

SISTEMA ADAPTATIVO A LA DOCENCIA Y APRENDIZAJE BASADO EN PRÁCTICAS

Federico Botella¹, Enrique Lazcorreta¹, Pascual González², Antonio Fernández-Caballero² y José Manuel Gascuña²

1: Instituto Universitario “Centro de Investigación Operativa”
Universidad Miguel Hernández
Av. de la Universidad s/n, 03202, Elche (Alicante)
e-mail: {federico, enrique}@umh.es, web: <http://cio.umh.es>

2: Instituto de Investigación en Informática de Albacete
Escuela Politécnica Superior de Albacete
Universidad de Castilla-La Mancha
Campus Universitario de Albacete, 02071, Albacete
e-mail: {pgonzalez, caballer,jmgascuena}@dsi.uclm.es, web: <http://www.i3a.uclm.es>

Palabras clave: Sistemas Hipermedia Adaptativos, e-learning.

Resumen. *El aprendizaje de un lenguaje de programación se refuerza con la resolución de prácticas. La creación, clasificación y gestión de cuestiones puede ser un trabajo arduo para el profesor, si queremos que éstas puedan adaptarse al nivel de cada alumno y en cada momento del curso. Este trabajo expone un sistema que permite al profesor generar muchas cuestiones sin preocuparse de su clasificación y gestión, recibiendo información del sistema sobre la adaptación de las cuestiones a las necesidades de aprendizaje de sus alumnos. Este sistema también permite la selección de cuestiones a resolver adaptada a los conocimientos de cada alumno.*

1. INTRODUCCIÓN

Los Sistemas Hipermedia Adaptativos aplicados a la educación a distancia permiten al alumno recorrer de manera no secuencial los contenidos de un curso adaptándose a sus conocimientos y necesidades en cualquier circunstancia. Sin embargo, delegar en el alumno la toma de decisiones sobre qué itinerario seguir en sus estudios no es aconsejable en muchas ocasiones, idea de la cual surgen los Sistemas Tutores Inteligentes [5,6,7,8], que permiten adaptar los itinerarios formativos de la materia estudiada al ritmo de trabajo y conocimientos

ya adquiridos por el alumno, partiendo de la experiencia del profesor y del propio sistema en su interacción con otros alumnos de la misma materia.

La adaptatividad que presentan muchos de estos sistemas es una imposición de cómo y qué debe estudiarse, no permitiendo al alumno seguir adelante si no supera antes ciertas pruebas de conocimiento [3].

Otros enfoques permiten al alumno elegir su propio itinerario [4,1], pero el sistema que lo tutoriza no puede determinar con claridad qué tópicos son realmente conocidos por el alumno si no descendemos al nivel de concepto (un tópico sería un conjunto de conceptos). A diferencia de un tutor humano, a un sistema informático no le resulta sencillo averiguar qué parte de una cuestión no entiende (o no es capaz de resolver) el alumno; sin embargo, el sistema informático es capaz de recordar cómo planteó el alumno cada una de las cuestiones que ha intentado resolver a lo largo del curso.

A diferencia de los sistemas adaptativos, los sistemas adaptables permiten ser configurados por el usuario para adaptarse a sus necesidades. El sistema expuesto en este trabajo es adaptable-adaptativo, ya que permite que el alumno indique cuál es su estado de ánimo y su predisposición a trabajar en ese momento (adaptable) y, en función de sus apetencias y de sus conocimientos sobre la materia en estudio, el sistema le sugerirá la realización de las cuestiones que mejor se adapten a su perfil (adaptativo).

2. SISTEMA DE CUESTIONES AUTO-ADAPTATIVAS

El sistema propuesto se basa en un diccionario de los conceptos que forman el curso, una colección de cuestiones, un analizador de código y la interacción de los alumnos del curso con el sistema. Todas las cuestiones son autocorrectivas (mediante un valor o una selección múltiple) de modo que el sistema pueda verificar si el alumno ha sido capaz de resolverla..

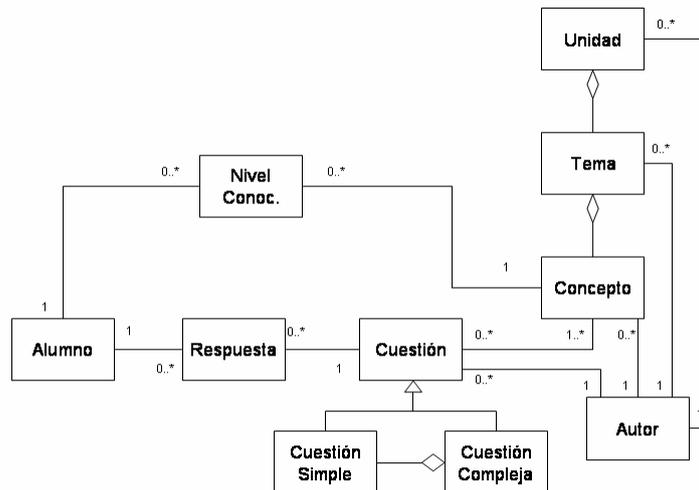


Figura 1. Modelo de clases del sistema.

En la Figura 1 se observan los tres componentes fundamentales del sistema: material teórico, práctico e interacción de los alumnos con el sistema. El material teórico es introducido por el profesor mediante conceptos que son asociados al currículo del curso. El material práctico consiste en cuestiones (simples y complejas) que también son introducidas por el profesor pero se asocian al material teórico por el sistema. La interacción de los alumnos con el sistema se realiza anotando los aciertos y fallos de cada alumno al intentar resolver una cuestión, así como el instante y la duración en cada intento. Esta interacción permite al sistema determinar el nivel del alumno en cada uno de los conceptos tratados en el curso, así como la dificultad que presenta cada cuestión al intentar ser resuelta por los diferentes alumnos del curso.

2.1. Cuestiones Auto-Adaptativas

Las cuestiones auto-adaptativas son cuestiones del tipo expuesto en [1], formadas por un enunciado, un código fuente y la respuesta correcta, que puede consistir en un valor único o un conjunto de opciones. Un ejemplo de estas cuestiones en un curso de programación en lenguaje C consistiría en preguntar *¿Cuál es el valor final de i?* a partir del código `int i = 6; if (i % 3) i += 2;` e indicar al sistema que la respuesta correcta es 6.

Al recibirla el sistema la analizará y la guardará en la correspondiente tabla, junto con información sobre los conceptos que contiene, en este caso `int`, `=`, `if`, `%`, y `+=`.

La dificultad de la cuestión será determinada por el propio sistema en función de los conceptos que contiene y el número de aciertos y fallos obtenidos cuando los alumnos intenten resolverla. Esta dificultad será utilizada por el sistema para sugerir al alumno cuestiones adaptadas al nivel que éste tenga y a su concentración actual.

2.2. Adaptación al profesor

El primer problema que surge al intentar implementar este sistema es el trabajo que puede suponer al profesor la creación, clasificación y gestión de las cuestiones, entre otras cosas por que al descender al nivel de concepto necesitaremos un gran número de cuestiones que puedan adaptarse al conocimiento conceptual (generalmente bastante disperso) de un gran número de alumnos que, además, pueden elegir su propio currículo.

La adaptación al profesor se logra liberándole de las dos últimas tareas de modo que sólo tenga que escribir los conceptos y las cuestiones.

Cuando el profesor entra en el sistema (ver Figura 2) puede consultar el material que ya contiene el curso, comprobando qué conceptos son tratados en las cuestiones ya incluidas, o introducir nuevo material. Al introducir un concepto nuevo el sistema comprueba que no exista previamente, lo inserta en su caso y comprueba si el concepto está incluido en las cuestiones que ya contiene el sistema, en cuyo caso actualiza la información sobre las cuestiones y sobre el nivel de los alumnos que ya han contestado a dichas cuestiones. Al insertar una nueva cuestión el sistema comprueba qué conceptos contiene y muestra esta información al profesor. Si el sistema no ha reconocido un concepto es debido a que no ha sido introducido aún con el material teórico del curso, siendo el profesor el responsable de

introducirlo. El análisis del uso de las cuestiones creadas también es labor del sistema. Una vez escrita una cuestión no es necesario eliminarla del sistema, aunque el profesor quiera añadir otra de características similares, el objetivo es conseguir un gran número de cuestiones que puedan adaptarse a las necesidades de aprendizaje de cada uno de los alumnos de un curso.

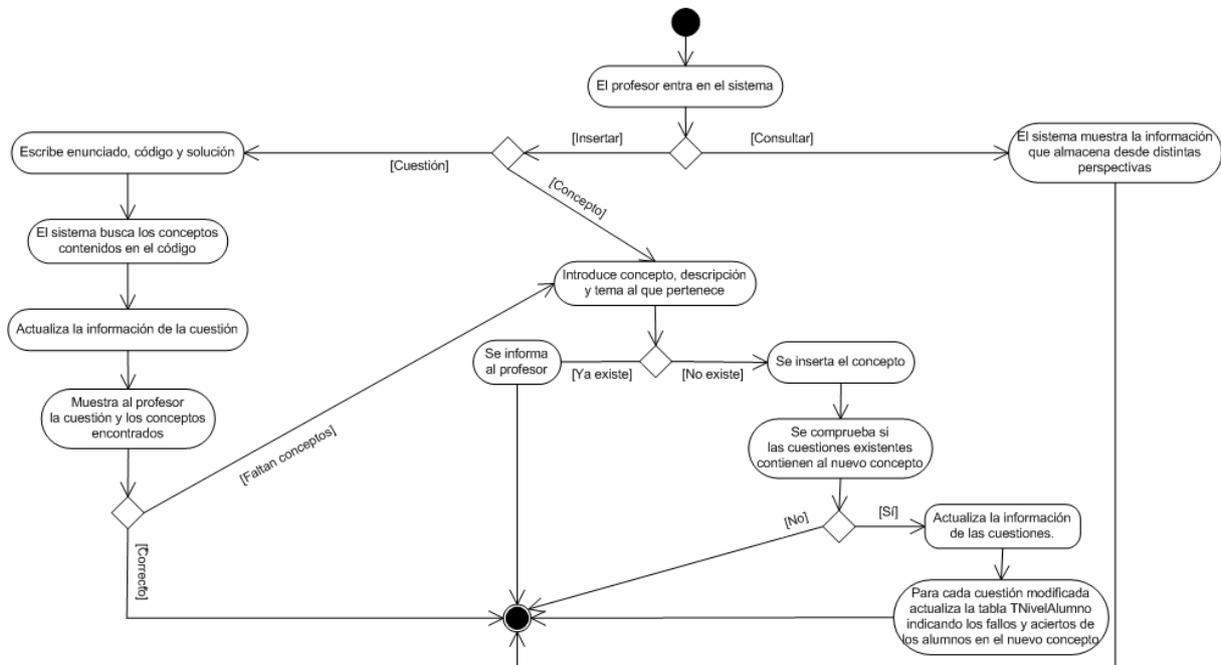


Figura 2. Interacción del profesor con el sistema.

Las carencias del sistema a la hora de ofrecer cuestiones adaptadas a cualquier alumno generan información que es enviada al profesor para que pueda insertar nuevas cuestiones sin tener que revisar manualmente las cuestiones ya disponibles en el curso.

2.3. Adaptación al alumno

Cuando un alumno quiere iniciar una sesión de prácticas se le presenta una sencilla interfaz con la que el sistema entenderá qué tipo de ejercicios quiere resolver el alumno: los concernientes a un concepto objetivo y a otros conceptos que el alumno conozca, o todos los de un tema o un conjunto de ellos, los que peor domina el alumno o incluso los que mejor domina. También el alumno puede indicar si está dispuesto a resolver cuestiones difíciles (quiere esforzarse en ese instante) o fáciles (no quiere esforzarse demasiado pero quiere aprovechar el tiempo que dispone).

Una vez recibida esta información el sistema comprueba los conocimientos demostrados por el alumno en la resolución de cuestiones y la dificultad de las mismas, mostrándole las cuestiones ordenadas en grupos de 10 según su adaptación al requerimiento actual del alumno, permitiendo que sea él quien decida qué cuestión desea resolver en cada instante.

Al terminar de resolver una cuestión se actualizará la información sobre el alumno y se volverá a sugerir cuestiones hasta que el alumno decida terminar su sesión de prácticas. La resolución de cuestiones por parte de los alumnos reportará la suficiente información al sistema para determinar automáticamente la dificultad de cada cuestión.

3. CUESTIONES AUTO-ADAPTATIVAS COMPLEJAS

El primer planteamiento del sistema nos lleva a la creación de cuestiones simples que contengan un número mínimo de conceptos de modo que el sistema sea capaz de detectar adecuadamente qué conceptos comprende bien el alumno y cuáles no. Sin embargo, el aprendizaje de un lenguaje de programación no puede basarse en la práctica de cuestiones simples. Sobre todo si el alumno quiere preparar un examen final en el que se entremezclan muchos de los conceptos del curso en un único código.

La información obtenida por el sistema cuando un alumno no es capaz de resolver una cuestión compleja podría dar lugar a confusión pues se considera que el alumno no conoce ninguno de los conceptos incluidos en la cuestión, por lo que para gestionar las cuestiones complejas se ha diseñado una estrategia diferente basada en la segmentación de la cuestión, por parte del profesor, en bloques menores de código (sub-cuestiones) que permitan identificar los conocimientos y carencias del alumno en cada concepto de la cuestión (Ver Figura 3). Si el alumno no resuelve satisfactoriamente la cuestión completa se le propone que la resuelva de forma desmembrada mostrándole secuencialmente bloques del código y preguntándole el valor de una o más variables, tanto si acierta como si no se le presenta la solución correcta y se le plantea resolver el siguiente bloque de código, anotando sus fallos y aciertos en cada sub-cuestión como si de cuestiones simples se tratara.

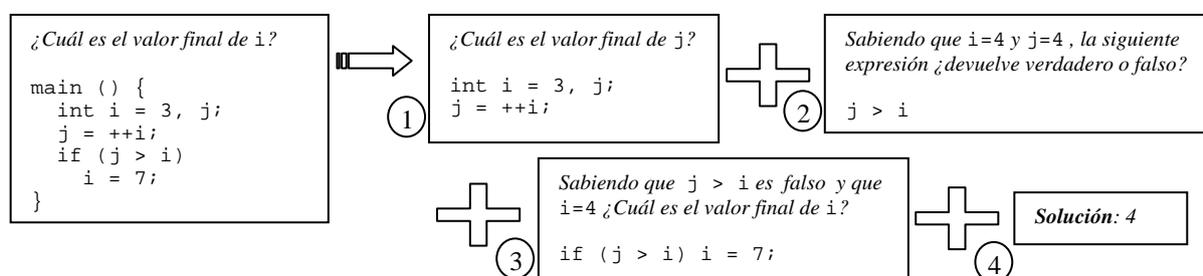


Figura 3. Depuración de una cuestión compleja.

4. CONCLUSIONES Y TRABAJO FUTURO

La parametrización de cuestiones propuesta en [2] dotaría al sistema de mayor potencia pues permitiría a los alumnos repetir cuestiones aunque el sistema le haya dado la respuesta correcta en un intento fallido.

La minería de datos permitirá descubrir patrones de comportamiento de los alumnos que, cruzados con la nota oficial obtenida por cada uno de ellos en un examen parcial o final

permitirá sugerir automáticamente estrategias de aprendizaje a los nuevos alumnos. La dificultad de cada cuestión se determina actualmente en base a todos los intentos de resolverla, pero la realidad es que una cuestión que es *difícil* al principio del curso puede ser *fácil* al final del mismo. Se estudiará un refinamiento del cálculo de la dificultad que tenga en cuenta el patrón del alumno que quiere resolver cuestiones.

AGRADECIMIENTOS

Este trabajo está financiado, en parte, por la ayuda de la Universidad Miguel Hernández de Elche – Bancaja R.R.1256/04 y por el proyecto de la Junta de Comunidades de Castilla-La Mancha PAI06-0093.

REFERENCIAS

- [1] P. Brusilovsky, S. Sosnovsky y O. Shcherbinina, *Quiz Guide: Increasing the Educational Value of Individualized Self-Assessment Quizzes with Adaptive Navigation Support*, Actas del World Conference of E-Learning (E-Learn), Washington DC (2004), pp. 1806-1813.
- [2] S. Sosnovsky, O. Shcherbinina y P. Brusilovsky, *Web-based parameterized questions as a tool for learning*, Actas del World Conference on E-Learning (E-Learn), Phoenix, Arizona (2003), pp. 309-316
- [3] A. Fernández-Caballero, J. M. Gascueña, F. Botella y E. Lazcorreta, *Distance Learning by Intelligent Tutoring System*, Actas del VII International Conference on Enterprise Information Systems (ICEIS), Miami (2005), vol. 5, pp. 75-82.
- [4] C. J. Butz, S. Hua y R. B. Maguire, *Bits: a Bayesian Intelligent Tutoring System for Computer Programming*, Actas del Western Canadian Conference on Computing Education (WCCCE), Kelowna BC (2004), pp. 179-186.
- [5] Tang, T.Y., Wu, A. *The implementation of a multi-agent intelligent tutoring system for the learning of computer programming*. Proceedings of 16th IFIP World Computer Congress-International Conference on Educational Uses of Communication and Information Technology, ICEUT 2000.
- [6] Capuano, N., Marsella, M., Salerno, S. *ABITS: An agent based Intelligent Tutoring System for distance learning*. Proceedings of the International Workshop on Adaptive and Intelligent Web-Based Education Systems, ITS 2000.
- [7] Peña, C.I., Marzo, J.L., de la Rosa, J.L. *Intelligent agents in a teaching and learning environment on the Web*. Proceedings of the 2nd IEEE International Conference on Advanced Learning Technologies, ICALT2002.
- [8] Hospers, M., Kroezen, E., Nijholt, A., op den Akker, H.J.A., Heylen, D. *An agent-based intelligent tutoring system for nurse education*. In Applications of Intelligent Agents in Health Care, J. Nealon and A. Moreno (eds), (2003), pp. 143-159.