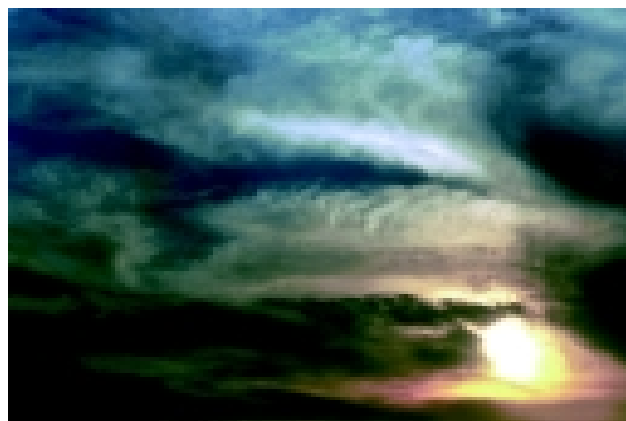


*Se desarrolla un Sistema Tutorial Inteligente para la Instrucción de Técnicas de Sintonización de Sistemas de Control (STISIN).*



# Sistema Tutorial Inteligente para el Autoentrenamiento en Sintonización de Sistemas de Control (SITSIN)

*Guillermo Romero Jiménez y  
María Concepción Pérez  
Ocampo*

## Resumen

**E**n este trabajo se describen el diseño, el desarrollo y la validación de un sistema tutorial inteligente orientado a la instrucción de técnicas de sintonización de sistemas de control. Este sistema está basado en sistemas desarrollados anteriormente en la Unidad de de Simulación del Instituto de Investigaciones Eléctricas (IIE) [Olayo, Roberto, 1998; Gutiérrez Eduardo, 1998; Buenabad, María de los Ángeles, 1996].

El sistema diseñado, como en [Olayo, Roberto, 1998 y Gutiérrez, Eduardo, 1998] cuenta con cuatro módulos: de conocimiento, del modelo del estudiante, de tutor y de interfaz (figura 1), característica principal que permite ubicar a este sistema en el contexto de los sistemas tutoriales inteligentes [Petrushin, U.A., 1995;

Itoh, Yukihiro *et al.*]. En este sistema en particular sólo se modificó el módulo de conocimiento, pues se aprovecha la estructura existente en Olayo (1998) para incorporar un nuevo dominio de aplicación: el de las técnicas de sintonización de sistemas de control. El sistema mantiene la característica de que también puede utilizarse como un sistema de consulta.

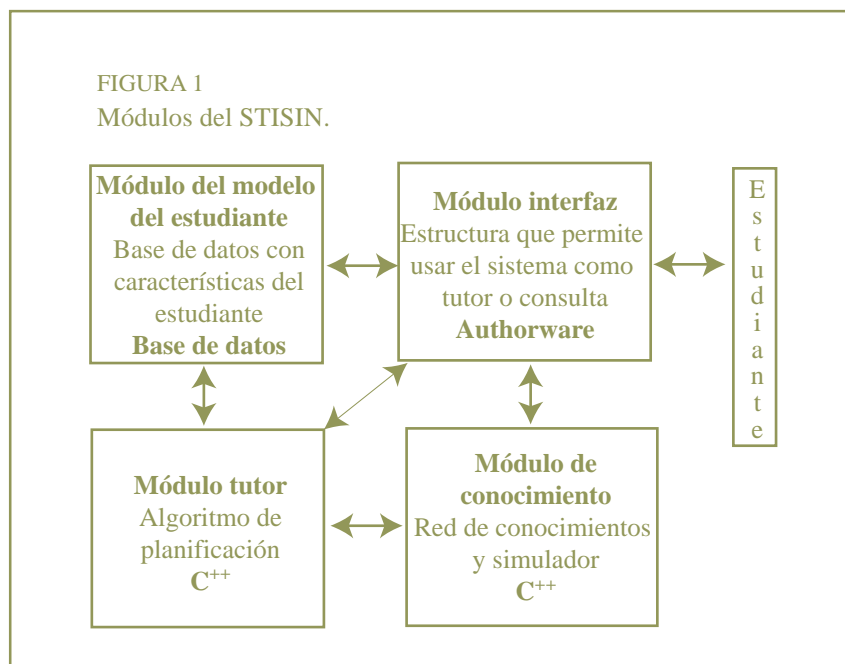
Además del diseño y validación del sistema tutorial, al seguir la metodología de elaboración del sistema se evaluó el grado de “generalidad” del sistema desarrollado por Olayo (1998), tomando en cuenta la evaluación y cuantificación de métricas que la ingeniería de *software* propone.

## Introducción

En la unidad de Simulación se tiene una línea de Investigación para el desarrollo de sistemas tutoriales inteligentes enfocados al entrenamiento de operadores de centrales de generación de energía eléctrica.

Dentro de esta línea se desarrolla este proyecto con el fin obtener un Sistema Tutorial Inteligente para la Instrucción de Técnicas de Sintonización de Sistemas de Control (STISIN).

Este sistema se basa en otros sistemas desarrollados anteriormente en la Unidad de Simulación. Del sistema de



Buenabad (1996), se observó la estructura para que el STISIN pudiera funcionar como sistema de consulta además de tutor. Se reutilizó el material multimedia (voz) que Buenabad diseñó para las lecciones.

Gutiérrez (1998) diseñó un sistema tutorial inteligente enfocado al sistema eléctrico de energía. La estructura de los módulos de su sistema fue básico para el diseño del STISIN.

Finalmente, Olayo (1998) diseñó y construyó un sistema tutorial inteligente enfocado a instruir operadores de reguladores automáticos de tensión (SITRAT), por lo que se constituyó en la estructura base (sólo se modificó el módulo de conocimiento) del STISIN.

A continuación se describe cada uno de los módulos del STISIN y las técnicas utilizadas para su implementación.

#### **Módulo de conocimiento**

La función del módulo de conocimiento es la de representar a un experto, en este caso, en técnicas de

sintonización de sistemas de control.

El módulo de conocimiento consiste en una red de conceptos como la describe Petrushin (1995) y otros autores. Cada concepto se relaciona con los demás de tres maneras: general, específica y requisito. Estas relaciones son utilizadas por el módulo tutor para planificar la secuencia de lecciones que mostrará al estudiante. La red se muestra en la figura 2.

*El STISIN se concibió bajo la premisa de desarrollar un sistema que se acercara lo más posible a una forma de enseñanza de acuerdo con la teoría del campo cognoscitivo [Gutiérrez, Eduardo, 1998]. Esto es, que el estudiante pudiera interactuar con el sistema sin que éste se impusiera siempre en el proceso de enseñanza-aprendizaje.*

Cada lección tiene además un número de créditos que se le asigna

dependiendo de la extensión y dificultad de la misma.

El material de instrucción, es decir, las pantallas de las lecciones, residen en el módulo interfaz. Cada nodo de la red de conocimientos tiene un atributo que permite relacionar el nodo con el material de instrucción respectivo.

#### **Módulo tutor**

El STISIN se concibió bajo la premisa de desarrollar un sistema que se acercara lo más posible a una forma de enseñanza de acuerdo con la teoría del campo cognoscitivo [Gutiérrez, Eduardo, 1998]. Esto es, que el estudiante pudiera interactuar con el sistema sin que éste se impusiera siempre en el proceso de enseñanza-aprendizaje. Por ello se diseñó una estrategia de enseñanza en la que el estudiante escoge la estructura general del curso de tres alternativas:

1. Primero los conceptos básicos, después sintonización, y, finalmente, los procedimientos de sintonización.
2. Primero sintonización, después los conceptos básicos y, finalmente, los procedimientos de sintonización.
3. Primero los procedimientos de sintonización, después sintonización y, finalmente, los conceptos básicos.

El módulo tutor planifica las lecciones que mostrará cada sesión. En una sesión, al estudiante se le presenta un menú de lecciones, él puede escoger el orden en que las ve y las veces que quiere verlas. Cuando considera que ha aprendido el material, solicita un examen para diagnosticar su nivel de comprensión de las lecciones.

El algoritmo de planificación consiste en ir presentando las lecciones conforme a la estructura que el estudiante eligió inicialmente. Cada lección tiene un cierto número de créditos y en el módulo del modelo del

estudiante cada estudiante tiene un máximo de créditos que se le puede presentar en una sesión. Así que la limitante en cuanto a la cantidad de lecciones es precisamente el máximo número de créditos. El estudiante puede variar este número al terminar cada sesión.

Cuando una lección no es comprendida, se procede a presentar las lecciones que se relacionan con ella para que el estudiante pueda entenderla. Para ello, el módulo tutor utiliza la red de conocimientos que reside en el módulo de conocimiento (figura 1).

Primero se muestran lecciones más generales. En principio se supone que es más fácil aprender cuando las cosas se ven de lo general a lo particular; sin embargo, mucha gente está acostumbrada a aprender de lo específico a lo general. Así pues, si después de mostrar las lecciones más generales el estudiante sigue sin comprender, se procede a mostrar las lecciones más específicas.

Si aún así el estudiante no comprendiera esa lección en particular, el sistema se la salta e intenta seguir con las demás lecciones. En este caso se denomina a la lección "no comprendida y sin recursos". Al final se vuelven a mostrar las lecciones que no fueron comprendidas.

La arquitectura de enseñanza utilizada en la parte de prácticas es la de simulación [Pressman, Roger, 1996]. En el resto del sistema, las lecciones se muestran como páginas diseñadas con multimedia, es decir que contienen sonidos, gráficas en movimiento, etcétera, pero podría utilizarse otra de las arquitecturas descritas por Schank (1990).

### ***Módulo del modelo del estudiante***

La modelación del estudiante o el usuario es un tema de frontera en el cual se han hecho muchos estudios

[Petrushin, V.A., 1995; Itoh, Yukihiro *et al.*, Shank, Roger, 1990]. Petrushin (1995) hace una descripción de diferentes métodos de modelar al estudiante.

En el caso del STISIN, en el módulo del modelo del estudiante se utilizó el método de sobreposición en la red de conocimientos; se guarda el nombre de usuario y contraseña de cada estudiante así como la situación del estudiante en cada lección. Además, cada estudiante tiene un límite de créditos que se le puede asignar en una sesión.

La situación del estudiante en cada lección se describe con los siguientes valores:

- 0 No se ha visto la lección.
- 1 No se ha entendido la lección.
- 2 No se ha entendido la lección y ya no se tienen recursos.
- 3 Lección comprendida.

*La arquitectura de enseñanza utilizada en la parte de prácticas es la de simulación [Pressman Roger, 1996]. En el resto del sistema, las lecciones se muestran como páginas diseñadas con multimedia, es decir que contienen sonidos, gráficas en movimiento, etcétera. Pero podría utilizarse otra de las arquitecturas descritas por Schank [(1990)].*

El módulo del modelo del estudiante se guarda en una base de datos. Cada vez que el sistema inicia se carga esa base de datos y se insertan los valores de cada lección en el nodo correspondiente de la red de conocimientos (figura 2).

### ***Módulo interfaz***

Finalmente, el módulo interfaz consiste

en una serie de pantallas arregladas para facilitar la navegación a través de ellas como si fuera una página Web. De hecho, el sistema puede funcionar de dos maneras: modo consulta y modo tutor. El modo consulta permite ver los temas a partir de una serie de menús diseñadas por hipervínculos (esta estructura ya la tiene el sistema diseñado por Olayo). El modo tutor permite que la interfaz interactúe con el módulo tutor, éste le indica al módulo interfaz las lecciones que se van a mostrar en la sesión para que se le presente el menú al estudiante (figura 1).

El menú permite la navegación por hipervínculos a las pantallas respectivas de cada lección. Tras ver el material de instrucción de cada lección, el estudiante regresa al menú donde puede escoger otra lección o, si se siente preparado, para realizar el examen de la sesión.

El examen contiene preguntas de todas las lecciones que se vieron en esa sesión. Después de cada pregunta, el módulo interfaz se comunica con el módulo tutor para informarle de los resultados de cada pregunta. El módulo tutor, a su vez, hace las modificaciones correspondientes al módulo del modelo del estudiante (figura 1).

### **Programación**

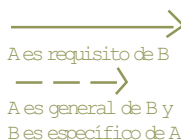
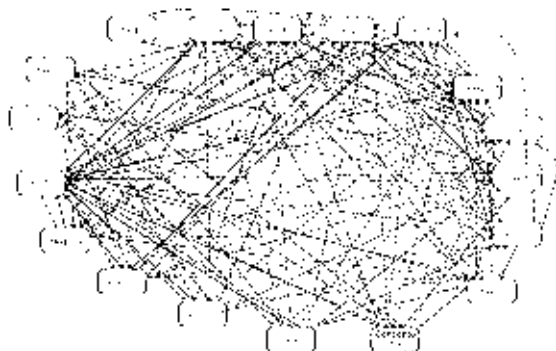
La programación del módulo de conocimiento se hizo en C++ con el compilador de Microsoft. Se utilizó programación orientada a objetos. Cada nodo de la red es un objeto de la clase CLeccion que tiene atributos para el número de créditos, el valor guardado en el módulo del modelo del estudiante, el número de la pantalla correspondiente en el Módulo Interfaz y las listas de apuntadores a las lecciones específicas, generales y de requisito.

### **Evaluación de la generalidad del STI de Olayo**

Para poder estimar la generalidad del

FIGURA 2

Red de conocimientos del módulo de conocimiento del STISIN.



- Concepto 0 Conceptos generales.  
 Concepto 1 Parámetros de control.  
 Concepto 2 Diagrama de bloque del controlador PI.  
 Concepto 3 Diagrama de bloque de PID (implementación paralela). Derivative on the error.  
 Concepto 4 Diagrama de bloque de PID (implementación paralela). Derivative on the measurement.  
 Concepto 5 Diagrama de bloques de PID (implementación de series).  
 Concepto 6 Sintonización de sistemas de control.  
 Concepto 7 Ajuste de la ganancia del controlador.  
 Concepto 8 Ajuste de la integral.  
 Concepto 9 Ajuste de la derivativa.  
 Concepto 10 Reglas de dedo, general, temperatura, nivel, flujo, presión.  
 Concepto 11 Ziegler Nichols básico.  
 Concepto 12 Modelo basado en aproximación.  
 Concepto 13 Cohen Coon (1953).  
 Concepto 14 Time integral performance criteria.  
 Concepto 15 Autosintonización y los tipos de autosintonización.  
 Concepto 16 Exact.  
 Concepto 17 Relay feedback auto-tuning (fisher control).

sistema tutorial inteligente de Olayo (1998), se realizaron pruebas de evaluación a través de la cuantificación por métricas [Pressman, Roger, 1996].

El proceso de evaluación se llevó a cabo durante el desarrollo del nuevo dominio de aplicación siguiendo la siguiente temática:

- a) Desglose de conceptos del dominio de aplicación.
- b) Conformación de la

arquitectura del modelo del estudiante.

c) Diseño de la agenda de lecciones.

d) Rediseño de interfaz.

e) Generación de lecciones.

f) Control de flujo a través de macromedia authorware.

Los factores y las características que se percibieron se dividen de la siguiente forma:

Características operativas

- Corrección.
- Fiabilidad.
- Eficiencia.
- Integridad.
- Facilidad de uso.

Capacidad para soportar cambios

- Facilidad de mantenimiento.
- Flexibilidad.
- Facilidad de prueba.

Adaptabilidad a nuevos cambios

- Portabilidad.
- Reusabilidad.
- Interoperabilidad.

Para identificar el factor de calidad en líneas de código tenemos estos factores:

- Facilidad de auditoría.
- Exactitud.
- Completitud.
- Complejidad.
- Estandarización en los datos.
- Eficiencia en la ejecución.
- Facilidad de exportación.
- Generalidad.
- Independencia de *hardware*.

*El STISIN es un prototipo que cumple con especificaciones importantes para un sistema tutorial inteligente. El sistema funciona como sistema de consulta y como sistema tutorial.*

Con el fin de poder realizar la cuantificación parametrizada se prorrataron las métricas de acuerdo con factores y características que se muestran en el cuadro 1.

La cuantificación parametrizada de métricas y el factor de calidad de las líneas de código (FCLC) ayudan a calcular el (FcR) factor de calidad del *software* durante el proceso de reutilización.

$$FcR = \text{Sum (métricas de calidad de$$

CUADRO 1

<div style="border: 1px solid black; padding: 5px; width: fit-content; margin-bottom: 5px;">Dominio de aplicación</div> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin-bottom: 5px;">Líneas de código</div> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin-bottom: 5px;">Factor de calidad</div>	Métrica de calidad de <i>software</i>										
	Corrección Fiabilidad Eficiencia Integridad Factibilidad de mantenimiento Flexibilidad Facilidad de prueba Reusabilidad Facilidad de interoperación Facilidad de uso										
	Facilidad de auditoría								0.5		0.5
	Exactitud			0.5							0.5
	Completitud								0.75		0.75
	Complejidad						0.3				0.25
	Estandarización en los datos				0.9					0.75	1.65
	Eficiencia en la ejecución										
	Facilidad de expansión	0.5		0.25	0.25						1
	Generalidad	0.75	0.75	0.1					0.1		1.7
	Independencia del <i>hardware</i>			0.5						0.75	1.25
	Modularidad		0.1	0.1	0.1		0.1	0.1			0.5
	Facilidad de operación				0.5			0.5			1
	Seguridad		0.25		0.1	0.25					0.85
	Valor S	0.1	0.1						0.1	0.1	0.4
	0.85	1.7	0.6	1.05	1.9	0.4	0.6	0.6	1.6	1.1	10.35

S valor de la métrica

100% completo

50% moderado

FcR=Sum (S)

FcR=10.35\* 768/100

LC líneas de código aprox. = 768

90% bast

75% considerable

25% poco

10% mínimo 0% nulo

x LC/100

79.488

$software\ FCLc ) \times Lc /100$

Donde Sum es la suma cuantificada de las métricas FCLc.

Lc es el número estimado de líneas de código en general.

100% grado máximo de calidad

de esta forma tenemos que para el sistema de Olayo:

FcR= 10.35 x 768 /100

siendo 79.48% FcR

**Conclusiones**

El STISIN es un prototipo que cumple con especificaciones importantes para un sistema tutorial inteligente. El sistema funciona como sistema de consulta y como sistema tutorial.

La estructura modular del STI, permite que su adaptación a otros dominios de aplicación sea más fácil, como en este caso, pues únicamente se

cambió la especialidad de RAT por la de técnicas de sintonización de sistemas de control.

La generalidad del STI de Olayo en una evaluación cualitativa se ubica en 80% de reutilización de la estructura para poder desarrollar otro STI diferente, al menos en lo que al dominio de aplicación se refiere. Se siguió utilizando la misma estrategia de enseñanza. El cambio de la estrategia de enseñanza, como lo asienta Olayo

en la parte de conclusiones de su trabajo, también se facilitarían, gracias a la estructura modular que presenta su diseño; es decir, se podría seguir utilizando la misma estructura de la red de conocimientos.

Como en el caso del STIRAT, el STISIN es un intento por tener un sistema tutorial que permita que el estudiante intervenga en las decisiones con respecto a la forma de lograr el objetivo. Esto compromete al estudiante y lo motiva a aprender.

El STISIN forma parte de la línea de investigación de sistemas tutoriales de la Unidad de Simulación y, como tal, se le han integrado los avances que en su momento tuvieron otros prototipos. Igualmente, se prevé que vendrán nuevos prototipos para mejorar el STISIN. Sin embargo, tanto el STISIN, como el STIRAT, constituyen un avance importante hacia un sistema tutorial inteligente enfocado a la capacitación.

### Agradecimientos

Los autores agradecen la ayuda y las observaciones al proyecto, al doctor José Montoya Murillo y al ingeniero Roberto Olayo Valles.

### Referencias

Buenabad, A. A., *Desarrollo de un sistema de capacitación dirigido a operadores de centrales generadoras de energía utilizando técnicas de realidad virtual y multimedia*, reporte preparado para el Programa de Adiestramiento en Investigación Tecnológica, Instituto de Investigaciones Eléctricas, 1996, México.

Gutiérrez, E.F., *Desarrollo, implantación y validación de un algoritmo de toma de decisiones basado en un modelo de estudiante para definir estrategias de enseñanza en sistemas tutoriales*, tesis de licenciatura, 1998, México.

Itoh, Yukihiro, Takahashi, Isamu, Konishi y Tatsuhiro, *On recognition of student's actions in ILE*, Faculty of Engineering, Shizuoka University, Japón.

Olayo, R., *"Sistema tutorial inteligente para autoentrenamiento en RAT"*, reporte preparado para el Programa de

Adiestramiento en Investigación Tecnológica, Instituto de Investigaciones Eléctricas, 1998, México.

Petrushin, V. A., "Intelligent tutoring systems: architecture and methods of implementation. A survey", en *Journal of Computer and Systems Sciences International*, 33(1), 1995, pp. 117-139.

Pressman, R.S., *Ingeniería de software. Un enfoque práctico*, McGraw Hill, 1996, pp. 43-73.

Shank, R.C., *Teaching architectures*, The Institute for the Learning Sciences, Northwestern University, 1990.

### GUILLERMO ROMERO JIMÉNEZ



*Ingeniero industrial en electrónica (1981) y maestro en ingeniería eléctrica (1984) por el Instituto Tecnológico de La Laguna. Obtuvo el doctorado en control automático en el Instituto Nacional Politécnico de Grenoble, Francia, en 1996. En 1984 ingresó al IIE. Actualmente es investigador de la Unidad de Simulación en donde colabora con el grupo de modelos de sistemas eléctricos y de control. Ha sido catedrático del Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM), campus Morelos, y profesor auxiliar en el Centro de Graduados del Tecnológico de la Laguna. Es autor de diversos artículos sobre el modelado en tiempo real de sistemas de control y sobre el modelado semicualitativo de procesos.*

### MARIA CONCEPCION PÉREZ OCAMPO

*Licenciada en informática (1996) por el Instituto Tecnológico de Puebla, participó en el V y VI Verano de la Investigación Científica con los proyectos de "Sistemas Expertos" y "Sistemas Operativos para Redes de Computadoras" y en el Congreso de Ciencia Joven con el Tema "GSERA, Generador de Sistemas Expertos"; realizó una estancia de Adiestramiento en Investigación Tecnológica en el IIE (1998) en la Unidad de Resultados de Simulación.*