

MODELO DEL INSTRUCTOR EXPLICITO EN UN SISTEMA TUTORIAL INTELIGENTE

Constanza R. Huapaya¹, Graciela M. Arona¹ y Francisco A. Lizarralde¹

Abstract — Los Sistemas Tutoriales Inteligentes (STI) son uno de los resultados más importantes producidos por la Inteligencia Artificial en Educación. Su creación es de alta complejidad, tanto pedagógica como computacional. Las herramientas de autoría son una alternativa eficaz para su implementación. El objetivo de este trabajo es definir un modelo del instructor, a fin de acrecentar la eficacia del sistema. El autor, experto en teorías cognitivas y diseño instruccional crea un STI. Este producto puede ser utilizado por instructores de distintos niveles de experticia. Son ellos quienes efectivamente lo usan en un aula con un grupo de estudiantes en diferentes sesiones. Se ha estudiado la interacción sistema-instructor para abstraer las características más relevantes. En especial, un instructor de temas de ingeniería podría ser asistido por un STI a fin de establecer los objetivos instruccionales de una sesión, evaluar el estado de conocimiento de sus alumnos, y valorar el éxito de su gestión.

Palabras clave — Inteligencia Artificial en Educación, Modelo del Instructor, Sistemas Tutoriales Inteligentes.

INTRODUCCION

La enseñanza puede ser entendida como un tipo especial de comunicación humana. Los maestros construyen la riqueza de esa comunicación a partir de su práctica cotidiana. A fin de mejorar la enseñanza, la comunidad de Inteligencia Artificial en Educación investiga teorías, metodologías y técnicas con la finalidad de construir herramientas que asistan al maestro en su actividad diaria. El principal foco de interés actual de esta investigación se basa en la imitación de la actividad adaptativa y comunicacional de un maestro humano.

SISTEMAS TUTORIALES INTELIGENTES

Para alcanzar el objetivo enunciado, se construyen ambientes de enseñanza, como los Sistemas Tutoriales Inteligentes [1]-[2], cuyo objetivo es transformarse en "maestros computacionales". El término "inteligente" en STI se refiere a la habilidad del sistema para saber qué enseñar, cuándo enseñarlo, y cómo enseñarlo.

Un STI debe mostrar capacidad de entendimiento, aprendizaje, razonamiento y resolución de problemas. Además, debe ser capaz de identificar las fortalezas y debilidades de un aprendiz, y establecer un plan

instruccional para adaptarse a cada estudiante real. Asimismo, manipula información sobre el proceso de aprendizaje de cada estudiante (como estilo de aprendizaje preferido y conceptos erróneos adquiridos). La instrucción debe ajustarse a las necesidades del estudiante, i.e., tiene que adaptarse a cada uno de ellos [3]. La capacidad de adaptarse diferencia a los STIs de las herramientas tradicionales como los CAIs (Computer Aided Instruction).

No obstante, la construcción de STIs reales todavía es ardua y costosa [4]. Es necesario profundizar el análisis del conocimiento experto involucrado en un STI a fin de ser capturado en un sistema computacional.

Durante los últimos veinte años se ha trabajado sobre la siguiente arquitectura básica (ver figura 1): un STI posee componentes para modelizar el conocimiento a enseñar (plasmado en el **modelo del dominio**), el seguimiento de la actividad del estudiante (o **modelo del estudiante**), el conocimiento pedagógico (que será plasmado en un **plan instruccional**), y la **interfaz** de comunicación (con el estudiante y con el profesor).

Una vez elegido el tópico que un estudiante determinado debe aprender, comienza una sesión tutorial donde el STI entrega el material instruccional siguiendo los lineamientos que el modelo pedagógico y su instructor verdadero decidieron para él (ella). Asimismo el plan instruccional considera las características personales de la

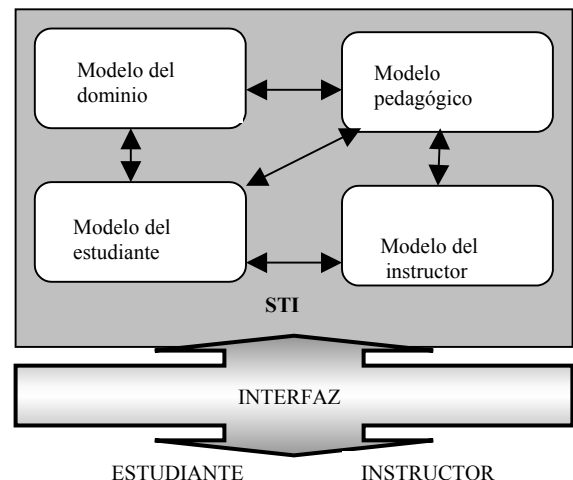


FIGURA. 1
ARQUITECTURA BASICA DE UN SISTEMA TUTORIAL INTELIGENTE.

¹ Constanza R. Huapaya, Graciela M. Arona y Francisco A. Lizarralde, Facultad de Ingeniería, Universidad Nacional de Mar del Plata, Juan B. Justo 4302, 7600 Mar del Plata, Buenos Aires, Argentina, huapaya@fi.mdp.edu.ar, graron@fi.mdp.edu.ar, flizarra@fi.mdp.edu.ar

estudiante a fin de adaptar apropiadamente la enseñanza del tópico a su perfil.

Antes de presentar el problema que nos preocupa, i.e., el modelo del instructor, mostramos someramente los modelos del dominio y del estudiante.

MODELO DEL DOMINIO

Un sistema tutorial es creado para enseñar un tema en particular, por lo tanto debe tener suficiente conocimiento del tema y además de la forma en que se enseña. Además este conocimiento debe estar estructurado de manera tal que el sistema pueda ser flexible. Los factores que hay que tener en cuenta son: cómo organizar los ítems; cómo relacionarlos con el avance del aprendizaje del alumno; cómo relacionarlos con el material de enseñanza.

La ontología propuesta en el ámbito conceptual se divide en dos estructuras la descripción del dominio y la organización del material de enseñanza.

La metodología de representación mediante mapas conceptuales, ha sido utilizada con éxito para el diseño instruccional [5]. Como descripción del dominio los nodos representan los tópicos a enseñar y los arcos las relaciones de precedencia, inclusión o pertenencia que se desee establecer entre tópicos. Esta estructura permite al sistema seleccionar el nodo a enseñar considerando el cumplimiento de los tópicos que lo preceden y según la estrategia de enseñanza que se quiera aplicar (de lo general a lo particular o viceversa)[6]. La enseñanza de distintos tipos de contenidos requiere distintas metodologías [7] [8], el sistema necesita reconocer estas diferencias. Se han establecido tres tipos de nodos: hecho, concepto y procedimiento [9], [10], [11]. La granularidad con que se defina este mapa conceptual esta directamente relacionado con la granularidad con que el sistema enseñará.

Relacionado con cada nodo del mapa de tópicos, el material de enseñanza se clasifica por el tipo de instrucción que brinda (explicación, ejemplo, ejercicio, test, ayuda) y por características propias (importancia, dificultad, tiempo). Es presentado al estudiante considerando su perfil y actuaciones anteriores representados en el modelo del estudiante [6].

MODELO DEL ESTUDIANTE

El modelo del estudiante es una descripción declarativa de las características de las actividades de aprendizaje de los estudiantes [12]. Por ejemplo, el sistema puede usar un *modelo overlay* básico en el cual se registra el entendimiento del sistema sobre el conocimiento del estudiante en un tópico determinado. En este modelo, la respuesta del estudiante es analizada para evaluarla convenientemente.

Nuestro modelo del estudiante trata la caracterización del estudiante (datos personales y perfil del estudiante), conocimiento (almacenamiento de los temas ya conocidos, objetivos instruccionales adquiridos, equivocaciones

cometidas y material didáctico usado) e historia (información de la última sesión). El modelo se ha organizado en dos módulos principales: *característica del aprendiz* y *sesión del estudiante*, el primero se refiere a las características personales y la evolución del conocimiento del estudiante a largo plazo, el segundo considera a la sesión globalmente para terminar analizando el “camino” recorrido y el nivel de rendimiento alcanzado por el alumno. Los resultados de ese análisis actualizan su perfil y su evolución personal (ver figura 2).

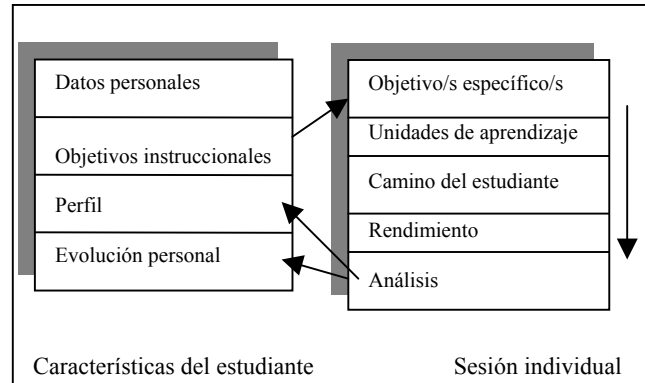


FIGURA 2
MODELO DEL ESTUDIANTE

El comportamiento del STI depende fuertemente del modelo del estudiante, sin embargo éste solo puede representar una serie de fotografías instantáneas de la actividad del alumno en determinados momentos. La consecuencia inmediata es el registro incompleto del proceso de aprendizaje individual de cada estudiante y por lo tanto solo disponer de modelos del estudiante inexactos.

EL PROBLEMA A TRATAR

La competencia en la enseñanza de un sistema tutor computacional es todavía un aspecto importante que ha tenido un desarrollo moderado. Una crítica centrada en este tema la dio Ohlsson [13] en los 80s donde pone de manifiesto el *rango limitado* y la poca *adaptabilidad* de las acciones tutoras en comparación con la riqueza de las tácticas de los maestros humanos.

La motivación de la crítica parece estar fundamentada en la pobre habilidad comunicacional de un STI. Esta es limitada, principalmente debida a los propios problemas de los sistemas computacionales. Actualmente existen pocos ejemplos de tutores computacionales que se "expresan" usando diálogos y gestos (como los usados por agentes pedagógicos animados) [14].

Asimismo, la tendencia actual sostiene que la manifestación de un sistema tutor no se debe reducir a la presentación de explicaciones o indicar cual es la respuesta correcta. Tanto el instructor como los estudiantes deberían co-construir las explicaciones y juntos seguir las mismas

líneas de razonamiento. Los STIs deben ser lo suficientemente flexibles y adaptativos para manipular dificultades como *fallas* (el tutor no entiende una respuesta del estudiante, el estudiante puede responder a una pregunta de manera inesperada o la táctica tutorial no trabaja apropiadamente), *revisión de tácticas* (p.e. el comportamiento de los estudiantes indican que el tutor puede saltar pasos en la explicación) y tratar de manipular la *ambigüedad* que presentan las respuestas del estudiante en su significado.

Por otro lado, el sistema tutorial debe "interpretar" la respuesta del estudiante. Esta captación de la actividad del alumno opera con información incompleta y muy a menudo altamente incierta (inexactitud del modelo del estudiante). En consecuencia los tutores deberían poseer métodos fuertes para manipular la incertidumbre. Estos métodos, idealmente deberían ser óptimos, i.e., usando las observaciones sobre los estudiantes, los métodos tendrían que garantizar la ejecución óptima de las acciones tutoriales.

A fin de tratar, en parte, la problemática presentada, centrada en la relación comunicacional del STI con sus usuarios, se propone un modelo del instructor explícito.

MODELO DEL INSTRUCTOR

La enseñanza humano-a-humano raramente incorpora estrategias tutoriales ideales. Algunos de estos mecanismos pueden ser simulados en computadora, (como el método Socrático) mientras que otros, como aquellos donde intervienen el afecto o emoción, son muy difíciles de lograrlo considerando el actual desarrollo de la tecnología computacional.

La actividad pedagógica de un STI, hasta ahora, se ha traducido en colecciones de reglas que solo trabaja relativamente bien en la práctica. En parte, esto se debe a que no existe una enciclopedia científica donde consultar las mejores heurísticas tutoriales. En su lugar, se debe adoptar diferentes métodos de enseñanza porque diferentes estudiantes utilizan diferentes estilos de aprendizaje. La riqueza del conocimiento del instructor real se debe tomar como una de las fuentes más importantes donde extraer las líneas principales del conocimiento pedagógico de un STI.

Un autor, el creador de un STI específico en un dominio particular, suele encontrar varias dificultades cuando desarrolla un STI (por ejemplo, qué criterio usar para considerar que un tópico determinado ha sido aprendido o cuál es la granularidad apropiada de las estrategias tutoriales), en consecuencia puede dar información inconsistente a la herramienta que se traducirá en un comportamiento problemático del STI producido. En este punto se debe considerar como crucial el nivel de experticia del autor. Sin embargo, el problema principal aparece cuando un instructor debe interactuar con el STI producido por algún autor. Si el instructor real no lo acepta, no lo usará apropiadamente en una clase real. Esto nos lleva al problema de la adaptación las estrategias tutoriales

incrustadas en la planificación instruccional y su asociación con las tácticas pedagógicas del instructor, i.e., como adaptar esas estrategias a los estudiantes reales.

En este punto debemos considerar *como* construir un STI. Esta tarea involucra el esfuerzo conjunto de expertos en ciencias cognitivas, tecnología instruccional y desarrollo de sistemas computacionales. Tanto el autor como el instructor (eventualmente puede ser la misma persona) generalmente no son expertos en informática. Por lo tanto es deseable disponer de sistemas computacionales que asistan amigablemente a los creadores del tutor computacional. Precisamente, las *herramientas de autoría* constituyen un tipo de sistema de adquisición de conocimiento a partir de autores/maestros/instructores/ entrenadores. A fin de asistir al autor, el sistema de autoría debería poseer conocimiento basado en diversas teorías generales, como las de diseño instruccional y psicología cognitiva.

Las herramientas de autoría actuales no son inteligentes. No poseen modelos de representación declarativa de su conocimiento sobre como construir STIs. Un ejemplo de los sistemas comerciales es Authoware v 4.0 de Macromedia. La inteligencia de las herramientas se manifiesta a través de la inspección dinámica de las descripciones declarativas de su conocimiento a fin de adaptar su comportamiento a las circunstancias bajo las cuales opera. La manipulación de ese conocimiento se hace a través de los modelos que posee el sistema.

En muchos STIs el *modelo del instructor* está implícitamente incluido y solo recientemente ha sido desarrollado explícitamente [13]. La actividad de un instructor involucra tanto conocimiento sobre el tema del dominio como sobre la pedagogía propia de esa área del conocimiento. A partir de la observación de la actividad de los instructores puede deducirse que la funcionalidad aportada por este modelo comprende los siguientes aspectos:

- Mejoramiento en la adaptación y realismo de las estrategias tutoriales.
- Provisión de ayuda inteligente individualizada para el autor/instructor.
- Refinamiento en el diseño de STIs particulares.
- Aliento a instructores novicios para que se involucren en el diseño de sistemas tutoriales
- Mejoramiento en la coordinación y cooperación entre varios instructores durante la creación del mismo STI.
- Registro de la actividad de cada instructor.

Con este modelo, los instructores de variado nivel de experiencia pueden interactuar con la herramienta de autoría a fin de diseñar e implementar sistemas tutoriales, asistido tanto por sus colegas a través del registro almacenado de su actividad como por la misma herramienta que, además puede usar teorías de diseño instruccional (como las de Gagné o Merrill).

Para que el modelo alcance la funcionalidad enunciada (ver figura 3) debe capturar información sobre las preferencias del instructor en estrategias tutoriales, sus

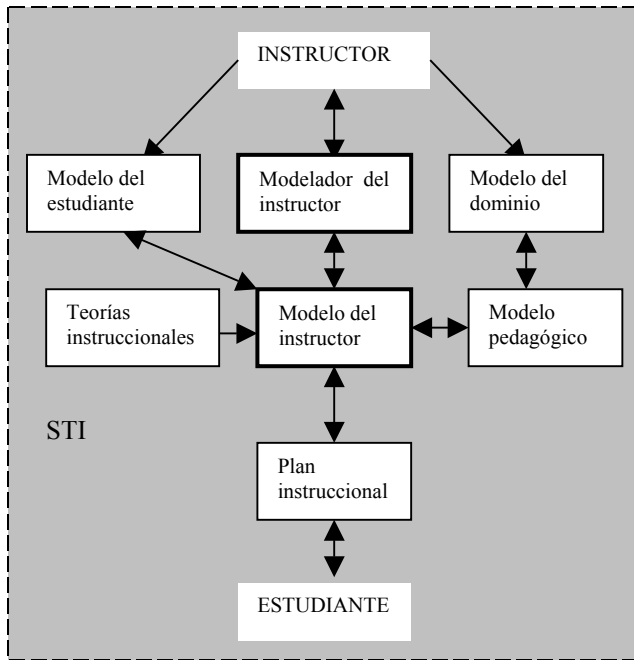


FIGURA 3

ARQUITECTURA DE STI, CON EL MODELO DEL INSTRUCTOR, CREADO POR LA HERRAMIENTA DE AUTORIA

intereses y actividades usuales con respecto a estilos de enseñanza preferidos por sus estudiantes, como su nivel de experticia (“experto”, “principiante”, etc.) tanto en la enseñanza en un dominio particular como en el proceso de creación de un STI.

Para que el modelo alcance la funcionalidad enunciada (ver figura 3) debe capturar información sobre las preferencias del instructor en estrategias tutoriales, sus intereses y actividades usuales con respecto a estilos de enseñanza preferidos por sus estudiantes, como su nivel de experticia (“experto”, “principiante”, etc.) tanto en la enseñanza en un dominio particular como en el proceso de creación de un STI.

Durante la etapa de creación del STI, los requerimientos pedidos por el sistema al autor (concretados a través de un modelador) pueden ser hechos siguiendo dos modalidades: a través de *preguntas directas* sobre sus preferencias personales y hábitos comunes (p.e. estilo de enseñanza o periodicidad de exámenes) o a través de la *observación de la actividad del instructor* registrando los cambios sobre el perfil del instructor.

Un autor/instructor, cuyo nivel de experiencia puede ser muy variado, puede comenzar proponiendo un curso inicial, donde inicializa las estructuras básicas de los modelos del *dominio* y el material instruccional asociado, *estudiante* (grupo de estudiantes que tomarán el curso), parámetros del *plan instruccional* (pedagogía del tema del curso). El curso resultante puede ser revisado posteriormente, tanto por el mismo instructor como por otros.

Si el nuevo instructor posee un nivel superior de

experticia, puede mejorar el diseño del curso. Si por el contrario, su nivel es inferior, puede aprender sobre desarrollo de cursos tutoriales inteligentes. I.e., el diseño e implementación de un STI queda sometido a un proceso de *refinamiento iterativo*, cuyo resultado será alcanzar sistemas tutoriales de mayor calidad, incentivando la cooperación de los autores.

Cuando se considera que el curso posee un nivel de refinamiento global apropiado, este es llevado a las aulas para su utilización con estudiantes reales. En este punto, el STI debe ajustarse a las condiciones reales dentro de las cuales se activará como por ejemplo, nivel de conocimiento de los alumnos, preferencias del instructor en esa aula, características del aula de computación, cantidad de estudiantes, etc. De esto modo se acrecienta el nivel de adaptabilidad de las estrategias tutoriales.

Por otro lado, la asistencia al autor/instructor atiende dos aspectos:

Diseño instruccional: el modelo debe adquirir información sobre las habilidades individuales en el desarrollo de sistemas tutoriales de cada instructor a fin de proveerlo con ayuda específica en el diseño instruccional. Un instructor principiante necesita mas ayuda que uno experimentado.

Uso del software: a fin de ampliar la cantidad de instructores que se acercan a los instructores computacionales es importante que la interfaz sea muy amigable y brinde ayuda en cada paso del desarrollo.

El modelo del instructor examina la actividad de cada instructor así como la de los estudiantes a fin de verificar la eficacia de las estrategias tutoriales del plan instruccional. El resultado de ese análisis contribuye a mejorar el conocimiento pedagógico del sistema a través del modelo propuesto.

MODELO DEL INSTRUCTOR EN CARRERAS DE INGENIERIA

La enseñanza de la Ingeniería se puede beneficiar con la inclusión de este modelo. Uno de los aspectos más importantes es la diseminación de los STIs en las aulas fundamentada en el acercamiento de estos sistemas al instructor real y no al revés. La adaptación realista al estilo de trabajo de cada instructor hace a estos sistemas más atractivos.

Hay temas que se enseñan en las carreras de Ingeniería que pueden ser controversiales en su inclusión y/o nivel de profundidad, por ejemplo, para la enseñanza de la modelización de diversos fenómenos propios del área (p.e. funcionamiento de redes eléctricas o el envío de gas natural licuado en tanques refrigerados) conviene un análisis top-down o bottom-up?. Mediante el modelo del instructor diversos profesores pueden concordar sobre el tema. A su vez, los docentes auxiliares pueden intervenir, inspeccionando la implementación del curso y aportando su punto de vista. Asimismo, la adaptación al estilo de

aprendizaje de cada estudiante será mas realista porque todos los docentes del curso aportarán a la mejora de la enseñanza en forma concordante.

CONCLUSION

La inclusión del modelo del instructor mejora el comportamiento global de un sistema tutorial. Se ha enriquecido la interacción con los usuarios del sistema y las relaciones con los otros modelos del STI. La inspección de la actividad detallada del estudiante por parte de su instructor agrega realismo a la actividad tutorial computacional y da lugar a una importante retroalimentación sobre las estrategias tutoriales. Asimismo el análisis detallado del dominio por varios instructores contribuye a su mejor construcción.

Sin embargo, falta evaluar mas fehacientemente el comportamiento de STIs con la nueva arquitectura. Esto se logrará cuando gran cantidad de estos sistemas funcionen en las aulas. Falta comprobar que el comportamiento instruccional mejora fuertemente con respecto a sistemas que no contienen el modelo explícitamente, i.e., verificar si el instructor computacional ha disminuido la brecha existente con el instructor real. Las pocas experiencias en este sentido así lo señalan.

REFERENCIAS

- [1] Hartley, J.,R, Sleeman, D.,H,. "Towards intelligent teaching systems" *International Journal of Man-Machine Studies*. Vol 5 1973, pags 215-236.
- [2] Wenger, E., "Artificial Intelligence and Tutoring Systems". *Los Altos, CA.*, 1987.
- [3] Du Boulay, B., Luckin, R, "Modelling human teaching tactics and strategies for tutoring systems". *International Journal of Artificial Intelligence in Education*. Vol 12 No 3, 2001, pags 235-256.
- [4] Murray, T, "Authoring Intelligent Tutoring Systems: An Analysis of the State of the Art". *International Journal of Artificial Intelligence in Education*. Vol 10 No 1 ,1999 , pags. 98-129.
- [5] Jonassen, D, Reeves, T, Hong, N, Harvey, D, Peters, K., "Concept Mapping as Cognitive Learning and Assessment Tools". *International Journal of Interactive Learning Research*. Vol.8. No 3-4, 1997, pags. 289-308
- [6] Arona, G, M, Huapaya, C, R, "La Representación del Dominio para la Generación de Sistemas Tutoriales Inteligentes". *Revista Internacio-nal Información Tecnológica*. Vol. 13 No 1, 2002, pags.185-192
- [7] Bloom, B, "Taxonomy of Educational Objectives". *The Classification of Educational Goals: Handbook I, Cognitive Domain*. New York; Toronto: Longmans, Green. 1956.
- [8] Gagné, R, "The Conditions of Learning and Theory of Instruction". *New York: Holt, Rinehart & Winston*. 1985.
- [9] Murray, T, "Special Purpose Ontologies and the Representation of Pedagogical Knowledge", *Proceedings of the International Conference on the Learning Sciences*. AACE, Charlottesville, VA. 1996.
- [10] Reigeluth, C, "Green Book .Module 3: Concept Classification" (www.indiana.edu/~idtheorymethods/m3.html). 1999.
- [11] Zabala, A, Los enfoques didácticos, en Coll, Salvador y otros "El constructivismo en el aula". Barcelona 1993.
- [12] Sison, R, Shimura, M, "Student Modeling and Machine Learning" *International Journal of Artificial Intelligence in Education*. Vol 9 No 1, 1998, pags 128-158
- [13] Ohlsson, S, "Some Principles of Intelligent Tutoring" en Lawler & Yazdani Eds. *Learning Environment and Tutoring Systems* Vol 1. Ablex: Norwood, NJ, 1987, pags 203-237.
- [14] Johnson, W, L, Rickel, J, "Steve: An animated pedagogical agent for procedural training in virtual environments". *SIGART Bulletin* 8, 1998, pags. 16-21.
- [15] Virvou, M, Moundridou, M,. "Adding an Instructor Modelling Com-ponent to the Architecture of ITS Authoring Tools". *International Journal of Artificial Intelligence in Education*. Vol 12, No 2 ,2001 pags 185-211