

Una propuesta docente basada en test de múltiple respuesta para mejorar el desarrollo de las sesiones en el aula

Sandra Baldassarri, Pedro Álvarez
Departamento de Informática e Ingeniería de Sistemas
Universidad de Zaragoza
sandra@unizar.es, alvaper@unizar.es

Resumen

La motivación e implicación de los estudiantes en las sesiones teórico-prácticas que tienen lugar en el aula condicionan el éxito o fracaso de esta actividad docente. Durante los últimos dos cursos, la actitud en el aula de los estudiantes de una asignatura de máster ha empeorado de manera drástica, impactando en el proceso y resultados de aprendizaje. Después de analizar las causas con los propios estudiantes se propone un cambio metodológico basado en combinar lecciones breves y auto-contenidas con test de auto-evaluación. Estos test pretenden ser un instrumento motivador y una herramienta de “feedback” continuo para los estudiantes. Para que esta aproximación no suponga un trabajo extra para el profesor se ha programado un sistema semántico de generación automática de test de múltiple respuesta.

Abstract

Students' motivation and participation in the theoretical-practical classroom sessions determine the success or failure of the teaching activity. For the last two courses, the attitude in class of the students of a master subject has gotten worse, which affects the process and the results of their learning. After analyzing the reasons with the students, a methodological change based in combining short and self-contained lessons with self-assessment tests, is proposed. These tests intend to be a motivating instrument and a tool for continuous feedback for the students. In order not to overload teachers with this approach, we developed a semantic system for automatic multiple-choice test generation.

Palabras clave

Sesiones en el aula, motivación del estudiante, test de múltiple respuesta, evaluación del aprendizaje

1. Introducción

Las sesiones teórico-prácticas en el aula son una actividad habitual en muchas asignaturas de grado y máster. En este tipo de sesiones la actitud del profesor y su forma de impartir clase juegan un papel clave a la hora de motivar a los estudiantes. Esta tarea no siempre resulta sencilla, ni siquiera depende únicamente del profesor, pero es vital para el buen desarrollo del curso y la consecución de los objetivos de la asignatura.

En estudios de máster las sesiones en el aula suelen ser más dinámicas y participativas. Los estudiantes están motivados por sí mismos, tienen una mejor predisposición al aprendizaje y están interesados en los contenidos. Además, el tamaño reducido de los grupos favorece aún más esta actitud positiva. Hasta hace dos años éste era el ambiente que caracterizaba las sesiones de la asignatura de *Tecnologías y Modelos para el Desarrollo de Aplicaciones Distribuidas* (TMDAD), impartida en el segundo semestre del Máster Universitario de Ingeniería Informática de la Universidad de Zaragoza. Sin embargo, estos dos últimos cursos la actitud de los estudiantes ha cambiado. La impresión de los profesores es que “desconectan” con demasiada facilidad. Según ellos mismos reconocen, este problema se debe principalmente al cansancio acumulado (algunos están todo el día en la facultad y otros compaginan estudios con trabajo en empresa o actividad investigadora) y al hábito generalizado de aprovechar el tiempo de clase para avanzar en paralelo en las tareas no presenciales que tienen que realizar como parte de ésta u otras asignaturas del programa. Esta falta de atención ha afectado negativamente a los resultados obtenidos por los estudiantes en el test final de la asignatura, dado que éste constaba de cuestiones fundamentalmente tratadas en el aula. Según los criterios de evaluación, no superar este test implica suspender la asignatura.

El objetivo de esta experiencia de innovación es principalmente reconducir la actitud de los estudiantes en el aula, teniendo presente que hay ciertas circunstancias que son ajenas al desarrollo de la asignatura (los horarios, el cansancio, la gestión del tiempo

que realiza cada estudiante, etc). La solución propuesta consiste en poner a los estudiantes en una situación en la que ellos mismos se vean obligados a prestar atención y tener una actitud participativa, pero que a la vez este esfuerzo tenga una recompensa inmediata. La idea intuitiva es sustituir el test final de la asignatura por la realización de un test breve en cada sesión de aula (con una duración de entre 5 y 10 minutos) [1]. Las cuestiones de los test estarán restringidas a los contenidos y discusiones de la correspondiente sesión y su calificación contribuirá a la nota final en la asignatura. Además, el estudiante podrá acceder a las respuestas correctas y comprender en qué falló y qué conocimientos debería revisar utilizando el material disponible. De la misma forma, el profesor podrá analizar el desarrollo de la sesión a partir del resultado de los test.

Esta solución no debe suponer una carga extra de trabajo para el profesor. Por este motivo se ha programado un sistema software que genere automáticamente los test, como alternativa a otras aproximaciones [5]. El sistema asiste en el proceso de crear y evaluar test de múltiple respuesta. Este tipo de test consiste de un conjunto de preguntas donde cada una de ellas tiene cuatro posibles respuestas, una correcta y tres incorrectas. Estas últimas reciben el nombre de distractores. Aunque existen sistemas similares en lo funcional, la solución programada es novedosa desde tres puntos de vista: crea los test a partir de bases de conocimiento disponibles en Internet, consigue mejorar la calidad de los distractores mezclando técnicas semánticas y algoritmos heurísticos, y ofrece flexibilidad en torno al formato de las preguntas que se pueden plantear [3, 2, 4].

A nivel de contenidos el artículo se estructura en tres secciones. En la sección 2 se presenta la asignatura y los detalles de la experiencia. En la sección 3 se describe el sistema software programado. Finalmente, se discuten unas breves conclusiones.

2. Descripción general de la experiencia

En esta sección se describe de forma breve la asignatura objeto de la innovación y el proyecto piloto.

2.1. Organización de la asignatura

TMDAD es una asignatura obligatoria de 6 créditos ECTS. Está organizada en dos bloques temáticos. El primero está dedicado fundamentalmente a las técnicas de ingeniería y las tecnologías que se aplican actualmente en el diseño y desarrollo de las aplicaciones distribuidas. El segundo aborda el problema de configurar y ejecutar estas aplicaciones en distintos entornos de computación (principalmente, en la nube) desde un

punto de vista tecnológico, económico y de negocio.

La asignatura consta de tres tipos de actividades: sesiones teórico-prácticas en el aula (30 sesiones de 1 hora), prácticas en el laboratorio (5 sesiones de 3 horas), y un proyecto en grupo que consiste en implementar un sistema distribuido a partir del trabajo realizado en el laboratorio (la dedicación estimada es de 90 horas por estudiante). Las sesiones en el aula se distribuyen por igual entre los dos bloques temáticos de la asignatura (15 horas por bloque); mientras que en el caso de las prácticas, tres sesiones corresponden al bloque de tecnologías y dos al de entornos de ejecución.

La evaluación consiste en un test final relacionado con los contenidos introducidos en las sesiones en el aula (30 % de la nota final) y la defensa y demostración del proyecto (70 %). Para superar la asignatura es necesario obtener una calificación igual o superior a 5,0 en las dos pruebas. Sólo en ese caso, la calificación final será de acuerdo a la ponderación establecida.

2.2. Puesta en práctica

Durante el curso 2017-2018, la asignatura TMDAD tiene matriculados 13 estudiantes. La estructura en dos bloques de la asignatura favorece el objetivo de evaluar bajo condiciones similares si la propuesta de innovación supone una mejora. A este respecto, el primer bloque se va a impartir y evaluar conforme el modelo anterior, es decir, realizando un único test de conocimientos al final del curso (éste valdrá un 15 % de la nota de la asignatura), mientras que el segundo bloque se evaluará por medio de un test por sesión (el conjunto de todos los test valdrá el otro 15 %).

El segundo bloque de la asignatura introduce los fundamentos y tecnologías existentes para configurar y ejecutar aplicaciones distribuidas en entorno de computación tipo clúster, grid y en la nube. También presenta experiencias y casos prácticos de aplicaciones reales que ayuden a los estudiantes a entender la metodología a seguir para trabajar con este tipo de entornos y los problemas habituales (e inesperados) que es necesario afrontar. Todo este conocimiento se imparte en 11 lecciones: 9 de ellas consisten de una única sesión, 1 requiere dos sesiones, y la última lección del curso está programada para 4 sesiones. En base a esta organización de lecciones se han programado 10 test de evaluación (la última lección no es evaluada porque consiste en un caso práctico donde los estudiantes actúan como consultores de servicios en la nube).

Cada test va a constar de 10 preguntas de selección múltiple, donde cada pregunta tendrá cuatro posibles respuestas alternativas y únicamente una será la correcta. El nivel de dificultad de las preguntas de un test concreto no será homogéneo con el fin de evaluar los distintos niveles de conocimiento y aprendizaje adquiridos por el estudiante. De las 10 preguntas, 6 serán sen-

cillas (la respuesta habrá aparecido explícitamente en el material usado durante la sesión), 2 de complejidad media (requerirán que el estudiante haya comprendido determinadas reflexiones/discusiones que han tenido lugar en el aula), y 2 complejas (requieren un nivel de análisis extra por parte del estudiante a partir de lo presentado/discutido). El valor de todas las preguntas de un test será el mismo (1 punto por pregunta), y La nota final del segundo bloque será la media aritmética del resultado de los 10 test programados.

3. Sistema de generación automática de test

A continuación se describe el proceso de creación de un test y el sistema que asiste al profesor.

3.1. Proceso de elaboración de un test

El proceso de elaborar un test consta de cuatro fases secuenciales. Durante estas fases el profesor interactúa con el sistema software desarrollado para redactar las preguntas y su respuesta correcta. El sistema genera automáticamente los distractores necesarios para crear un test de múltiple respuesta y, posteriormente, publica este test en el servicio *Google Form*.

En la primera fase, el profesor redacta el test. Las preguntas son del tipo *single-word answers*, es decir, la respuesta correcta puede ser una palabra o un concepto concreto. El sistema software tiene una interfaz Web que facilita la tarea de redacción y permite configurar la dificultad del test (o de cada pregunta individual).

Durante la segunda fase, el sistema genera automáticamente el conjunto de distractores necesarios para cada pregunta. Estos distractores se calculan aplicando distintos algoritmos semánticos y técnicas heurísticas sobre varias bases de conocimiento semántico disponibles en Internet (WordNet, WordNik y Wikipedia). El proceso de selección de distractores depende de la calidad de estos y el nivel de dificultad del test.

En la tercera fase, el sistema muestra al profesor el test generado. El docente tiene la posibilidad de cambiar los distractores propuestos como respuestas incorrectas (el propio sistema le ofrece una lista de alternativas) y/o aceptar el test. Una vez aceptado, el sistema publica el test en el servicio *Google Form*, devolviendo la correspondiente URL pública. De esta forma, los estudiantes pueden acceder y realizar el test desde cualquier navegador y el profesor consultar los resultados.

3.2. Generación de distractores

El componente clave del sistema desarrollado es el generador de distractores (o respuestas incorrectas). La

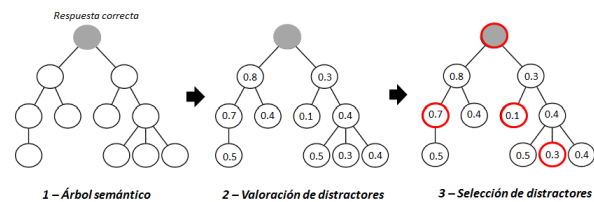


Figura 1: Generación/selección de distractores

figura 1 describe el método programado para generar y seleccionar los distractores de una pregunta.

El punto de partida es la respuesta correcta especificada por el profesor. A partir de ésta se crea un árbol que contiene un conjunto de conceptos que están relacionados a nivel semántico con la respuesta (paso 1). Cada nodo del árbol corresponde con uno de estos conceptos, siendo la raíz la respuesta correcta. Para la creación del árbol semántico se aplica un algoritmo en profundidad que interacciona con las bases de conocimiento accesibles en la Web. Dado un nodo cualquiera, sus hijos son conceptos que son “próximos” semánticamente a la respuesta del nodo padre. La profundidad del árbol es un parámetro de configuración del sistema.

El árbol semántico representa un conjunto de candidatos a distractores. En el segundo paso del método se aplican una serie de heurísticas que valoran la calidad de estos distractores. A cada nodo del árbol se le asigna un valor comprendido entre el 0,0 (el concepto del nodo es un distractor de muy baja calidad) y 1,0 (el distractor es equivalente a la respuesta correcta). Estas heurísticas son complejas, pero en esencia valoran un concepto según: su grado de similaridad semántica con la respuesta correcta, su importancia en el contexto de la pregunta, el nivel de profundidad del árbol en el que aparece, y si aparece en una o varias bases de conocimiento, entre otras cuestiones.

Finalmente, el tercer paso consiste en la selección de los distractores de la pregunta. Esta selección depende del grado de complejidad del test configurado por el profesor. El sistema actual permite cuatro niveles de dificultad: sencillo, intermedio, complejo y muy complejo. Por ejemplo, en el nivel intermedio los algoritmos de selección eligen las siguientes (cuatro) posibles respuestas para la pregunta: la respuesta correcta (nodo raíz), un distractor de calidad alta (con un valor en torno al 0,7) y dos distractores de baja calidad. El objetivo es que el estudiante rápidamente descarte dos posibles opciones y tenga dudas entre las otras dos, entre las que se encuentra la respuesta correcta. En base a estos criterios, para la pregunta “*The cluster virtualization is the (blank) of cloud computing*” las opciones generadas (dificultad intermedia) son “*origin*” (respuesta correcta), “*ancestry*” (distractor de alta calidad), y

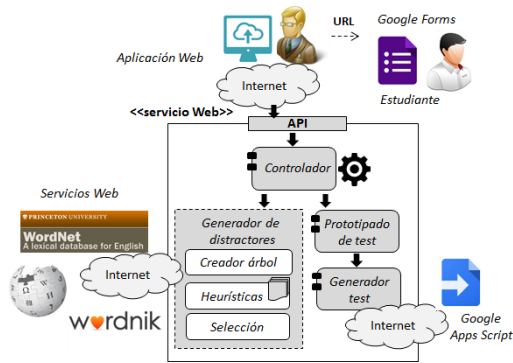


Figura 2: Componentes del sistema software

“drive” y “descent” (ambos de baja calidad).

3.3. Diseño e implementación del sistema

La figura 2 muestra los componentes del sistema de generación de test. La solución consiste en una aplicación Web que asiste al profesor en el proceso de creación y un servicio Web responsable de este proceso.

Internamente, el servicio está compuesto por cuatro componentes. Un *controlador* que atiende las peticiones de los usuarios y es responsable de crear un proceso independiente para gestionar la elaboración completa de cada test. Este proceso interactúa con el componente de *generación de distractores* para crear las respuestas alternativas correspondientes a cada pregunta. Este generador crea los árboles semánticos a partir de las distintas bases de conocimiento integradas (WordNet, WordNik y Wikipedia), aplica las técnicas heurísticas y selecciona los distractores. El coste computacional de este proceso de generación es elevado y ha requerido el uso de técnicas y algoritmos de programación paralela. Posteriormente, a partir de los distractores seleccionados, el componente de *prototipo de test* elabora una propuesta de enunciado (preguntas y respuestas) y gestiona el proceso de modificación y aceptación por parte del profesor. Finalmente, el componente de *creación de test* interactúa con *Google Forms* para crear la versión definitiva del test. El diseño de este último componente está pensado para integrar en un futuro herramientas de publicación de test alternativas a partir de los enunciados actuales.

Desde un punto de vista tecnológico, la programación del sistema está basada en *Node.js*, *AngularJS* y *MongoDB*. Además, la interfaz del servicio está basada en tecnología *Swagger* para facilitar su integración

4. Conclusiones

Esta experiencia de innovación pretende recuperar el interés, la motivación y la participación de los estudiantes en un futuro en otros sistemas accesibles vía Red. Los test no suponen una carga extra para el estudiante y tenemos la esperanza de que tampoco les genere una respuesta negativa en forma de estrés, ansiedad o presión añadida. Por otro lado, las contribuciones técnicas del sistema programado suponen un avance con respecto a otras soluciones existentes, aunque queda pendiente para un futuro evaluar la calidad e idoneidad de los distractores generados. Finalmente, resaltar que la solución podría reutilizarse para cualquier asignatura.

Agradecimientos

Este trabajo está financiado por un Proyecto de Innovación Docente de la Universidad de Zaragoza, PII-DUZ_17_078.

Referencias

- [1] O. Adesope, D.A. Trevisan y N. Sundararajan. Rethinking the Use of Tests: A Meta-Analysis of Practice Testing. *Review of Educational Research*, 87(3):659–701, 2017.
- [2] F. Aggrey, R. Kuo, M. Chang y Kinshuk. Online test system to reduce teachers’ workload for items and test preparation. *Innovations in Smart Learning*, Springer, 215–219, 2017.
- [3] R. Conejo, E. Guzmán y M. Trella. The SIETTE automatic assessment environment. *Int. Journal of Artificial Intelligence in Education*, 26(1):270–292, 2016.
- [4] R. Mitkov, L.A. Ha, A. Varga y L. Rello. Semantic similarity of distractors in multiple-choice test: extrinsic evaluation. En *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pp. 49–56, 2009.
- [5] J.L. Risco-Martín, M. Sánchez-Élez y I. Pardines. Experiencia educativa para fomentar el aprendizaje autónomo a través de preguntas tipo test generadas por los alumnos. En *Actas de las XVII Jornadas de Enseñanza Universitaria de Informática, Jenui 2011*, páginas 283–291, Sevilla, Julio 2011.