



Inteligencia Artificial. Revista Iberoamericana
de Inteligencia Artificial

ISSN: 1137-3601

revista@aepia.org

Asociación Española para la Inteligencia
Artificial
España

Vizcaíno, Aurora

Un estudiante simulado que detecta y corrige situaciones negativas en entornos colaborativos
Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial, vol. 8, núm. 23, verano, 2004, p.

0

Asociación Española para la Inteligencia Artificial
Valencia, España

Disponible en: <http://www.redalyc.org/articulo.oa?id=92502314>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal

Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

Un Estudiante Simulado que Detecta y Corrige Situaciones Negativas en Entornos Colaborativos

Aurora Vizcaíno

Escuela Superior de Informática
Universidad de Castilla-La Mancha
Paseo de la Universidad, 4, 13071 Ciudad Real (España)
aurora.vizcaino@uclm.es

Resumen

Durante el aprendizaje colaborativo, soportado o no por computadora, pueden surgir situaciones que disminuyen las ventajas de esta forma de aprendizaje o que impiden que éstas aparezcan. Este artículo propone una solución para controlar los efectos de tales situaciones en ambientes colaborativos apoyados por computadora. Concretamente, describe el diseño y comportamiento de un estudiante virtual, al que se le llama “estudiante simulado”, encargado de detectar la ocurrencia de tres situaciones negativas: alumnos pasivos, conversaciones fuera de lugar y problemas en la resolución de los ejercicios. Los experimentos realizados para validar la eficiencia del estudiante simulado demostraron que en la mayoría de los casos detectó y evitó las tres situaciones negativas.

Palabras clave: Aprendizaje colaborativo soportado por computador, estudiantes simulados, modelo del estudiante y del grupo.

1. Introducción

Numerosos estudios han demostrado las ventajas que ofrece aprender de forma colaborativa. Cuando los alumnos trabajan en grupo suelen estar más atentos al proceso de aprendizaje (Damon y Phelps, 1989) ya que sus intervenciones y sus fallos repercuten en los resultados del grupo. Además, el aprendizaje que se obtiene, a menudo, es más duradero y significativo puesto que es fruto de los intercambios de opinión y la reflexión de los miembros del grupo (Soller et al., 1998). Por otro lado, el aprendizaje colaborativo aumenta las

habilidades sociales de los estudiantes (Blaye y Light, 1995).

Sin embargo, como se explica en Dillenbourg (1999), no basta con reunir a un grupo de estudiantes y encomendarle una tarea para que surja un aprendizaje colaborativo eficiente, es decir, para que surja un ambiente óptimo para la discusión, reflexión, comunicación y colaboración.

Desafortunadamente, a veces, diferentes comportamientos impiden que los beneficios del aprendizaje colaborativo tengan lugar. A

continuación se detallan algunos de dichos comportamientos, (una lista más completa se encuentra en Vizcaíno (2001)).

- Sentimiento individualista. Esta situación tiene lugar cuando un miembro del grupo se considera más inteligente o capacitado que los demás para resolver los problemas o la tarea encomendada, por lo que trabaja ignorando las opiniones del resto de los compañeros y de forma individualizada. Esta situación suele provocar un sentimiento de frustración en el resto de componentes del grupo.
- Pasividad. Se dice que hay pasividad en el grupo cuando uno o varios alumnos no colaboran y no toman parte en la realización de las tareas. Esta circunstancia tiene un doble efecto negativo. Por un lado la persona pasiva aprende menos que en el caso de aportar ideas y recibir las críticas y opiniones de los compañeros. Por otro lado, el grupo se siente incómodo con una persona que no participa.
- Conversaciones fuera de lugar. A veces, cuando se trabaja en grupo se entablan conversaciones que no están relacionadas con el tema que se pretende aprender. Este hecho puede implicar pérdidas considerables de tiempo tardándose, en consecuencia, mucho más en completar la tarea.

Desde hace poco más de una década, las nuevas tecnologías, proporcionan apoyo para el aprendizaje colaborativo (Lund, Baker y Baron 1996). Sin embargo, los primeros sistemas desarrollados se centraron principalmente en proporcionar mecanismos para fomentar la comunicación, olvidándose de los aspectos pedagógicos.

En la actualidad este problema se está subsanando gracias a nuevos sistemas que aparte de mecanismos de comunicación ofrecen un apoyo pedagógico. Por ejemplo GRACILE (Ayala and Yano, 1995) se basa en la teoría de la zona de desarrollo próximo de Vygotsky. El sistema EduAgents de Hietala (1996) utiliza dos tipos de agentes que aplican distintos enfoques, uno constructivista y otro conductista. No obstante, todavía queda una tarea pendiente, controlar que situaciones negativas similares a las previamente citadas no ocurran, y que el sistema se encargue de anularlas en caso de que aparezcan. Este es el objetivo del estudiante simulado que se presenta en este artículo. Concretamente, el estudiante simulado se “enfrenta” a tres situaciones negativas: conversaciones fuera de lugar, pasividad

y problemas en la resolución de ejercicios. El resto del artículo está estructurado de la siguiente forma: El apartado dos describe por qué se eligió un estudiante simulado para realizar esta tarea. La sección tres describe cómo trabaja dicho estudiante simulado en una aplicación real, HabiPro. Para ayudar a entender las actuaciones del estudiante simulado se explica brevemente el funcionamiento de HabiPro y su arquitectura. En el cuarto apartado se describe un experimento realizado para comprobar la eficiencia del estudiante simulado detectando las situaciones negativas. Finalmente, las conclusiones son presentadas en la sección quinta.

2. ¿Por qué un Estudiante Simulado?

El objetivo que se pretendía con este trabajo era monitorizar el comportamiento de los estudiantes con el fin de detectar situaciones negativas y corregirlas, pero sin que los alumnos se sintieran observados por un profesor real. La primera idea que surgió fue diseñar un profesor virtual que actuara cuando fuera necesario. Sin embargo, esta idea fue desechada al considerar que las personas, en general, trabajan y colaboran más espontáneamente cuando el jefe o el profesor no forma parte del grupo (Dillenbourg, 1999). El hecho de trabajar con un profesor, aunque sea virtual, podría disminuir el rendimiento de los alumnos, ya que como se demuestra en Goodman et al. (1998), la interacción disminuye cuando los alumnos trabajan con el jefe o un profesor. Además, cuando los alumnos reciben un consejo de un profesor no suelen cuestionar si es correcto o no, simplemente lo creen; en cambio, cuando una propuesta proviene de un compañero surge una discusión y una reflexión que ayuda a que el alumno experimente una aprendizaje significativo.

En consecuencia, el estudiante simulado que se presenta juega un rol de alumno que participa dando su opinión y sólo cuando detecta que pueden surgir situaciones negativas actúa tratando de evitarlas tomando el rol de un estudiante “responsable”.

Otros autores han usado con anterioridad la idea de estudiante simulado en sus aplicaciones: Chan (1991) desarrolló el primer sistema que usó un estudiante artificial, el cual realizaba las mismas tareas que el estudiante humano, ambos intercambiaban ideas sobre lo que un profesor les iba enseñando (Chan y Chou, 1995). Otro ejemplo es mostrado en Ramírez (1998), donde un estudiante artificial es enseñado por un alumno real. Sin embargo, ninguna de las aplicaciones existentes se centra en evitar las tres situaciones negativas que se han comentado.

3. El Estudiante Simulado en HabiPro

El estudiante simulado está integrado en una aplicación síncrona y distribuida llamada HabiPro. A continuación se describen algunos aspectos sobre el sistema, necesarios para entender posteriormente el comportamiento del estudiante simulado.

HabiPro pretende desarrollar en los alumnos buenos hábitos de programación. Como muestra la Figura 1, la interfaz de la aplicación está dividida en dos partes. En la parte derecha se encuentra un chat donde los alumnos, incluyendo el estudiante simulado, escriben su opinión. En la esquina inferior derecha hay una pequeña pantalla que indica los nombres de las personas que están conectadas al sistema en ese momento. En la parte de la izquierda hay una ventana principal donde se muestra el ejercicio que hay que resolver. Las tres ventanas más pequeñas que hay debajo son para que los alumnos escriban las soluciones que proponen. Si un estudiante quiere mandar a sus compañeros la solución que ha escrito en la ventana titulada “write your answer” debe pulsar el botón “send”. Entonces, los compañeros verán que debajo del rótulo “answer 2” ó “answer 3” ha llegado una aportación.

Los alumnos usan la ventana del chat para discutir qué solución de las propuestas (no es obligatorio que todos los alumnos propongan una solución) consideran la correcta. Cuando el grupo llega a un acuerdo, todos los alumnos deben presionar el botón “check” que hay al lado de la ventana que contiene

la solución que el grupo considera correcta. El sistema no evaluará la solución, indicando si es o no correcta, hasta que todos los alumnos pulsen el botón “check”. De esta forma se obliga a que los alumnos tengan que tomar una decisión y decantarse sólo por una solución. Además, se logra que todos los alumnos vayan siempre por la misma pregunta. En una versión previa era suficiente con que una persona pulsara el botón “check” para que el sistema comprobara la respuesta. Cuando la solución era correcta el sistema pasaba automáticamente a la siguiente pregunta en todos los computadores. Con frecuencia, los alumnos se quejaban porque no habían tenido tiempo para ver la solución que se había “chequeado” o para reflexionar sobre ella (Vizcaíno et al., 2000). En el caso de comprobar una solución errónea el sistema da la opción de intentarlo de nuevo o de consultar pistas.

3.1 Arquitectura de HabiPro

HabiPro está formado por una interfaz, la misma para todos los estudiantes, sean o no simulados; de hecho, las aportaciones que están etiquetadas como “Alumno3” en la Figura 1 fueron realizadas por el estudiante simulado.

HabiPro tiene un gestor de información que analiza las entradas recibidas y las distribuye entre los Modelos de Estudiantes (ME), el Modelo del Grupo (MG) y/o el modelo del estudiante simulado (en inglés Simulated Student Behaviour Model, SSBM).

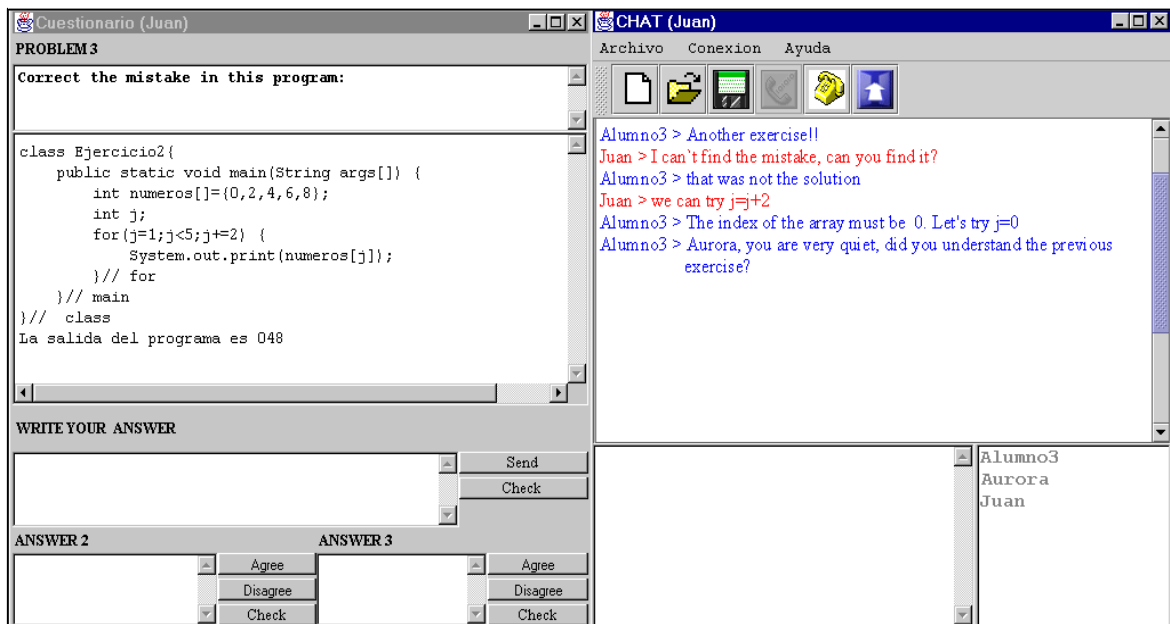


Figura 1. Interfaz de HabiPro

3.1.1 El Modelo del Estudiante

Un ME genérico está formado por un conjunto de entidades que contienen información sobre el alumno. La información que un ME almacena depende de los objetivos del sistema. El ME que hemos diseñado contiene las siguientes entidades:

Frecuencia de interacción: Número de veces que el alumno interacciona con el sistema. Puesto que la aplicación permite dos tipos de interacción, a través del chat o de la ventana de respuestas, el sistema controla cuántas veces los alumnos intervienen en cada tipo. Es importante distinguir entre los dos tipos para poder determinar qué rol desempeña cada alumno. Por ejemplo, un alumno puede introducir soluciones a menudo en la venta de respuestas pero quizás no use demasiado el chat, caso en el cual el alumno no se debe considerar pasivo.

Densidad de interacción: La densidad de interacción es el número de letras escritas en el chat por un alumno en cada ejercicio. Es decir, cuando el sistema pasa al siguiente ejercicio el contador de densidad de interacción de cada alumno es inicializado a cero. Este hecho evita que un estudiante que estuviera comenzando una conducta pasiva fuera ignorado por el estudiante simulado por tener una alta densidad de interacción en los primeros ejercicios.

La densidad de interacción es un indicador muy consultado por el estudiante simulado a la hora de detectar alumnos pasivos. Imaginemos que un alumno tiene una frecuencia de interacción en el chat normal pero lo utiliza solo para aportaciones muy breves del estilo “sí”, “no”, “ok”; en este caso el bajo valor de densidad de interacción le indicará al estudiante simulado que un alumno puede tener una conducta pasiva.

Nivel de conocimiento: Una de las ventajas de que el sistema tenga una ventana de respuesta por alumno es que le permite averiguar el conocimiento que cada uno tiene.

Cada vez que un estudiante propone una solución, se compara con el conjunto de posibles soluciones correctas ubicadas en el módulo llamado Conocimiento del Dominio (se describe en el siguiente apartado). Si la solución es correcta se le da una puntuación de 1 punto. En el caso de aproximarse a la solución, por ejemplo, si la respuesta correcta es “ $j < 3$ ” y se propone “ $j < 2$ ” se asignaría una puntuación menor, por ejemplo 0.5 puntos y 0 para otras soluciones más alejadas de la solución correcta (esta puntuación sólo es consultada por el estudiante simulado, los alumnos la desconocen).

Con frecuencia los alumnos con comportamientos pasivos tienen un nivel de conocimiento bajo: en

este caso el estudiante simulado trata de ayudarles explicando las aportaciones que él hace o incluso simula que él no ha entendido la propuesta de otro compañero y le pregunta directamente con el objetivo de que los alumnos con menos conocimiento aprendan. En la Figura 1 la última aportación del estudiante simulado muestra otra posible intervención de éste preguntándole directamente a una alumna pasiva si entendió el ejercicio anterior.

3.1.2 El Modelo del Grupo

El MG se define como una forma de representar las características que definen al grupo como un todo (Paiva, 1997).

El MG diseñado para HabiPro contiene las siguientes entidades:

Interacción del Grupo: Se calcula sumando todas las frecuencias de interacción individuales y denota la participación global del grupo. Este indicador es útil para comparar el grado de participación de cada estudiante con respecto a todo el grupo. El valor de la interacción del grupo dividido entre el número de personas representa la media de interacción del grupo, factor utilizado para detectar estudiantes con una frecuencia de interacción baja, es decir, por debajo de la media.

Densidad del Grupo: Se calcula sumando todas las densidades de interacción. La densidad del grupo es usado por el estudiante simulado para detectar los roles de los alumnos. Por ejemplo, si un grupo tiene una densidad de grupo de 100 y un estudiante A de 75 y el estudiante B de 25, se presupone que el estudiante A está jugando un papel de líder, o al menos es mucho más activo que B, por lo que este último podría ser un candidato a alumno pasivo. Este indicador también se puede usar para comparar el nivel de comunicación de distintos grupos.

Conocimiento del Grupo: Se infiere a partir de los acuerdos del grupo. Las soluciones que son comprobadas por el sistema han sido elegidas por todo el grupo, por lo que dichas soluciones son las que indican el conocimiento que el grupo tiene.

Preferencias: El sistema, con la finalidad de adaptarse a las preferencias de los usuarios, detecta en qué tipo de ejercicios los alumnos obtienen mejores rendimientos y qué tipo de ayuda prefieren.

3.2 El SSBM

El SSBM es el componente más importante de la arquitectura puesto que determina cómo debe actuar el estudiante simulado.

El SSBM está formado por cinco componentes (Figura 2): el Detector de Problemas, el Módulo

Pedagógico, el Historial de la Sesión, el Conocimiento sobre el Dominio y el Generador de Acciones.

Conocimiento sobre el dominio: Contiene información sobre la materia a aprender, por ejemplo los ejercicios y sus soluciones correctas y aproximadas. En caso de querer usar HabiPro en otro dominio, por ejemplo matemáticas, el contenido de este módulo tendría que ser modificado.

Detector de Problemas: A través de la información recibida desde los modelos del estudiante, el modelo del grupo y el conocimiento sobre el dominio el Detector de Problemas comprueba si está surgiendo alguna situación negativa. Este módulo (ver Figura 2) está formado por tres subcomponentes: El Detector de Conversaciones “off-topic”, el Detector de Problemas de Aprendizaje y el Detector de Pasividad. El primero controla conversaciones fuera de lugar (en inglés “off-topic”). A partir de ahora siempre nos referiremos a ellas como conversaciones “off-topic”. El segundo controla el proceso de aprendizaje del grupo. El último módulo controla la participación de cada estudiante con el objetivo de detectar comportamientos pasivos. La ventaja de tener módulos separados es que se pueden modificar individualmente y además se podrían añadir nuevos o cambiar la funcionalidad de uno sin que interviniera en el funcionamiento del resto de los módulos.

Módulo Pedagógico: Indica qué estrategias debe realizar el estudiante simulado para evitar situaciones conflictivas que previamente fueron detectadas por el Detector de Problemas. Antes de proponer una estrategia, el Módulo Pedagógico analiza la naturaleza del problema, las características del grupo y del alumno/s involucrado/s en éste y el Historial de la Sesión, el cual almacena todas las intervenciones realizadas tanto por los estudiantes reales como por el estudiante simulado. Al tener

almacenadas todas las intervenciones del estudiante simulado, el sistema puede controlar que no repita las mismas frases, evitando así parecer un robot que siempre reacciona igual ante comportamientos semejantes.

El Módulo Pedagógico puede, por ejemplo, indicar la conveniencia de motivar a un alumno que tiene una conducta pasiva, o de reforzar el aprendizaje de un concepto que los alumnos no han entendido totalmente. Estos ejemplos están relacionados con los problemas de estudiantes pasivos y el aprendizaje de los alumnos, pero también puede referirse a estrategias que intenten evitar conversaciones “off-topic”. La Figura 2 muestra tres pequeños submódulos del Módulo Pedagógico que contienen diferentes estrategias para cada tipo de problema.

Generador de Acciones: El estudiante simulado puede realizar diferentes acciones para cada estrategia que el Módulo Pedagógico posee. Según la información que el Generador de Acciones reciba desde el Historial de la sesión, el ME, el MG, el Módulo Pedagógico y el dominio, decidirá realizar una acción u otra. Por ejemplo, si el Detector de Problemas detectara que hay un alumno pasivo, el Módulo Pedagógico podría decidir que el estudiante simulado debe invitar al alumno a tomar parte en la resolución de los problemas, pero será el Generador de Acciones el que decida cómo concretamente el estudiante simulado debe invitar al alumno, por ejemplo planteándole una pregunta.

Puesto que cuando se describió el ME y el MG se esbozó cómo el estudiante simulado detecta conductas pasivas, el próximo apartado se centra en explicar el comportamiento del estudiante simulado ante “off-topic conversations” y problemas de aprendizaje.

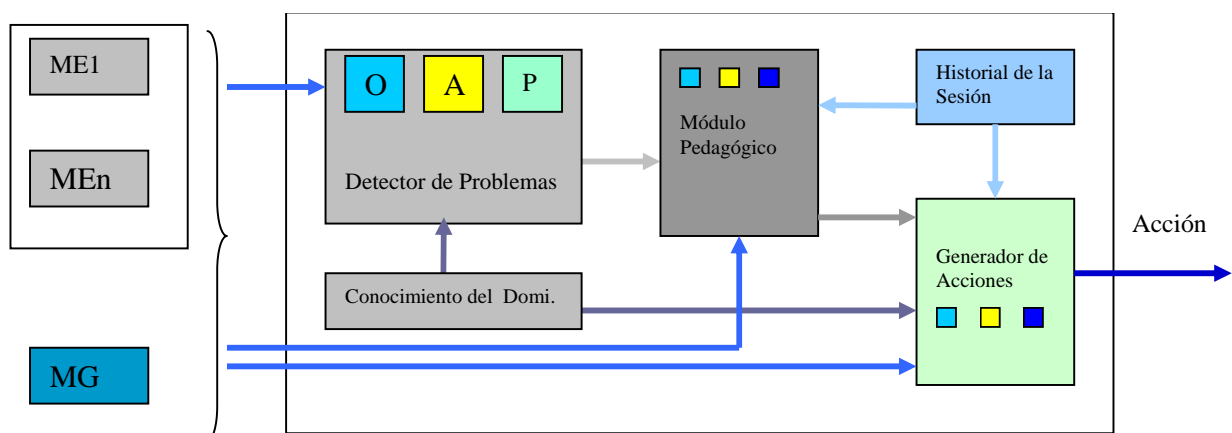


Figura 2. Arquitectura del SSBM

3.3 Funcionamiento del Estudiante Simulado

Para controlar si surgen conversaciones “off-topic”, el estudiante simulado utiliza un componente del Detector de Conversaciones “off-topic” llamado Procesador de Conversaciones, el cual consulta diferentes bases de datos:

- Una base de datos general que contiene palabras relacionadas con la resolución de problemas en programación, por ejemplo: problema, solución, o bucle.
- Una base de datos específica que almacena palabras que se usan en la programación con Java, por ejemplo: main, try, o catch.
- Una base de datos llamada “playful database” la cual guarda palabras relacionadas con conversaciones típicas de los alumnos cuando hablan de conversaciones “off-topic”. Por ejemplo, fútbol en el caso de alumnos españoles. Por supuesto, las bases de datos pueden ser modificadas dependiendo del entorno donde HabiPro sea utilizado.

Cada vez que los estudiantes escriben una frase en la ventana del chat, el Procesador de Conversaciones comprueba si la frase tiene palabras contenidas en la base de datos específica. En el caso de que ninguna palabra sea encontrada en esta base de datos, el segundo paso es buscar en la base de datos general. Si tampoco coincide ninguna palabra, el estudiante simulado supone que una conversación “off-topic” está comenzando. Cuando esta situación ocurre en una o dos frases consecutivas el estudiante simulado no interviene puesto que pequeñas conversaciones “off-topic” ayudan a que los alumnos se relajen y se encuentren más cómodos en el grupo. Sin embargo, si la conversación tiene más de tres aportaciones el estudiante simulado debe actuar.

El objetivo de la “playful database” es que el estudiante simulado pueda “saber” de qué están hablando los alumnos, de forma que él pudiera hacer alguna aportación sobre el tema y cerrar después la conversación para redirigir la atención de los alumnos, nuevamente hacia los ejercicios. Por ejemplo, si se detecta que los alumnos están hablando de fútbol el estudiante simulado podría decir: “A mí no me gusta el fútbol, vamos a seguir con los ejercicios”.

Cuando no es posible saber sobre qué tema están hablando, el estudiante simulado suele intentar captar la atención de los alumnos sugiriendo una solución al ejercicio. Nuestro grupo realizó un experimento con profesores tratando de imitar los roles del estudiante simulado. El experimento

demostró que las intervenciones más exitosas terminando conversaciones “off-topic” eran aquellas que proponían soluciones a los ejercicios, ignorando que los alumnos estaban realizando comentarios sobre otros temas (Vizcaíno y Prieto, 2000). Por esta razón, para cada ejercicio, el estudiante simulado tiene un conjunto de frases que puede utilizar para cerrar una conversación “off-topic”.

El estudiante simulado también está encargado de actuar cuando los alumnos tienen problemas evidentes para encontrar la solución de un ejercicio. El Detector de Problemas de Aprendizaje monitoriza cuántas propuestas erróneas se comprueban por ejercicio y en qué grado se aproximan a la solución correcta. Cuando los alumnos comprueban sucesivamente dos soluciones erróneas el estudiante simulado interviene proponiendo en el chat alguna solución parecida a la real. Por ejemplo, puede indicar “yo no creo que el error del programa esté ahí, quizás el fallo sea del bucle”. Para cada ejercicio el estudiante simulado posee un conjunto de pistas que puede utilizar según la ocasión.

4. Evaluación

Con el objetivo de comprobar si el estudiante simulado actuaba cuando debía y controlaba las situaciones problemáticas se llevó a cabo un experimento con cuarenta y cuatro alumnos matriculados en el primer curso de la asignatura Introducción a la Programación de la carrera de Computación de Ciudad Real. Grupos de dos alumnos más el estudiante simulado tenían que resolver distintos problemas de programación durante una hora. Todas las actuaciones eran grabadas en logs que posteriormente se estudiaron para analizar el comportamiento del estudiante simulado.

El análisis de los datos demostró que las actuaciones del estudiante simulado para ayudar a encontrar la solución fueron útiles en un 93.8 % de los casos (61 veces de 65), en el resto de casos los alumnos ignoraron la sugerencia que el estudiante simulado proponía.

Los históricos de las conversaciones mostraron que los alumnos comenzaron catorce veces conversaciones “off-topic”. Doce de ellas fueron detectadas por el estudiante simulado y pudo detener once, por lo tanto realizó correctamente su trabajo en un 78.6 % de los casos. Los datos mostraron que en general una intervención del estudiante simulado era suficiente para devolver la atención de los alumnos a los problemas.

Con respecto a los alumnos pasivos, el estudio demostró que siempre que el estudiante simulado intervenía para motivar a un alumno que supuestamente tenía comportamiento pasivo éste comenzaba a aportar ideas. Además, sólo en un caso un alumno pasivo repitió su conducta y el estudiante simulado volvió a actuar.

Un experimento similar se llevó a cabo, con los mismos alumnos, pero usando HabiPro sin el estudiante simulado. Este experimento demostró que los alumnos resolvían más ejercicios usando la versión que contenía el estudiante simulado.

5. Conclusiones

El aprendizaje colaborativo proporciona numerosos beneficios, sin embargo, para que éstos tengan lugar deben evitarse determinadas situaciones negativas. Este artículo demuestra el modo en que un estudiante simulado puede ayudar a detectar y controlar determinadas situaciones no deseables, concretamente, conversaciones “off-topic”, comportamientos pasivos en los alumnos y problemas para encontrar la solución de los ejercicios. Puesto que el diseño del estudiante simulado es modular, podrían añadirse otros módulos encargados de controlar situaciones diferentes.

Por otra parte, el experimento demostró que las actuaciones del estudiante simulado eran eficientes en la mayoría de los casos. Además, los alumnos resolvían más ejercicios gracias a la colaboración del estudiante simulado. Éste era un resultado predecible, puesto que si el estudiante simulado actuaba correctamente evitando que los alumnos pierdan tiempo conversando de otros temas o intentando un gran número de veces soluciones erróneas a los problemas, el número de ejercicios que pueden resolver correctamente aumenta, obteniéndose mayor aprovechamiento del aprendizaje colaborativo.

Agradecimientos

Mi agradecimiento a Ben du Boulay, sin cuya colaboración este trabajo nunca se hubiera realizado. También quisiera agradecer a los revisores anónimos y a Macario Polo sus útiles consejos para mejorar este artículo.

Referencias

- Ayala, G., and Yano, Y.** GRACILE: A Framework for Collaborative Intelligent Learning Environments, *Journal of the Japanese Society of Artificial Intelligence*, Vol.10. No. 6. pp: 156-170. 1995.
- Blaye, A. and Light P.H.** Collaborative Problem Solving with HyperCard: The Influence of Peer Interaction on Planning and Information Handling Strategies. In *Computer Supported Collaborative Learning*, O'Malley (Ed.). Heidelberg: Springer-Verlag. 1995.
- Chan, T.-W.** Integration-kid: a learning companion system. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, Sydney, Australia, Morgan Kaufmann Publishers, Inc., 1094-1099, 1991.
- Chan, T.-W., and Chou, C.-Y.** Simulating a Learning Companion in Reciprocal Tutoring Systems. In *Conference on Computer Support for Collaborative Learning (CSCL'95)*. Lawrence Erlbaum Associates, Inc, 1995.
- Damon, W. and Phelps, E.** Critical distinctions among three approaches to peer education. *International Journal of Educational Research*, 13, 9-19. 1989.
- Dillenbourg, P.** Introduction: What Do You Mean By “Collaborative Learning”? Dillenbourg (Ed.). *Collaborative Learning Cognitive and Computational Approaches*. Elsevier Science, 1999.
- Goodman, B., Soller, A., Linton, F., and Gaimari, R.** Encouraging Student Reflection and Articulation using a Learning Companion. *International Journal of Artificial Intelligence in Education*, 9 (3-4), 1998.
- Hietala, P.** A prototype for a social learning system with intelligent agents. In *proceedings European Conference on Artificial Intelligence in Education*. Fundação Calastre Gulbenkian, Lisboa, Portugal, September. <http://www.uta.fi/ph/papers/EuroAIED96/EuroAIED96.html>, 1996.
- Lund, K., Baker, M., and Baron, M.** Modelling Dialogue and Beliefs as a Basis for Generating Guidance in a CSCL Environment. In *Proceedings of the Intelligence Tutoring Systems-96 Conference*, Montreal 206-214.1996.
- Ramirez, J.A.** Teaching a Learning Companion. In Ayala, G. (Ed.). *Proceedings of the International Workshop Current Trends and Applications of Artificial Intelligence in Education*. The Fourth Word Congress on Expert Systems, pp 83-89. Monterrey, México. 1998.
- Paiva, A.** Learner Modelling for Collaborative Learning Environments. In du Boulay and

- Mizoguchi (Eds.), *Proceedings of AI-ED 97*. Kobe, Japan, IOS Press, 215-230, 1997.
- Soller, A., Goodman, B., Linton, F., Gaimari, R.** Promoting Effective Peer Interaction in an Intelligent Collaborative Learning System. *Proceedings of the 4th International Conference on Intelligent Tutoring Systems (ITS 98)*. San Antonio, Texas, pp 186-195, 1998.
- Vizcaíno, A., and Prieto, M.** Examining the Effectiveness of New Technology in High School. In *Proceedings of the International Symposium of Technologies of Information and Communication in Engineering and Industry*. TICE'2000. Troyes, France, October 2000, pp355-360, 2000.
- Vizcaíno, A., Contreras, J., Favela, J., and Prieto, M.** An Adaptive, Collaborative Environment to Develop Good Habits in Programming. In *Proceedings of the 5th International Conference on Intelligent Tutoring Systems*. (ITS'00) LNCS Montreal, Canada, pp 262-271, 2000.
- Vizcaíno, A.** Enhancing Collaborative Learning Using a Simulated Student. PhD thesis. Universidad de Castilla-La Mancha. ISBN:84-84-27-176-5: ProQuest: Information and Learning. 2001.