

Un Modelo Híbrido para el Entrenamiento Adaptativo Basado en Ejercicios

María de los Ángeles Alonso¹, Argelio V. de la Cruz², Yira Muñoz¹, Gonzalo A. Torres¹

¹Centro de Investigación en Tecnologías de Información y Sistemas

Universidad Autónoma del Estado de Hidalgo

²Innovaciones Tecnológicas Avanzadas SA de CV

marial, yira, torres{ @uaeh.edu.mx }, argelioc@intelsoft.com.mx

Resumen

En el presente trabajo se propone una estrategia de ejercitación basada en un modelo híbrido que integra técnicas de Inteligencia Artificial, Bases de Datos y Programación Orientada a Objetos, donde los conceptos fundamentales son: *Ejercicio Prototipo* y *Nivel de Complejidad*. El ejercicio prototipo permite optimizar el código almacenado de una gran variedad de ejercicios del mismo tipo, mientras que el nivel de complejidad garantiza la adaptación del entrenamiento de acuerdo al nivel de preparación del estudiante. Para lograr la funcionalidad del modelo, se han definido diversas estructuras de representación del conocimiento que componen en tiempo de ejecución el enunciado y la solución al problema presentado, lo cual posibilita el diseño de ejercicios genéricos. Se incluye, además, el registro de todo el quehacer del aprendiz, de modo que puede servir de base para dar seguimiento a su proceso de aprendizaje y por consiguiente, para brindarle las recomendaciones necesarias.

Con este modelo de entrenamiento se logra ofrecer una gran diversidad de ejercicios que apoyen la etapa de reforzamiento y eviten el uso repetitivo de los mismos ejercicios, adaptar la ejercitación a las necesidades del alumno y brindar una asesoría desde el punto de vista que éste es orientado durante su preparación.

Palabras claves: Entrenamiento adaptativo, Modelo del estudiante, Tutorial inteligente, Sistema de ejercitación, Base de conocimiento.

1. Introducción

El entrenamiento, por su propia condición de instrumento destinado a la consecución de objetivos específicos, resulta ser una etapa imprescindible en el proceso de enseñanza-aprendizaje. De ello se desprende, que el diseño de un sistema educativo con características de ejercitación debe cumplir con algunos requerimientos para obtener un

material de calidad que permita cumplir los objetivos para los cuales fue diseñado.

De ser así, la ejercitación estimulará el desarrollo mental continuo, para alcanzar objetivos nuevos y más elevados. Por otro lado, al estar basado en un programa computarizado permitirá un *aprendizaje activo* del aprendiz, donde la solución de problemas y la toma de decisión sustituirán la memorización de la información. Además, y como aspecto de gran importancia, el aprendizaje estará basado en las necesidades reales de cada alumno y no en los objetivos del material [3], [7], [11], [12], [17].

Tomando en cuenta la necesidad de la ejercitación en algunas áreas de la educación como son: la matemática, la física y la química, se ha propuesto una clasificación de los materiales educativos, donde se hace referencia de manera particular a Sistemas de Ejercitación porque desempeñan un papel muy importante en el logro de habilidades y destrezas, en las que la ejercitación y la reorientación son elementos fundamentales [18], [20], [28].

Los sistemas de ejercitación desempeñan un papel muy importante porque permiten reforzar las dos fases finales del proceso de instrucción: aplicación y retroinformación. Sin embargo, para lograr que este tipo de sistema constituya una herramienta de apoyo apropiada, se deben evitar limitaciones como: poca interacción, repetición de ejercicios, desorientación en la estrategia de aprendizaje y desconocimiento, por parte de los actores implicados, del desempeño experimentado.

El modelo propuesto en este trabajo incide directamente sobre estos problemas. Evita el uso de Bases de Datos o plantillas para la

generación de ejercicios, lo cual además de minimizar el tiempo de desarrollo, reduce considerablemente las posibilidades de repetición de un ejercicio, no sólo de manera individual, sino también cuando se trabaja con un grupo de alumnos, por ejemplo en un laboratorio.

Por otra parte, el uso de niveles de complejidad en los mecanismos de generación dinámica de los ejercicios, permite crear un modelo del estudiante más cercano al nivel de conocimiento que en la realidad este posee.

Por último, dado que todo el quehacer del aprendiz es registrado en una base de datos, el sistema puede generar estadísticas y recomendaciones que ayudan tanto al estudiante como al profesor a establecer estrategias de aprendizaje que favorezcan este proceso educativo.

2. Antecedentes

Como resultado del impacto que ha tenido el uso de la computadora en el proceso de aprendizaje y en específico de la ejercitación, ha proliferado el desarrollo de materiales digitales que sirven de apoyo al proceso educativo [1], [5], [6], [9], [26]. Para la elaboración de software didáctico, las técnicas más utilizadas han sido las herramientas de autor tales como Director, Flash, Toolbook, Swish, por mencionar algunas, los lenguajes de programación convencionales como Java y C++ y algunas técnicas de Inteligencia Artificial como por ejemplo, los Sistemas Basados en Conocimiento conocidos tradicionalmente como Sistemas Expertos [13], [19]. Estas últimas han sido muy utilizadas para el desarrollo de Sistemas Tutoriales Inteligentes dada su capacidad de emular, de una manera muy natural, las experiencias pedagógicas y disciplinares del profesor [4], [23], [24].

Sin embargo, para lograr un material digital de entrenamiento con capacidades para retroalimentar y dar seguimiento al aprendizaje del estudiante, se requiere del uso de diversas técnicas para poder aprovechar las

ventajas que nos brindan cada una de forma individual. Con esta filosofía de trabajo se logra el desarrollo de sistemas híbridos que exhiben múltiples capacidades [10], [27], [29], [30], [32].

Basado en un modelo híbrido, el empleo de conocimiento, permitirá asesorar las acciones de los estudiantes explicando los errores que van cometiendo y/o los resultados de sus acciones y proporcionando a su vez, las oportunas ayudas y refuerzos [3], [4], [16].

Por otro lado, el manejo de los datos, como resultado de las interacciones con el sistema y los intereses propios de los aprendices, se tomará como base para presentar una interfaz sistema-usuario en correspondencia con sus necesidades. Toda esta información representa el modelo del estudiante que se genera en su navegación por el sistema [15], [12], [21], [25], [31].

3. Modelo de entrenamiento adaptativo basado en ejercicios

La arquitectura del modelo híbrido propuesto se muestra en la figura 1, la cual está constituida por cuatro componentes principales: una interfaz de interacción usuario-sistema, un módulo de comunicación, bases de datos y base de conocimiento

Los procedimientos de comunicación, programados en el paradigma orientado a objetos, mantienen, una relación de comunicación constante con el modelo del estudiante y con el esquema de ejercicios por niveles de complejidad para garantizar que al aprendiz se le presente el ejercicio adecuado a su nivel de preparación en el tema.

Las dos bases de datos proporcionan información que hace posible que el entrenamiento se adapte al progreso del usuario. La base Modelo del Estudiante se encarga de registrar la información generada por la interacción del usuario con el sistema de ejercitación, mientras que la base Modelo del Profesor describe el esquema de ejercicios por

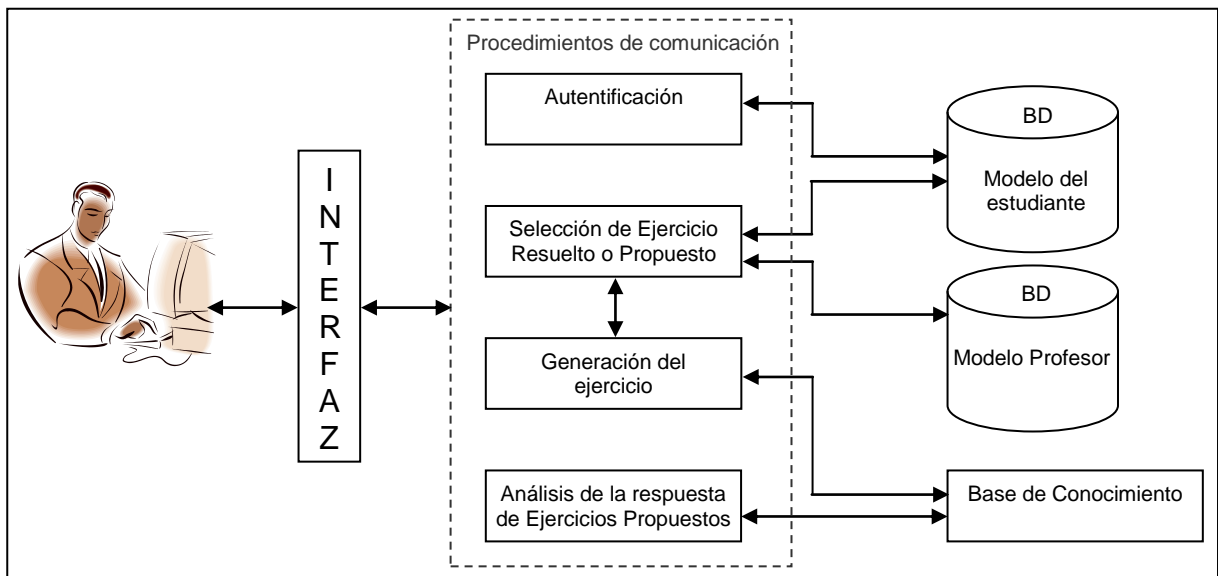


Figura 1. Arquitectura del método de entrenamiento adaptativo

niveles de complejidad, la forma de evaluar los ejercicios y transitar por los niveles definidos.

En la base de conocimiento se encuentra la estructura general del método de entrenamiento adaptativo y del contenido del tema para brindar las recomendaciones pertinentes.

Bajo esta concepción, las tareas ofrecidas al estudiante en un momento determinado podrán variar dependiendo de: su perfil, la estrategia actual de enseñanza, las acciones ejecutadas por el mismo cuando aborda tareas específicas y por el propio diseño del curso.

3.1 Esquema de ejercicios por niveles de complejidad

Una vez seleccionado los ejercicios de cada tema, el método de entrenamiento, obliga a definir un esquema para que los ejercicios estén organizados por niveles de complejidad. Su estructura se basa en separar los ejercicios sencillos, de los más complejos, para conformar así, dos grupos, llamados:

- Niveles de Complejidad Básicos (NCB), que comprende el grupo de los ejercicios que obligatoriamente debe conocer y dominar el estudiante en cada subtema y

- Niveles de Complejidad Avanzados (NCA) que contiene ejercicios más complejos que implican un nivel de conocimiento mayor por parte del estudiante. Sin embargo, se pone a consideración del profesor o incluso del estudiante mismo, si desea aprenderlos y dominarlos.

Dentro de cada nivel de complejidad básico y avanzado, a su vez, se especifican otros niveles de dificultad,. En la figura 2, se observa la estructura de los ejercicios por niveles complejidad que se puede presentar en un subtema dado dentro de una materia.

3.2 Niveles de complejidad

Una vez que se han definido cuales serán los ejercicios que conformarán los NCB y NCA se efectúa una organización más fina y detallada, que consiste en clasificar los niveles de complejidad, desde el más sencillo al más complejo, lo cual garantiza que el alumno pueda ir evolucionando paso a paso sin saltar etapas en el aprendizaje. Esta estrategia de entrenamiento permite obtener un ajuste más fino del modelo del estudiante

3.3 Concepto de ejercicio prototipo

Un prototipo de ejercicios es un código que se utiliza para generar ejercicios específicos en

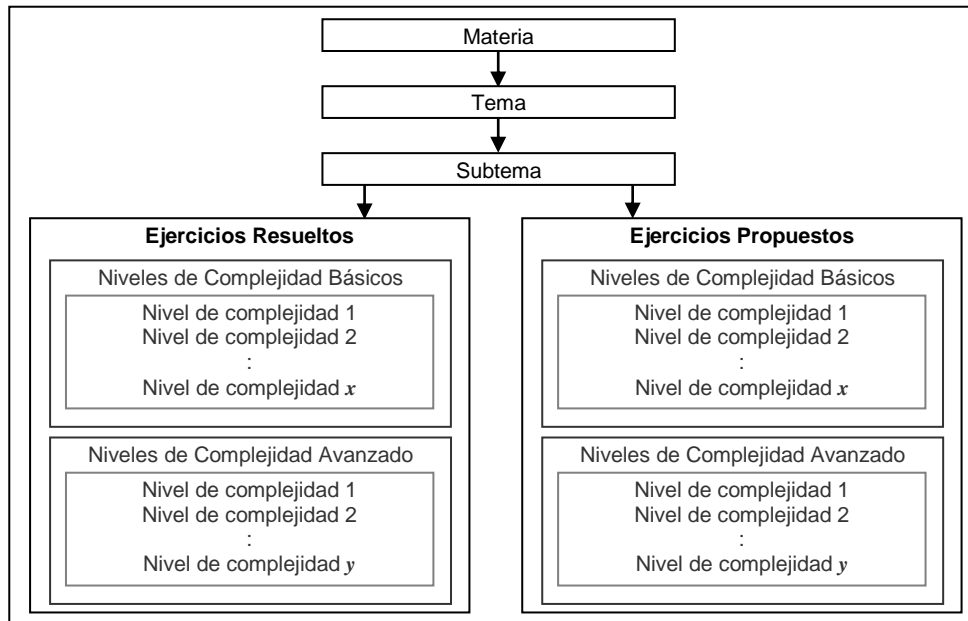


Figura 2. Esquema para la organización de los niveles de complejidad en un subtema

tiempo de ejecución. Para el mecanismo de generación de los ejercicios se necesitan, fundamentalmente, cuatro acciones:

1. Composición de la pregunta.- Se utiliza uno o más textos concatenados y la generación de valores que son asignados a las variables que forman parte del texto.
2. Construcción de la respuesta correcta.- Se compone el texto que muestra paso a paso el proceso de desarrollo que se realiza para obtener la solución del problema planteado.
3. Comparación de la solución correcta con la brindada por el usuario. El sistema calcula el resultado de ejercicio planteado y lo compara con la solución ofrecida por el alumno. El resultado de este paso permitirá al sistema tomar decisiones para el próximo paso.
4. Visualización de los resultados del proceso.

Considerando este mecanismo de generación de preguntas y construcción de respuestas en tiempo de ejecución, es posible contar con un sistema que presente una gran diversidad de ejercicios, sin necesidad de utilizar grandes

volumenes de información almacenada para este fin. Además, se le brinda al usuario la posibilidad de practicar con una gran cantidad de ejercicios, lo cual es un objetivo en los sistemas de práctica y ejercitación.

Cada prototipo de ejercicio tiene asociado un nivel de complejidad que se cuantifica a partir de 1, el cual equivale al mínimo grado de dificultad, hasta el nivel que sea necesario. Es importante mencionar que el número de niveles en cada uno de los temas es variable y además, que cada nivel es relativo a cada tema del contenido, es decir, dos ejercicios con el mismo nivel de complejidad pero pertenecientes a temas distintos pueden presentar complejidades diferentes desde el punto de vista práctico.

3.4 Tipos de Ejercicios

Con el fin de brindar diferentes formas de entrenar, en dependencia del nivel de conocimiento que se tenga sobre el tema, este modelo plantea tres tipos de ejercicios:

- *Ejercicios Resueltos*: Se le presenta al usuario un ejercicio y la explicación de su solución paso a paso.
- *Ejercicios Propuestos*: Se le propone un ejercicio al usuario para que lo resuelva e introduzca la solución. El usuario recibe

aprobación si contesta acertadamente o se le muestra la solución del ejercicio en caso de error.

- *Ejercicios Externos*: El estudiante le plantee al sistema un ejercicio para que el mismo se lo resuelva. Recibe solución o consejo del sistema referente al ejercicio planteado.

3.4.1 Ejercicios resueltos

Este tipo de ejercicios se les mostrará a aquellos estudiantes que están en fase de asimilación de conocimiento.

Los ejercicios resueltos constituyen problemas que el sistema le presenta al alumno donde se incluye la solución correcta. Regularmente, se visualizan en una sola pantalla mostrándose el planteamiento del problema y el desarrollo de su solución paso a paso con explicaciones, imágenes estáticas o animadas como ilustraciones en caso necesario.

Con estos ejercicios se pretende ayudar al usuario a introducirse en la resolución de problemas, mediante demostraciones detalladas.

El aprendiz podrá observar y analizar los ejercicios de este tipo, las veces que éste lo requiera. Sin embargo, el sistema de ejercitación se basará en el modelo del profesor definido para el entrenamiento. Cada vez que solicite un nuevo ejercicio, el modelo adaptativo considerará la cantidad de ejercicios que el alumno ha analizado dentro del nivel de complejidad actual y el tiempo dedicado a su aprendizaje, para examinar si le muestra un ejercicio de mayor complejidad o no. En el caso de que el modelo razone que no ha vencido el nivel bajo estudio, le presentará un nuevo ejercicio del mismo nivel pero con valores diferentes.

En la figura 3 se muestra un ejemplo del diseño de la pantalla de ejercicios resueltos. En ésta observamos que se cuenta con tres áreas donde se muestra: el tema al que corresponde el ejercicio, el planteamiento del problema y la solución detallada. Como se puede apreciar se

ejecutan las acciones 1, 2 y 4 del mecanismo de generación de ejercicios (epígrafe 3.3).

3.4.2 Ejercicios propuestos

Los propuestos son ejercicios que el sistema le presenta al usuario para que éste los resuelva. Éstos, al igual que los ejercicios resueltos se seleccionan de forma aleatoria en tiempo de ejecución y son construidos a partir de los ejercicios prototipo tomando en cuenta su nivel de dificultad.

Una vez que el usuario ha resuelto el ejercicio que el sistema le propuso, debe introducir la solución que obtuvo. Entonces el sistema evalúa si la respuesta obtenida por el usuario es la correcta o no. En caso de que la solución obtenida por el usuario sea incorrecta, el sistema le presenta una pantalla con consejos para resolver correctamente el ejercicio y le muestra paso a paso el desarrollo para llegar a la solución.

En este caso se necesitan las cuatro acciones definidas en el esquema general de funcionamiento de los prototipos de ejercicios.

3.4.3 Ejercicios externos

Existen dos formas de implementar el ejercicio externo:

1. *Fijo*. Cuando el sistema presenta el ejercicio con la posibilidad de que se introduzcan los valores por parte del alumno para que el sistema le realice los cálculos.
2. *Sugerido*. El usuario puede solicitar la solución de un problema para que se lo resuelva el sistema.

Para proceder con este último caso, el usuario deberá contestar una serie de preguntas que permitan al sistema detectar el prototipo de ejercicio al cual el problema planteado pertenece y que nivel de complejidad posee. Para desempeñar esta tarea se construirá una base de conocimiento que permitirá la ubicación del ejercicio dentro del sistema.

Una vez que el sistema haya detectado a qué prototipo de ejercicio pertenece, se genera una pantalla donde el alumno podrá introducir los valores correspondientes al caso particular

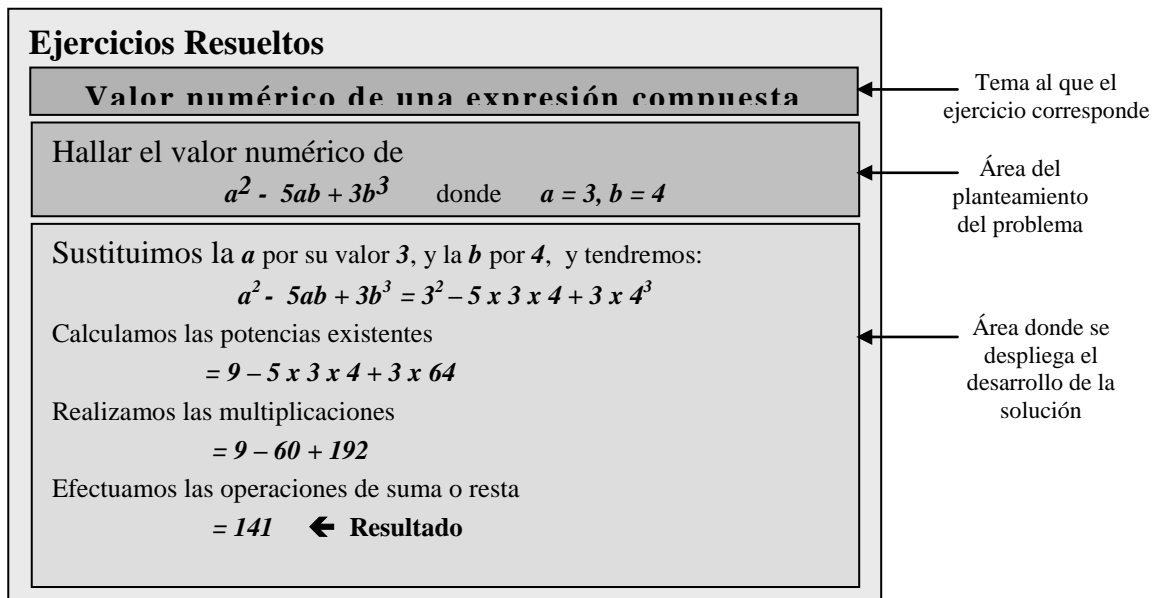


Figura 3. Estructura de la pantalla generada para un ejercicio resuelto

presentado. El sistema obtiene la solución y se le presenta el desarrollo de la solución paso por paso con la información referente al ejercicio que sugerido.

En ocasiones, puede ocurrir que el sistema no encuentre el tipo de prototipo solicitado. Para este caso, se generará un reporte o correo a los desarrolladores que les recomendará la conveniencia de incluir este prototipo de ejercicio.

Como se observa, en este caso se necesita un paso que consiste en la ubicación del prototipo que proporciona el usuario y además, se introduce una variante en la acción 1, puesto que los valores no se generan, sino que son introducidos por el propio usuario.

En resumen, para este tipo de ejercicio se ejecutan los pasos 1, 2 y 4 del mecanismo general definido para los prototipos de ejercicio.

4. HArIES: Lenguaje para el desarrollo de la Base de Conocimiento

HArIES (Hybrid Artificial Intelligent Expert System) es un lenguaje de representación del conocimiento de propósito general, por medio del cual se pueden construir y modificar las

bases de conocimiento referidas a cualquier dominio del quehacer humano. Presenta ambientes visuales para la edición de sus estructuras y tiene características similares a los lenguajes de programación convencional como Java y C++, todo lo cual facilita el desarrollo de las aplicaciones

En este ambiente se brindan un amplio número de estructuras para la representación del conocimiento que constituyen la vía que utiliza el desarrollador para almacenar la información y programar su funcionamiento [2], [3], [13], [14].

Para desarrollar un sistema educativo con HArIES se necesita construir una Base de Conocimiento donde se incluya toda la información relacionada con el contenido, metodología, organización, multimedia, ejercicios y evaluaciones, lo cual constituye el modelo pedagógico [8], [21], [22], [29].

El esquema de representación del conocimiento de este lenguaje es híbrido por cuanto integra varios de los esquemas clásicos: lógico, reglas de producción y marcos. Esta característica híbrida permite modelar muchos tipos de conocimiento y en ello radica su diversidad de representación.

Tres de las estructuras de representación del conocimiento que lo distinguen de otros lenguajes son la *Variable Imagen Sensible*, la *Base de Control* y los *Textos y Reglas de conclusión* [2], [14].

La variable imagen sensible es un marco que facilita la tarea del ingeniero del conocimiento en la creación de interfaces y el uso de multimedios. Con ella, es posible crear ambientes para presentar información utilizando imágenes, videos, sonidos, animaciones, hipermedias, gráficas, funciones y mensajes, sin necesidad de recurrir a la programación de estas funcionalidades.

Con esta estructura, se construyen las pantallas que se le van a mostrar al aprendiz, las cuales pueden incluir textos, menús o cajas de edición para solicitud de valores. Los textos pueden ser predefinidos o construido en tiempo de ejecución de acuerdo a las condiciones del contexto y a las respuestas del usuario. Los menús pueden ser de selección simple o múltiple y se pueden configurar en correspondencia al problema bajo estudio.

La Base de Control, también con características de un marco, es una estructura para diseñar las estrategias de control en dependencia de las necesidades del problema a emular. Permite diseñar la organización en tareas o módulos que tendrá la aplicación a desarrollar y el orden en que debe ejecutarse cada tarea. Así cada módulo podrá tener una estrategia de control propia, características de presentación personalizadas, condiciones a cumplir para su procesamiento y presentación de resultados independientes o no al resto de los módulos. Esta estructura, posibilita la obtención de disímiles comportamientos en una aplicación, con una labor de programación mínima.

En el lenguaje HAries, se puede considerar como un resultado a presentar al usuario, a un planteamiento complejo donde se expresa en lenguaje natural la conclusión a la que se arriba después de haber analizado el caso en cuestión. Este planteamiento puede estar constituido por palabras, frases y párrafos

completos que se componen a partir de textos predefinido y conocidos como textos de conclusión. Las reglas de conclusión, por su parte, son relaciones que permiten establecer las condiciones bajo las cuales tiene sentido la presentación de un resultado al usuario en tiempo de ejecución. Por medio de éstas es posible componer un texto combinando varios textos de conclusión, los cuales que se le mostrarán al usuario de acuerdo a la situación específica que se presenta producto de la interacción.

Estas cuatro estructuras del lenguaje, facilitan el desarrollo de los sistemas de ejercitación y en general, de los sistemas tutoriales inteligentes, dado su capacidad para de ajustarse dinámicamente a las condiciones del contexto y poder modelar un comportamiento adaptativo en las aplicaciones que se desarrollen con el lenguaje HAries.

5. Conclusiones

Con este modelo ha sido posible conjugar algunas de las condiciones fundamentales para los sistemas de ejercitación: cantidad de ejercicios, variedad en los formatos con que se presentan y retroinformación que reorienta con luz indirecta la acción del aprendiz.

La inclusión de niveles de complejidad dentro del modelo ha permitido crear un incentivo dentro del programa que motive al estudiante a realizar una cantidad significativa de ejercicios que hayan sido resueltos correctamente y sin ayuda, lo cual estimula el aprendizaje individual.

El mecanismo utilizado para el entrenamiento basado en ejercicios permite al sistema seleccionar la estrategia pedagógica más apropiada tomando en consideración el conocimiento del aprendiz.

Diversos sistemas desarrollados bajo este modelo en el área de las matemáticas, han sido utilizados en escuelas públicas y privadas en México con resultados muy satisfactorios.

Referencias

- [1] Alonso MA, De la Cruz AV; (2002) Serie Educativa Virtual. Revista Hífen. Edição Especial. Brasil, Pp. 7–12.
- [2] Alonso MA, de la Cruz AV, Gutiérrez A; (2004) Knowledge Representation Language: HArIES. En Memorias de la 8th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2004). Orlando, Florida. USA. Págs. 358-361.
- [3] Alonso MA, de la Cruz AV, Gutiérrez A; (2004) HArIES: Un Lenguaje para la Programación del Conocimiento con Facilidades para la Construcción de Material Educativo. Revista Iberoamericana de Sistemas, Cibernética e Informática. Vol. 2. Núm. 2. <http://www.iiisci.org/journal/risci/Contents.asp?var=&Previous=ISS8535>.
- [4] Anoluna A; (2007) Advances in intelligent tutoring systems: Problem-solving modes and model of hints. International Journal of Computers, Communications & Control, Vol.2, Pp. 48-55.
- [5] Aranda L, Torres S, Trella M, Conejo R; (2000) BabelWin: An environment for learning and monitoring reading and writing skills. Computers and Education in the 21st Century 79-91.
- [6] Benhayón MA, Briceño PC, Magallanes KD, Montezuma MA; (2007) Generación y Difusión del Conocimiento. Funcionalidades Básicas del Sistema de Gestión del Conocimiento de la Universidad Metropolitana: ESEGA. Revista Iberoamericana de Sistemas, Cibernética e Informática. Vol. 4. Núm. 1. <http://www.iiisci.org/journal/risci/Contents.asp?var=&Previous=ISS0803>.
- [7] Berlanga AJ, Garcia-Penalvo FJ; (2008) Learning Design in Adaptive Educational Hypermedia Systems. Journal of Universal Computer Science Vol.14 Pp. 3627-3647.
- [8] Castillo S; (2008) Pedagogical Proposal Based on Constructivism for the Optimal Use of ICT in the Teaching and Learning of Mathematics. Revista Latinoamericana de Investigación en Matemática Educativa-Relime Vol.11 Pp.171-194.
- [9] Chang KE, Chiao BC, Chen SW, Hsiao RS; (2000) A programming learning system for beginners — A completion strategy approach. IEEE Transactions on Education 43(2), Pp. 211–220.
- [10] Chen CM; (2009) Personalized E-learning system with self-regulated learning assisted mechanisms for promoting learning performance Expert Systems with Applications, Vol. 36, Pp. 8816-8829.
- [11] Cohen AM, Cuypers H, Jibeteau D, Spanbroek M (2006) Interactive learning and mathematical calculus. Mathematical Knowledge Management Book Series: Lecture Notes in Computer Science Vol.3863 Pp. 330-345.
- [12] Corbett A, McLaughlin M, Scarpinato KC; (2000) Modeling student knowledge: Cognitive tutors in high school and college. User Modeling and User-Adapted Interaction, Vol. 10, Pp. 81-108.
- [13] De la Cruz AV, Valdés JJ, Jocik E, Balsa J, Rodríguez A; (1993) Fundamentos y Práctica de la Construcción de Sistemas Expertos. Editorial Academia, La Habana, Cuba.
- [14] De la Cruz AV, Alonso MA; (2002) The HArIES environment (v6.00) for the development of intelligence systems. Revista Hífen. Edição Especial. Pp. 184–186. Brasil.
- [15] Faraco RA, Rosatelli MC; (2004) An approach of student modelling in a learning companion system. In: Proceedings of the 9th Ibero-American Conference on Artificial Intelligence (IBERAMIA 2004), Puebla, México. Pp. 891-900.
- [16] Fong-Lok L; (2002) Diagnosing students' Algebra Errors on the Web. In Proceedings of the International Conference on Computers in Education (ICCE'02) Pp.578- 579 Auckland, New Zealand.

- [17] Fong J, Kwan I, Ng M, Li I, Chan SK (2004) An application-oriented e-Learning system with self-monitoring and adaptive exercises. *Advances in Web-Based Learning - ICWL 2004 Book Series: Lecture Notes in Computer Science* Vol. 3143 Pp. 325-332.
- [18] Galvis A; (1994) *Ingeniería de Software Educativo*. Ediciones Uniandes. Segunda Impresión. Santa Fe de Bogotá, Colombia.
- [19] Giarratano J, Gary R; (2001) *Sistemas Expertos. Principios y Programación*. International Thomson Editores. Ed. 3.
- [20] González G; (2008) ¿Qué es Software educativo, cómo se clasifica y cuáles son sus características? *Profesor Interactivo* <http://profesorinteractivo.blogia.com/2008/042401--que-es-software-educativo-como-se-clasifica-y-cuales-son-sus-caracteristicas-.php>. Visitado: 3/05/2009.
- [21] Graf S, Lin T, Kinshuk J; (2008) The relationship between learning styles and cognitive traits - Getting additional information for improving student modeling. *Computers in Human Behavior*, Vol.24, Num. 2, Pp. 122-137.
- [22] Hernández Y, Sucar LE, Arroyo-Figueroa G; (2008) Building an Affective Model for Intelligent Tutoring Systems with Base on Teachers' Expertise. In: *Proceedings of the 7th Mexican International Conference on Artificial Intelligence (MICA I 2008)*, México City, Mexico, Pp. 754-764.
- [23] Heffernan NT, Koedinger KR; (2002) An Intelligent Tutoring System Incorporating a Model of an Experienced Human Tutor. In: *Proceedings of the 6th International Conference Intelligent Tutoring Systems* Pp. 596-608.
- [24] Keles A, Ocak R, Keles A, Gulu A; (2009) ZOSMAT: Web-based Intelligent Tutoring System for Teaching-Learning Process. *Expert Systems with Applications*, Vol. 36, Pp. 1229-1239, Part1.
- [25] Kelly D, Tangney B; (2006) Adapting to intelligence profile in an adaptive educational system. *Interacting with Computers* 18(3) 385-409.
- [26] Lesta L, Yacef K; (2002) An Intelligent Teaching Assistant System for Logic. In: *Proceedings of the 6th International Conference on Intelligent Tutoring Systems (ITS 2002)*, Biarritz, France Pp. 421-431.
- [27] Ministerio de Educación, Política Social y Deporte. Gobierno de España. *Multimedia Authoring for Language Tutors and Educational Development*. http://malted.cnice.mec.es/presentacion/proyecto_Malted.htm. Visitado 4/04/2009.
- [28] Pere Marquès; (2009) Clasificación del software educativo. *La Uno - Educación y TIC*. <http://escuelauno-nqn.blogspot.com/2009/01/clasificacin-del-software-educativo.html>. Visitado 3/05/2009.
- [29] Pritchard D. and Negoita MG; (2005) A Fuzzy-Ga Hybrid Technique for Optimization of Teaching Sequences Presented in ITSs. *Advances in Soft Computing* 2, 311-316.
- [30] Tomas AP, Leal JP; (2003) A CLP-based tool for computer aided generation and solving of maths exercises. In: *Proceedings of the 5th International Symposium on Practical Aspects of Declarative Languages (PADL 2003)*, Nevada, USA, Pp. 223-240.
- [31] Trinh TH, Buche C, Querrec R, Tisseau J; (2009) Modeling of Errors realized by a Human Learner in Virtual Environment for Training. *International Journal of Computers Communications & Control*, Vol.4, Pp. 73-81.
- [32] Virvou M., Moundridou M.; (2001) Student and instructor models: Two kinds of user model and their interaction in an ITS authoring tool. In: *Proceedings of the 8th International Conference on User Modeling*, Sonthofen, Germany, Pp. 158-167.