

Ambiente inteligente de aprendizaje con manejo afectivo para Java

Ramón Zatarain Cabada, María Lucía Barrón Estrada,
Francisco González Hernández, Raúl Oramas Bustillos

Instituto Tecnológico de Culiacán, Culiacán, Sinaloa,
México

{rzatarain, lbarron}@itculiacan.edu.mx

Resumen. En este artículo se presenta un ambiente inteligente de aprendizaje con manejo afectivo para Java llamado Java Sensei. El sistema está diseñado para ayudar a los estudiantes de programación a reforzar distintas áreas del conocimiento sobre Java. El sistema evalúa aspectos como el estado cognitivo y afectivo del estudiante para tomar las estrategias de intervención que realizara agente pedagógico, así como realiza procesos de adaptabilidad por medio de un sistema de recomendaciones. El manejo pedagógico se realizó con una perspectiva Example-Tracing con manejo afectivo.

Palabras clave: entornos inteligentes de aprendizaje, sistemas tutores inteligentes, computación afectiva, lenguajes de programación, redes neuronales, sistemas lógicos difusos, sistemas de recomendación.

1. Introducción

La enseñanza y aprendizaje de la programación ha sido un reto complejo para todas las universidades en cualquier país. Algunos factores que contribuyen a su complejidad son: los métodos de enseñanza empleados por los instructores, los métodos de estudio empleados por los estudiantes así como sus habilidades y actitudes, el proceso de aprendizaje único que tiene la programación y la influencia psicológica de parte de la sociedad hacia los estudiantes [1, 2, 3, 4].

Algunas investigaciones han propuesto y desarrollado metodologías y herramientas que ayudan a los estudiantes a aprender como programar. Por ejemplo la ACM tiene el Grupo de Interés Especial sobre la Educación en Ciencias Computacionales (SIGCSE por sus siglas en inglés) donde se discuten problemas relacionados al desarrollo, implementación y/o evaluación de programas y planes de estudio [5].

Hoy en día en el campo de la inteligencia artificial, los Sistemas Tutores Inteligentes (ITS por sus siglas en inglés) intentan simular a un tutor humano proveyendo instrucciones personalizadas que toman en cuenta no solo aspectos cognitivos del estudiante [6] si no también elementos afectivos [7].

Este artículo presenta la implementación de un Entorno Inteligente de Aprendizaje (ILE por sus siglas en inglés) para el aprendizaje del lenguaje de programación Java. Esta herramienta es nombrada *Java Sensei*, la cual intenta crear un entorno flexible e interactivo que considere los estados afectivos y cognitivos de los estudiantes. La organización del artículo es la siguiente: en la sección 2, describimos el diseño e implementación del sistema con su arquitectura, componentes y algoritmos. Los resultados y pruebas de la aplicación se muestran en la sección 3 y las conclusiones y trabajos futuros en la sección 4.

2. Diseño e implementación de Java Sensei

El ILE *Java Sensei* consiste de siete componentes: *Plataforma Afectiva*, *Módulo del Estudiante o Usuario*, *Agente Pedagógico*, *Módulo de Adaptación*, *Módulo de Enseñanza (Pedagógico)*, *Módulo del Experto* y *el Motor ITS (Sistema de Tutoría Inteligente/Afectivo)*.

2.1. Plataforma afectiva

En el módulo afectivo se creó una plataforma emocional basada en el ITS AutoTutor [9] donde se implementó un sistema lógico difuso y una red neuronal de propagación hacia atrás para el reconocimiento de emociones. La plataforma emocional representa un modelado tanto del tutor como del estudiante. Ambos componen las variables de entrada y salida que necesita el sistema lógico difuso. Estos modelos y variables son descritos a continuación.

Modelo del Estudiante: El modelo del estudiante representa la parte cognitiva y emocional del estudiante. Este consiste de dos aspectos cognitivos y dos emocionales explicadas en la Tabla 1.

Tabla 1. Variables de entrada que representan el modelo del estudiante.

Nombre de la variable de entrada	Definición
Emoción Actual	Es la emoción actual del estudiante cuando responde un ejercicio. La emoción es obtenida a través de la aplicación de reconocimiento emocional. Los posibles valores son: Feliz, Enojado, Triste, Sorprendido y Neutral.
Emoción Previa	Es la emoción previa del estudiante. Sus valores son del mismo tipo que en la Emoción Actual.
Habilidad Global del Estudiante	Es la eficiencia que posee el estudiante con respecto al curso en su totalidad.
Calidad de la Respuesta Actual	Es la eficiencia que tiene el estudiante con respecto al ejercicio que está resolviendo.

Acciones del Tutor: Las acciones del tutor representan la respuesta emocional de un tutor humano y están representadas como variables de salidas del sistema lógico difuso que son usadas por el componente del *Agente Pedagógico* y el *Motor ITS*. Ambos serán explicados más adelante. Las variables están definidas en la Tabla 2.

Tabla 2. Variables que representan las acciones del tutor.

Nombre de la variable de entrada	Definición
Retroalimentación	Este es un mensaje que da una perspectiva positiva, neutral o negativa del estudiante con respecto a su habilidad global y la calidad de sus respuestas (ambas variables mencionadas anteriormente). Representa la retroalimentación que daría un tutor humano cuando él sabe si el progreso académico es correcto o no.
Respuesta empática emocional	Son frases recibidas por el estudiante de parte del tutor estas dependerán de los estados emocionales presentados por el estudiante. Están categorizadas en: neutral, encantado, sorprendido, compasivo y escéptico
Expresión Facial	Es la expresión facial que tendrá que usar el Agente Pedagógico. Categorizadas en Encantado, Escéptico, Sorprendido y Neutral.
Intervención	Este valor representa si el agente pedagógico debe realizar una intervención (aparecer en pantalla) al estudiante.

2.2. Sistema lógico difuso y red neuronal

El sistema lógico difuso busca representar la forma en la que un tutor maneja varios escenarios relacionados con la situación emocional y cognitiva de un estudiante [10]. Estas situaciones tienen que hacerse con acciones predefinidas.

El sistema lógico difuso usa los valores emocionales obtenidos de la red neuronal la cual reconoce los estados afectivos del estudiante cuando este trabaja con el ILE *Java Sensei*. El sistema lógico difuso trabaja con cuatro variables de entrada difusa y tres de salida. Cuenta con un total de 144 reglas difusas construidas. Para definir esas reglas se usó el Lenguaje de Control Difuso, un lenguaje estandarizado para la creación de reglas difusas. La implementación fue hecha con *jFuzzyLogic* [11].

El sistema de reconocimiento emocional fue construido en tres pasos: El primer paso fue una implementación de un extractor de características de imágenes faciales. En el segundo paso se usó este extractor sobre un corpus de datos para poder obtener los datos y entrenar la red neuronal. Usamos algoritmos basados en Java implementados en NeuroPH [21] para realizar el entrenamiento, el clasificador usado en la red neuronal fue propagación del error hacia atrás. En el tercer paso se integró la red neuronal con el sistema lógico difuso. Ambos son parte del ILE. Para el entrenamiento y prueba de la red neuronal usamos el corpus RAFD (Radboud Faces Database), el cual es una base de datos con 8040 expresiones faciales diferentes que contienen un conjunto de 67 modelos incluyendo hombres y mujeres. Una vez que el estado emocional es extraído del estudiante, este es enviado al sistema lógico difuso. La Fig. 1 muestra los pasos seguidos en el reconocimiento y retroalimentación afectivos del sistema. En el primer paso del ILE este recibe los datos y colección de imágenes del estudiante desde el navegador. En el segundo paso el ILE inicia un ciclo en el cual las imágenes del rostro del estudiante son enviadas a la red neuronal y decide al final del ciclo la emoción actual. Esta emoción actual es obtenida basándose en la emoción más repetida del estudiante. En el tercer paso el ILE envía el modelo del estudiante actualizado al sistema lógico difuso el cual ejecuta las reglas difusas para obtener las salidas de las acciones del tutor. En este último paso, las acciones del tutor son enviadas al navegador para construir las acciones del *Agente Pedagógico*.

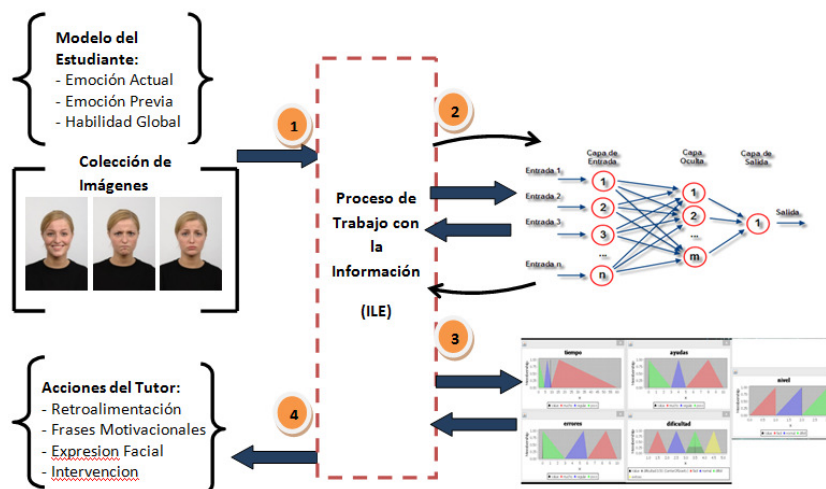


Fig. 1. Pasos seguidos por el sistema lógico difuso y la red neuronal.

2.3. Agente pedagógico

El agente pedagógico es responsable de transmitir los mensajes y expresiones que un tutor humano debería realizar. Para esto se le ha provisto con expresiones faciales y diálogos con los que pueda comunicarse con el estudiante. Esas expresiones están

basadas en las expresiones faciales investigadas por Ekman [12]. La posición de las cejas, boca y apertura de los ojos son tomadas en cuenta. Estas posiciones se utilizan como entrada en el programa FaceGen Modeller Free [13]. En la Fig. 2 se observan las cuatro expresiones faciales hechas para el agente pedagógico de *Java Sensei*. Estas expresiones están basadas en el trabajo realizado con el agente pedagógico de Auto-Tutor [10].

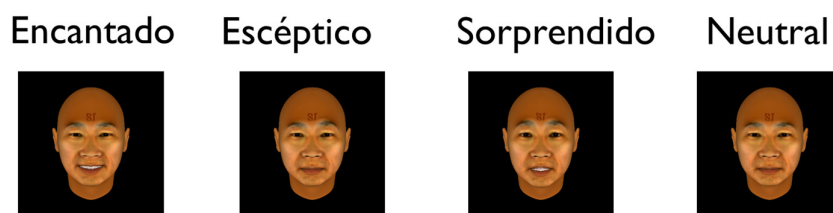


Fig. 2. Expresiones faciales del *Agente Pedagógico*.

2.4. Módulo de adaptación

Una característica de los ILE es que estos deben poder adaptarse al estudiante [8]. *Java Sensei* colecciona los puntajes producidos por los estudiantes al visitar los recursos o finalizar los ejercicios con los que cuenta el ILE. El puntaje de los ejercicios se mide usando la escala de Likert [14] donde el valor de uno indica que hay un desagrado total y el valor cinco indica que hay un agrado total. Cuando el estudiante ingresa por primera vez al sistema todos los ejercicios se inicializan con un valor de dos, con esto se coloca un valor que represente neutralidad en el gusto del estudiante. Para poder utilizar estos puntajes en la adaptabilidad creamos dos sistemas de recomendación uno para ejercicios y otro para recursos usando Apache Mahout [15]. El sistema implementa el enfoque K-Nearest Neighbor (k-NN) y la Correlación de Pearson. Entre más usuarios participen en el sistema, se generaran mejores recomendaciones y de este modo se reflejara una mejor adaptación hacia cada usuario.

2.5. Módulo pedagógico

El sistema usa un modelo pedagógico conocido como Problem-solving [6] en este los estudiantes aprenden resolviendo problemas los cuales cuentan con una cierta estructura. El sistema usa tres tipos de estrategias para solucionar los problemas. El tipo 1 es usado para evaluar conceptos teóricos con ejercicios de opción falso y verdadero. El tipo 2 presenta piezas de código que el estudiante tendrá que seleccionar para completar un programa completo y generar una salida deseada. El tipo 3 es una combinación de los dos tipos previos y permite crear ejercicios más complejos que involucren “n” número de pasos para alcanzar la solución.

2.6. Módulo del experto

Este modelo contiene el conocimiento del experto que el estudiante quiere aprender y practicar. El sistema representa el conocimiento usando archivos JSON (una

variación de los archivos XML). La representación del conocimiento usa la teoría de espacios del conocimiento [19]. El modelo del experto se representa en seis habilidad básicas que el estudiante deberá dominar para así ser este modelado por un gráfico que representa el espacio del conocimiento. Estas habilidades son: Introducción a Java, Variables y Cálculos, Estructuras de Selección, Estructuras de Repetición, Métodos y Arreglos.

2.7. Motor ITS

Un ILE combina técnicas hipermedia con otros métodos usados por los ITS. Se aplicó la teoría Example-Tracing [16,17] con el fin de implementar la parte ITS en *Java Sensei*. Los Example-Tracing ITS tienen la ventaja de proveer paso por paso una guía para los estudiantes y proveer múltiples estrategias para poder resolver un problema, esto incluye manejo soluciones óptimas y sub-óptimas como conceptos erróneos comunes. Los ejercicios Example-Tracing pueden ser fácilmente adaptados con el modelo pedagógico Problem-solving (Resolución de Problemas) y ser representados por un grafo que es atravesado por el estudiante.

Una de las principales contribuciones de este trabajo es la integración de información cognitiva y emocional en un grafo de comportamiento el cual es usado por el agente pedagógico para mostrar emoción y empatía hacia el estudiante. Con esto se crea un concepto llamado Affective Example-Tracing. La Fig. 3 muestra parte de un grafo de comportamiento.

A continuación, se describe el algoritmo que se implementó en el Affective Example-Tracing [18]:

- a) El ILE carga el archivo JSON del ejercicio que se va a resolver.
- b) Dentro del archivo JSON, un grafo de comportamiento es creado y el paso inicial es ejecutado.
- c) Los estudiantes navegan por el gráfico (a través de disparos de eventos como el clic del ratón), teniendo diferentes opciones que puedes realizar y diferentes caminos a elegir. Las opciones son:
 - i. Paso Inicial (PI)
 - ii. Paso Erróneo (PE)
 - iii. Paso Optimo (PO)
 - iv. Paso Sub-Optimo (PSO)
 - v. Paso Final Optimo (PFO)
 - vi. Paso Final Sub-Optimo (PFSO)
- d) Cada paso lanza dos eventos:
 - i. Evaluación de la emoción actual.
 - ii. Evaluación de la acción del tutor.

Cada vez que un estudiante escoge alguna de las opciones en el grafo, el motor llama al sistema lógico difuso para obtener las acciones del tutor (mensajes de retroalimentación, realizar una intervención, etc.) y así obtener la información emocional para que el *Agente Pedagógico* pueda construir su intervención y el *Motor ITS* construir la pertinente representación gráfica. En la figura 4 se ilustra un ejemplo en el

cual se observa como de lado izquierdo de la figura los estudiantes resuelven un problema en un camino óptimo y de lado derecho el estudiante llega a un camino erróneo.

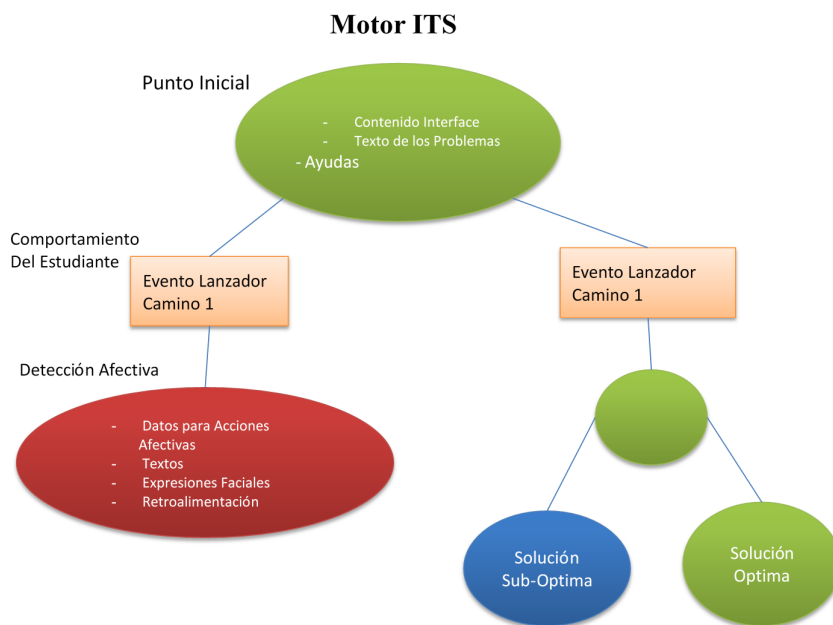


Fig. 3. Ejemplo de grafo de comportamiento.

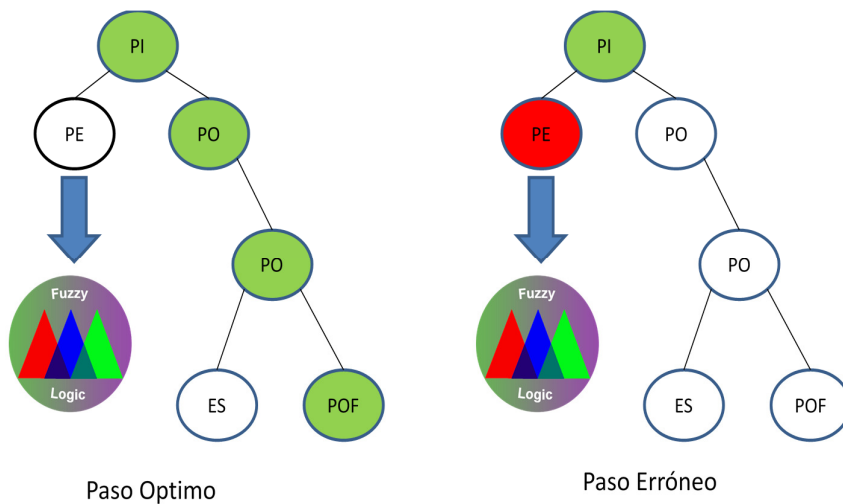


Fig. 4. Ejemplo de navegación a través del árbol.

3. Resultados y comparaciones

Para evaluar el ILE Java Sensei se consideran dos aspectos: la funcionalidad del software y la utilización del sistema con un grupo de estudiantes. Para evaluar la funcionalidad del software se comparó con otros tres sistemas similares al ILE Java-Sensei. Estos sistemas son JavaTutor System [23], JITS [20] y jLatte [24]. Para este trabajo, hemos decidido evaluar nueve características de un ILE (ITS) [6] utilizando una matriz ponderada utilizando la escala de Likert [14] de 1 a 5, donde 1 representa una característica incompleta y 5 una característica completa. La Tabla 3 muestra los resultados obtenidos de la evaluación de la funcionalidad del software.

Tabla 3. Tareas realizadas en la fase de pruebas.

Característica	JavaSensei	JavaTutor	JITS	jLatte
Generatividad	3	5	4	4
Conocimiento experto y del estudiante	5	5	5	5
Iniciativa mixta	2	5	1	1
Aprendizaje Interactivo	4	5	4	5
Modelo instruccional	5	5	5	5
Auto-mejora	1	1	1	1
Manejo de emociones	4	5	1	1
Sistema web	5	5	1	1
Sistema móvil	5	1	1	1
Totales:	34	37	23	24

Los sistemas evaluados cumplen con casi todas las características de un sistema ILE/ITS a excepción de las características 3 (iniciativa mixta) y 6 (auto-mejora). La característica 7 (el manejo de emociones) es un área reciente por lo tanto son pocos los sistemas que incorporan esta característica. Lo mismo para las características 8 y 9. Se espera que un futuro los ITS desarrollados en los últimos años migren como sistemas Web y proporcionen soporte para dispositivos móviles.

La evaluación del uso del sistema se llevó a cabo una prueba en el Instituto Tecnológico de Culiacán, con el grupo ESA de 8:00 a 9:00 A.M., de la asignatura Programación Orientada a Objetos (POO), en donde se buscó medir el impacto del ILE Java Sensei en el tema de herencia en la POO. La selección de los estudiantes no se realizó de manera aleatoria, sino por disponibilidad de los participantes. Para acceder al ILE, los estudiantes utilizaron un navegador web desde con soporte de HTML 5 y que cuente con cámara frontal esta podría ser una computadora de escritorio, laptop o teléfono inteligente, ingresando al sistema desde su cuenta de Facebook.

Siguiendo una de las metodología propuesta por Woolf [6] para la evaluación del ITS se diseñó una prueba donde se aplicaría un Pre-Test (Examen Diagnostico), posteriormente habría una sesión de uso del programa y un examen Post-Test. Para esto primero se impartió una introducción al ILE Java Sensei al grupo anteriormente mencionado y al día siguiente se aplicó un examen diagnóstico para evaluar aspectos teó-

ricos y prácticos sobre: conceptos herencia, tipo de herencia, atributos, métodos heredados, palabra clave extends, herencia simple, diferencia entre métodos overload-override, final, static, polimorfismo y la utilización del método super().

Posteriormente se dividió el grupo en dos partes, dónde el Grupo 1 sería el grupo experimental y utilizaría el ILE JavaSensei y el Grupo 2 representaría el grupo de control. La tabla 4 resume las actividades realizadas.

Tabla 4. Tareas realizadas en la fase de pruebas.

Fase	Tarea	Fecha	Participantes	¡Latte
1	Introducción al Ambiente ILE JavaSensei	24/03/2015	20	20 mins.
2	Examen Diagnóstico	25/03/2015	20	16 mins.
3	Utilización ILE JavaSensei	26/03/2015	10	50 mins.
4	Examen Post-Test	27/03/2015	20	20 mins.

Una vez que el Grupo 1 estudió el tema de herencia utilizando el ILE JavaSensei se procedió a aplicar el examen Post-Test a ambos grupos.

Al realizar el análisis del examen diagnóstico se obtuvo un 80% de aprobados y el 20% restante de reprobados. De manera similar, con el examen Post-Test se obtuvo un 85% de aprobados con el correspondiente 15% de reprobados. La figura 5 muestra los resultados obtenidos.



Fig. 5. Resultados de los exámenes diagnóstico y post-test.

De la gráfica anterior podemos observar una mejoría del 5% antes y después de los exámenes. Al realizar el análisis comparativo del Grupo 1 que utilizaron el ILE JavaSensei (ver tabla 4), el 20% mejoró su puntaje respecto al examen diagnóstico, el 40% no mostro cambio alguno y el 40% disminuyo su calificación.

4. Conclusiones y trabajo a futuro

Dados los resultados de las pruebas estadísticas realizadas, se puede considerar que la utilización de un Ambiente ILE tiene un efecto positivo en el aprendizaje de los

estudiantes. La variación negativa fue menor por lo que un ILE es una herramienta que ayuda a reforzar los distintos temas que el estudiante pueda estar cursando en su curso de programación, con las ventajas de la disponibilidad que un sistema web tiene.

Se espera a futuro poder adaptar un Módulo de colaboración social al ILE [22] ya que además de las emociones se ha comprobado la importancia de la interacción social de los estudiante, ya que esto lo ayuda a obtener herramientas para poder enfrentar mejor los retos que “desconoce” y tener un lugar donde pueda debatir sus ideas con respecto a los ejercicios y/o lecciones que está enfrentando.

Referencias

1. Matthíasdóttir, Á.: How to teach programming languages to novice students? Lecturing or not. In: International Conference on Computer Systems and Technologies-CompSysTech. (2006)
2. Jenkins, T.: On the difficulty of learning to program. In: Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences (2002)
3. Gomes, A. and A.J. Mendes: Learning to program-difficulties and solutions. In: International Conference on Engineering Education-ICEE (2007)
4. Bernard, M. and Bachu, E.: Enhancing the Metacognitive Skill of Novice Programmers through Collaborative Learning. In: Metacognition: Fundamentals, Applications, and Trends. (2015), Springer. p. 277–298
5. SIGCSE (2015) [cited 2015]; Available from: <http://www.sigcse.org/>
6. Woolf, B.P.: Building Intelligent Interactive Tutors. Amherst, Massachusetts, Estados Unidos: Morgan Kaufmann Publishers (2009)
7. Picard, R.W.: Affective computing. MIT press (2000)
8. Brusilovsky, P. and E. Millán: User models for adaptive hypermedia and adaptive educational systems. In: The adaptive web. Springer-Verlag (2007)
9. D’Mello, S., R.W. Picard, and A. Graesser: Toward an affect-sensitive AutoTutor. *IEEE Intelligent Systems*, 4, pp. 53–61 (2007)
10. D’Mello, S., et al.: AutoTutor detects and responds to learners affective and cognitive states. In: Workshop on Emotional and Cognitive Issues at the International Conference on Intelligent Tutoring Systems (2008)
11. Cingolani, P. and J. Alcalá-Fdez: jFuzzyLogic: a java library to design fuzzy logic controllers according to the standard for fuzzy control programming. *International Journal of Computational Intelligence Systems*, 6(sup1), pp. 61–75 (2013)
12. Ekman, P.: Facial Expressions. (1999)
13. Inversions, S.: FaceGen modeller: (Version 3.3) [computer software]. Toronto, ON: Singular Inversions (2008)
14. Likert, R.. A method of constructing an attitude scale. *Scaling: a sourcebook for behavioral scientists*. Chicago: Aldine, pp. 233–243 (1974)
15. Owen, S., et al.: Mahout in action. (2011)
16. Alevén, V., et al.: A new paradigm for intelligent tutoring systems: Example-tracing tutors. *International Journal of Artificial Intelligence in Education*, 19(2), pp. 105–154 (2009)
17. Alevén, V., et al.: The cognitive tutor authoring tools (CTAT): preliminary evaluation of efficiency gains. In: *Intelligent Tutoring Systems*, Springer (2006)
18. Kumar, R., et al.: Comparison of Algorithms for Automatically Building Example-Tracing Tutor Models. In: *Educational Data Mining* (2014)
19. Doignon, J. P., & Falzague, J. C.: Knowledge spaces. Springer (1999)

20. Wallis, M.: JavaTutor-A Remotely Collaborative, Real-Time Distributed Intelligent Tutoring System for Introductory Java Computer Programming. A Qualitative Analysis (2011)
21. Bruxella, J. Marv Dallfin, S. Sadhana, and S. Geetha: Categorization of Data Mining Tools Based on Their Types. *International Journal of Computer Science and Mobile Computing*, 3(3), pp. 445–452 (2014)
22. Soller, A.: Supporting social interaction in an intelligent collaborative learning system. *International Journal of Artificial Intelligence in Education (IJAIED)*, 12, pp. 40–62 (2001)
23. Grafsgaard, J. F., Wiggins, J. B., Vail, A. K., Boyer, K. E., Wiebe, E. N., & Lester, J. C.: The Additive Value of Multimodal Features for Predicting Engagement, Frustration, and Learning during Tutoring. In: *Proceedings of the 16th International Conference on Multimodal Interaction*, pp. 42–49, ACM (2014)
24. Holland, J., Mitrovic, A., & Martin, B.: J-LATTE: a Constraint-based Tutor for Java. (2009)