de transición). Un ejemplo de este tipo de redes bayesianas se ilustra en la Figura 3-3, la cual toma como red base la de la figura 3-1. La red bayesiana dinámica representa a la red bayesiana en dos estados de tiempo, tiempo=0 que representa los valores anteriores y tiempo=1 que tomaría el valor actual de las variables del modelo. De esta manera podemos estimar los valores actuales considerando los valores previos.

En la red bayesiana dinámica se realiza el proceso de inferencia típico de una red bayesiana, pero considerando los valores que mostraban los nodos con evidencias previas. Los parámetros que se tienen en una red bayesiana dinámica son las probabilidades iniciales P (Enfermedad), probabilidad de transición $P(\text{Enfermedad}_{t+1} | \text{Enfermedad})$ y probabilidad de

observación $P(\text{Fiebre} \mid \text{Enfermedad})$. Las probabilidades iniciales son obtenidas de la red base, la probabilidad de transición es la probabilidad de una variable dado su valor en el tiempo anterior y la probabilidad de observación es la probabilidad del hijo dado su padre.

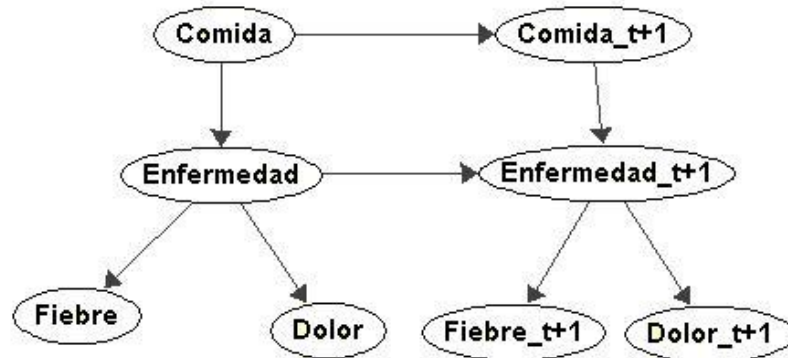


Figura 3-3 Ejemplo de una red bayesiana dinámica tomando como red base la Figura 3-1

Los parámetros que se observan en la red bayesiana dinámica son los siguientes:

- Probabilidades iniciales: $P(\text{Estado})$.- en nuestro caso la probabilidad inicial está dada por la importancia capturada en el temario.

- Probabilidades de transición: $P(\text{Estado}+1 \mid \text{Estado})$ La probabilidad de transición es la importancia que el maestro ha asignado ya sea al historial o al experimento actual.

Suposiciones Básicas

- Proceso markoviano - el estado actual sólo depende del estado anterior; es decir sólo hay arcos entre experimentos consecutivos.

- Proceso estacionario en el tiempo - las probabilidades de transición, $P(\text{Estado}+1 \mid \text{Estado})$, no cambian en el tiempo.

El modelo del estudiante generado por nuestra herramienta consiste en una red bayesiana dinámica, utilizando una representación más expresiva basada en modelos probabilistas relacionales, los cuales se describen en la siguiente sección.

3.3 Modelo probabilista relacional

Los modelos probabilistas relacionales (PRMs, por sus siglas en inglés) son una formalización de las redes bayesianas orientadas a objetos, que integran las fortalezas de los modelos probabilistas y la lógica relacional (Getoor, et al. 2007). Los MPRs son representaciones que se basan en los mismos principios de las redes bayesianas, pero se consideran más expresivos, debido a que cada clase es un objeto cuyos componentes son nodos de una red bayesiana. Estos modelos permiten representar varias redes bayesianas en el mismo modelo, debido a que se pueden instanciar de diversas formas.

En la Figura 3-4 se muestra un ejemplo de un MPR utilizado para el modelado de los estudiantes en laboratorios virtuales de robótica móvil (Noguez 2005). Este modelo incluye las clases: “estudiante,” “objetos de conocimiento,” “resultados de actividades” y “otras actividades”. Las líneas punteadas representan la relación entre las clases y las flechas las dependencias entre los atributos de las clases, que corresponde a los nodos de una red bayesiana. En esta estructura se observa como los nodos que componen cada clase pueden formar parte de una red bayesiana.

A diferencia de las redes bayesianas, en el modelo probabilista relacional se instancian sólo aquellos nodos que son utilizados para poder desarrollar el proceso de inferencia de la red; es decir, se generan múltiples

redes bayesianas que van de acuerdo al uso, permitiendo de esta forma hacer más eficiente el proceso de inferencia, ya que los valores que no interfieren no son analizados.

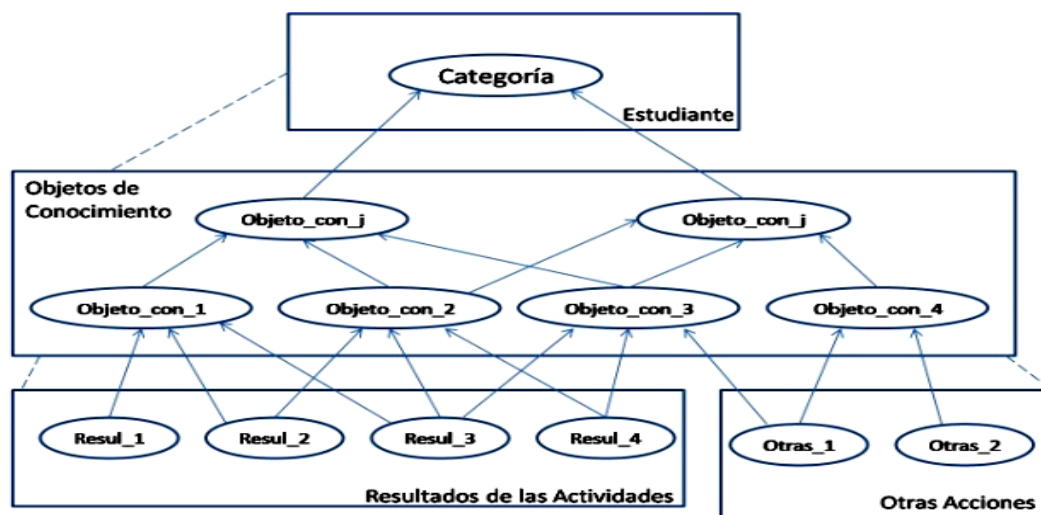


Figura 3-4 Ejemplo de un modelo probabilista relacional para representar el conocimiento de un estudiante (Noguez 2005).

El modelo probabilista relacional para representar el conocimiento de un estudiante propuesto por (Noguez 2005) es la base del modelo de esta investigación. Con base en este modelo se desarrolló un sistema de autoría que permite a un profesor (no experto en STI ni en modelos probabilistas) definir en forma muy sencilla el modelo del estudiante. Para ello, se incorporan aspectos de representación del problema que facilitan a un docente no experto en redes bayesianas el capturar la información de manera transparente. Esto se logra mediante el uso de formularios y esquemas basados en jerarquías de temas, sub-temas y conceptos, con probabilidades que están relacionadas con la cantidad de contenido de cada nivel con respecto a su nivel superior.

3.4 Tutores Inteligentes basados en modelos probabilistas

El uso de modelos probabilistas para el modelado del estudiante en STI ha sido estudiado por varios autores. El Sistema SE-Coach (Conati, et al. 1997) contiene un módulo para evaluar explicaciones del estudiante, mientras el estudiante aprende a través de ejemplos. El modelo de este sistema es una red bayesiana que incluye:

1. Una representación de la solución del ejemplo, la cual se genera automáticamente a partir del conjunto de reglas.
2. Los nodos que reflejan la lectura de acciones y la autoexplicación.

Otro ejemplo de este tipo de sistemas es HYDRIVE, que es un STI para el diagnóstico de fallas del sistema hidráulico en aviones F-15 (Mislevy y Gitomer, 1996). El sistema se basa en una red bayesiana para inferir el conocimiento con el que cuenta el estudiante. HYDRIVE clasifica a los estudiantes en “experto,” “bueno,” “ok,” y “novato,” donde cada caso está representado por variables probabilistas. El sistema DT-Tutor (Murray y VanLehn 2000) es un STI que emplea una red bayesiana para representar las creencias del tutor sobre los problemas que puede tener el estudiante relacionados con el conocimiento a adquirir. DT-Tutor utiliza un grafo de solución del problema, donde una red jerárquica de dependencias representa la solución del problema.

Conejo describe las ventajas de los enfoques bayesianos y propone a las redes bayesianas como una plataforma para el desarrollo de STI (Conejo, et al. 2001). López Puga y García García (2008), hacen una recopilación de los trabajos que han utilizado redes bayesianas, describiendo los resultados favorables analizados desde un punto de vista pedagógico.

El laboratorio virtual de robótica móvil del ITESM (ITESM 2008) desarrollado con base en la propuesta de Noguez (2005), cuenta con un modelo del estudiante basado en MPRs, el cual construye una red bayesiana para cada nivel de conocimiento del estudiante. La Figura 3-5 muestra la arquitectura de interacción entre el STI y el laboratorio virtual de robótica móvil, donde la parte de la categorización inicial es un módulo que inicializa las variables de la red bayesiana que utilizará para el modelado del estudiante.

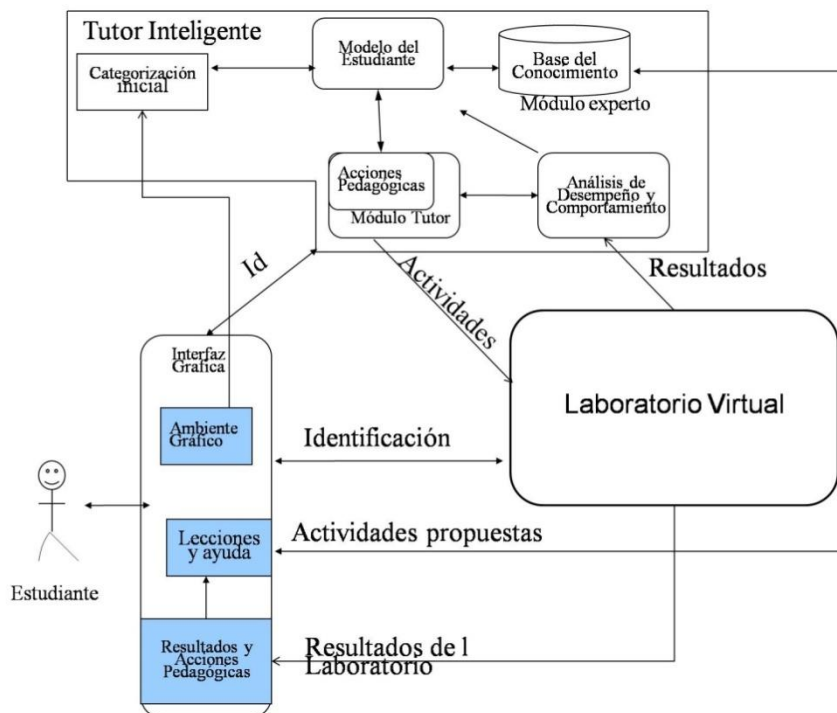


Figura 3-5 Arquitectura de interacción entre un tutor inteligente y un laboratorio virtual (Noguez, 2005). La figura muestra la arquitectura de trabajo del sistema de laboratorio virtual de robótica (ITESM 2008) en el cual se distingue los 3 componentes principales: Tutor Inteligente, Laboratorio Virtual y al Estudiante.

Los tutores inteligentes que se desarrollan con la herramienta de autoría propuesta en este trabajo de investigación, están estructurados para trabajar bajo un esquema similar al de la Figura 3-5 propuesta por Noguez, por lo que a continuación analizaremos el modelo propuesto por Noguez. En el capítulo 4 veremos los laboratorios virtuales y la importancia de incorporar un STI a un laboratorio virtual.

3.5 Modelo probabilista relacional del estudiante

El modelo probabilista relacional del estudiante propuesto por Noguez (2005) presenta diferentes ventajas, entre las que destacan la capacidad expresiva para modelar estudiantes en ambientes más complejos que los que permiten las redes bayesianas puras; esto gracias a la abstracción de las clases de los MPR. Otra característica importante de este modelo es que permite instanciar aquellos nodos que se van a utilizar por lo que se generan las instancias sólo cuando son utilizadas; esto permite la propagación de evidencia solo en la parte del modelo que está siendo usada. También, permite definir tablas de probabilidades condicionales para cada una de las categorías.

El modelo consiste en 7 clases: Estudiantes, Temas de conocimiento, Sub-temas de conocimiento, Conceptos básicos, Comportamiento de exploración, Resultados de actividades y Experimentos. Los cuales están asociados con el fin que el estudiante adquiera el mayor conocimiento posible basándose en los experimentos.

En la figura 3-6 se muestra el esqueleto de MRP que propone Noguez donde se ilustra que entre los enlaces es posible generar una red bayesiana para cada estudiante manejando valores de probabilidad independientes.

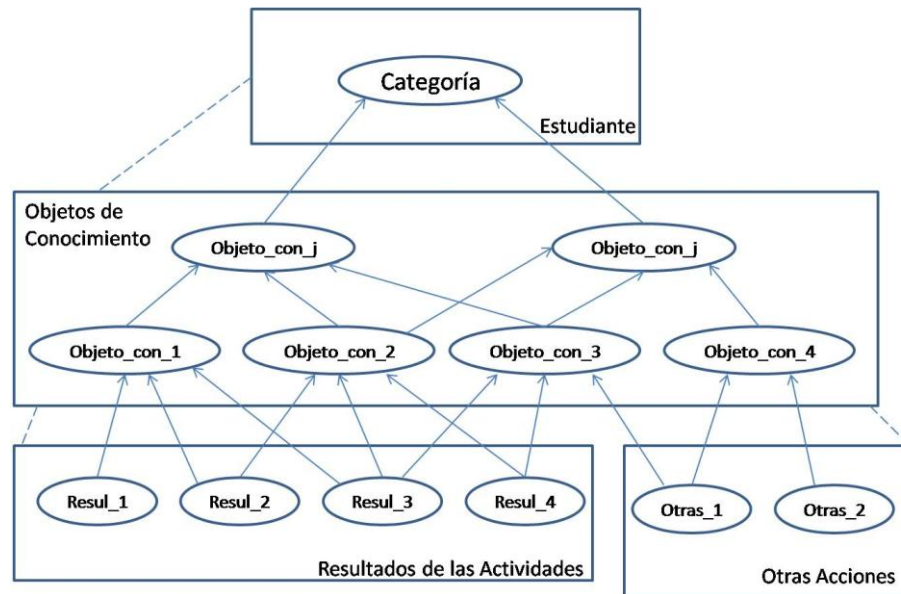


Figura 3-6 Esqueleto propuesto por Noguez(2005) para el ámbito de robótica

Los parámetros principales del modelo son al igual que en las redes bayesianas las TPC las cuales pueden ser diferentes en dependencia de la categoría del estudiante (novato, intermedio, avanzado), esto hace al modelo más flexible. Es importante aclarar que las TPC son definidas por docentes expertos en el tema a enseñar. Los métodos de inferencia que utiliza el modelo son los mismo que implementan las redes bayesianas; solo que el MRP permite hacer inferencias en un menor número de nodos ya que solo realiza la inferencia en la red que esta utilizando.

La importancia de este modelo es que permite su adaptación para otros ambientes de manera sencilla. En especial se puede utilizar para la interacción con laboratorios virtuales, ya que el modelo probabilista relacional del estudiante fue diseñado para laboratorios virtuales y aplicado con éxito en un laboratorio virtual de robótica (Noguez y Sucar 2005). Para más detalle del modelo ver Noguez (2005).

3.6 ELVIRA

Elvira es un software diseñado para la edición y evaluación de modelos gráficos probabilistas, específicamente redes bayesianas y diagramas de influencia (Elvira 2001). Elvira cuenta con un formato propio para la codificación de los modelos, un lector-intérprete para los modelos, una interfaz gráfica de construcción, opciones de modelos canónicos (puertas OR, AND, MAX, etc.) y algoritmos de inferencia, entre otras aplicaciones. Elvira está desarrollado en JAVA 2.0 y su código se encuentra libre para ser descargado en su página web (Elvira 2001).

En este trabajo se utilizan los módulos de inferencia del sistema Elvira para determinar las probabilidades de las diferentes variables incluidas en el modelo del estudiante, como se detalla más adelante. Se utiliza su estructura para definir y manipular la red bayesiana que sirve de base para el modelado del estudiante.

3.7 Resumen del capítulo

En este capítulo se mostraron las características de los modelos gráficos probabilistas, la forma de representación que utilizan y la manera en que manejan la incertidumbre mediante el uso de valores probabilísticos para las variables del modelo. Se describieron algunas ventajas de utilizar los modelos gráficos probabilistas en el diseño de un sistema tutor inteligente. También se habló del uso de modelos probabilistas relacionales para el modelado del estudiante en sistemas tutores inteligentes para ambientes virtuales.

En el siguiente capítulo se describen a los laboratorios virtuales, y la importancia de incorporar un STI, lo que implica la necesidad de desarrollar herramientas de autoría que permitan construir STIs para laboratorios virtuales.

Capítulo 4

Laboratorios Virtuales

4.1 Definición de laboratorio virtual

Un laboratorio virtual (LV) es un ambiente heterogéneo simulado o teleoperado, que permite a un grupo de personas trabajar en dicho laboratorio para la solución de problemas (Wagner 1999). En un laboratorio virtual un estudiante interactúa para resolver ejercicios propuestos por algún docente y la retroalimentación que recibe la evalúa el propio estudiante cuando analiza los resultados que obtuvo de algún experimento.

Los laboratorios virtuales permiten compartir recursos y evitar costos que se tendrían al operar un laboratorio real. En la mayoría de los laboratorios virtuales se realizan experimentos para los cuales el estudiante proporciona los elementos para que el sistema simule o realice el experimento. Se trata de un sistema de interacción donde el estudiante indica que valores tomarán las variables involucradas en el experimento para que el sistema procese la información.

Se puede ver a un laboratorio virtual, como un entorno en el cual los estudiantes pueden experimentar a través del sistema; esto permite a los

profesores identificar los conocimientos a reforzar para tener mejores resultados en los diferentes experimentos. El laboratorio permite al estudiante resolver actividades especificadas por los profesores, para que después el profesor, el propio estudiante u otro sistema evalúe el resultado de las actividades.

El uso de laboratorios virtuales anexan dos ventajas principales al proceso enseñanza aprendizaje primero permite desarrollar un aprendizaje constructivista y segundo fomenta la capacidad de análisis y pensamiento crítico en el estudiante.

La principal desventaja del uso de laboratorios virtuales para enseñanza es que en ocasiones el estudiante no tiene claro si fue exitoso o no en la resolución de un experimento; es decir, no hay una retroalimentación instantánea, sino que el usuario debe interpretar los resultados. Este problema puede ser resuelto si incorporamos un STI al laboratorio virtual.

4.2 Clasificación de laboratorios virtuales

Los laboratorios virtuales se clasifican en dos grupos principales, laboratorios virtuales simulados y laboratorios virtuales remotos. En estas dos categorías podemos clasificar todos los laboratorios virtuales, aunque existen casos que combinan ambos. Enseguida se explica cada uno.

Laboratorios Virtuales Simulados. Este tipo de laboratorio consiste en un modelo simulado del material utilizado para realizar los experimentos, así como del ambiente de experimentación (Joler and Christodoulou 2001). La principal ventaja de este tipo de laboratorios es proporcionar accesos

ilimitados al laboratorio, sin poner en riesgo al estudiante, ni al equipo a utilizar.

Laboratorios Virtuales Remotos. Estos laboratorios permiten el acceso remoto a un laboratorio real mediante una computadora que trabaja en el laboratorio la cual permite realizar un experimento y observar sus resultados (Massino and Sheridan 1994). En estos ambientes el estudiante tiene la ventaja de estar operando un equipo real y se enfrentará a situaciones que son difíciles de modelar en un simulador. Otra característica importante de este tipo de laboratorios es que se puede tener acceso al laboratorio fuera de los horarios de operación de los centros educativos.

4.3 Ventajas de la utilización de laboratorios virtuales

Existen varias ventajas de utilizar laboratorios virtuales; diversos autores muestran características diferentes, aunque señalan que los laboratorios virtuales no remplazarán a los laboratorios reales. Algunas de las ventajas más importantes son:

Reducción de costos. Si un estudiante o una institución de educación no cuenta con los recursos necesarios físicamente, es posible utilizar laboratorios virtuales para realizar los experimentos. Con lo anterior se evita comprar equipo que puede ser muy costoso.

Promoción de la transferencia tecnológica. El *software* desarrollado y depurado de un laboratorio virtual puede ser añadido a una biblioteca y ser utilizado por diversos estudiantes.

Ampliación del tiempo de acceso. Debido a que los laboratorios virtuales trabajan a través de sistemas de *software* se pueden colocar en un

servidor para tener acceso en todo momento, permitiendo así que se elimine el problema del horario de oficina de asistir al laboratorio cuando se encuentre el responsable del mismo.

Comparar Resultados de Experimentos. Se pueden hacer comparaciones entre los resultados de diversos estudiantes o del mismo estudiante, permitiendo analizar las estrategias y los errores cometidos para una próxima retroalimentación.

Reducir riesgos al recibir capacitación. Utilizando entornos virtuales se reduce la posibilidad de algún accidente al operar equipos peligrosos. Un ejemplo común son los simuladores de vuelo donde son entrenados los pilotos.

Incorporar estrategias didácticas. Al hablar de estrategias didácticas podemos hacer referencia a la incorporación en la educación de tutores inteligentes, ayudas en línea, monitoreo remoto por parte del profesor, etc. Por tal motivo los laboratorios virtuales se ven como el entorno ideal de trabajo de un STI, ya que permitirá dar al estudiante una retroalimentación continua analizando el resultado de sus experimentos de manera constante. Algunos autores han trabajado en el desarrollo de laboratorios virtuales incorporando características de los STI. La Tabla 4-1 tomada de Noguez (2005) muestra algunos ejemplos de laboratorios virtuales que incorporan técnicas de los STIs.

Los laboratorios virtuales han mostrado tener muchas ventajas para mejorar el proceso enseñanza-aprendizaje. Sin embargo, actualmente es necesario incorporar alguna forma de retroalimentar al estudiante durante el uso de un laboratorio virtual; esto lleva a generar nuevos laboratorios con estas características o a agregar características a los laboratorios existentes que permitan esta retroalimentación. La ventaja de desarrollar una

herramienta de autoría para STI para LVs existentes es que el profesor responsable del curso pueda implementar un STI que permita al estudiante recibir una retroalimentación adecuada sin necesidad de esperar a que el profesor revise cada experimento.

Tabla 4-1 Ejemplos de laboratorios virtuales que utilizan facilidades de los STI (Noguez 2005)

| Sistema | Representación del conocimiento | Modelo del Estudiante | Método de enseñanza | Planificador o Tutor | Tipo de interfaz |
|-----------------|---|-----------------------|-------------------------|---|------------------------------------|
| TESS | Simulación. Bloques de Funciones. Maquina de Edo. | No Existe | Lecciones Ejercicios | Estático | Hipermedia Animaciones |
| MILAS | Simulación | Profesor real | Guía didáctica fija | No existe | Gráfica Sonora |
| GLUCOMAT | Simulación | Sistema Experto. | Basado en Problemas | Dinámico con módulo de decisiones | Gráfica, comunicación por mail. |
| CMOS | Simulación | Multiagente | Basado en problemas | No Existe. Presenta resumen de experto | Multimedia |
| VES | Simulación Ecocardiográfica | Basado en ontología. | Basado en casos | No existe | Gráficas 3d de interacción |
| ITESM RM | Simulación | Basado en MPRs | Basado en problemas | Basado en reglas | Graficas 3d Texto |

A continuación se realiza una breve descripción de los laboratorios de la tabla 4-1:

TESS

TESS es un sistema tutorial basado en hipertextos para el diseño de controladores lógicos programables (PLC por sus siglas en ingles). El estudiante recibe un problema de programación a resolver. El modulo tutorial incluye un módulo de aprendizaje que explica los elementos distintos que

componen al experimento. Este sistema se utiliza como complemento al curso presencial de PLCs.

MILAS

MILAS es un laboratorio virtual basado en simuladores. Esta diseñado con fines de entrenamiento y capacitación de personal para la operación de taladros. El sistema cuenta con guías, exámenes y materiales didácticos; pero carece de un modelo del estudiante.

GLUCOMAT

GLUCOMAT es un tutor inteligente aplicado a un laboratorio virtual para la enseñanza de un sistema dinámico complejo, basado en un modelo matemático para la regulación homeostática de la glucosa. Cuenta con un sistema experto que se compone de los tres módulos principales de un STI. El sistema proporciona retroalimentación al finalizar el experimento con base en los resultados.

CMOS

El proyecto CMOS (Cockpit Maintenance Operations Simulator) desarrolla un ambiente de entrenamiento moderno para operaciones de mantenimiento de equipo de aviación. El sistema cuenta con un STI que administra los materiales a enseñar e interactúa de forma dinámica con los estudiantes mediante diversos experimentos simulados donde existe un agente computacional para cada una de las áreas. Este sistema brinda retroalimentación si y solo si el estudiante lo solicita explícitamente, de lo contrario simplemente almacena los resultados.

VES (Virtual Electrocardiogram System)

Es un sistema aplicado a medicina que permite el modelado del corazón en ambientes gráficos basados en simulación. El sistema tiene acoplado a un STI el modulo tutor se basa en una descripción de un caso clínico particular. Y la retroalimentación es con una comparación entre el estado actual y el caso particular.

ITESM'RM

Este es un laboratorio virtual para el ámbito de robótica móvil, cuenta con un STI desarrollado para el laboratorio virtual específicamente, cuyo modelo del estudiante está basado en los MPR y el modulo tutor se basa en reglas que determinan las acciones a tomar dados los resultados de los experimentos. La parte de los experimentos es presentada mediante una interfaz simulada en JAVA 3D.

4.4 Resumen del capítulo

En este capítulo se describieron las características de los laboratorios virtuales, los diferentes tipos de laboratorios virtuales existentes y las ventajas de implementar un laboratorio virtual; además se mencionó como el uso de laboratorios virtuales puede mejorar el proceso enseñanza-aprendizaje. Los laboratorios virtuales requieren que los estudiantes reciban una retroalimentación adecuada, para tal tarea se propone usar STIs que interactúen con el estudiante y proporcionen dicha retroalimentación.

Dada la complejidad y tiempo requerido para el desarrollo de STIs, proponemos el desarrollo de una herramienta de autoría que facilite su construcción e incorporación a laboratorios virtuales.

En el capítulo siguiente hablaremos sobre el desarrollo de la herramienta de autoría propuesta, las características que tiene y los requisitos para su implementación.

Capítulo 5

Desarrollo de la Herramienta de Autoría

5.1 Descripción General

La herramienta de autoría para tutores inteligentes consiste en un sistema para desarrollar STI's que puedan monitorear a un estudiante que se encuentre trabajando en un laboratorio virtual. Además, el STI debe ser capaz de proponer experimentos y lecciones en caso de que el estudiante requiera conocimiento adicional sobre un tema.

La herramienta desarrollada está basada en el modelo propuesto por Noguez (2005), para el modelado de estudiantes en laboratorios virtuales de robótica móvil. Las modificaciones realizadas en esta investigación al modelo de Noguez nos permiten generalizar la herramienta para laboratorios virtuales de distintas áreas, siempre y cuando se ajusten a los requerimientos que se describen en el apéndice A de generación de laboratorios virtuales. El esquema general de trabajo de la herramienta de autoría se describe en la

Figura 5-1 y el esquema de uso del producto generado por la herramienta en la figura 5-2.

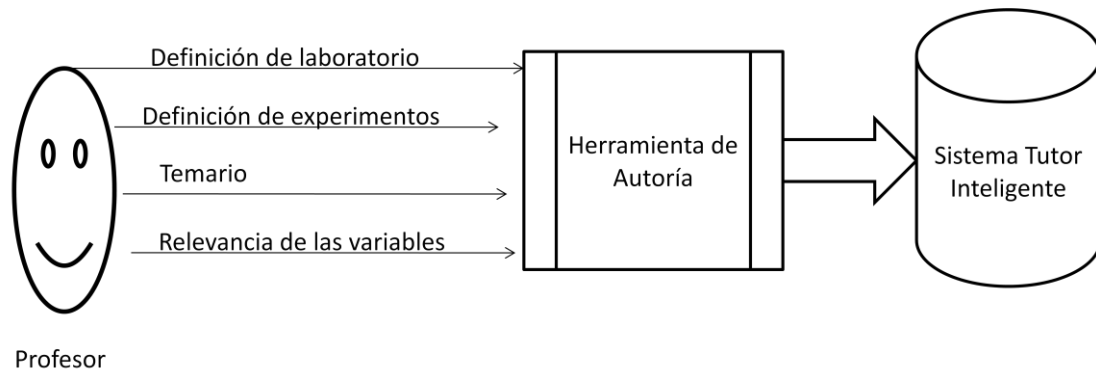


Figura 5-1 Generación de un nuevo STI. La generación de un nuevo STI implica que el docente proporcione información sobre el laboratorio (nombre y variables), definición de experimentos (nombre, descripción y variables a evaluar en cada experimento), temario, y relevancia de las variables (pesado de importancia para la evaluación).

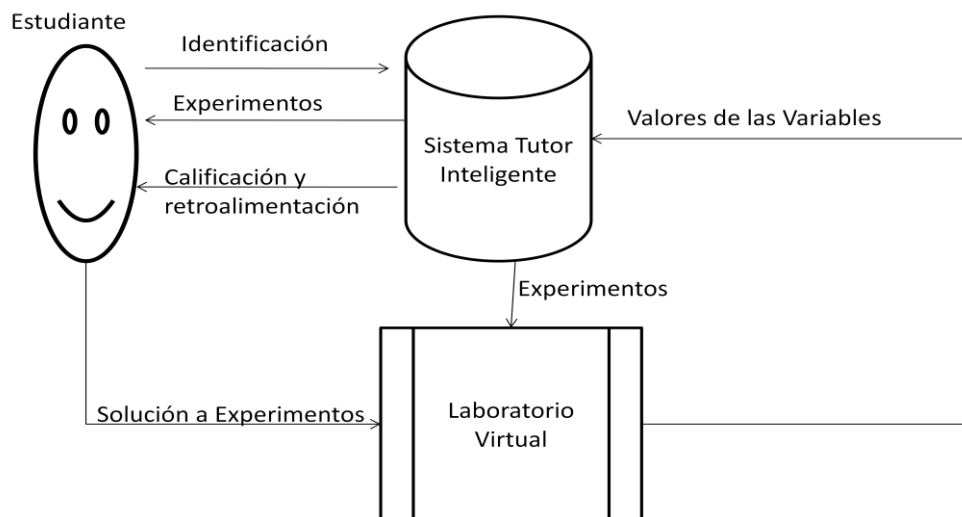


Figura 5-2 Esquema de trabajo del STI con el laboratorio virtual. El STI requiere que el estudiante se identifique, le asigna un experimento que el estudiante debe resolver en el laboratorio virtual, luego evalúa el resultado y proporciona al estudiante la retroalimentación adecuada, ya sea un nuevo experimento o una lección.

La herramienta requiere de la existencia de un laboratorio virtual (LV), del cual extrae los resultados obtenidos por un estudiante que interactúe con el laboratorio. La herramienta proporciona una interfaz de captura para dar de alta toda la información asociada al laboratorio; dicha interfaz permite capturar la localización del laboratorio, los experimentos que se pueden resolver, así como las variables que se encuentran asociadas a cada experimento. La figura 5-3 ilustra la interacción entre los actores que participan en el proceso de construcción del STI y su posterior utilización por el estudiante. A continuación se describen los procesos que permiten la relación entre actores y que se muestran la figura 5-3:

1. Capturar definición.- Genera los archivos necesarios para la definición de un STI, se capturan los temarios, los experimentos del laboratorio, los valores para las reglas del módulo tutor, así como los pesos de las variables que generan la base de conocimiento. En esta parte el docente configura al STI.
2. Generar tutor inteligente.- La herramienta utiliza los archivos de configuración del paso 1 para generar el STI que se acopla al laboratorio virtual.
3. Inicializar al estudiante.- El estudiante se identifica con el STI para empezar a trabajar. Si el estudiante no está registrado, el STI genera archivos de configuración del nuevo estudiante, si este utiliza el modelo del estudiante correspondiente
4. Activar experimento.- El STI identifica el nivel del estudiante, le asigna un experimento de manera aleatoria correspondiente al nivel del estudiante y se muestra la descripción del experimento.

5. Mostrar experimento.- El STI despliega la interfaz del laboratorio para que el estudiante realice el experimento.
6. Resolver experimento.-El estudiante realiza los experimentos en el laboratorio virtual.
7. Enviar resultado del experimento.- El LV envía los valores que el estudiante le asigno a cada variables asociada al .
8. Enviar lecciones.- El STI compara el resultado de las variables con los valores esperados para dicho experimento, las califica y evalúa en el modelo del estudiante propagando la evidencia obtenida de las calificaciones. Se utilizan los nuevos valores de conocimiento obtenidos por el modelo del estudiante para determinar el nuevo nivel del estudiante y decidir si envía una lección al estudiante o un nuevo experimento.

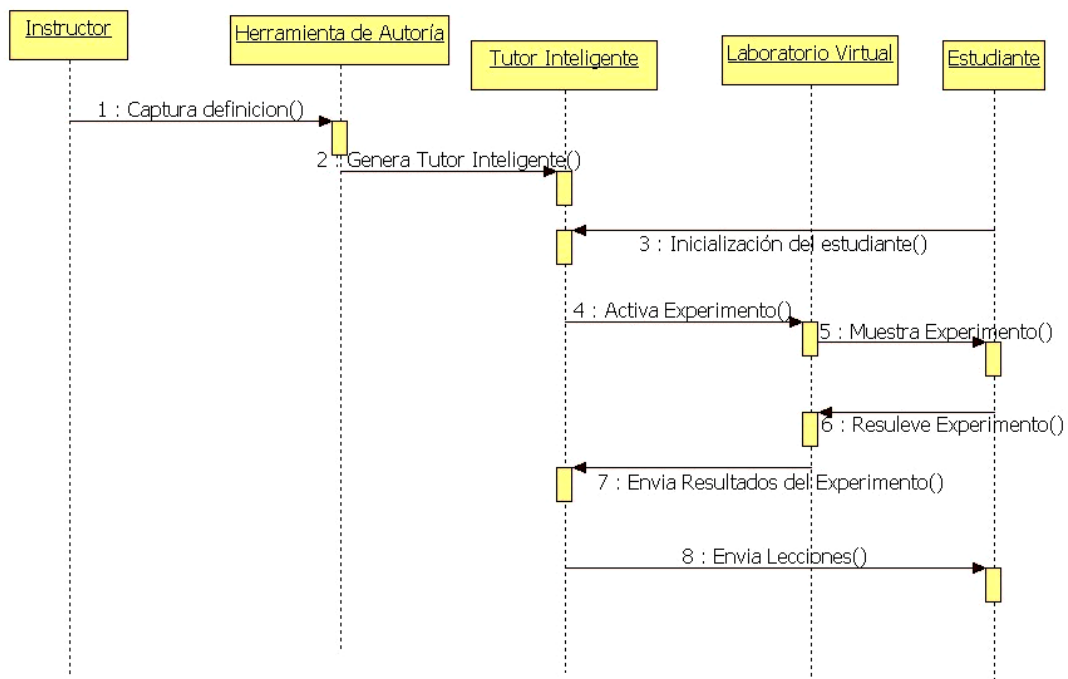


Figura 5-3 Secuencia de interacciones entre los actores que participan en el proceso de un STI.

La parte principal del STI generado por la herramienta es una red bayesiana definida a partir del modelo de MPRs. Dicha red se genera mediante la captura de un temario jerárquico conteniendo temas, sub-temas y conceptos, tomando en cuenta las aportaciones de cada nivel hacia su nivel superior. Este aspecto se detalla en la sección 5.3.

Como parte de la herramienta de autoría se desarrollaron módulos que permiten diseñar cada parte de los componentes de un STI, de acuerdo al modelo propuesto por Murray (1999) la figura 5-4 muestra el esquema general de trabajo entre el laboratorio virtual, el STI y el LV. La herramienta contiene los siguientes módulos:

Módulo Experto o base de conocimiento: este módulo almacena y estructura la información que se va a enseñar: lecciones, experimentos, valores esperados, variables a evaluar, mecanismos de evaluación y el mapa jerárquico de todo el temario.

Módulo Estudiante: modela al estudiante mediante una red bayesiana dinámica que incluye tablas de probabilidades. Esto representa el estado cognitivo del estudiante.

Módulo Tutor: Almacena las estrategias pedagógicas, para proporcionar lecciones al estudiante o seleccionar nuevos experimentos a realizar. Este se implementa en la herramienta mediante reglas de producción.

Interfaz: Es la parte del tutor inteligente a la que tiene acceso el estudiante. Está parte muestra las lecciones y se enlaza con el laboratorio para realizar los experimentos o actividades.

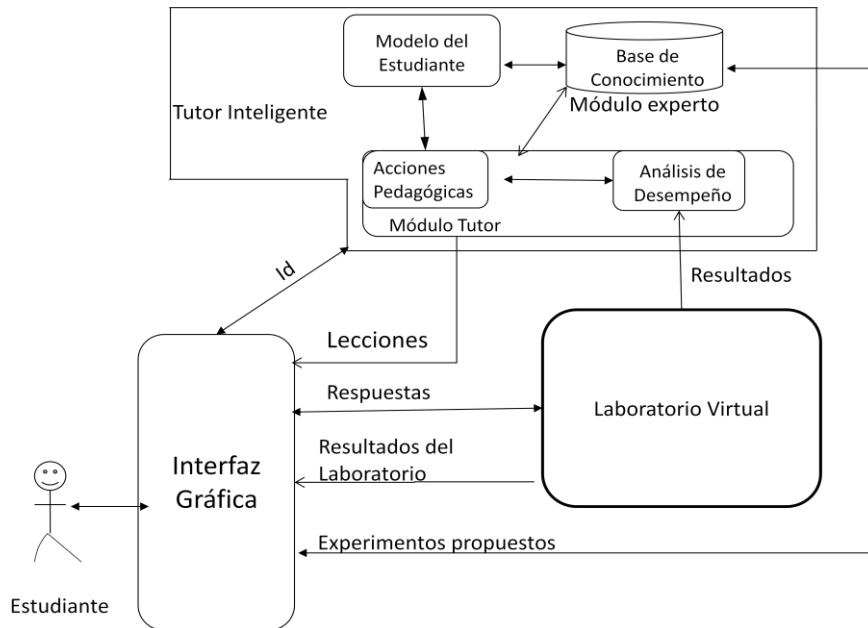


Figura 5-4 Esquema general de trabajo. La parte correspondiente al STI generado por HEDEA está en el recuadro titulado tutor inteligente donde se distinguen los componentes, mientras que la parte inferior hace referencia a la interfaz entre el tutor y el laboratorio virtual.

5.2 Descripción de los módulos

En esta sección se describe cada uno de los módulos principales de la herramienta, las estrategias utilizadas para su desarrollo y la funcionalidad que brindan como herramienta de autoría.

5.2.1 Interfaz con el Laboratorio Virtual

El laboratorio virtual es un ambiente donde se realizan experimentos a través de un simulador (se asume que el instructor cuenta con dicho ambiente). Es en ese sistema donde el estudiante resolverá los experimentos que el docente proponga. Para enlazar el laboratorio virtual y el sistema tutor inteligente, es necesario definir un conjunto de variables que se monitorean

al realizar un experimento y su valor se envía al STI. Asimismo es necesario que el docente tenga conocimiento sobre los nombres de las variables que pueden ser monitoreadas en el laboratorio virtual. Estas variables serán el enlace principal de comunicación entre el laboratorio virtual y el tutor Inteligente.

La Figura 5-5 muestra la pantalla que captura los datos necesarios para configurar un laboratorio virtual utilizando HEDEA. Los archivos de configuración requieren que se identifique el laboratorio virtual con un nombre, el cual generará un archivo con dicho nombre en el cual se almacenaran los datos del laboratorio, el nombre del laboratorio se utiliza para configurar todos los archivos requeridos en el STI. El docente debe tener definidos los experimentos que puede desarrollar un estudiante en el laboratorio. La información a capturar de cada experimento incluye: nombre, descripción, nivel, número de variables, nombre de las variables y valor esperado de cada variable.

Configuración de Laboratorio

Capture ó Seleccione Laboratorio

Nombre:

Experimento Categoría Variables a evaluar

| Actividad | Descripcion | Categoría | Numero de Variables |
|-----------|-------------|-----------|---------------------|
|-----------|-------------|-----------|---------------------|

| Variables | Valor Esperado |
|-----------|----------------|
|-----------|----------------|

Figura 5-5 Pantalla de captura de un Laboratorio Virtual. Esta pantalla permite capturar las características para configurar un nuevo laboratorio virtual o abrir un laboratorio virtual existente para el que se quiere construir el STI.

5.2.2 Base de conocimientos

La base de conocimientos define el contenido temático de los conocimientos necesarios para el desarrollo de los experimentos en el LV y es la base para generar el modelo del estudiante basado en MPRs.

El primer paso para generar la base de conocimiento es capturar el temario del curso, con el cual se tendrá un esqueleto del mapa conceptual y sus relaciones; la figura 5-6 muestra un ejemplo de este tipo de mapas conceptuales. En ese mismo lugar se captura el porcentaje de conocimiento que aporta cada elemento a su nodo padre; utilizando esta estrategia se evita que el docente maneje los conceptos de redes bayesianas y probabilidades. Además la pantalla permite asociar la captura de los contenidos del curso con las lecciones que contengan lo que se tiene que enseñar de cada lección, ver Figura 5-7. La base de conocimiento es representada en forma jerárquica con base a los temas, subtemas y conceptos asociados a las lecciones que se debe enseñar para cada nivel.

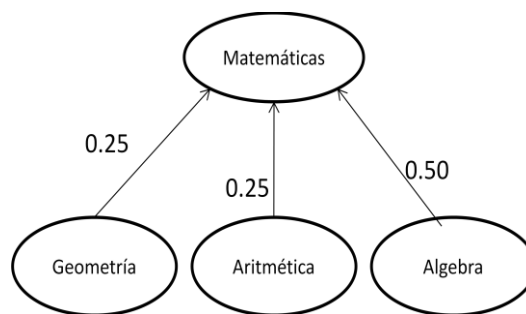


Figura 5-6 Ejemplo de mapa conceptual capturado con pesos. Cada arco indica el porcentaje de conocimiento que aporta para su padre

En la figura 5-6 se muestra un ejemplo del mapa conceptual que el docente tiene que capturar y una estructura de pesos donde la suma de los pesos es igual a 1.

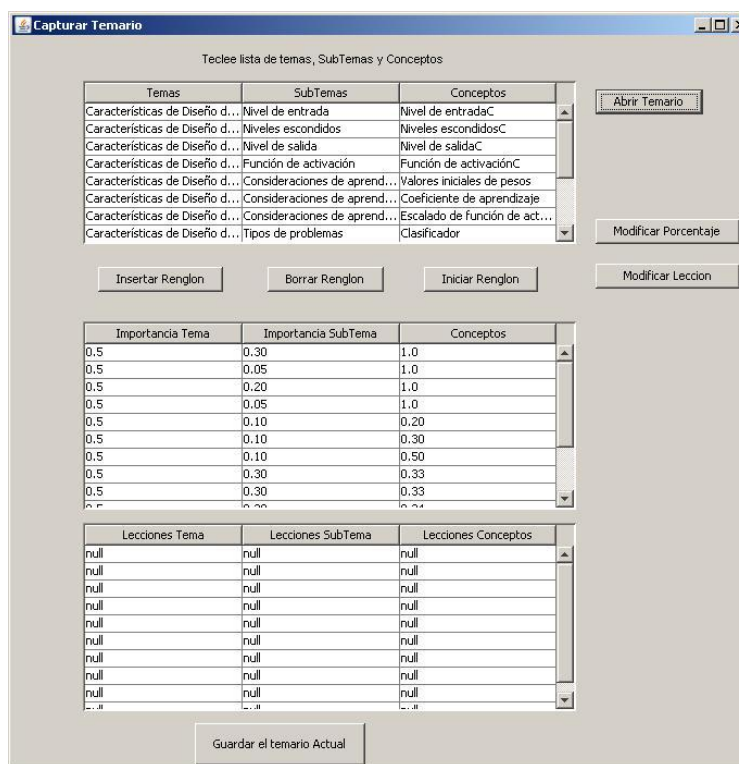


Figura 5-7 Captura de Temario. La ilustración muestra la pantalla para capturar la lista de temas, subtemas y conceptos, así como los valores de importancia y las lecciones relacionadas a cada nodo.

Los valores de importancia son asignados durante la captura de cada uno de los temas, subtemas y conceptos, así el temario capturado cuenta con toda la información necesaria para crear una representación grafica.

Después de haber capturado el temario, se cuenta con un modelo conceptual de tres niveles jerárquicos (Temas, Sub-temas y Conceptos), el cual es el esqueleto base de la red bayesiana que se utilizará. En la figura 5-8 se muestra una parte de dicho modelo para el experimento de redes neuronales.

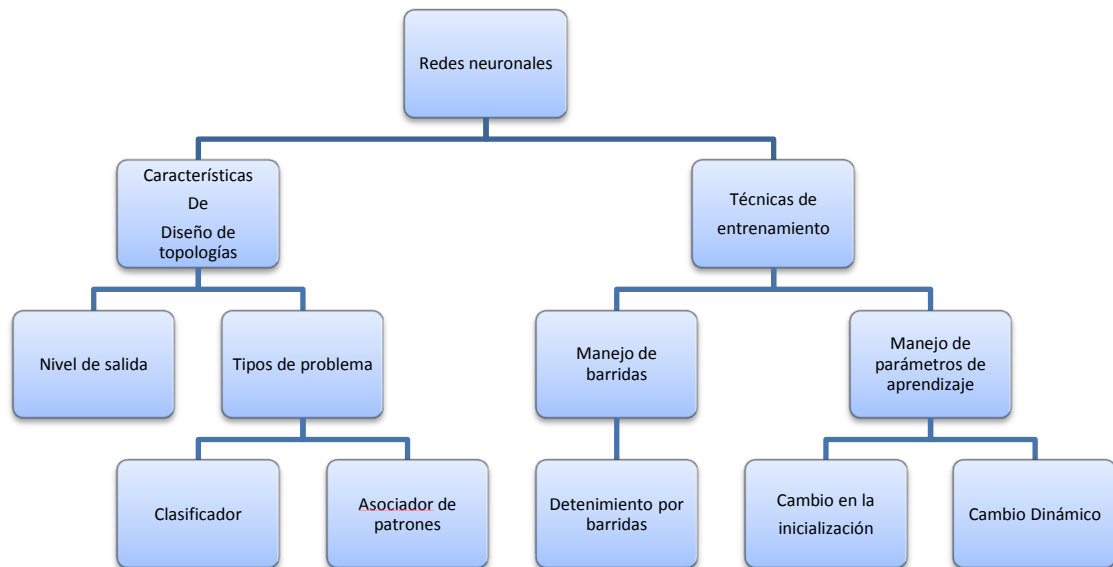


Figura 5-8 Ejemplo de mapa jerárquico en el ámbito de redes neuronales

Ya que se tiene el modelo se asocian los conceptos a las variables de los experimentos que producen el aprendizaje de los temas a enseñar. Para realizar esta tarea, se despliega al profesor una pantalla que permite asociar las variables de cada problema a la rama del temario que corresponden; es decir a cada concepto a aprender del esquema de la figura 5-8 se asocian una o varias variables de experimentos. Esto se hace mediante la pantalla que se ve en la Figura 5-9. Con esta asociación se tienen todos los datos necesarios para definir la red bayesiana completa; la cual es requerida como base del modelo del estudiante.



Figura 5-9 Enlace del laboratorio virtual con el temario. Esta pantalla permite enlazar las variables de cada experimento con la sección del temario correspondiente. Con esto se complementa la información necesaria para la generación del modelo gráfico probabilista a utilizar en el STI.

5.2.3 Modelo del Estudiante

El modelo del estudiante es una de las partes esenciales en el diseño de sistemas tutores inteligentes, debido a que dicho modelo representa el nivel de conocimiento del estudiante en un momento determinado. Para realizar esta tarea es necesario construir este modelo para cada estudiante, a partir de la base de conocimiento definida previamente, este proceso se explica en la sección 5.3.

El modelo del estudiante generado en forma de una red bayesiana es almacenado en archivos de texto con el formato que requiere ELVIRA, para realizar posteriormente el proceso de inferencia. La Figura 5-10 muestra la pantalla donde se captura las relaciones entre temas, subtemas, conceptos y variables cuyo peso varía para cada concepto; la herramienta proporciona

una distribución uniforme de probabilidad para las variables, la cual puede ser modificada por el usuario. Con los datos capturados se define la red bayesiana, ver sección 5.3.

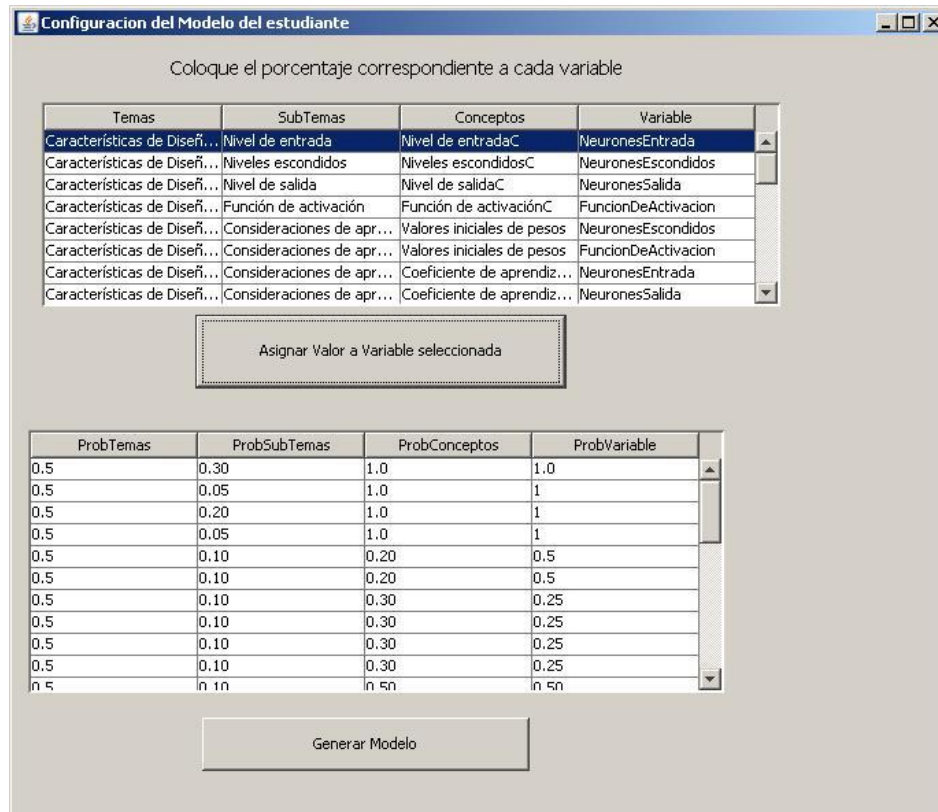


Figura 5-10 Captura del modelo del estudiante. En esta pantalla se permite la captura a los valores de importancia de las variables de los experimentos para poder adquirir algún conocimiento. Estos valores son necesarios para inicializar las tablas de probabilidad del modelo.

5.3 Generación del Modelo del estudiante

La generación del modelo del estudiante basado en una red bayesiana es uno de los puntos clave para el desarrollo de la herramienta de autoría. Primero fue necesario generar la información que se requiere para construir el modelo:

- Nombre de los Nodos
- Enlace entre los nodos
- Tablas de Probabilidades Condicionales
- Valores Iniciales de las redes bayesianas
- Método de inferencia a utilizar

Enseguida se describe como se obtienen estos campos:

Nombre de los Nodos: El nombre de los nodos se obtiene a partir de cada uno de los niveles de la jerarquía del temario del curso: temas, subtemas, conceptos y variables, los cuales son capturados durante la generación del modelo del estudiante y la base del conocimiento; hay tantos nodos como componentes del temario.

Enlace entre los nodos: Indica la relaciones jerárquicas entre nodos padres y nodos hijos. Estas relaciones no son capturadas directamente por el usuario, sino que se generan mediante una búsqueda de dependencias en la jerarquía de temario, la cual fue capturada en el módulo de generación de base de conocimiento. La búsqueda consiste en analizar la tabla e ir enlazando los nodos que compartan los mismos padres en cada uno de los niveles de temas, subtemas y conceptos.

La tabla de probabilidades condicionales: se genera mediante algoritmos que permiten calcular todas las combinaciones de los valores de entrada de cada nodo, haciendo que la suma sea igual a 1. Dichas probabilidades se calculan en función del “peso” de cada elemento (variable,

concepto, sub-tema, tema) con respecto a su nivel superior, ver ejemplo más adelante.

Valores Iniciales: corresponden al valor inicial de cada variable en el modelo. Indican que el estudiante no ha realizado ningún experimento, por lo que los valores son lo más bajos permitidos. Estos valores nunca son propagados, ya que la primera propagación se ejecuta cuando ya se realizó algún experimento.

Método de inferencia a utilizar: se utilizará la función general de propagación con el método de inferencia de eliminación de variable (*Variable Elimination*[VE](Dechter 1996)). La explicación de este método se encuentra en el anexo E.

Generación de la Red Bayesiana

La red bayesiana que representa al modelo del estudiante, tiene una estructura jerárquica análoga a cada elemento en el temario. Hay un nodo para cada tema, sub-tema, concepto y variable; y ligas que corresponden a las relaciones entre ellas.

Después de la construcción de una red bayesiana (en formato Elvira), se realizan inferencias basadas en los valores de los nodos hojas (correspondiente a las variables de los experimentos), los cuales se inicializan con las calificaciones obtenidas por el estudiante después de haber resuelto un experimento. Estas calificaciones son calculadas por el tutor inteligente basándose en los valores esperados proporcionados por el docente durante la configuración del laboratorio virtual.

En el modelo se considera la experiencia previa del estudiante (experimento anterior) y los resultados del experimento actual, para

actualizar el estado del estudiante. Para esto, el modelo se transforma en una red bayesiana dinámica.

Para generar la red dinámica se requiere tener un modelo base generado previamente y remplazar los valores de los nodos anteriores por los valores obtenidos de la inferencia anterior. Este es el núcleo del proceso de inferencia que se utiliza en este trabajo, ya que permite considerar los resultados previos para tomar una decisión sobre el estado de conocimiento actual del estudiante.

5.3.1 Ejemplo de generación de una red bayesiana desde el temario.

A continuación se ilustra el proceso de generar una red bayesiana para una parte del STI de redes neuronales que se presenta más adelante.

- 1) Se captura el temario (temas, sub-temas, conceptos) del curso, figura 5-11.

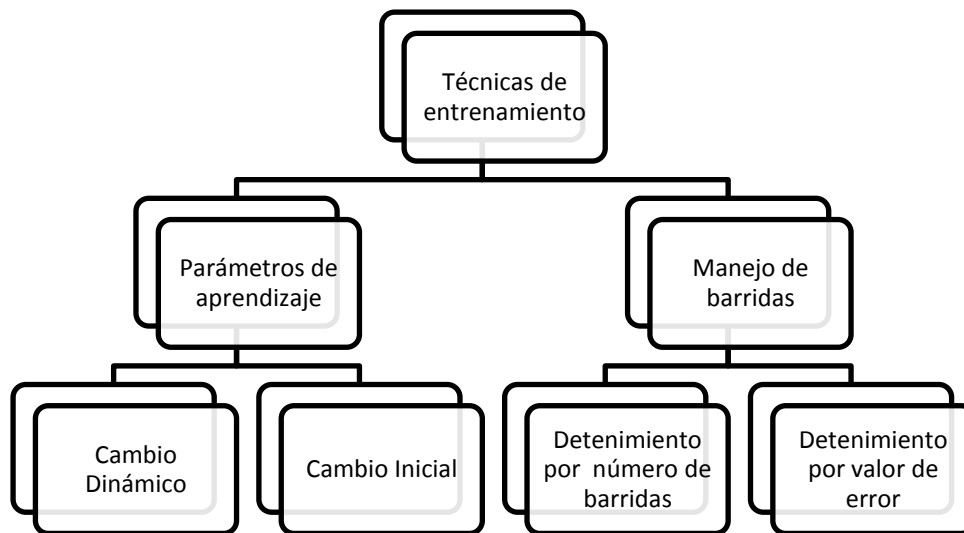


Figura 5-11 Ejemplo de temario capturado para el caso de redes neuronales (parcial).

- 2) Se especifica la “importancia” de cada sub-tema, concepto para el nivel superior, lo que se transforma a probabilidades condicionales ver tabla 5-1.

Tabla 5-1 Ejemplo de probabilidades del Temario de la figura 5-11
En la tabla se muestran los nombres de los nodos y su relación de temas y subtemas. Vemos que los valores de las variables suman 1.

| Probabilidades Técnicas de entrenamiento | |
|---|-----|
| Parámetros de aprendizaje | 0.7 |
| Manejo de barridas | 0.3 |
| Probabilidades Parámetros de aprendizaje | |
| Cambio Dinámico | 0.7 |
| Cambio inicial | 0.3 |
| Probabilidades manejo de barridas | |
| Paro por numero de barridas | 0.5 |
| Paro por valor de error | 0.5 |



Figura 5-12 Ejemplo de grafo generado a partir del temario capturado.

- 3) Se genera la estructura de la red bayesiana:
- a) Se generan los nodos uno por cada tema, subtema y conceptos, sin repetir nombres.

b) Partiendo de los conceptos se empiezan a generar enlaces a sus subtemas relacionados.

c) Se generan enlaces de los subtemas a los temas relacionados.

El grafo mostrado en la Figura 5-12 muestra la estructura básica de la red bayesiana que corresponde al temario desarrollada para el ejemplo.

4) Se generan las tablas de probabilidad condicional para cada variable de la red bayesiana:

a) Se generan todas las combinaciones posibles de los valores de los padres hacia los hijos, que son “sabe” y “no sabe” de los nodos involucrados, con lo cual se generará una tabla como la de la Figura 5-13 (aún sin valores).

b) Para el llenado de valor se suman el valor de todos los nodos cuyo valor sea “sabe”. En la figura 5-13 tenemos que los valores de “Parámetros de entrenamiento” es 0.7 y “manejo de barridas” 0.3, por lo tanto cuando se dice que sabe en ambos tiene un valor el nodo hijo de 1.0, Si sólo uno de los valores (0.3 y 0.7) estuviera presente el valor de sabe del nodo hijo será el valor del padre con valor de sabe, y si ambos estuvieran en no sabe, el valor se inicializa con valores determinados de 0.0 a sabe y 1 a no sabe.

| Parámetros ... | Sabe | Sabe | No Sabe | No Sabe |
|----------------|------|---------|---------|---------|
| Manejo de b... | Sabe | no Sabe | Sabe | no Sabe |
| Sabe | 1.0 | 0.7 | 0.3 | 0.0 |
| No Sabe | 0.0 | 0.3 | 0.7 | 1.0 |

Figura 5-13 Ejemplo de una tabla de Probabilidad Condicional para la variable técnicas de entrenamiento. Los valores de la tabla indican que si un estudiante contesta de manera correcta a las dos variables la probabilidad que domine ese concepto es del 100%. Si solo contesta una de las variables la probabilidad de dominar dicho concepto estará dada por la importancia de la variable contestada de manera correcta.

Con esta estrategia es posible generar, todas las tablas de probabilidad condicional mediante la captura de sólo la parte de la aportación de conocimiento del tema, subtema y concepto; permitiendo así que el docente no requiera saber sobre redes bayesianas y probabilidades condicionales.

5) Generación del modelo dinámico:

De manera similar se genera la red bayesiana dinámica que tome en cuenta el aprendizaje previo. Dicha red requiere generar dos nodos extras por cada nodo en el mapa conceptual. Uno para el valor anterior y un nodo temporal para calcular el valor siguiente. Un ejemplo de esto se muestra en la Figura 5-14. Esta red permite que el estudiante avance mediante la acumulación de diferentes experimentos resueltos en el laboratorio actualizando su estado de conocimiento según los resultados.

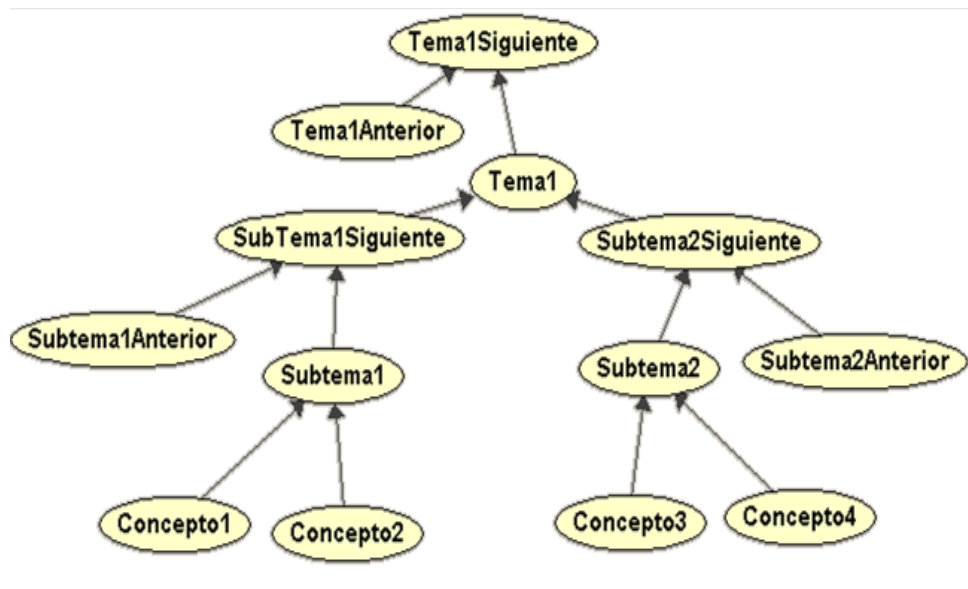


Figura 5-14 Red bayesiana dinámica que considera el conocimiento previo, basada en el modelo de la figura 5.12

Resumiendo, con base al proceso descrito anteriormente se genera una red bayesiana dinámica que representa el modelo del estudiante; la cual tiene las siguientes características:

Red Base

- Los nodos tienen valores de “sabe y no sabe”
- Los nodos hojas están relacionados con las variables de los experimentos y reciben su valor en función de una calificación entre 0 y 100.
- Las tablas de probabilidad de la red base son llenadas durante la definición del temario por el docente, y está basada en el porcentaje del contenido temático que brinde al nodo padre; es decir el valor pesado de conocimiento del nodo. Ver sección 5.2.3

La figura 5-15 muestra un ejemplo de cómo se distribuye la red base donde cada nodo cuenta con una TPC en función del número de valores de entrada.



Figura 5-15 Ejemplo de cómo se mira la red base (parcial solo 2 niveles)

Red de transición.

Tiene los mismos componentes que la red base y se le añade lo siguiente:

- Un nodo “valor anterior” que se inicializa con valores de experimentos anteriores del mismo estudiante

- Un nodo “valor siguiente” que permite fusionar el valor del historial del estudiante con los resultados del experimento actual para el cálculo de nuevos valores.

La figura 5-16 muestra la red de transición para el ejemplo de la figura 5-15. Los nodos con terminación siguiente, toman en cuenta los valores anteriores obtenidos por el estudiante en otros experimentos y el valor siguiente será almacenado en el modelo para ser tomado en cuenta en el próximo experimento a resolver. Las TPC de los nodos con terminación siguiente está determinada por la importancia que se da a resultados anteriores y actuales.

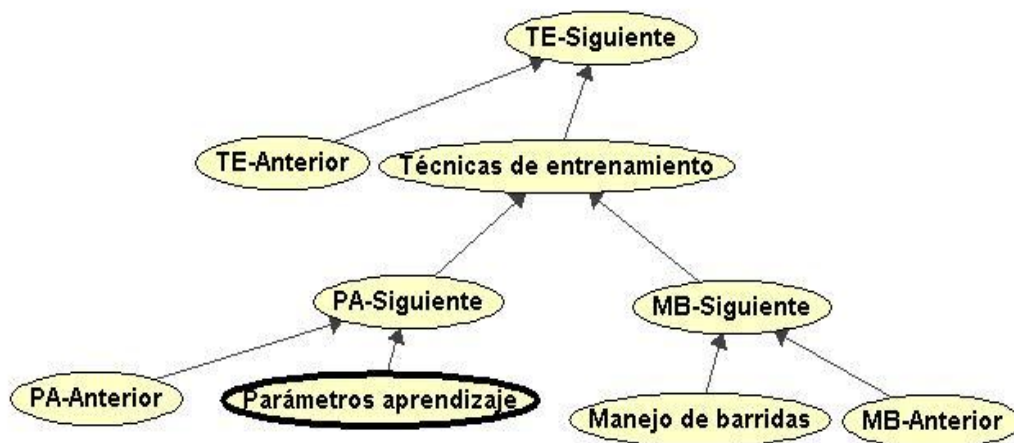


Figura 5-16 Ejemplo de red de transición tomando la figura 5-15 como base

Enseguida se presenta el algoritmo que se utiliza para llevar a cabo el proceso de inferencia de la red bayesiana que modela al estudiante.

5.4 Proceso de inferencia

Se tiene una red bayesiana generada de la distribución de temas, subtemas, conceptos y variables. Dicha red está en espera de los resultados de algún experimento; ya que la red es utilizada para actualizar el modelo del estudiante después de cada experimento.

1. Primer experimento.

- 1.1. Se inicializa las variables con los valores obtenidos del resultado de del experimento en el laboratorio.
- 1.2. Se propaga la evidencia utilizando el algoritmo de eliminación de variable.
- 1.3. Se obtienen los valores de cada nodo superior
- 1.4. Se almacenan los resultados.

2. Extender Modelo a red bayesiana dinámica

- 2.1. Se coloca la red base con sus valores iniciales
- 2.2. A cada nivel no hoja de la red se le añaden dos nodos de valor anterior y valor siguiente
- 2.3. Se inicializa los nodos valor anterior de cada nivel con los valores acumulados
- 2.4. Los nodos valor siguiente son los que se almacenarán en los próximos experimentos

3. Siguiete experimento.

- 3.1. Recibe resultados de experimento
 - 3.1.1. Se inicializan las variables con los valores obtenidos de las respuestas del laboratorio y los datos anteriores.
 - 3.1.2. Se propaga la evidencia.
 - 3.1.3. Se obtienen los valores de cada nodo superior
 - 3.1.4. Se almacenan los resultados

4. El paso 3 se repiten para cada experimento subsecuente.

El algoritmo descrito anteriormente es la secuencia de pasos que se utilizará cada vez que el STI reciba información sobre un nuevo experimento resuelto por el estudiante, donde actualiza el modelo y genera una nueva configuración del modelo del estudiante introduciendo los valores de las inferencias previas.

De esta manera, después de que un estudiante realiza un experimento, se tiene un el modelo la probabilidad de que el estudiante maneje adecuadamente cada aspecto del temario a diferentes niveles. Con base en el modelo, el tutor decide que es necesario que el estudiante estudie para fortalecer su conocimiento, así como que nuevo experimento le recomienda realizar. Ese se realiza mediante las reglas almacenadas en el módulo tutor.

5.5 Módulo Tutor

El módulo tutor permite definir valores para cada nivel del estudiante, Principiantes, intermedio y avanzado. Con los cuales se forman reglas de la forma “Si<resultado> << entonces [Lección]”. Con estas reglas se permite asignar lecciones si el estudiante no ha alcanzado el nivel de conocimiento necesario para cambiar de categoría.

Las reglas están formadas de dos partes, para cada categoría se propone un valor de conocimiento necesario para ser promovido o cambiar de categoría. Si se está enseñando un concepto por ejemplo se puede poner como regla que ese concepto se obtenga un conocimiento del 80% para principiantes y 90 % para avanzados; Así, cuando un estudiante realiza el experimento la regla compara si la categoría es principiante y tiene más del

80% cambia la lección a otro concepto. Y si es avanzado y tiene menos del 90% repite una lección similar para que mejore el conocimiento de ese concepto.

Si el estudiante alcanza el nivel de conocimiento requerido, entonces el modelo sigue buscando en la red bayesiana si falta algún valor por cumplir. Si el valor total es mayor que el requerido de su nivel, el estudiante se cambia al siguiente nivel. Si el valor total del curso es menor entonces se asigna una nueva lección y un nuevo experimento del mismo nivel. A continuación se describen los procesos de asignación de experimentos y lecciones.

5.5.1 Proceso de asignación y ejecución de experimentos.

En esta sección se mostrará el esquema de trabajo sobre el STI, para la asignación de experimentos a un estudiante desde dos perspectivas; un estudiante comienza un experimento nuevo y un estudiante continúa con el siguiente experimento. La Figura 5-17 muestra el esquema general para comenzar a trabajar con el STI dentro de un laboratorio virtual, además en la figura se pueden identificar las acciones que tiene que realizar cada componente

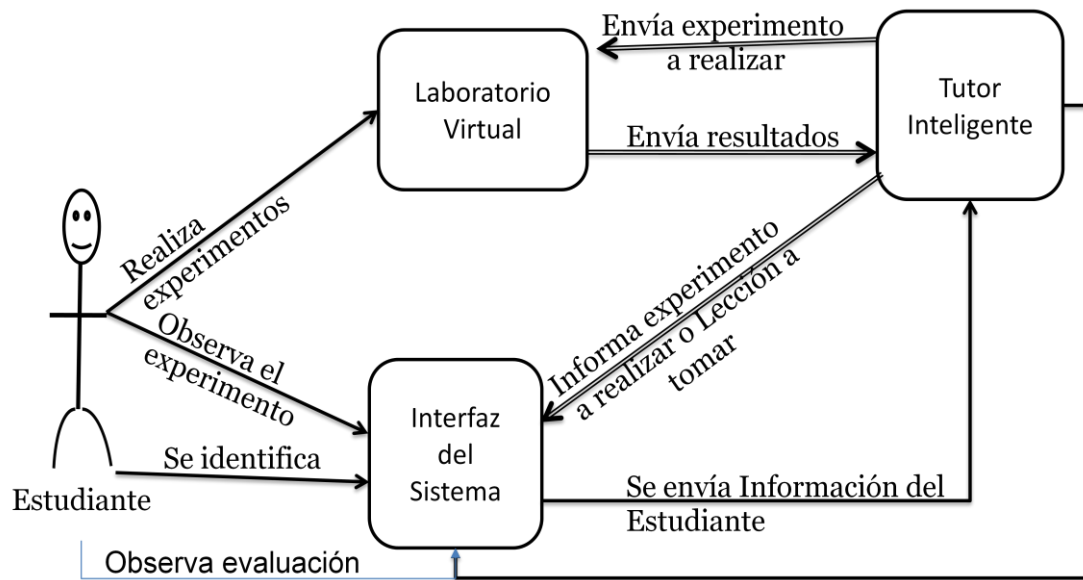


Figura 5-17 Caso Genérico para comenzar experimento. Este caso muestra las acciones para un experimento nuevo, se ve al estudiante interactuando tanto con el sistema como con el laboratorio virtual.

Mecanismo de Asignación de experimentos:

1. El tutor Identifica al estudiante.
2. Identifica la categoría, basado en los valores de experimentos anteriores o “Principiante” si es el primer experimento.
3. Selecciona un experimento del módulo dominio.
4. Se envía la información de que experimento se debe realizar.
5. El tutor queda en espera de los resultados del experimento.

Después de que el STI ha realizado la asignación anterior, el estudiante se encuentra trabajando directamente en el laboratorio virtual, respondiendo el experimento que fue propuesto por el tutor. Al terminar dicho experimento el estudiante avisa al STI para que lo califique y asigne un nuevo experimento o termine su sesión. Así que el STI continúa con el proceso:

6. El STI Identifica al estudiante y el experimento que realizó.

7. El STI evalúa a través del módulo tutor de el mismo los resultados del experimento en el laboratorio.
8. Se proporciona retroalimentación necesaria, nuevas lecciones o se repite la lección.
9. Se asigna un nuevo experimento basado en el modelo del estudiante.
10. Se envía la información de que experimento se debe realizar.
11. El tutor queda en espera de los resultados de los experimentos.

Así el tutor siempre está en espera de que el estudiante realice un nuevo experimento o termine la sesión.

5.6 Utilización del STI Generado

Para utilizar al STI generado se deberá registrar un estudiante. El registro de un estudiante genera una copia del modelo del estudiante y de los archivos de configuración. Estos archivos se almacenan en una carpeta con el nombre del estudiante. Cada vez que el estudiante resuelve un experimento se actualiza su modelo y se modifican los valores del estado de conocimiento.

5.7 Descripción del método desarrollado

Para desarrollar un sistema tutor inteligente utilizando la herramienta propuesta en esta investigación, se requiere de la realización de una serie de pasos previos y posteriores al sólo hecho de manejar la herramienta de autoría. La Figura 5-18 muestra el esquema general de trabajo para la herramienta. En la primera parte del esquema se describen las actividades que tiene que hacer el docente antes de empezar a trabajar con la herramienta de autoría, para mayor información ver apéndices B. Después

de que el docente realizó el trabajo de diseñar experimentos, coloca el temario en forma de mapa conceptual y realiza el pesado correspondiente a cada nodo, se procede a trabajar dentro de la herramienta de autoría. La segunda parte del diagrama de la Figura 5-18 ilustra los pasos que se tienen que realizar para poder desarrollar un STI.

Para poder desarrollar un STI el docente tiene que capturar en la herramienta de autoría la información correspondiente al mapa conceptual, sus pesos, las variables del laboratorio, así como asignarle un nombre al laboratorio con el que trabajara el STI. Los pasos a seguir dentro de la herramienta son configurar un laboratorio, después capturar un temario. Ya que se tiene el temario capturado se procede a generar la base de conocimiento en la cual se almacenaran las lecciones que se van a enseñar. Después de generar la base de conocimiento se genera el modelo del estudiante, donde se genera una red bayesiana dinámica que será utilizada para inferir el conocimiento que un estudiante tiene. Por último se capturan las reglas pedagógicas del módulo tutor, que en este caso son valores para cada una de las categorías. Al final se genera el STI que utilizara los módulos anteriores.

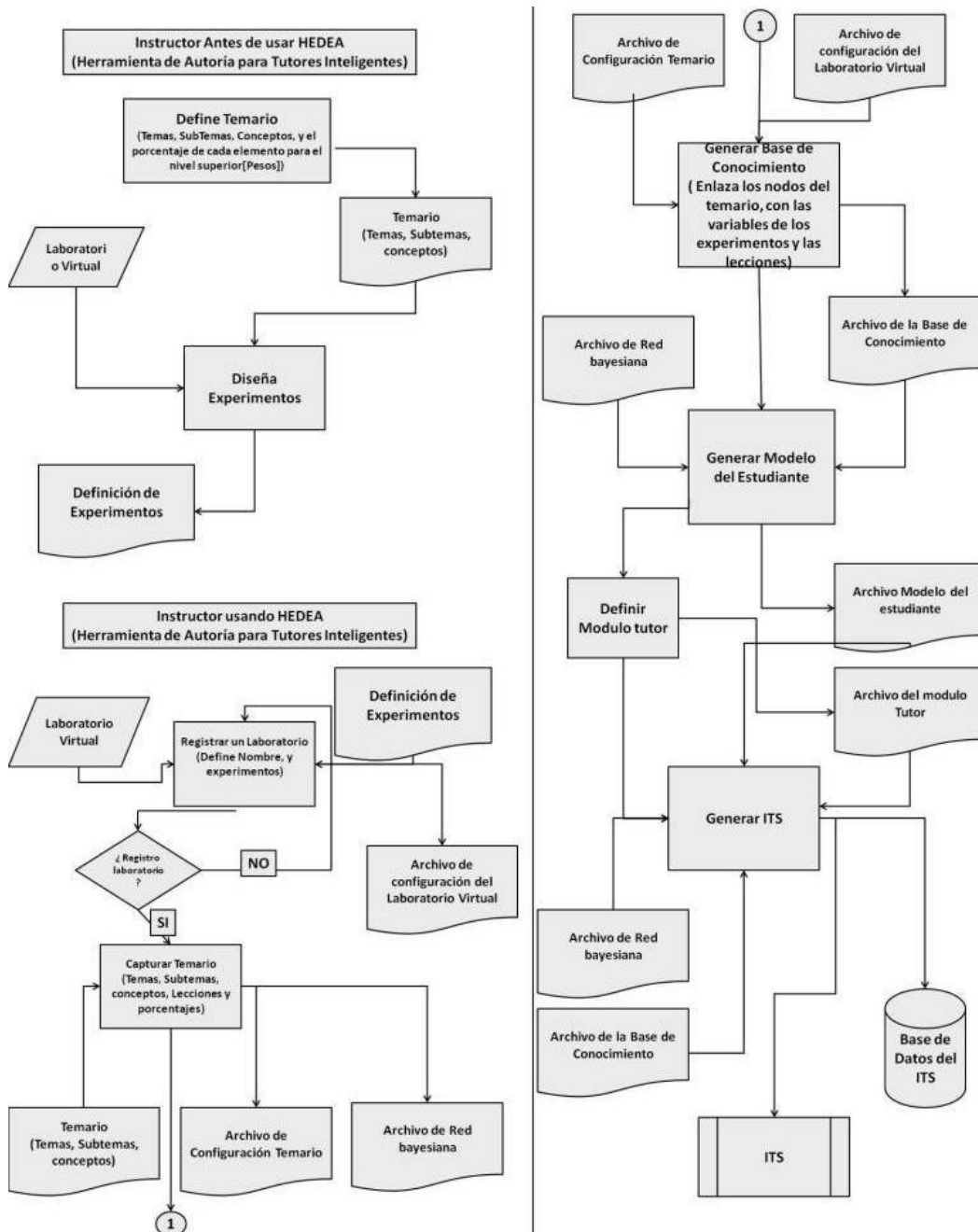


Figura 5-18 Esquema general del proceso de generación de un STI. En el esquema se distinguen dos partes el trabajo de docente antes de utilizar la herramienta y el trabajo del docente ya utilizando la herramienta.

5.8 Resumen del capítulo

En este capítulo se describió la herramienta de autoría desarrollada, se analizaron las estrategias utilizadas para la generación automática del modelo del estudiante y la construcción del tutor. Además, se analizó el funcionamiento del STI que genera la herramienta, así como los procedimientos para la actualización dinámica del modelo del estudiante. Por último se realizó una explicación general del esquema de trabajo de la herramienta denotando las actividades que tiene que hacer el docente antes de empezar a construir un STI. En el siguiente capítulo hablaremos de las técnicas de evaluación y veremos los resultados obtenidos con la herramienta desarrollada.

Capítulo 6

Evaluación de la Herramienta

La evaluación de la herramienta de autoría se puede realizar desde tres diferentes perspectivas:

1. Usabilidad: Evaluación de la capacidad de la herramienta para definir y generar de manera automática a un tutor inteligente. Esta evaluación es hecha por los instructores, usuarios de la herramienta.

2. Efectividad del modelo del estudiante: la evaluación del modelo de inferencias del MPR del módulo estudiante utilizado para representar el conocimiento del estudiante. Esto permite verificar si los valores de probabilidad obtenidos son “razonables” de acuerdo al instructor.

3. Efectividad del Tutor Inteligente: Evaluación de la funcionalidad del tutor inteligente generado por la herramienta, con respecto a su correcta representación del conocimiento adquirido por el estudiante y su impacto en el proceso de aprendizaje. Esto implica evaluar el STI generado en un grupo de estudiantes; analizando el impacto del tutor en el aprendizaje.

Enseguida se presentan los resultados de las evaluaciones.

Para poder realizar esta evaluación se generó un caso de estudio completo de un laboratorio y tutor de redes neuronales.

6.1 Caso de estudio

Como parte de las evaluaciones se diseñó un caso de estudio basado en redes neuronales para un pequeño laboratorio virtual programado en MatLab y que permite hacer experimentos con redes de retropropagación. El temario a enseñar se compone de 2 temas, 8 sub-temas y 14 conceptos. El número de variables a evaluar es de 6 variables distintas lo que generaría un modelo base con 30 nodos, más un nodo que proporciona la evaluación global del conocimiento del estudiante.

El apéndice B muestra el temario diseñado y los experimentos que fueron utilizados en este caso de estudio.

6.2 Evaluación de usabilidad de la herramienta

La evaluación de funcionalidad en la herramienta construida se realizó enfocada a la usabilidad. De acuerdo con Ferré y colaboradores (2001), usabilidad puede ser evaluada por 5 atributos: aprendizaje, eficiencia, retención del usuario, tasa de error y satisfacción. Debido a que la herramienta aun es un prototipo, se construyó un cuestionario enfocado a evaluar los aspectos de aprendizaje (facilidad con la que un usuario puede aprender a utilizar la herramienta de autoría), y satisfacción (opinión subjetiva del usuario sobre todo el entorno utilizado). La Tabla 6-1 muestra el cuestionario diseñado para la evaluación en el cual cada pregunta tiene asignado un porcentaje a la izquierda. El valor más alto que se puede obtener del cuestionario es 3.0.

Tabla 6-1 Cuestionario de usabilidad de la herramienta de autoría

| Número de pregunta | Descripción de la pregunta | Importancia | Atributo evaluado |
|--------------------|--|-------------|-------------------|
| 1 | Construir un temario con conceptos pesados es una tarea fácil. | 0.2 | Aprendizaje |
| 2 | He aprendido a utilizar HEDEA muy rápidamente | 0.2 | Aprendizaje |
| 3 | Creo que el STI creado por HEDEA hará que mis alumnos aprendan más rápidamente | 0.2 | Satisfacción |
| 4 | Fue fácil seguir las instrucciones de la herramienta. | 0.1 | Aprendizaje |
| 5 | Los pasos para registrar mis experimentos en HEDEA fue fácil. | 0.1 | Satisfacción |
| 6 | Me gustaría recomendar HEDEA a mis colegas | 0.1 | Satisfacción |
| 7 | HEDEA me permite corregir o cambiar mis entradas fácilmente. | 0.1 | Satisfacción |

Dos usuarios voluntarios no experto en desarrollo de STI's realizaron el proceso de creación de STI y evaluaron la usabilidad de la herramienta mediante el cuestionario de usabilidad, asignando las calificaciones que se muestran en la Tabla 6-2. La calificación global obtenida por la herramienta fue de 2.3 la cual podemos considerar buena y se identificaron algunas posibles mejoras. Además de estas evaluaciones, implementamos un caso de estudio y la evaluación de usabilidad obtiene resultados similares. A continuación se describen los aspectos resultantes de esta evaluación identificando los puntos positivos y negativos.

Aspectos positivos:

- La herramienta permite construir el modelo sin necesidad de conocer los conceptos de inteligencia artificial.

- El uso del temario con conceptos pesados permite que se pueda adaptar la herramienta para diversas áreas.
- La captura de los experimentos a realizar en el laboratorio es sencilla y permite enlazar fácilmente al STI con el LV.

Aspectos por mejorar :

- Falta mejorar la tolerancia a fallas de captura de datos haciendo más amigable la corrección de errores. Esto porque la corrección de errores durante la captura fue considerada como una debilidad.
- Deben mejorarse las instrucciones de uso de la herramienta que no son suficientemente claras.

Tabla 6-2 Resultado del cuestionario de usabilidad aplicado

| Número de pregunta | Descripción de la pregunta | Importancia | Atributo evaluado | Respuesta obtenida | Subtotal |
|---|--|-------------|-------------------|--------------------|------------|
| 1 | Construir un temario con conceptos pesados es una tarea fácil. | 0.2 | Aprendizaje | 3 | 0.6 |
| 2 | He aprendido a utilizar HEDEA muy rápidamente | 0.2 | Aprendizaje | 3 | 0.6 |
| 3 | Creo que el STI creado por HEDEA hará que mis alumnos aprendan más rápidamente | 0.2 | Satisfacción | 2 | 0.4 |
| 4 | Fue fácil seguir las instrucciones de la herramienta. | 0.1 | Aprendizaje | 1 | 0.1 |
| 5 | Registrar mis experimentos en HEDEA fue fácil. | 0.1 | Satisfacción | 2 | 0.2 |
| 6 | Me gustaría recomendar HEDEA a mis colegas | 0.1 | Satisfacción | 3 | 0.3 |
| 7 | HEDEA me permite corregir o cambiar mis entradas fácilmente. | 0.1 | Satisfacción | 1 | 0.1 |
| Calificación TOTAL (valor máximo 3): | | | | | 2.3 |

6.3 Evaluación del modelo del estudiante

En esta evaluación se monitorean los resultados que tiene en el nivel de conocimiento un estudiante a partir de las calificaciones obtenidas en cada variable. Estas acciones permiten hacer pruebas sobre el modelo propuesto y analizar si requieren ajustarse los valores de peso para las evidencias anteriores así como los experimentos actuales. Es decir, que los resultados obtenidos dependen tanto de la estructura generada por la herramienta, como de los pesos dados por el instructor. Para estas pruebas se diseñó una interfaz que permite capturar calificaciones y da como resultado el valor en los nodos del temario en el modelo del estudiante (red bayesiana). La Tabla 6-3 muestra los resultados de las variables para un caso; en el cual se tiene ciertas probabilidades iniciales que se actualizan con base en los resultados de un experimento y mediante la inferencia en el modelo.

Tabla 6-3 Resultados de las variables de RNA de un caso básico. La tabla muestra las calificaciones obtenidas para cada variable en escala de 0 a 1. Para este caso se inicializaron las variables del curso con 1, luego de un experimento se guardo el conocimiento y se volvió a propagar nueva evidencia. Podemos ver que la segunda interacción a pesar de tener valores menores tiene una mejor evaluación porque toma en cuenta también el resultado previo. Con estas acciones se permite la actualización dinámica del conocimiento adquirido.

| Variable | Calificación | Variable | Calificación |
|-----------------------|--------------|-----------------------|--------------|
| Neuronas de entrada | 1 | Neuronas de entrada | .8 |
| Neuronas escondidos | 1 | Neuronas escondidos | .9 |
| Neuronas de Salida | 1 | Neuronas de Salida | .95 |
| Función de activación | 1 | Función de activación | 1 |
| Desempeño obtenidos | 1 | Desempeño obtenidos | .6 |
| Total de cambios | 1 | Total de cambios | .98 |
| Evaluación→ | 0.75 | Evaluación2→ | 0.815 |

La tabla 6-3 muestra las dos primeras interacciones de un estudiante con el sistema y los resultados sobre el conocimiento general del curso dadas las respuestas, donde 1 es el valor más alto. Debe notarse que aunque en el segundo resultado no se obtuvo una calificación perfecta el valor sigue aumentando, ya que toma en cuenta los resultados previos.

Un ejemplo de evaluación será llevar lo más bajo posible un valor de algún concepto y desplegar el resultado que se obtiene. La tabla 6-4 muestra la evaluación de un estudiante que no domina un concepto y a pesar que obtiene calificaciones excelentes en los otros conceptos, el conocimiento global del estudiante se mantiene fijo porque necesita adquirir el conocimiento de los otros nodos.

Tabla 6-4 Evaluación de estudiante que comete errores en un concepto. La tabla muestra un ejemplo para el caso de estudio de redes neuronales donde los conceptos que fallan representan el 15% del total del curso. Aquí podemos ver como el modelo del estudiante va actualizándose hasta que se estabiliza.

| Variable | Calificación | Variable | Calificación |
|-----------------------|--------------|--|--------------|
| Neuronas de entrada | .99 | Neuronas de entrada | .99 |
| Neuronas escondidos | .99 | Neuronas escondidos | .99 |
| Neuronas de Salida | .99 | Neuronas de Salida | .99 |
| Función de activación | .99 | Función de activación | .99 |
| Desempeño obtenido | .05 | Desempeño obtenido | .05 |
| Total de cambios | .05 | Total de cambios | .05 |
| Evaluación INICIAL→ | 0.67 | Evaluación después de 10 experimentos→ | 0.848 |

La tabla muestra los resultados de un estudiante que contesta correctamente, excepto los experimentos sobre manejo total de barridas. Los experimentos se repitieron 10 veces llevando a la evaluación total a 0.848. Si vemos en el temario el concepto manejo total de barridas representa un 15% del conocimiento, que debe tener el estudiante, por lo tanto vemos que el

modelo de inferencia refleja los resultados esperados en base al temario capturado.

De acuerdo a los parámetros establecidos en este ejemplo, un estudiante que inicia requiere resolver de manera correcta 4 experimentos consecutivos para alcanzar el valor máximo de conocimiento que es mayor a 0.95. Esta evaluación puede ser hecha por el instructor a través de un software de apoyo que se ha incluido en la herramienta, usando experimentos de prueba, y alimentando directamente las variables con los resultados sin necesidad de ejecutarlos en un laboratorio virtual. El instructor puede evaluar la sensibilidad de cada variable sobre el conocimiento adquirido de un estudiante y ajustarla de acuerdo a los resultados obtenidos.

6.4 Evaluación del Tutor Inteligente

Para poder evaluar un STI generado mediante la herramienta sería necesario realizar un estudio de un grupo de estudiante utilizando el tutor. Esto está más allá del alcance de esta tesis. Sin embargo, podemos considerar resultados previos de STI similares generados manualmente por Noguez y Sucar (2005) para el dominio de robótica móvil, donde se muestra que el aprendizaje de los estudiantes mejora al incorporar el STI al laboratorio virtual. Consideramos que podemos esperar resultados similares con base en la herramienta; los cuales dependen también de la capacidad del docente y de las facilidades del laboratorio virtual.

6.5 Pruebas de desempeño

Para las pruebas de desempeño se midió el tiempo de respuesta del software en una computadora con procesador Intel Pentium Dual-Core a 1.73 GHz y 1 GB de memoria RAM. Los procesos evaluados dentro de la

herramienta de autoría son: generar base de conocimiento, generar temario, generar modelo del estudiante y generación del tutor. Dentro del STI se evaluaron los procesos de: generar evaluación y asignar lecciones. Estos procesos son considerados los más relevantes, ya que son donde la herramienta genera los módulos principales del STI y donde el STI realiza el proceso de asignación, evaluación y actualización del modelo del estudiante después de haber resuelto un experimento.

Tabla 6-5 Tiempos de Respuesta de los procesos principales de la herramienta

| Herramienta de autoría | Nombre de la Pantalla | | Tiempo(ms) |
|------------------------------------|-----------------------|------------------------------|------------|
| | En la | Generar Base de Conocimiento | |
| Generar Temario | | | 875 |
| Generar Módulo estudiante | | | 1426 |
| Generar Archivos de configuración | | | 1059 |
| Generación de las reglas del tutor | | | 25 |
| En el Tutor | Generar Evaluación | | 1042 |
| | Asignar Lecciones | | 987 |

Las pruebas se realizaron con los datos descritos en el anexo B para RNA. Se realizaron los procesos y se midió el tiempo de respuesta para cada proceso; los tiempos aquí medidos son sin tomar en cuenta los tiempos de captura de los datos. El número de nodos que se generan en la red bayesiana dinámica para el ejemplo descrito en el anexo B es de 93 nodos; mientras que la red Bayesiana inicial es de 31 nodos. El tiempo total de generación de los archivos que componen dichas redes es de 1.4 segundos.

En el caso de la fase de generación del STI el tiempo total es de aproximadamente tres segundos, lo cual consideramos aceptable, ya que en principio solo se realiza una vez. En la fase de operación del tutor los

tiempos obtenidos son razonables, ya que un estudiante esperaría aproximadamente un segundo en recibir la retroalimentación para un experimento que ha generado.

6.6 Resumen del Capítulo

En este capítulo se abordaron diferentes estrategias de evaluación para la herramienta desarrollada: usabilidad, efectividad del modelo del estudiante y efectividad del tutor. En cuanto a la usabilidad la encuesta muestra que la herramienta cumple con los propósitos de desarrollar un modelo del estudiante a partir de un contenido temático, pero falta mejorar la tolerancia para corregir errores como lo mostro la encuesta de usabilidad. La eficiencia en la evaluación del modelo del estudiante en cuanto a su consistencia fue positiva debido a que los resultados arrojan los valores que habían sido previstos. Respecto a la evaluación de la efectividad del tutor este aspecto queda fuera del alcance de este trabajo, aunque resultados previos con modelos similares producen cierta evidencia de que el impacto en el aprendizaje es positivo. En el siguiente capítulo hablaremos de las principales contribuciones de esta investigación y de los trabajos futuros.

Capítulo 7

Conclusiones y Trabajo Futuro

7.1 Conclusiones

El uso de sistemas tutores inteligentes se ha popularizado en los últimos años; sin embargo, su construcción requiere de una gran cantidad de tiempo y personal capacitado. Para agilizar la construcción de STIs se han desarrollado herramientas de autoría. Las herramientas desarrolladas a la fecha están restringidas a ciertas clase de tutores y no se aplican directamente a laboratorios virtuales.

La herramienta de autoría que se ha desarrollado en esta investigación permite a los profesores generar sistemas tutores inteligentes basados en un modelo probabilístico; los cuales son obtenidos de una jerarquía de temas, subtemas y conceptos (temario del curso). El STI generado por la herramienta es utilizado como apoyo a un laboratorio virtual existente. La herramienta facilita a un docente la generación de un STI, sin requerir que sea experto en tutores o modelos probabilistas. El aspecto central de la herramienta es la generación automática de un MPR a partir de la información dada por el profesor. El MPR permite posteriormente modelar el estado de conocimiento del estudiante y proveerle asesoría personalizada.

Se ataca la facilidad de uso haciendo transparente para el docente que el modelo de la red bayesiana obtiene sus valores de probabilidad basado en el porcentaje del contenido temático. El tipo de STI generado puede ser utilizado en cualquier situación donde se tenga un laboratorio virtual funcionando. El sistema tutor inteligente se comunica al laboratorio virtual mediante las variables que se desean evaluar en cada experimento. Con lo anterior la herramienta es de uso general para cualquier laboratorio virtual que incluya variables a monitorear en los aspectos a ser evaluados.

La herramienta permite que cualquier profesor que utiliza un laboratorio virtual y desea incorporar un STI, pueda hacerlo sin necesidad de manejar las técnicas de inteligencia artificial necesarias ni el dominio de un lenguaje de programación.

La herramienta de autoría fue evaluada mediante la generación de un STI para un laboratorio virtual de redes neuronales y dos pruebas de usabilidad a personas voluntarias en Base de datos y Robótica. Se evaluaron tres aspectos: Usabilidad, inferencias en el modelo del estudiante y evaluación del tutor. Mediante las pruebas que se realizaron se considera que la herramienta tiene una usabilidad aceptable y permite generar al modelo de forma transparente al docente; sin embargo, falta por mejorar las instrucciones y la tolerancia a errores de captura. En cuanto al modelo del estudiante se hicieron pruebas que demuestran que los valores obtenidos mediante propagación probabilística en el modelo generado es coherente con los resultados esperados. La evaluación de tutor inteligente queda como trabajo futuro, pero de manera indirecta gracias, a las pruebas realizadas para este tipo de tutores en laboratorios virtuales se esperan buenos resultados.

7.2 Aportaciones

Las principales aportaciones de esta investigación son:

- a) Una herramienta de autoría que permite construir un STI basado en modelos probabilísticos a partir de un esquema jerárquico de: temas, subtemas y conceptos. Dicho STI está enfocado a trabajar con laboratorios virtuales.
- b) La construcción automática y transparente del modelo del estudiante que utiliza redes bayesianas; sin necesidad de que el docente conozca los conceptos de las mismas. Además, se implementó una red bayesiana dinámica que permite hacer uso de las evidencias de resultados de experimentos realizados anteriormente y poder actualizar el modelo de forma dinámica.

7.3 Trabajo Futuro

Parte del trabajo futuro de esta investigación consisten en el diseño y generación del módulo tutor, que incorpore estrategias y modelos pedagógicos eficientes para ser utilizados en el STI. Con este tipo de tutores se pueden establecer sistemas de monitoreo más complejos. La evaluación de desempeño tanto del tutor como de la herramienta, requiere por lo menos de la duración de un curso completo, a fin de identificar cuáles son los avances de los estudiantes que utilizaron el STI y compararlos con los que no lo utilizaron. Asimismo, se requiere evaluar la satisfacción del docente con las estrategias seguidas por el STI durante la implantación del sistema en un grupo de estudiantes controlado.

Otras extensiones importantes para la herramienta de autoría es la generalización de la misma para diseñar sistemas tutores inteligentes, para ambientes web, móviles y diversos sistemas operativos; asimismo, se requiere diseñar estrategias que permitan ensamblar los STI generados por la herramienta a plataformas de enseñanza o gestores de contenido para cursos mediante la web. Otras posibles mejoras son: desarrollar una interfaz de captura mediante iconos que permita al usuario dibujar el mapa conceptual; así como extender la versión actual a más de tres niveles de profundidad en la enseñanza.

Referencias

Aleven, Vicent, Bruce M. McLaren, Jonathan Sewall, and Kenneth R. Koedinger. "The Cognitive Tutor Authoring Tools (CTAT): Preliminary Evaluation of Efficiency Gains." *Proceedings of the 8th International Conference on intelligent Tutoring Systems*. Jhongli, Taiwan, 2006. 26-30.

Béjar, Javier. «Modelos Probabilistas.» *Notas del curso de IA, 2005-2006*. España: Universidad Politecnica de Cataluña, 18 de Diciembre de 2005.

Conati, C., A. Gertner, k. Vanlehn, and M. Druzel. "On line student modeling of coached problem solving using Bayesian networks." *Using Modeling: Proceedings of Sixth International Conference*. New York, USA: Springer Verlag, 1997. 231-242.

Conejo, Ricardo, Eva Millán, Jose Luis Pérez de la Cruz, and Mónica Trella. "Modelado del alumno: Un enfoque bayesiano." *Revista Iberoamericana de Inteligencia Artificial*, 2001: 50-58.

Dechter, R. «Bucket elimination: A unifying framework for probabilistic inference.» *Twelfth Conf. on Uncertainty in Artificial Intelligence*. Portland, Oregon,: E. Horvits and F. Jensen, 1996. 211--219.

Elvira. *Elvira*. 01 01, 2001. <http://leo.ugr.es/elvira/> (accessed 01 18, 2009).

Ferré, X., N. Juristo, H. Windl, y L. Constantine. «Usability Basics for Software Developers.» *IEEE Software* 18,1, 2001: 22-28.

Ferrero, B., A. Arruate, I. Fernandez-Castro, and M. Urretavizcaya. "Herramientas de Autor para Enseñanza y Diagnóstico: Iris-D." *Revista Iberoamericana de Inteligencia Artificial No. 12*, 2001: 13-28.

Fortin, Mikaël, Jean-François Lebeau, Amir Abdessemed, François Courtemanche, and André Mayers. "A Standard Method of Developing User Interfaces for a Generic ITS Framework." In *Intelligent Tutoring Systems*, by Beverley P. Woolf, Esma Aïmeur, Roger Nkambou and Susanne Lajoie, 312-322. Berlin, Alemania: Springer Berlin / Heidelberg, 2008.

Freedman, R., S. S. Ali, and S. McRoy. "What is an Intelligent Tutoring System?" *Intelligence (Intelligence)* 11, no. 3 (2000): 15-16.

Friedman, Nir, Lise Getoor, Daphne Koller, and Avi Pfeffer. "Learning Probabilistic Relational Models." *International Joint Conferences on Artificial Intelligence, IJCAI*, 1999: 1300-1309.

Getoor, Lise, Nir Friedman, Daphne Koller, Avi Pfeffer, and Benjamin Taskar. "Probabilistic Relational Models." In *An Introduction to Statistical Relational Learning*, by L. Getoor and B. Taskar, 129-174. MIT Press, 2007.

Gómez-Gil, Pilar. *Diseño de Laboratorio Virtual Básico de Redes Neuronales de Retropropagación*. Reporte Técnico, Neuronal Net Research Lab: Instituto Nacional de Astrofísica, Óptica y Electrónica. Coordinación de Ciencias Computacionales, 2009.

Hämäläinen, Wilhelmiina, and Mikko Vinni. "Comparison of Machine Learning Methods for Intelligent Tutoring Systems." In *Intelligent Tutoring Systems*, by Ikeda Mitsuru, Ashley Kevin D. and Chan Tak-Wai, 525-534. Jhongli, Taiwan: Springer Berlin / Heidelberg, 2006.

Hernández, Y., and J. Noguez. "Affective Behavior in Intelligent Tutoring Systems for Virtual Laboratories." *12th International Conference on Artificial Intelligence in Education, AIED 2005*. Amsterdam, Países Bajos, 2005. 960-962.

IEEE. *Standard for Learning Technology. Learning Technology System Architecture*. USA: IEEE, 2003.

ITESM. *Laboratorio Virtual de Robotica*. 2008. <http://elearning.ccm.itesm.mx:8080/VirtualLaboratory/> (último acceso: 18 de 05 de 2009).

Joler, M., and C.G. Christodoulou. "Virtual Laboratory, Instruments and Simulations Remotely Controlled via Internet." *Antennas and Propagation Society IEEE Intl. Sym.*, 2001: 388-391.

López Puga, Jorge, and Juan García García. "Sistemas de Tutorización Inteligente Basados en Redes Bayesianas." *Revista Electrónica de Metodología Aplicada*, 2008: 13-25.

Massino, M., and T.B. Sheridan. "Teleoperator Performance with varying force and visual feedback." *Human Factors*, 1994: 145-157.

Mislevy, R., y D. Gitomer. *The role of probability-based inference in an intelligent tutoring system*. Technical Report, National center for research on evaluation., 1996.

Murray, R., y K. VanLehn. «DT Tutor: A Decision-Theoric, Dynamic Approach for Optimal Selection of Tutorial Acions.» *Proceedings of 5th International Conference on Intelligent Tutoring Systems*. Springer Verlag, 2000. 153-162.

Murray, Tom. "Authoring Intelligent Tutoring Systems: An Analysis of the State of Art." *International Journal of Artificial Intelligence in Education*, 1999: 98-129.

Nkambou, Roger. «Managing Inference Process in Student Modelling for Intelligent Tutoring Systems.» *Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence*. Washington, DC.: ICTAI. IEEE Computer Society,, 1999. 19.

Noguez, Julieta. *Modelo probabilista relacional del estudiante para laboratios Virtuales, Tesis Doctoral*. Cuernavaca, Morelos: ITESM, Capus Cuernavaca, 2005.

Noguez, Julieta, y L. Enrique Sucar. «A Probabilistic Relational Student Model for Virtual Laboratories.» *Sixth Mexican International Conference on Computer Science (ENC'05)*. 2005. 2-9.

Salgueiro, Fernando, Zulma Cataldi, Fernando Lage, and Ramon García-Martínez. "Sistemas Tutores Inteligentes: Redes Neuronales para Selecccion del Protocolo Pedagógico." *IV Workshop de Tecnología Informática Aplicada en Educacion*. Buenos Aires, Argentina, 2005. 255-266.

Sánchez, Vila, and Penín Lama. "Monografía: Técnicas de la inteligencia artificial aplicadas a la educacion." *Revista Iberoamericana de Inteligencia Artificial*, 2007: 7-12.

Sun Microsystems. *Netbeans*. Septiembre de 2008. <http://www.netbeans.org/> (último acceso: 01 de 09 de 2008).

VanLenh, Kurt. "Student Modeling." In *Foundations of Intelligent Tutoring Systems*, by C. Martha Polson and J. Jeffrey Richardson, 55-78. Hillsdale, NJ, USA: LEA, 1988.

Wagner, B. «From Computer Based Teaching to Virtual Laboratories in Automatic Control.» *Frontiers in Education Conference*. San Juan Puerto Rico: ASEE/IEEE, 1999. Sesion 13d6-6.

Zatarain Cabada, Ramón, María Lucía Barrón Estrada, Barrientos Eduardo Urías, Moisés Osorio Velásquez, and Carlos Alberto Reyes García. "Multiple Intelligence Tutoring Systems for Mobile Learners." *Eighth IEEE International*

Conference on Advanced Learning Technologies. Santander, Cantabria, Spain: IEEE, 2008. 652-653.

Apéndices

A.-GUIAS PARA EL DISEÑO DEL LABORATORIO

(Gómez-Gil 2009)

Este laboratorio se utilizará en conjunción con una herramienta automática de autoría de tutores inteligentes basada en MPR, llamada HEDEA [Romero 2008]. A fin de que el laboratorio sea directamente utilizable por la herramienta, y que a partir de éste pueda construirse un tutor inteligente, su diseño debe cumplir con las siguientes condiciones:

1. Debe definirse un temario que describa los componentes del conocimiento que el laboratorio (y por consiguiente el tutor inteligente) refuerza.
2. La definición del temario debe tener solamente 3 niveles de detalle, que llamaremos “tema,” “sub-tema” y “concepto.”
3. Cada nivel del temario (tema, sub-tema o concepto) tendrá asignado un porcentaje relacionado a la cantidad de aporte de dicho nivel al conocimiento total del alumno (a). Este porcentaje será introducido por el instructor al momento de capturar el temario.
4. El tutor inteligente mantendrá en todo momento información del conocimiento de cada uno de los alumnos que lo utilizan.
5. Cada alumno(a) tendrá asociada una categoría de conocimiento, la cual puede ser “principiante,” “intermedio” o “avanzado.” Inicialmente, la categoría asignada por el tutor inteligente es “principiante.”
6. El laboratorio virtual estará formado por una serie de experimentos. Cada experimento consistirá de los siguientes componentes:
 - a. **Nombre.** Identifica al experimento de manera única
 - b. **Nivel del experimento.** Puede tomar los valores: básico, intermedio o avanzado.
 - c. **Objetivo.** Contiene una pequeña redacción escrita por el instructor sobre las partes del conocimiento que se verán probadas y/o afectadas con el experimento.

- d. **VARIABLES A EVALUAR.** Contiene información de las variables que servirán para evaluar el conocimiento del estudiante. Por cada variable se debe incluir:
 - i. el *nombre* de la variable,
 - ii. el *umbral* de su valor permitido (valor mínimo aceptable) ó el rango de su valor permitido (en caso de ser numérica), o un conjunto de valores permitidos en caso de ser numérica o alfanumérica
 - e. **Escenario.** Es una breve descripción de la información que se presenta al estudiante, así como el medio ambiente sobre el que se desarrolla el experimento.
 - f. **Descripción narrativa.** Aquella dada al estudiante acerca del experimento, donde se le explica el problema a resolver y se le indican las condiciones de evaluación del experimento.
7. El instructor(a) definirá una serie de lecciones asociadas a conceptos, subtemas o temas, según desee. Si el/la estudiante no satisface un mínimo de conocimiento de alguno de éstos, el tutorial presentará al estudiante la lección correspondiente que previamente asignó el instructor(a). El tutor inteligente generado por HEDEA se encargará de decidir automáticamente si el/la estudiante satisface o no el conocimiento a nivel tema, subtema y/o concepto.
8. Cada experimento en el laboratorio virtual tendrá asignado un conjunto de reglas, las cuales se construyen automáticamente a partir de las definiciones dadas para cada variable del experimento basándose en:
- a. El umbral de su valor permitido (valor mínimo aceptable) ó,
 - b. El rango de su valor permitido (en caso de ser numérica), o
 - c. Un conjunto de valores permitidos en caso de ser numérica o alfanumérica.

REFERENCIAS

Romero-Inzunza, Mario Alberto. *“Diseño y Desarrollo de una Herramienta de Autoría para Tutores Inteligentes aplicables a ambientes educativos virtuales utilizando MPR’s y POO.” Propuesta tesis de maestría, coordinación de computación, INAOE. Junio 2008*

Romero-Inzunza, Mario Alberto. *“Diseño y Desarrollo de una Herramienta de Autoría para Tutores Inteligentes aplicables a ambientes educativos virtuales utilizando MPR’s y POO.” Cartel de avances otoño 2008, coordinación de computación, INAOE. Diciembre 2008*

Romero-Inzunza, Mario Alberto “Shell neuronal de retro-propagación”, trabajo de programación del curso: “Tópicos Selectos: Redes neuronales artificiales.” *Coordinación de computación, INAOE. Primavera 2008.*

Apéndice B

Caso de Estudio.

A continuación se enlista una tabla que contiene los temas, subtemas y conceptos relacionados a la enseñanza de redes neuronales de retropropagación.

| TEMA | % Conocimiento | SUBTEMA | % Conocimiento | CONCEPTO | % Conocimiento | NOMBRE DE LA LECCIÓN ASIGNADA |
|--|--------------------|---|--------------------|--|--------------------|-------------------------------|
| 1. Características de Diseño de topologías | 50% | | | | | Topologías.ppt |
| | | 1.1 Nivel de entrada | 30% | | | |
| | | 1.2 Niveles escondidos | 5% | | | |
| | | 1.3 Nivel de salida | 20% | | | |
| | | 1.4 Función de activación | 5% | | | |
| | | 1.5 Consideraciones de aprendizaje | 10% | | | parametrosApre n.ppt |
| | | | | 1.5.1 Valores iniciales de pesos | 20% | |
| | | | | 1.5.2 Coeficiente de aprendizaje | 30% | |
| | | | | 1.5.3 Escalado de función de activación | 50% | |
| | | | | | Total: 100% | |
| | | 1.6 Tipos de problemas | 30% | | | tiposProblemas.ppt |
| | | | | 1.6.1 Clasificador | 33% | |
| | | | | 1.6.2 Asociador de patrones | 33% | |
| | | | | 1.6.3 Aproximador de funciones | 34% | |
| | | | | | Total: 100% | |
| | | | Total: 100% | | | |
| 2. Técnicas de entrenamiento | 50% | | | | | Entrenamiento.ppt |
| | | 2.1 Manejo de parámetros de aprendizaje | 70% | | | |
| | | | | 2.1.1 Cambio dinámico de parámetro | 70% | CambiParametros.ppt |
| | | | | 2.1.2 Cambio en la inicialización de pesos | 30% | |
| | | | | | Total: 100% | |
| | | 2.2 Manejo del total de barridas | 30% | | | |
| | | | | 2.2.1 Paro por máximo de barridas | 50% | |
| | | | | 2.2.2 Paro por valor del error | 50% | |
| | | | | | Total: 100% | |
| | | | Total: 100% | | | |
| | Total: 100% | | | | | |

EXPERIMENTOS PARA EL LABORATORIO VIRTUAL

EXPERIMENTO 1:

- **Nombre.**
TOPOLOGIA CORRECTA
- **Nivel del experimento (básico, intermedio o avanzado)**
BÁSICO
- **Objetivo**
Determinar las topologías correctas para una RNA dada la descripción narrativa de un problema

- **Variables a evaluar.**

| Nombre de la variable | Rango del valor permitido |
|-----------------------|---------------------------|
| Neuronas-entrada | de 2 a 2 |
| Neuronas-salida | De 1 a 1 |
| Neuronas-escondidos | Mayor que 2 |
| Función-activación | “Lineal” |

- **Escenario.**
 - Presentar al estudiante:
 - Una descripción por default de una topología de una red neuronal alimentada hacia adelante con un nivel escondido
 - El texto de descripción del problema dado en el siguiente punto
 - Permitir al estudiante cambiar valores y avisar al sistema cuando está lista la topología que desea.
 - Al terminar, el laboratorio envía los resultados de las variables al tutorial.

- **Descripción dada al estudiante acerca del experimento**

“Se desea utilizar una red neuronal alimentada hacia adelante con un nivel escondido, para calcular el valor de una función escalar $g(x,y)$. Se tienen 100 ejemplos de evaluación de esta función, que contienen los valores de entrada de x , y y su correspondiente valor de salida. Estos valores de salida están en el rango $[-30,15]$. Dibuje la arquitectura correcta de esta red. La descripción de la arquitectura incluye el número de neuronas de entrada, escondidos, de salida y tipo de función de activación en el último nivel”

EXPERIMENTO 2:

- **Nombre.**
ANIMALES
- **Nivel del experimento (básico, intermedio o avanzado)**
INTERMEDIO
- **Objetivo**
Determinar las topologías correctas para una RNA dada la descripción narrativa del problema.
- **Variables a evaluar.**

| Nombre de la variable | rango del valor permitido |
|-----------------------|---------------------------|
| Neuronas-entrada | de 5 a 5 |
| Neuronas-salida | De 3 a 3 |
| Neuronas-escondidos | Mayor a 5 |
| Función-activación | “sigmoide” |

- **Escenario.**
 - Presentar al estudiante:
 - Una descripción por default de una arquitectura de red
 - El texto de descripción del problema
 - Permitir al estudiante cambiar valores y avisar al sistema cuando está listo
 - El laboratorio envía los resultados de las variables al tutorial
- **Descripción dada al estudiante acerca del experimento**

“Se quiere usar una RNA con un nivel escondido que permita distinguir entre 3 tipos de animales. Se tiene un conjunto de entrenamiento que incluye las siguientes medidas de cada animal:

 - Peso
 - Altura
 - Color
 - Número de patas
 - Tipo de comida (1 = herbívoro, 2 = carnívoro, 3 = otro)

EXPERIMENTO 3:

- **Nombre.**
LEDS
- **Nivel del experimento (básico, intermedio o avanzado)**
AVANZADO
- **Objetivo**
Determinar si el/la estudiante es capaz de mejorar el desempeño de una RNA a través de la modificación de los parámetros de ésta.
- **Variables a evaluar.**

| Nombre de la variable | rango del valor permitido |
|-----------------------|---------------------------|
| Desempeño obtenido | Menor o igual a 1.0E-3 |
| Total_cambios | Menor a 20 |

- **Escenario.**
 - El sistema permite al estudiante modificar todos los parámetros que desee de la red neuronal, y ejecutar el proceso de entrenamiento de la red, a lo más 5 veces.
 - El sistema toma nota del número de cambios efectuados y almacena la eficiencia obtenida por la red (MSE alcanzado) en cada caso.
 - Al terminar los 5 intentos, el sistema pregunta al estudiante que corrida considera la mejor y envía el desempeño obtenido y total de cambios para el caso elegido por el/la estudiante.
- **Descripción dada al estudiante acerca del experimento**
“Se quiere hacer una red que reciba un dígito (número entre 0 y 9), representado en binario (4 bits) y lo convierta a su representación en un sistema de leds. Se cuenta con un archivo de entrenamiento, llamado leds.txt que tiene 10 renglones, en cada uno está el número en binario (4 bits) y su correspondiente representación en el sistema de leds (7 bits)”

Apéndice C

Manual de usuario de la herramienta de autoría para sistemas tutores inteligentes (HEDEA)

Autor: Mario Alberto Romero Inzunza

Instituto Nacional de Astrofísica, Óptica y Electrónica

Coordinación de Ciencias computacionales

ÍNDICE

| | |
|---|-----|
| 1.-INTRODUCCIÓN..... | 98 |
| 2.-CONFIGURANDO LABORATORIO NUEVO | 100 |
| 2.1 CAPTURANDO EXPERIMENTOS..... | 101 |
| 3.-CONFIGURANDO LABORATORIO EXISTENTE | 102 |
| 4.-GENERAR BASE DE CONOCIMIENTO:..... | 103 |
| 6.-GENERANDO LA BASE DE CONOCIMIENTO | 105 |
| 7.-GENERAR MÓDULO ESTUDIANTE | 105 |
| 6.-GENERAR MÓDULO TUTOR | 106 |

1.-Introducción

La herramienta de Autoría para Sistemas Tutores Inteligentes es un software desarrollado en Java, con el que se pueden diseñar Sistemas Tutores Inteligentes (STI) sin necesidad de que el docente tenga conocimientos sobre las técnicas de inteligencia artificial utilizadas para construirlos como redes bayesianas y modelos probabilistas, los cuales son la base para el núcleo de los STIs que se pueden generar.

La presente versión de la herramienta requiere de la existencia un laboratorio virtual donde se realizaran los experimentos, conocer las variables que se pueden monitorear en el laboratorio.

Además es necesario que el docente tenga estructurado el contenido del curso de forma jerárquica en los niveles de Temas, Subtemas, Conceptos y Variables asociadas a los experimentos.

El diagrama de las figuras C-1 y C-2 muestran una perspectiva general de la secuencia de trabajo que se realiza.

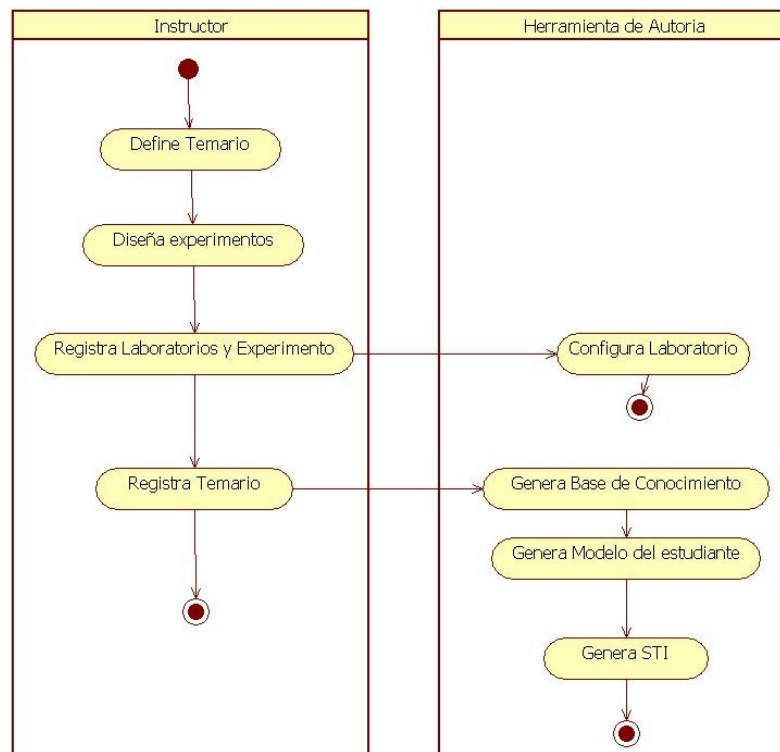


Figura C-8-1 Proceso General de trabajo

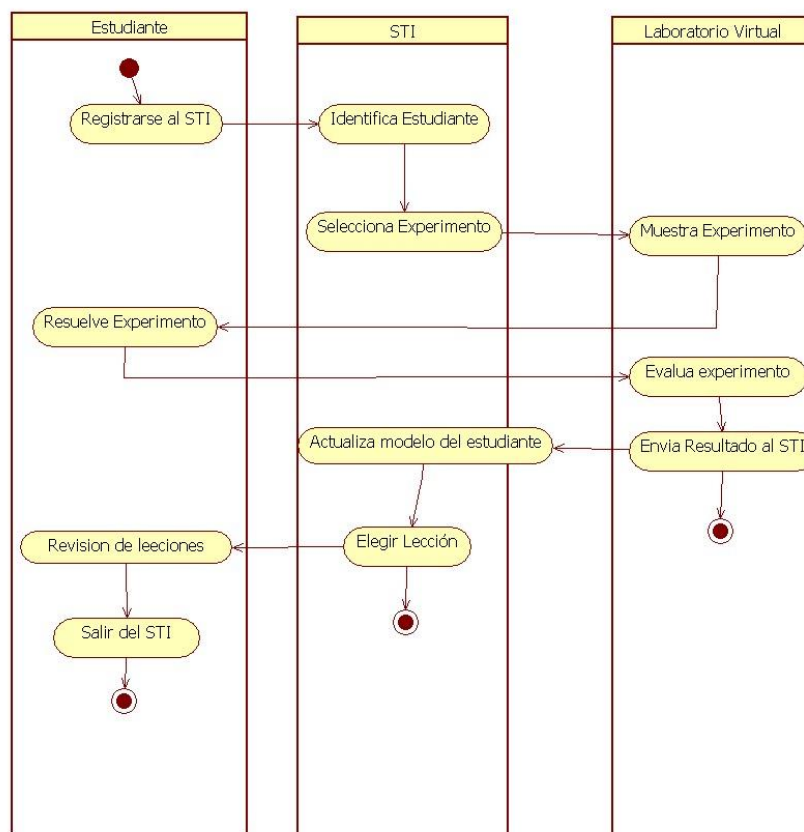


Figura C-8-2 Proceso General en el STI

En la figura C-1 se puede observar que el docente requiere realizar algunas actividades muy importantes antes de empezar a utilizar la herramienta entre las que se destacan:

- Definir Temario
- Definir Experimentos
- Registrar laboratorio virtual
- Registrar las variables a evaluar en el laboratorio virtual.

En esta misma figura se observa cada una de las actividades que tiene que ejecutar la herramienta de autoría para ir generando los archivos de configuración para el STI.

En la figura C-2 se ilustra de manera general como sucede el proceso de comunicación entre el STI, el laboratorio virtual y el estudiante, así el usuario puede hacerse de una idea de cómo funcionara el STI que se genere.

La figura C-3 muestra la pantalla que aparece al ejecutar HEDEA. Para empezar a elaborar un sistema tutor inteligente con la herramienta de autoría es necesario configurar un laboratorio virtual.

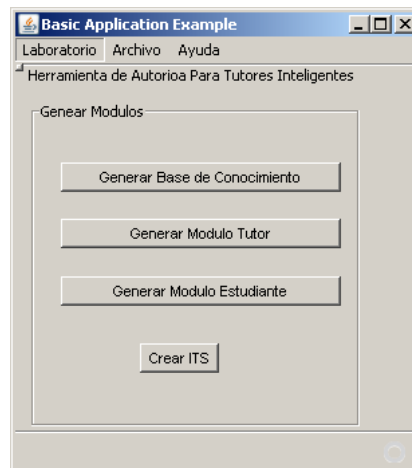


Figura C-8-3 Pantalla inicial.

2.-Configurando Laboratorio Nuevo

En la figura c-3 se muestra la pantalla de configuración para el laboratorio, para lo cual es necesario tener estructurado los experimentos, así como el número de variables a evaluar, el nombre de las variables y el valor esperado de la variable en cada experimento. El nombre del laboratorio no lleva “.” Ni caracteres especiales, solo el nombre en texto alfanumérico. La extensión de los archivos de laboratorio es .LABI, la cual no es capturada, los archivos de configuración son almacenados en la carpeta laboratorios.

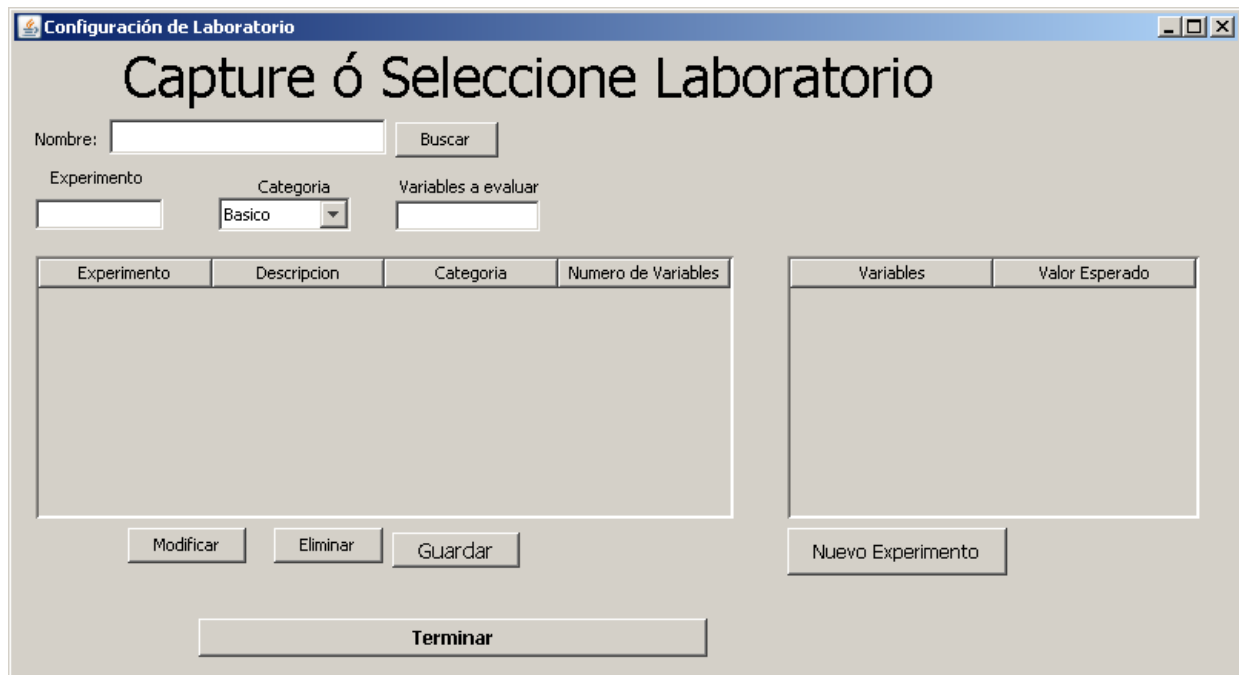


Figura C-8-4 Pantalla de configuración del laboratorio

2.1 Capturando Experimentos.

Para capturar un experimento se teclea un nombre de experimento, el nivel al que pertenece y el número de variables que se evaluarán en ese experimento.

Después en la parte derecha de la pantalla se capturarán los nombres de las variables con las siguientes características, las variables deben ser caracteres alfanuméricos sin caracteres especiales permitidos solo el guión chaparro (_) y sin espacios en blanco. En esta misma sección se captura el valor esperado para la variable las cuales deben coincidir con alguna de las siguientes formas: Variables de texto, numéricas o rangos de número. La única restricción para colocar una variable de rango es que se capture primero el valor menor seguido de un (guion) – y luego el valor mayor del rango. Para valores sin límite sugiero colocar un valor extremo por ejemplo “mayor que 5 “capturaríamos “5-100000”.

Al finalizar de configurar un experimento hacemos clic en nuevo experimento con lo cual nos permite capturar una descripción de experimento, la cual puede ser una archivo que diga en qué consiste el experimento. Por último se hace clic en guardar para almacenar los cambios y empezar a trabajar con el laboratorio.

3.-Configurando Laboratorio Existente

Si se cuenta con un laboratorio existente, se hace clic en el botón buscar de la pantalla en la figura 2 desplegándose la pantalla de la figura 3.

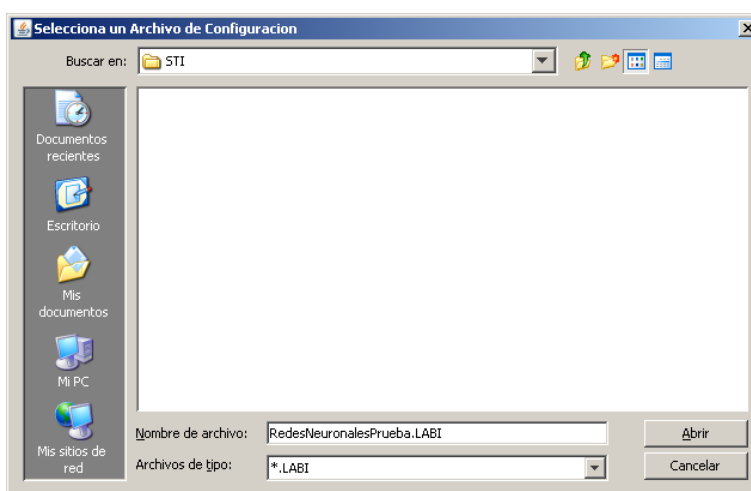


Figura C.-8-5 Pantalla de selección de laboratorios existentes

Una vez seleccionado un laboratorio se carga el listado de experimentos existentes en el laboratorio, y se puede proceder a capturar nuevos experimentos como se describió en la sección anterior o se cierra la pantalla para empezar a trabajar en la herramienta de autoría.

Al finalizar de editar un laboratorio es necesario que se haga clic en el botón de guardar.

Nota: Si no se activa algún laboratorio no se puede trabajar con la herramienta de autoría, esto debido a que los Tutores Inteligentes generados, se

encuentran asociados a un laboratorio. El laboratorio activo es aquel que se está configurando si no se da clic al botón guardar se trabajara con la última versión almacenada.

4.-Generar Base de conocimiento:

La figura C-6 muestra la pantalla de configuración de la base de conocimiento.



Figura C-8-6 Pantalla para generar la base de conocimiento

En este módulo se captura toda la información relacionada con el temario (Temas, Sub-Temas y Conceptos),

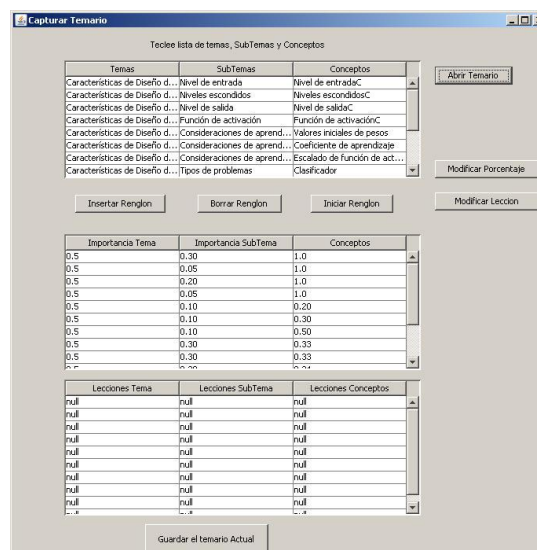


Figura C-8-7 Captura de temario

Como primer paso se tiene que capturar un temario, los niveles de jerarquía son: Temas, Sub-Temas y Conceptos. Para el proceso de captura se tiene que hacer mediante los conceptos siguiendo la siguiente estrategia. Se teclea el tema al que pertenece el concepto, luego el subtema y al final el nombre del concepto. Para añadir más conceptos del mismo subtema simplemente se presiona la tecla enter. Si se desea añadir un nuevo sub-tema se posiciona el mouse sobre el último sub-tema añadido y se presiona enter. Y si desea un nuevo tema solo se coloca el cursor en la primera celda del último renglón añadido y se presiona enter.

Durante la captura de un nuevo, tema, Subtema o concepto aparecerá una ventana desplegable solicitando información sobre: Porcentaje de conocimiento que aporta al padre, y lugar donde acceder a la información a enseñar (Notas de las lecciones).

Las pantallas de captura de esa información son las siguientes:

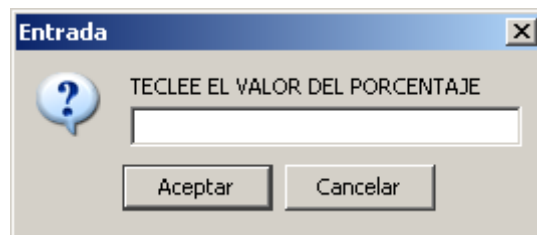


Figura C-8-8 Captura el porcentaje de conocimiento hacia el nodo padre

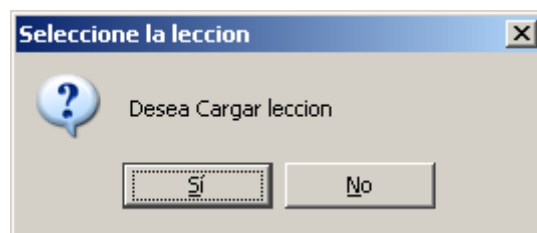


Figura C-8-9 Pregunta sobre lección

Si no se desea añadir información de la lección se hace clic en no y simplemente se deja en blanco para futuras modificaciones. Si se dice que si se selecciona la ubicación del archivo a desplegar.

En este momento ya se tiene un mapa jerárquico de temas, sub-temas y conceptos el cual es utilizado como la base para accionar al STI.

6.-Generando la base de conocimiento

Ya que se tiene capturado el temario se empieza a relacionar cada una de las variables con las hojas del mapa jerárquico de temas, subtemas y conceptos las cuales están dadas por los experimentos del laboratorio.

7.-Generar Módulo Estudiante

Esta sección se utiliza para terminar de configurar al modelo del estudiante en el cual se tiene que capturar los datos de la aportación de cada variable hacia su padre. Aquí es importante notar que la misma variable puede asignar conocimiento diferente dependiendo cuales sean los padres en la jerarquía.

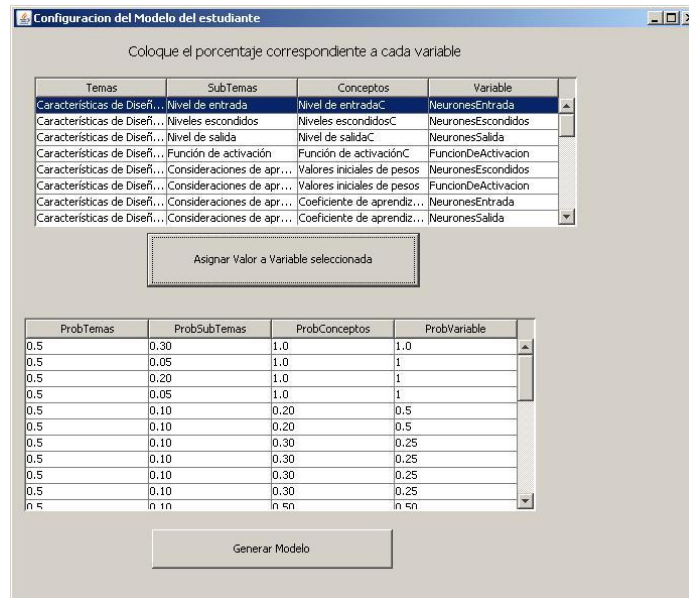


Figura C-8-10 Modelo del estudiante

En esta parte se permite configurar el porcentaje de aportación de cada variable. Al terminar de capturar todos los porcentajes. Se tiene que hacer clic en Generar Modelo. Por defecto el sistema genera una distribución uniforme entre las variables, haciendo un aporte igual para cada variable. Es decir las variables tienen la misma importancia en todos los casos. Si se desea seguir con esta idea se hace clic en generar modelo sin modificar nada.

8.-Generar Módulo Tutor

El módulo tutor, permite anotar un valor para considerar que una lección ha sido aprobada para cada categoría. En esta parte solo se coloca el valor mínimo necesario para deducir que un estudiante domina el conocimiento de esa categoría.

Con los pasos anteriores ya se tendría generado los archivos de configuración del STI. Solo hacemos clic en el botón generar ITS y se creara una carpeta con el nombre del laboratorio concatenado a la palabra tutor inteligente.

Esta carpeta se almacenara en la ruta de configuración donde se almaceno la herramienta de autoría.

Apéndice D

Archivos de Configuración

Los archivos de configuración son el mecanismo por el cual se inicializa al STI, estos archivos son archivos de texto plano con una terminación que indica el fin de cada sección.

Se generan archivos para:

- Laboratorio Virtual
- Base de Conocimiento
- Tabla de Probabilidad
- Modelo del Estudiante
- Red Bayesiana Básica
- Red Bayesiana Dinámica.

Archivo Laboratorio Virtual.- este archivo tiene una extensión .LABI, con la cual se identifica y el nombre de archivo está formado por el nombre del laboratorio virtual que representa. La estructura del archivo contiene separado por el carácter de “;” (punto y coma) la siguiente información:

- Experimento: contiene un nombre para el experimento
- Descripción: la ubicación de una archivo describa al estudiante que experimento tiene que realizar en el laboratorio virtual.
- Categoría: Nivel para el experimento (Básico, Intermedio, Avanzado).
- Numero de Variables y Nombre de las Variables a evaluar.
- Valor Esperado de cada Variable a evaluar.

Archivo Base de Conocimiento: Este archivo contiene la tabla capturada por el usuario, solo que le coloca una extensión .ELVIRA y es utilizado para generar las redes bayesianas básicas.

Archivo de Probabilidad: Contiene los valores de las probabilidades correspondientes uno a uno al archivo de base de conocimiento, este archivo se identifica con la extensión .Prob.

Archivo Modelo del estudiante.- Este archivo tiene una extensión .std y en él se encuentran los valores de peso para cada variable que puede ser monitoreada en el laboratorio virtual. Este archivo y los archivos de la base de conocimiento son utilizados para generar los archivos de las redes bayesianas.

Archivos Red Bayesiana Básica y Red Bayesiana Dinámica, estos archivos están estructurados en un formato especial requerido por ELVIRA, en el cual se extrajo la información del modelo del estudiante y de la base de conocimiento, permitiendo generar un archivo de extensión .ELV llamado mapaConceptual concatenado con el nombre del laboratorio al que pertenece y otro con la leyenda temporal, haciendo referencia a la red bayesiana dinámica.

Apéndice E

Algoritmo de Eliminación de Variable

La herramienta de autoría desarrollada en esta investigación genera archivos de configuración para el sistema tutor inteligente, el cual utiliza un algoritmo de propagación probabilista llamado *Variable Elimination* (Dechter 1996). El objetivo de la inferencia probabilística según Béjar (2005) es calcular la distribución de probabilidad a posteriori de un conjunto de variables dada la observación de un evento (Variables del Experimento en ámbito de esta investigación).

A continuación se describe el algoritmo de Variable Elimination. Para empezar se define:

Entradas

F → la lista de probabilidades condicionales en la Red Bayesiana

X → Una lista de variables a evaluar

Y → Lista de variables observadas

Y_0 → Valores de las variables observadas

P → Un orden para eliminar las variables fuera de X unión Y, es decir variables ocultas.

Salida:

$$P(X|Y=Y_0)$$

El algoritmo VE regresa un conjunto de probabilidades dado los valores de las variables, obtiene su nombre debido a que elimina las variables no observadas convirtiéndolas en factores y solo toma en cuenta las variables observadas para hacer los cálculos de las probabilidades finales. La Tabla F-1 muestra Pseudocódigo del Algoritmo VE.

Tabla F-1 Algoritmo de eliminación de variable (VE por sus siglas en ingles)

| |
|--|
| Función VE(X,Y,Y ₀ ,F) |
| factores=[]; P=Invierte(F)) |
| para cada variable en P hacer |
| factores=concatena(factores, CalculaFactor(variable, Y ₀)) |
| si la variable no esta en X unión Y |
| factores=CalculaVariableOcultas (variable, factores) |
| Fin_Para |
| return NORMALIZA(Multiplicación(factores)) |

Las funciones llamadas por el algoritmo se explican a continuación:

CalculaVariableOcultas.- Esta función multiplica los factores y los suma todos con respecto a la variable oculta es decir da por hecho todo los posibles valores de la variable.

CalculaFactor.-Genera el factor del valor correspondiente a la variable en la función.

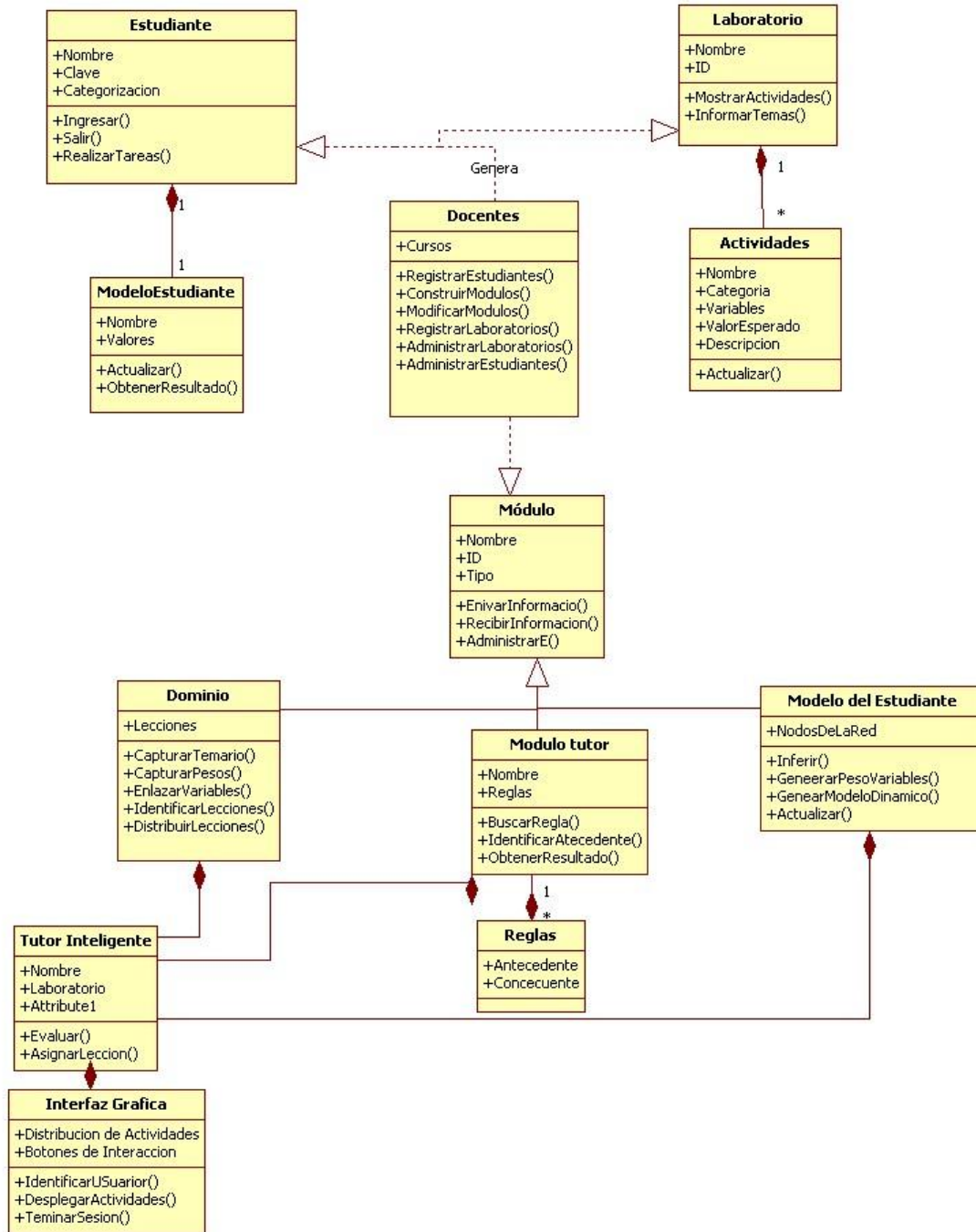
NORMALIZA.-lleva los valores de 0 a 1 en términos de probabilidad.

Al terminar el algoritmo regresa el valor normalizado de la probabilidad de la variables en X dado el conjunto de variables Y.

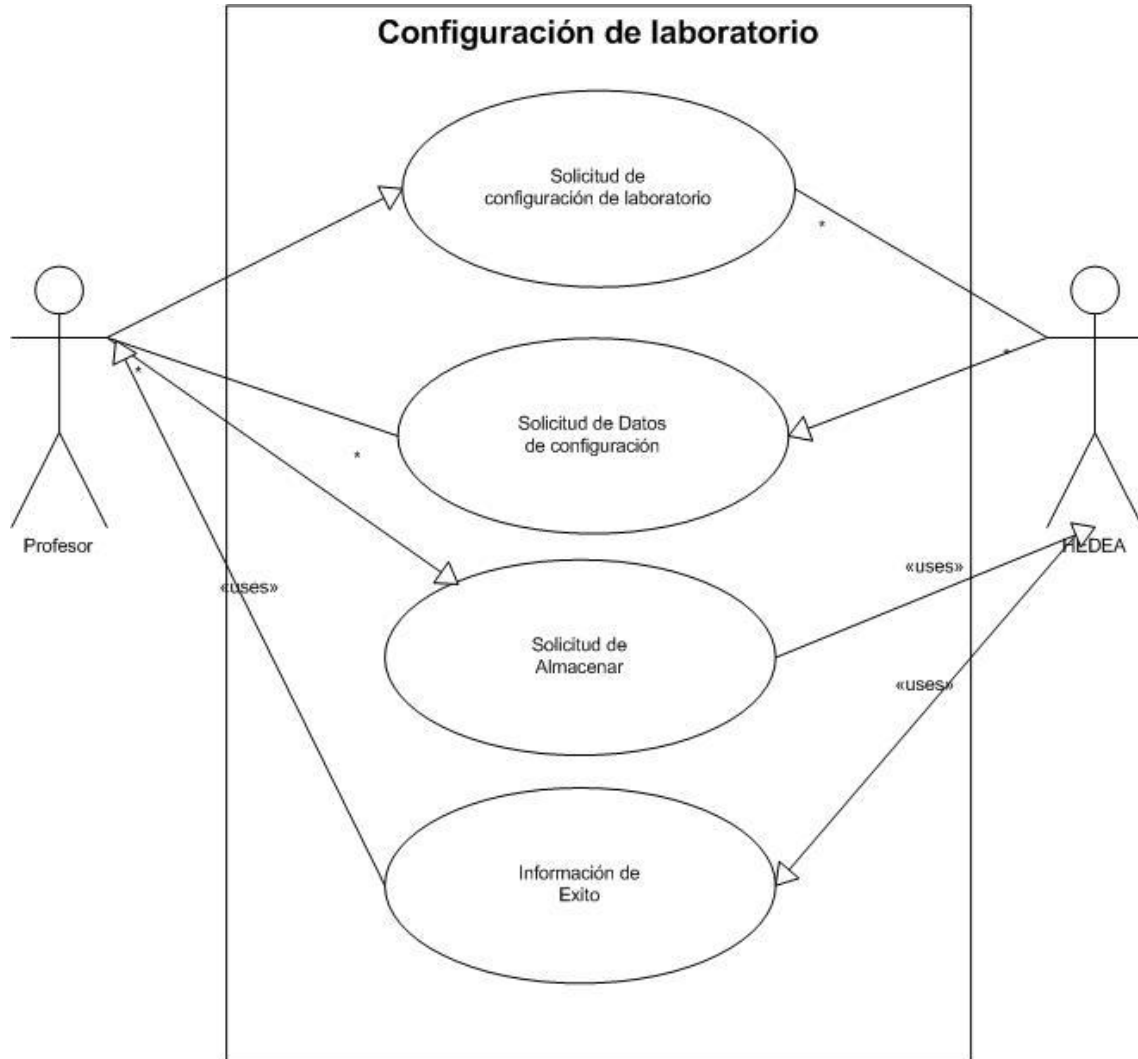
Apéndice F

Diseño de Software

Diagrama de Clases de HEDEA

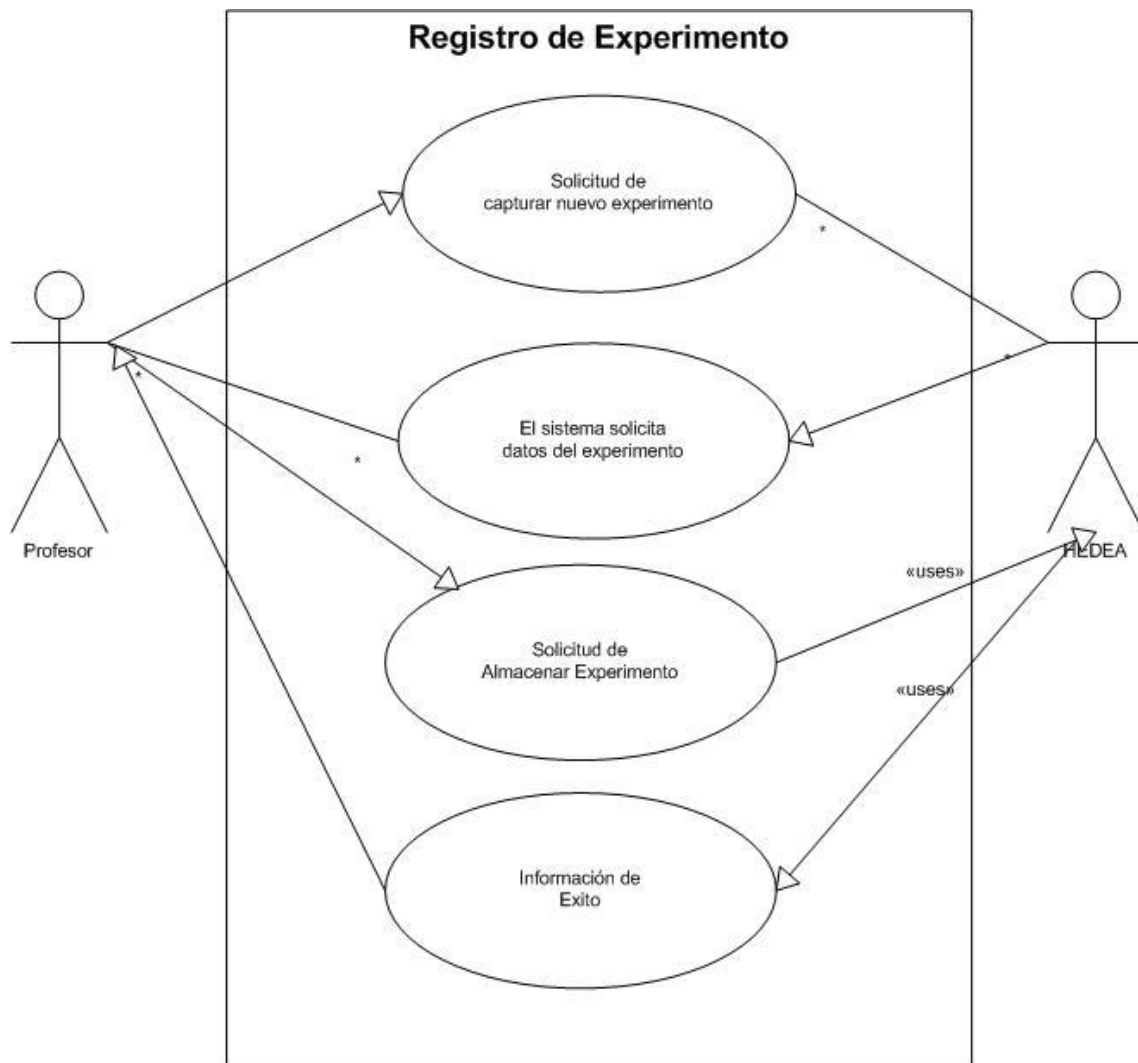


CASOS DE USO DE HEDEA

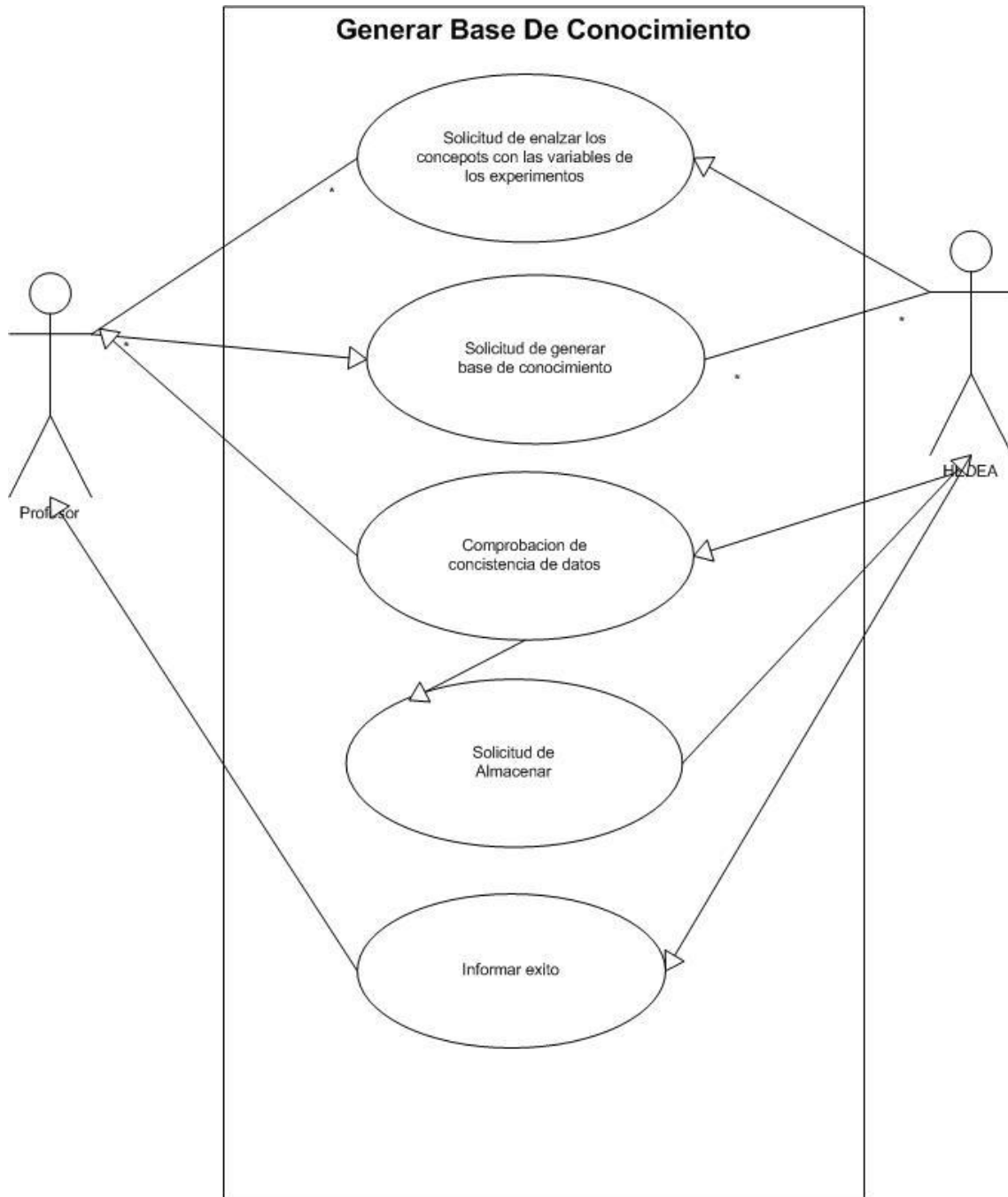


| | | |
|-----------------------------|--|---|
| Nombre | Configuración de Laboratorio | |
| Objetivos asociados | Generar los archivos para asociar un laboratorio virtual a un tutor inteligente con HEDEA | |
| Requisitos asociados | Datos del Laboratorio, descripción de experimentos, Variables a evaluar y Valores esperados | |
| Descripción | El Sistema deberá permitir la captura de los datos necesarios para que se pueda configurar el laboratorio virtual. | |
| Precondición | El usuario tiene todos los datos asociados al laboratorio virtual | |
| Secuencia Normal | Paso | Acción |
| | 1 | El usuario solicita configuración de un nuevo laboratorio virtual |
| | 2 | El sistema solicita los siguientes datos: nombre del laboratorio. |
| | 3 | El usuario solicita guardar el nuevo laboratorio |
| | 4 | El sistema comprueba la consistencia de los datos. |
| | 5 | El sistema almacena los datos proporcionados |
| | 6 | El sistema informa del éxito del proceso |
| Postcondición | El laboratorio esta almacenado y el STI que se configure estará asociado a este laboratorio. | |
| Excepciones | Paso | Acción |
| | 1 | Si ya existe el laboratorio, se abre un navegador de archivos, se selecciona el laboratorio y se carga la información de manera automática |
| | 2 | El usuario solicita modificar laboratorio existente, el sistema permite cambiar o agregar nuevos experimentos al laboratorio solicitando la información de: nombre de experimento, descripción, variables a evaluar, valores esperados. |
| | 4 | Si falta algún dato el sistema lo solicita. |
| Rendimiento | Paso | Cota de tiempo |
| | 5 | Menos de 1 segundo |
| Frecuencia esperada | 1 vez por cada laboratorio virtual a utilizar. | |

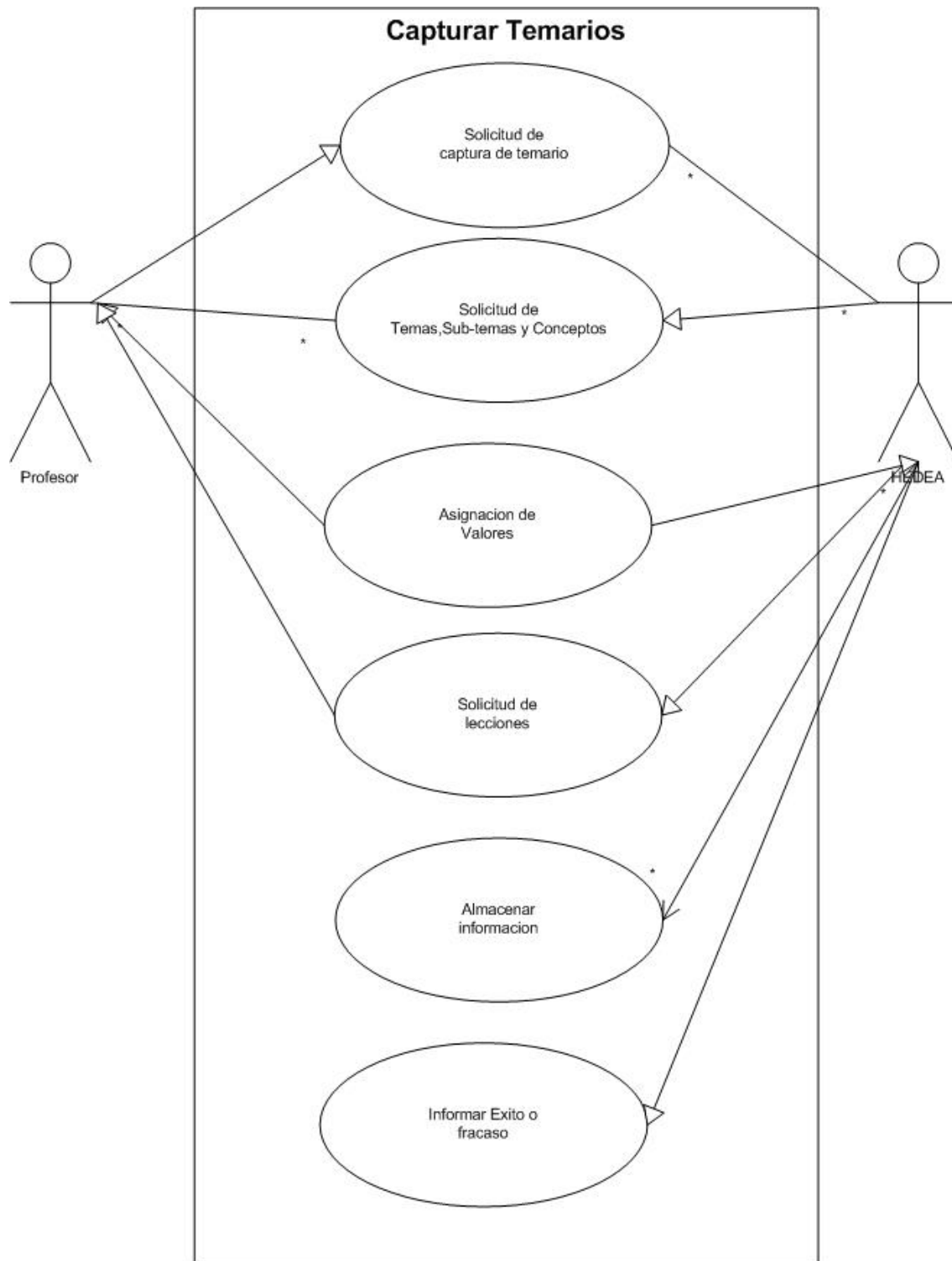
| | |
|--------------------------------|---|
| Prioridad de Desarrollo | Alta |
| Comentarios | La frecuencia será mayor durante las primeras configuraciones debido a modificaciones en los datos. |



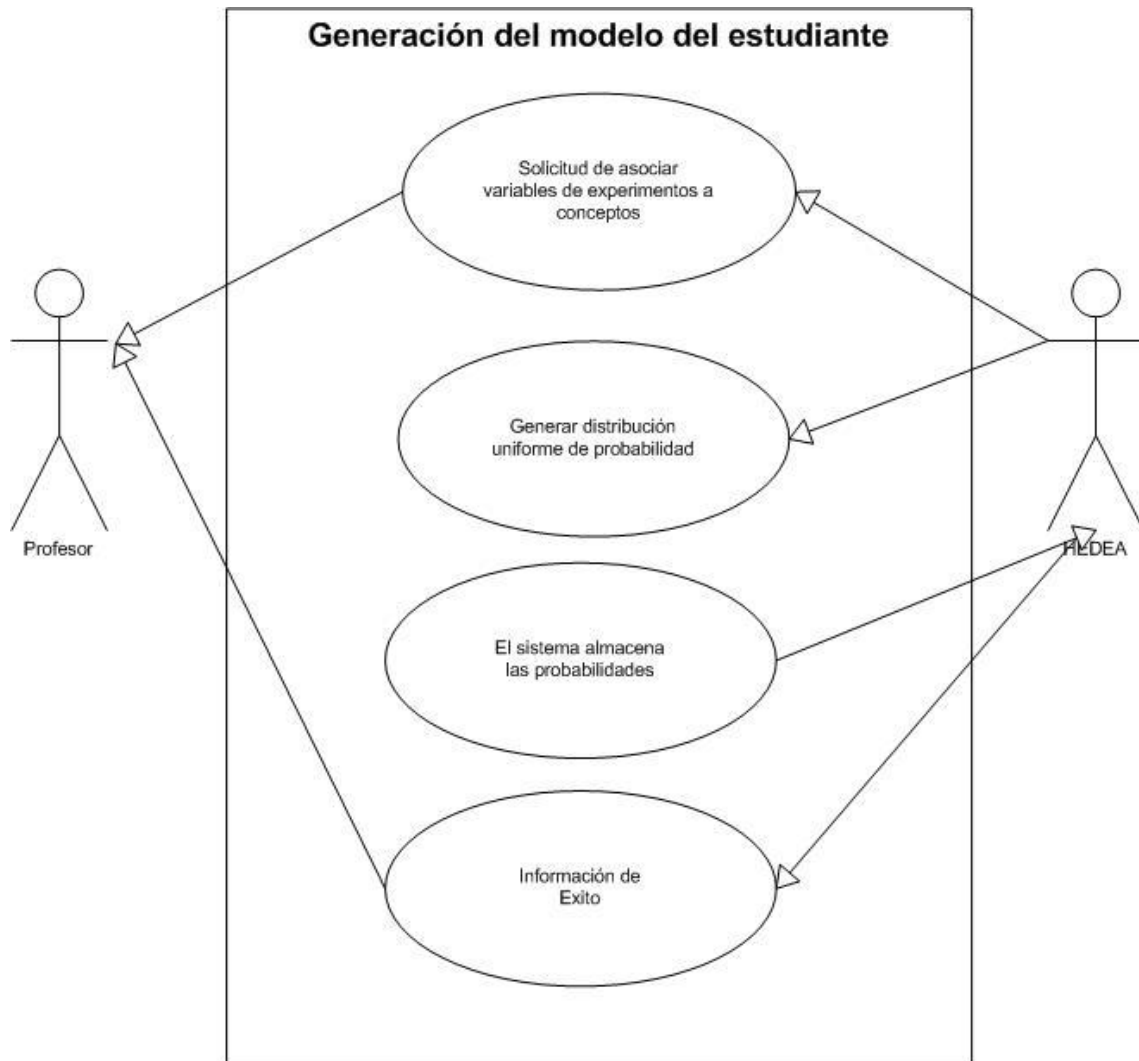
| | | |
|--------------------------------|--|---|
| Nombre | Registro de Experimento | |
| Objetivos asociados | Almacenar y modifica experimentos de un laboratorio existente | |
| Requisitos asociados | Un laboratorio virtual registrado, la información asociada a los experimentos. | |
| Descripción | Se generan nuevas entradas a la lista de experimentos que puede resolver un estudiante en un laboratorio virtual | |
| Precondición | Se ha cargado el laboratorio virtual. | |
| Secuencia Normal | Paso | Acción |
| | 1 | El usuario pide capturar un nuevo experimento |
| | 2 | El sistema solicita los siguientes datos: nombre de experimento, número de variables, variables a evaluar, valor esperado de las variables. descripción de experimentos |
| | 3 | El usuario solicita guardar los cambios |
| | 4 | El Sistema comprueba la consistencia de los datos. |
| | 5 | El sistema almacena los datos proporcionados, |
| | 6 | El sistema informa del éxito del proceso |
| Postcondición | El laboratorio está almacenado con los experimentos configurados. | |
| Excepciones | Paso | Acción |
| | 1 | Si el usuario desea modificar un experimento solo indica la opción de modificar al experimento. |
| | 2 | Si el usuario se solicitó modificar solo se cambia el elemento a modificar |
| | 4 | Si falta algún dato el sistema lo solicita. |
| Rendimiento | Paso | Cota de tiempo |
| | 5 | Menos de 1 segundo |
| Frecuencia esperada | 1 vez por cada experimento a capturar y modificar. | |
| Prioridad de desarrollo | Alta | |
| Comentarios | La frecuencia depende del número de experimentos que contiene el laboratorio virtual. | |



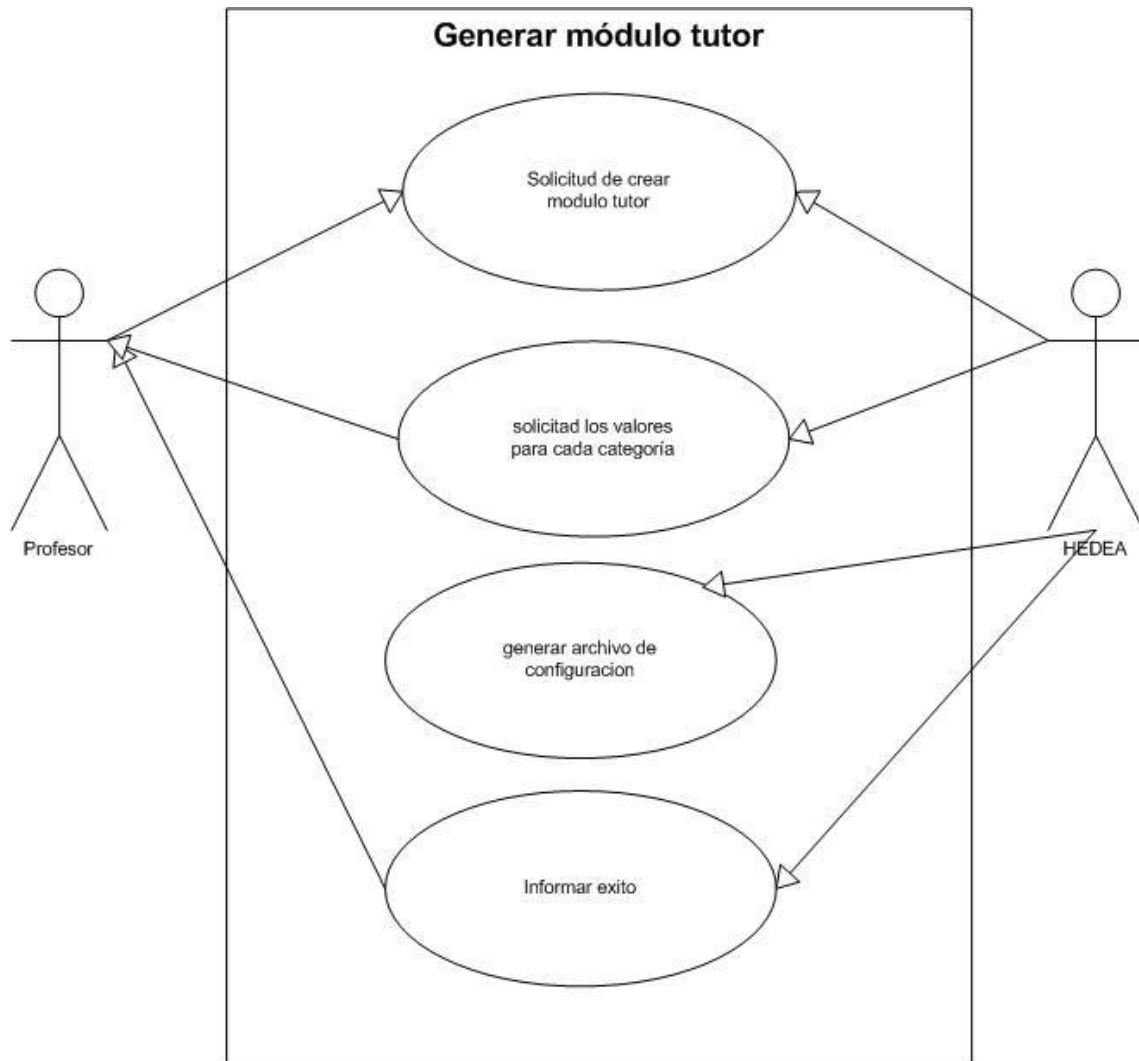
| | | |
|--------------------------------|---|--|
| Nombre | Generar Base de Conocimiento | |
| Objetivos asociados | Almacenar y modificar el listado de temas, subtemas y conceptos del temario asociado al STI. Así como las lecciones que se administraran. | |
| Requisitos asociados | Haber cargado un laboratorio virtual, contar con el temario del curso capturado, haber capturado los experimentos | |
| Descripción | Se captura el temario en forma de temas, subtemas y conceptos. Se asocia a los diversos experimentos mediante el uso de las variables de los experimentos previamente capturados. | |
| Precondición | Se han cargado los experimentos del laboratorio virtual y el temario. | |
| Secuencia Normal | Paso | Acción |
| | 1 | El sistema solicita enlazar cada rama del temario con las variables de los experimentos. |
| | 2 | El usuario solicita generar base de conocimiento. |
| | 3 | El Sistema comprueba la consistencia de los datos. |
| | 4 | El sistema almacena los datos del temario. |
| | 5 | El sistema informa del éxito del proceso. |
| Postcondición | Se cuenta con el temario almacenado | |
| Excepciones | Paso | Acción |
| | 1 | Si no se selecciona una rama del experimento se vuelve a solicitar. |
| | 4 | Si falta algún dato el sistema lo solicita. |
| Rendimiento | Paso | Cota de tiempo |
| | 5 | un segundo |
| Frecuencia esperada | 1 vez por STI a construir | |
| Prioridad de desarrollo | alta | |
| Comentarios | | |



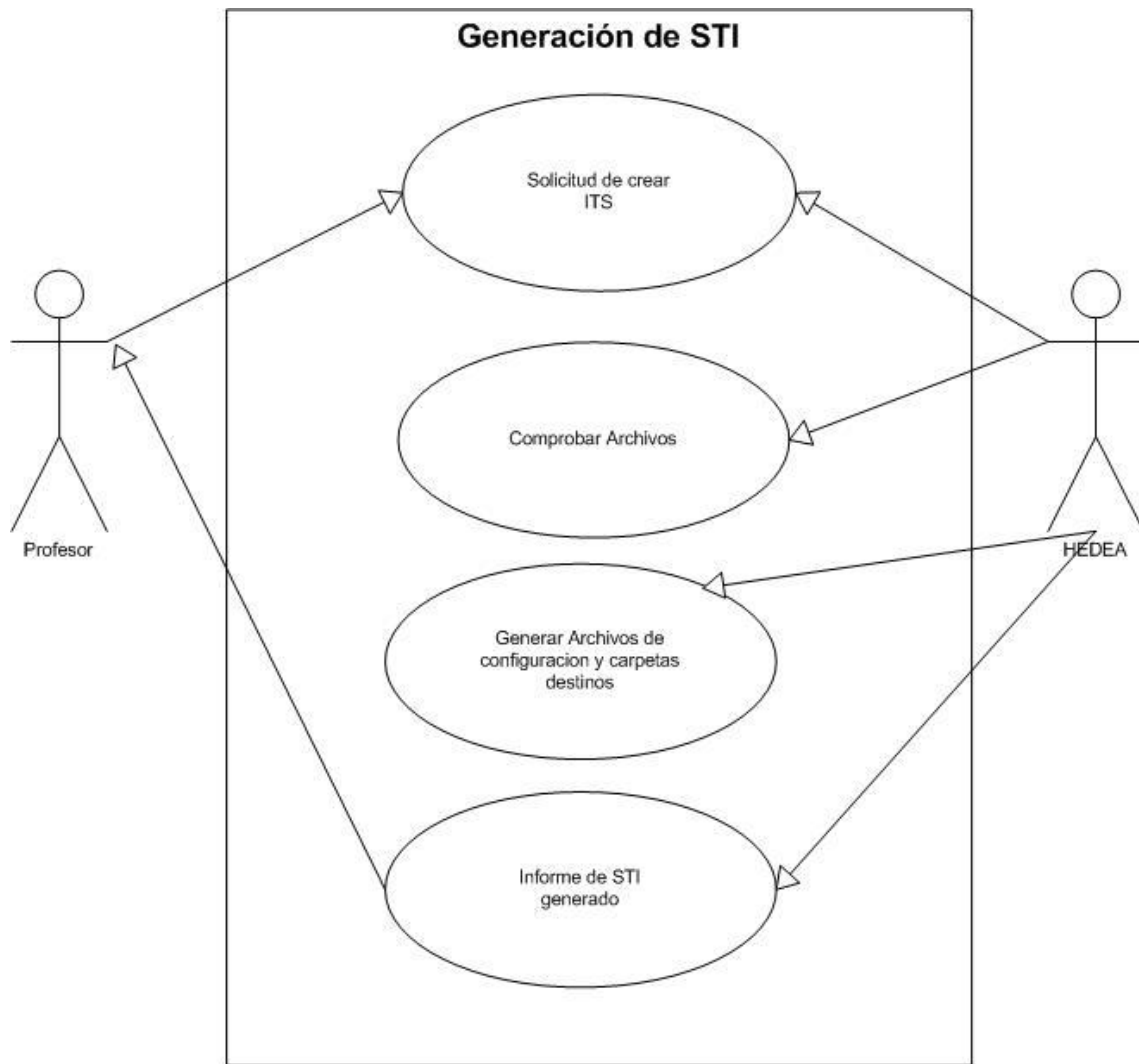
| | | |
|--------------------------------|---|--|
| Nombre | Capturar Temarios | |
| Objetivos asociados | Almacenar la descripción de temas, subtemas y conceptos asociados a un STI. | |
| Requisitos asociados | Un laboratorio virtual configurado, el listado de temario elaborado por el docente | |
| Descripción | Se captura el temario en forma de mapa jerárquico. | |
| Precondición | Se ha configurado el laboratorio virtual. | |
| Secuencia Normal | Paso | Acción |
| | 1 | El usuario solicita cargar temario |
| | 2 | El sistema solicita información de nombre de tema, subtema o concepto. |
| | 3 | El sistema solicita el peso asociado al nodo configurado del temario. |
| | 4 | El sistema solicita la lección relacionada al nodo. |
| | 5 | El sistema almacena los datos proporcionados, |
| | 6 | El sistema informa del éxito o fracaso del proceso. |
| Postcondición | Se tiene almacenado un mapa jerárquico de temas, subtemas y conceptos y está listo para asociar cada rama a las variables de los experimentos. | |
| Excepciones | Paso | Acción |
| | 5 | El sistema devuelve error si faltan datos |
| Rendimiento | Paso | Cota de tiempo |
| | 5 | Menos de 1 segundo |
| Frecuencia esperada | 1 vez por cada STI a generar | |
| Prioridad de desarrollo | Alta | |
| Comentarios | Esta es una de las tareas más fuertes para la generación del STI. El tiempo que requerirá invertir el usuario para completarlo es grande; pero el tiempo del sistema es pequeño . | |



| | | |
|--------------------------------|--|---|
| Nombre | Generación de modelo del estudiante. | |
| Objetivos asociados | Establecer los pesos de las variables asociadas a cada rama del temario. Cargar los pesos de las ramas con una suma igual a uno. | |
| Requisitos asociados | Haber cargado un laboratorio. | |
| Descripción | Se genera el archivo de configuración con los pesos asociados. Se generan las redes bayesianas que representa el conocimiento de un estudiante | |
| Precondición | Haber generado la base de conocimiento | |
| Secuencia Normal | Paso | Acción |
| | 1 | Se despliega el temario con las variables asociadas por separado a cada rama. |
| | 2 | El sistema genera una distribución uniforme de probabilidad para cada variable. |
| | 3 | El Sistema comprueba que la suma de probabilidades sea igual a uno. |
| | 4 | El sistema almacena las probabilidades. |
| | 5 | El sistema informa del éxito o fracaso del proceso |
| Postcondición | Se cuenta con el archivo de base de conocimiento y el modelo del estudiante. | |
| Excepciones | Paso | Acción |
| | 2 | El usuario decide que valores utilizar para cada variable. |
| | 4 | Si falta algún dato se vuelve a Solicitar. |
| Rendimiento | Paso | Cota de tiempo |
| | 5 | 1 segundo |
| Frecuencia esperada | 1 vez por cada STI | |
| Prioridad de desarrollo | Alta | |
| Comentarios | Esta es el componente principal del STI | |



| | | |
|--------------------------------|--|--|
| Nombre | Generar Módulo Tutor | |
| Objetivos asociados | Almacena las reglas de las acciones a tomar según el resultado de los experimentos realizados por un estudiante. | |
| Requisitos asociados | Un laboratorio virtual configurado | |
| Descripción | Se capturan valores de peso para los conceptos que debe alcanzar un estudiante para ser considerado promovido. | |
| Precondición | Se ha cargado el laboratorio virtual, el temario y los experimentos y se ha generado el modelo del estudiante. | |
| Secuencia Normal | Paso | Acción |
| | 1 | El usuario teclea un valor de 0-1 para cada uno de las categorías del estudiante: Principiante, intermedio y avanzado. |
| | 2 | El sistema comprueba valores este en el rango de 0-1. |
| | 3 | El sistema genera los archivos de configuración. |
| Postcondición | Se cuenta con los archivos necesarios para generar el STI. | |
| Excepciones | Paso | Acción |
| | 1 | Solicitar los valores por niveles y por nodos de la base de conocimiento. |
| Rendimiento | Paso | Cota de tiempo |
| | 3 | Menos de 1 segundo |
| Frecuencia esperada | 1 vez por cada STI a generar. | |
| Prioridad de desarrollo | Alta | |
| Comentarios | Estos valores son los que se utilizan para proporcionar lecciones. Por ejemplo si calificación < .85 entonces LECCIÓN. | |



| | | |
|--------------------------------|---|--|
| Nombre | Generar STI | |
| Objetivos asociados | Generar los archivos para que funcione el STI. Colocar el STI en una carpeta. | |
| Requisitos asociados | Un laboratorio virtual configurado, haber generado el modelo del estudiante, la base de conocimiento y el modulo tutor. | |
| Descripción | Se asocian los archivos de configuración al software de un STI genérico desarrollado. | |
| Precondición | Se ha cargado el laboratorio virtual y generado los archivos de configuración. | |
| Secuencia Normal | Paso | Acción |
| | 1 | El usuario solicita crear STI. |
| | 2 | El sistema comprueba los archivos. |
| | 3 | El sistema genera una carpeta con el nombre del laboratorio y las palabras tutor inteligente. |
| | 4 | El sistema hace una copia de los archivos de configuración. |
| | 5 | El sistema crea una carpeta llamada tutor inteligente, la cual contiene el archivo .jar asociado al tutor. |
| | 6 | El sistema informa del éxito del tutor. |
| Postcondición | Se cuenta con los archivos necesarios para generar el STI. | |
| Excepciones | Paso | Acción |
| | 2 | Si falta algún archivo se marca un error y se solicita comprobar. |
| Rendimiento | Paso | Cota de tiempo |
| | 3 | Menos de 2 segundos |
| Frecuencia esperada | 1 vez por STI a generar | |
| Prioridad de desarrollo | Alta | |
| Comentarios | Esta parte requiere de acceso de sistema para poder guardar y generar carpetas. | |

Tabla de acrónimos

STI - Sistema Tutor Inteligente

ITS - Intelligent Tutoring System

PRM -Probabilistic relational model

MPR - Modelo Probabilista Relacional

LV - Laboratorio Virtual

HEDEA - Herramienta de Autoría desarrollada en la Tesis

IA - Inteligencia Artificial

TPC- Tabla de probabilidad condicional

RNA- Redes Neuronales artificiales