



BOTBLOQ: Ecosistema integral para el diseño, fabricación y programación de robots DIY

Proyecto Financiado por el Centro de Desarrollo Tecnológico
Industrial (CDTI)

EXPEDIENTE: IDI-20150289

Cofinanciado por el Fondo Europeo de Desarrollo Regional
(FEDER) a través

del Programa Operativo Plurirregional de Crecimiento
Inteligente 2014-2020

ACRÓNIMO DEL PROYECTO: BOTBLOQ



Centro para el
Desarrollo
Tecnológico
Industrial



UNIÓN EUROPEA

Fondo Europeo de
Desarrollo Regional (FEDER)

Una manera de hacer Europa

ENTREGABLE E.6.1.2.1 Revisión Sistemática de la Literatura actual sobre Sistemas Tutores Inteligentes Adaptativos

RESUMEN DEL DOCUMENTO

BOTBLOQ: Ecosistema integral para el diseño, fabricación y programación de robots DIY

El presente documento contiene una revisión sistemática de la literatura actual sobre el tema de los Sistema Tutoriales Inteligentes (STI) haciendo énfasis en el aspecto adaptativo de los mismos. Teniendo en cuenta su importancia en el contexto del proyecto BOTBLOQ, se incluyeron secciones sobre los STI para la programación de robots y sobre los algoritmos adaptativos utilizados en los STI.

BOTBLOQ: Ecosistema integral para el diseño, fabricación y programación de robots DIY

SISTEMAS TUTORIALES INTELIGENTES (STI)	4
SISTEMAS TUTORIALES INTELIGENTES PARA LA PROGRAMACIÓN DE ROBOTS	5
SISTEMAS TUTORIALES INTELIGENTES ADAPTATIVOS	9
ALGORITMOS ADAPTATIVOS	12
BAYESIAN KNOWLEDGE TRACING (BKT)	12
ITEM RESPONSE THEORY (IRT)	13
MULTI-ARMED BANDIT (MAB) PROBLEM	15
INTERACTION NETWORKS	16
PETRI-NET	18
PIGEON ALGORITHM	18
REFERENCIAS	19

1. SISTEMAS TUTORIALES INTELIGENTES (STI)

Un Sistema Tutorial Inteligente (STI) es un software informático basado en Inteligencia Artificial (IA) que puede razonar utilizando modelos de conocimiento orientados a promover y evaluar el aprendizaje, los cuales están diseñados para simular el comportamiento y orientación de un tutor humano [61, 62, 63]. La función principal de un STI es adaptarse al alumno mediante la comprensión de sus estados cognitivos, meta-cognitivos y afectivos.

Un STI puede ayudar a los estudiantes a estudiar una variedad de temas haciendo preguntas, analizando respuestas y ofreciendo instrucción personalizada. Dos de las formas más importantes en que se distinguen estos sistemas de otros tipos de sistemas de instrucción asistida por computadora son: 1) que pueden interpretar respuestas complejas de los estudiantes y 2) aprender a medida que operan.

Con el fin de cumplir sus funciones, el software crea y mantiene un perfil con la información de cada estudiante, estimando el grado de maestría de cada uno de ellos. Estos sistemas pueden adaptar el modo en que desarrollan la tutoría en tiempo real, siguiendo las diferencias en las estrategias individuales de los estudiantes o ajustando su base de conocimiento para una interacción más efectiva con todos los estudiantes.

El término “Sistema Tutorial Inteligente” fue acuñado por David Sleeman y John Seely Brown [72], a partir de reconocer la evolución de los sistemas educativos basados en Instrucción Asistida por Ordenador (Computer Assisted Instruction CAI) en sistemas de Instrucción Inteligente Asistida por Computadora (ICAI) y enfatizando el enfoque en el aprendizaje individual. En 1987, Wenger [73] proporcionó una descripción detallada del concepto de STI.

Los Sistemas Tutoriales Inteligentes alcanzaron una gran atención a finales de los ochenta al mismo tiempo coincidiendo con el auge en el desarrollo de los sistemas expertos [68]. La mayoría de los STI emplean tecnología basadas en los sistemas expertos, en particular, los llamados sistemas de producción y la representación del conocimiento humano.

Los STI se organizan mediante módulos para modelar el conocimiento sobre 1) el dominio a tratar, 2) el alumno, y 3) las estrategias de enseñanza o pedagógicas [68].

Los componentes principales de un STI son: un módulo experto (o de dominio), un módulo de estudiante y un módulo. El módulo experto debe ser capaz de resolver los problemas que el módulo tutor somete a los estudiantes. El módulo tutor controla la interacción con el alumno, en base a su conocimiento docente y comparando el comportamiento del alumno y la representación del conocimiento del dominio. El

BOTBLOQ: Ecosistema integral para el diseño, fabricación y programación de robots DIY

modulo del estudiante refleja lo que el sistema puede inferir acerca del estado cognitivo del estudiante.

Anderson et al. [64] establecieron ocho principios para el diseño de tutor inteligente:

1. Representar la competencia del estudiante como un conjunto de producción.
2. Comunicar la estructura de objetivos subyacente a la solución de problemas.
3. Proporcionar instrucción en el contexto de resolución de los problemas.
4. Promover una comprensión abstracta del conocimiento sobre la resolución de problemas.
5. Minimizar la carga de la memoria de trabajo.
6. Proporcionar retroalimentación inmediata sobre los errores.
7. Ajustar el volumen y complejidad de la instrucción con el aprendizaje.
8. Facilitar aproximaciones sucesivas a la habilidad que se quiere alcanzar

A los cuales se unió un principio general, el cual según sus creadores Corbett et al. [65] gobierna el diseño del tutor inteligente: *un sistema tutorial inteligente debe permitir al estudiante trabajar en la resolución de problemas buscando obtener una conclusión exitosa.*

2. SISTEMAS TUTORIALES INTELIGENTES PARA LA PROGRAMACIÓN DE ROBOTS

A continuación, mostraremos algunos ejemplos sobre STI vinculados con la programación de robots.

Fernández et al [35] nos da una perspectiva de la relación existente en 2013 entre los métodos de aprendizaje e-Learning y los estudios de Ingeniería en España, dando ejemplos sobre varios temas de estudios de ingeniería, haciendo hincapié en los buenos resultados obtenidos en las experiencias mencionadas. Entre los diferentes ejemplos que menciona encontramos dos relacionados con la temática de la programación de robots.

Rosado [36] describe un sistema que permite a los estudiantes de ingeniería realizar prácticas en un laboratorio a través de un navegador web, lo cual posibilita utilizar un robot industrial a distancia, pudiendo programarlo y visualizar el resultado de la programación realizada en tiempo real, así como interactuar con el mismo. En dichas

BOTBLOQ: Ecosistema integral para el diseño, fabricación y programación de robots DIY

prácticas, el estudiante reserva el robot durante 30 minutos, pudiendo descargar y ejecutar el programa y visualizarlo a través de una cámara IP. El servidor web está programado para ofrecer estadísticas de utilización. El diseño del sistema es fácilmente adaptable a cualquier tipo de experimento para los que se requiere utilizar hardware de difícil acceso, haciendo rentables los altos costos de un robot industrial.

Vargas et al [37] exponen los resultados obtenidos mediante dos laboratorios de experimentación orientados a la enseñanza de materias del área de Ingeniería de Sistemas y Automática: el control de un sistema de tres tanques y la programación de un robot, destacando el alto grado de satisfacción obtenido.

FIRST (For Inspiration and Recognition of Science and Technology) es una organización juvenil internacional no lucrativa fundada en 1989 para motivar a los estudiantes a interesarse en estudios de STEM (ciencia, tecnología, ingeniería y matemáticas). FIRST organiza un concurso anual de robots, denominado FRC (FIRST Robotics Competition) para estudiantes de entre 8 y 12 años. El sistema de control comprende los componentes electrónicos y las herramientas de software que los equipos usan para controlar sus robots.

En [60] se realizó un análisis evaluativo del sistema de control utilizado en FRC. Una de las herramientas que se consideraron fue el FRC Simulator, el cual sirve para probar código en un robot virtual. Dicho simulador fue considerado de utilidad por los diferentes equipos, aunque se le señalaron problemas de incompatibilidad con robots personalizados o ciertos lenguajes de programación.

Otra herramienta analizada fue RobotBuilder, basada en los lenguajes de programación Java y C ++, la cual permite a los equipos diseñar visualmente su framework de código de robot y luego exportar ese framework a código real. La evaluación de la misma resultó muy satisfactoria por parte de los equipos que la usaron.

Uno de los aspectos que reviste especial interés con relación a los sistemas tutoriales inteligentes (STI) son los modelos de los estudiantes. Dichos modelos suelen centrarse en la corrección de las respuestas de los estudiantes [38]. Este enfoque se debe a que la forma más fácil de recoger y evaluar las respuestas de los estudiantes eran las preguntas de opción múltiple. Gracias a los avances en la tecnología, actualmente es relativamente fácil crear actividades interactivas complejas de resolución de problemas. En tales entornos, es útil analizar no sólo la corrección de las respuestas de los estudiantes, sino también el tiempo empleado en el proceso de resolución.

Con respecto al modelado del aprendizaje, hay una extensa investigación sobre curvas de aprendizaje (por ejemplo, [41, 42]). En el contexto de los sistemas de tutoría inteligente, el enfoque más utilizado es el trazado del conocimiento bayesiano [43],

BOTBLOQ: Ecosistema integral para el diseño, fabricación y programación de robots DIY

que modela la probabilidad de que un estudiante haya aprendido una habilidad particular durante una secuencia de intentos.

En este sentido se enfocan los trabajos de Jarusek y Pelanek [39, 40], los cuales modelan el desempeño de los estudiantes en ejercicios donde el único criterio de desempeño es el tiempo para resolver un problema. Ejemplos de estos ejercicios son rompecabezas lógicos como Sudoku o problemas de programación y matemáticas adecuadamente formulados. Los datos que emplean son tiempos de resolución de problemas de estudiantes reales que utilizan el sistema Problem Solving Tutor basado en la web para practicar la resolución de ejercicios. El sistema contiene problemas informáticos sobre programación de robots, números binarios, programación en C y Python, filtrado de datos mediante expresiones regulares, y crucigramas de números binarios.

Los problemas de programación de robots son particularmente populares. En ellos, los estudiantes tienen que programar un robot para realizar una tarea particular. Estos problemas tienen sintaxis y semántica muy simples, pero pueden proporcionar problemas desafiantes que ilustran conceptos importantes (particularmente sobre recursión).

Jarusek y Pelanek [44] describen un modelo que asume una relación lineal entre la capacidad de resolución de problemas y el logaritmo del tiempo requerido para resolver un problema (i.e. una relación exponencial entre capacidad y tiempo), relacionando el modelo con la teoría de respuesta al ítem y el filtrado colaborativo. La teoría de respuesta al ítem (TRI) o modelo de Rasch [45] se utiliza principalmente en pruebas adaptativas computarizadas para predecir la probabilidad de una respuesta correcta seleccionando un elemento de prueba adecuado. Es una teoría psicométrica, utilizada para la construcción de test y pruebas psicológicas, que describe la relación entre un conjunto de datos obtenidos en un proceso de medición (las respuestas a los ítems de una prueba) con determinadas variables latentes, como por ejemplo los rasgos de personalidad de los sujetos a quienes se ha administrado. El tipo de variables o propiedades latentes que se pretende medir pueden ser cuantitativas o cualitativas

La realidad aumentada (RA) proporciona un medio para complementar visualmente el mundo físico con imágenes generadas por computadora e interactividad. Mediante el uso de tecnologías de visualización, la visión de un usuario puede ser superpuesta en tiempo real con anotaciones, información o gráficos generados por ordenador. La realidad virtual (RV) sitúa a un usuario dentro de un mundo totalmente generado por ordenador. RA es una alternativa que preserva la experiencia natural al mismo tiempo que permite muchos de los beneficios del reino virtual.

La RA tiene muchas aplicaciones en la educación [46] entre otras áreas. Exponer a los usuarios a la información virtual permite un nivel de control sobre su entorno percibido de otra manera inalcanzable. La información que brinda la RA se obtiene in

BOTBLOQ: Ecosistema integral para el diseño, fabricación y programación de robots DIY

situ y distrae menos al usuario que si se proporciona la misma información en papel o en una pantalla independiente. Ampliando este paradigma, la realidad aumentada espacial (RAE) [47] usa proyectores en lugar de pantallas portables para mostrar contenido virtual.

Zaeh y Vogl [48] emplearon proyectores láser para la programación de robots industriales, diseñando una interfaz de usuario mediante la cual un usuario podría controlar un dispositivo de entrada rastreado para delinear una ruta a un robot. Para ello, el proyector muestra la ruta a seguir mientras el robot realiza su tarea siguiendo la trayectoria indicada. Las ganancias en interactividad entre el programador y el robot facilitadas por la RAE ayudan a producir una reducción de hasta 80% en el tiempo de programación del robot, en comparación con los métodos ordinarios".

Un objetivo fundamental de la investigación informática ha sido y es la creación automatizada de agentes inteligentes autónomos para su uso en simulaciones tácticas y aplicaciones robóticas. Dichos agentes son programas que pueden realizar tareas de control 'inteligentes' (de tipo humano) con habilidades motoras y tomar decisiones sobre cómo y cuándo utilizar esas habilidades cuando se realiza una tarea compleja o se realiza una misión.

Las técnicas de aprendizaje automático han sido utilizadas en diversas ocasiones para hacer que los agentes tácticos aprendan mientras ejecutan en un ambiente determinado en situaciones de simulaciones militares, juegos de computadora o fútbol de RoboCup.

Uno de los grandes desafíos de la robótica hoy en día es aprender de los usuarios humanos que son inexpertos en la interacción con los robots, pero que tienen las habilidades para enseñar con flexibilidad a otros seres humanos y a los niños en particular.

La Programación de Robots mediante la Demostración (Programming by Demonstration, PbD) se ha convertido en un tema central de la robótica que se extiende a través de áreas de investigación general como la interacción humano-robot, aprendizaje automatizado, visión por ordenador y control de motores. PbD [49] es un poderoso mecanismo para reducir la complejidad de los espacios de búsqueda para el aprendizaje, ya sea iniciando la búsqueda desde la buena solución conocida o eliminando del espacio de búsqueda las malas soluciones.

Por otra parte, la programación explícita y tediosa de una tarea por un usuario humano puede ser minimizada o eliminada. Por último, estudiar y modelar el

BOTBLOQ: Ecosistema integral para el diseño, fabricación y programación de robots DIY

acoplamiento entre la percepción y la acción, nos ayuda a entender el mecanismo que sustenta dicho acople.

Stein et al describen en su artículo [50] la creación de agentes que pueden aprender a realizar una tarea observando a un experto, luego dar vuelta sin problemas y enseñar la misma tarea a una persona menos competente. Estos agentes son enseñados a través de la observación de la actuación de los expertos y luego refinados a través de la práctica sin supervisión de la tarea, todo ello en un entorno simulado.

El diseño de un paradigma de interacción humano-robot eficaz es particularmente importante para tareas complejas como la manipulación de múltiples robots que requieren que ambos, el humano y el robot, trabajen juntos de manera unida. Aunque el aumento del número de robots puede ampliar el área que los robots pueden cubrir dentro de un período limitado de tiempo, una pobre interfaz humano-robot comprometerá el rendimiento del equipo de robots.

Existen varios paradigmas para aumentar la eficacia de la interacción humano-robot [51]: 1) interfaces multimodales en las que el usuario controla los robots utilizando voz y gesto; 2) Interfaces configurables que permiten al usuario crear nuevos comandos mostrándolos; 3) interfaces adaptativas que reducen la carga de trabajo del operador según sea necesario a través del aumento de la autonomía del robot.

Las interfaces basadas en la gesticulación pueden ser muy intuitivas para la manipulación y el control remoto de los brazos del robot [52]. Pieska et al. desarrolló un sistema basado en gestos para ayudar a operadores inexpertos en tareas de programación de robots industriales. La interfaz resultó útil para usuarios con diferentes niveles de experiencia en programación de robots [53].

Stein y Gonzalez [54] investigan el entrenamiento de robots mediante la interacción humana directa a través del uso de tecnología cenestésica (basada en el tacto) conocida como háptica. Sus resultados cualitativos indican que dicha interfaz resulta útil para apoyar el proceso de formación, dando al humano una mejor comprensión de las acciones de los agentes, y proporcionar un mecanismo para crear y mejorar los agentes (así como posiblemente validar su rendimiento).

Kucukyilmaz et al. [59] investigan el uso de tecnología háptica para la comunicación entre un ser humano y un robot que están cooperando en una tarea con el objetivo de que el robot pueda inferir las intenciones de su colaborador humano y ajustar los niveles de control en consecuencia, mediante negociaciones entre ellos, a través de este dispositivo.

BOTBLOQ: Ecosistema integral para el diseño, fabricación y programación de robots DIY

Aleotti et al. [56] utiliza tecnología cenestésica bajo un enfoque de Programación por Demostración, donde la vibración indica al instructor lo que el sistema físico no puede hacer. Eguchi et al. [57] utiliza esta misma tecnología para mostrar al instructor las diferencias entre el sistema virtual y el robot real para ser entrenado.

Calinon y Billard [58] reportan resultados exitosos con un sistema que usa la guía física del maestro del brazo del robot para ayudarlo a aprender un nuevo movimiento. Grollman y Billard [55] utilizan demostraciones fallidas para enseñar al robot lo que no debe hacer.

3. SISTEMAS TUTORIALES INTELIGENTES ADAPTATIVOS

Los Sistemas de Aprendizaje Adaptativo (Adaptive Learning Systems, ALS) son una nueva generación de Sistemas Tutoriales Inteligentes que utilizan la tecnología, las teorías de aprendizaje y los conocimientos psicológicos para adaptar aspectos de dichos sistemas tales como presentación, contenido, evaluación, etc. al nivel de conocimiento y las características personales de los alumnos.

En los años 80, Bloom [31] llevó a cabo experimentos con diversos grupos de alumnos en diferentes situaciones de aprendizaje. Así llegó a la conclusión de que la forma ideal de instrucción es cuando el instructor solo tiene a su cargo de 1-3 personas, ya que de esta forma es más factible adaptar la instrucción a las características y preferencias de cada estudiante en particular. A pesar de ello, la tutoría individual siempre ha sido difícil de realizar en el entorno educativo del mundo real debido a la gran diferencia entre el número de instructores y estudiantes y los altos costos de proporcionar un instructor para cada alumno. El término "inteligente" aplicado a los sistemas tutoriales, según Elsom-Cook [32], indica que estos son capaces de desarrollar comportamientos que, en el caso de ser llevados a cabo por un humano, este sería catalogado como un "buen profesor".

Por su parte, Sonwalkar [33] plantea que un sistema de aprendizaje adaptativo debe proporcionar características tales como:

- Un marco pedagógico que permita variaciones en el proceso de instrucción.
- Vías de enseñanza adaptadas a diferentes estilos y estrategias de aprendizaje.
- Retroalimentación inteligente y continua a los estudiantes.
- Monitorear el progreso educativo, generar informes y permitir la orientación efectiva del profesorado.
- Integrar diferentes estrategias como la interacción, el uso de facilidades multimedia y el aprendizaje en un contexto pedagógico.

Muchos autores consideran que los sistemas tutoriales están compuestos por dos bucles [24]:

- El bucle externo se encarga de ayudar al estudiante a seleccionar la próxima tarea que deberá llevar en su aprendizaje. La tarea consiste generalmente en resolver un problema complejo de varios pasos.
- El bucle interior se ejecuta para cada paso de la tarea con el objetivo de darle sugerencias y comentarios al estudiante con relación al problema que está resolviendo.

BOTBLOQ: Ecosistema integral para el diseño, fabricación y programación de robots DIY

Para decidir cuál debe ser la próxima tarea que debe ejecutar el estudiante se han utilizado cuatro métodos básicos:

- I. El bucle exterior muestra un menú y permite al estudiante seleccionar la siguiente tarea.
- II. El bucle exterior asigna las tareas en una secuencia fija y el estudiante debe terminar la tarea actual para poder avanzar.
- III. El bucle exterior implementa una estrategia basada en el dominio del estudiante [31], para lo cual el curso se estructura mediante como una secuencia de niveles de dificultad. Cuando un estudiante está trabajando en un nivel particular, el sistema le sigue asignando tareas del mismo nivel hasta que el estudiante demuestre que lo domina. Sólo entonces se le permite que pase al siguiente nivel. De esta forma, algunos estudiantes pueden terminar el plan de estudios antes que otros.
- IV. Para cada tarea, se asume que el sistema sabe qué componentes de conocimiento se ejercitan con la misma. Para cada uno de dichos componentes de conocimiento, el tutor mantiene un estimado del grado de dominio que ha mostrado el estudiante del mismo. Cuando el estudiante completa una tarea y el sistema debe seleccionar la siguiente, este elige dicha tarea teniendo en cuenta los componentes de conocimientos de las diferentes posibles tareas y los componentes que ya domina el estudiante. Esta estrategia se conoce como macro adaptación [34].

Una dirección interesante para futuras investigaciones sería analizar no solo la complejidad de las tareas y la capacidad del estudiante para resolverlas correctamente, sino también el tiempo empleado para ello [2].

A continuación, se exponen algunas plataformas de aprendizaje ampliamente utilizadas por gran número de alumnos. Todas ellas son adaptables y capaces de personalizar la educación de los alumnos.

- I. Plataforma de Aprendizaje Adaptativa Knewton [29]: fundada en 2008, la empresa se asocia con Pearson Education en 2011. Ha desarrollado una plataforma para personalizar el contenido educativo, la cual puede detectar y registrar acciones de los usuarios tales como si contestan las preguntas correctamente o no, la cantidad de tiempo que requieren para hacerlo, el estilo de aprendizaje que mejor se adapta a cada usuario para aprender un concepto, o los movimientos del ratón. Todas estas acciones se registran en el perfil del estudiante y se comparan con su historia previa y con las de otros estudiantes en situaciones similares.

- II. Plataforma de Educación ALEKS [28]: desarrollada a partir de 1994, fue adquirida por McGraw-Hill Education en 2013. Al inicio cada alumno debe contestar una prueba con 20-30 preguntas. La prueba se adapta a cada alumno según lo que este va contestando a cada pregunta. ALEKS utiliza la teoría de espacios de conocimiento para comprender los temas que domina el estudiante
- III. Plataforma Smart Sparrow [27] es una compañía que surge a partir de la Universidad de Nueva Gales del Sur en Sydney, Australia. Cuenta con tres niveles de adaptación:
 - a. el sistema ofrece retroalimentación inteligente e inmediata a los estudiantes con dificultades mientras realizan una tarea.
 - b. el sistema puede detectar el nivel de conocimientos actual del alumno, sus errores de concepto, y la capacidad de aprendizaje, adaptando el nivel de dificultad y la entrega de los materiales de acuerdo con dicha valoración.
 - c. el sistema da a los profesores la posibilidad de adaptar su enseñanza
- IV. Corporación D2L [33] (Desire2Learn hasta 2014), ofrece diversas herramientas para crear cursos, así como un foro de discusión para estudiantes e instructores. La plataforma utiliza pre-evaluación y post-evaluación, utilizando los resultados para determinar la trayectoria de aprendizaje de cada alumno y adaptar su contenido.
- V. DreamBox Learning [30] es un proveedor de software en línea que se centra en la educación matemática en el nivel de escuela primaria y secundaria. Utiliza animaciones y juegos para enseñar matemáticas y se basa en el principio de "aprender haciendo".

4. ALGORITMOS ADAPTATIVOS

Un algoritmo adaptativo es aquel que puede cambiar su comportamiento durante su ejecución en base a información disponible en su entorno o en el propio programa. Dicha información puede corresponder a recursos computacionales disponibles o provenir del historial de datos recibidos recientemente. Su papel en el desarrollo de Sistemas Tutoriales Inteligentes (STI) es muy variado e importante. A continuación, recopilamos algunas experiencias y aplicaciones de los mismos en el contexto de los STI.

4.1. Bayesian Knowledge Tracing (BKT)

Bayesian Knowledge Tracing (BKT) es uno de los métodos de modelado de estudiantes más ampliamente adoptados en los Sistemas Tutoriales Inteligentes (ITSs). BKT aprovecha la información sobre las observaciones de las interacciones registradas (por ejemplo, correctas, incorrectas) de los estudiantes con el sistema para mantener actualizado la estimación del conocimiento latente del estudiante (por ejemplo, no aprendido, aprendido).

El modelo BKT [34] es un Modelo de Markov Oculto de dos estados (HMM) caracterizado por cinco elementos básicos:

- I. N , el número de diferentes tipos de estado de conocimiento oculto;
- II. M , el número de diferentes tipos de observación del comportamiento de los estudiantes;
- III. Π , la distribución de estado inicial $P(S_0)$;
- IV. T , la probabilidad de transición de estado $P(S_{t+1} | S_t)$ y
- V. E , la probabilidad de emisión $P(O_t | S_t)$.

Obsérvese que N y M están predefinidos antes del entrenamiento, mientras que Π , T y E son aprendidos de la secuencia de observación de los estudiantes. Los arcos entre los nodos representan su dependencia condicional. Sin embargo, el modelo BKT no tiene en cuenta las intervenciones instruccionales. Las intervenciones instruccionales son acciones iniciadas por el sistema que guía la actividad de aprendizaje del estudiante. Dos intervenciones instruccionales comunes son:

- Elicit (obtener): representa preguntarle a un estudiante cuál es el siguiente paso y
- Tell (decir): significa entregar contenido educativo a través de una declaración escrita que revela el siguiente paso.

Lin y Chi [22] plantearon la hipótesis de que las intervenciones instruccionales pueden afectar los estados latentes del estudiante de manera diferente según el tipo de intervención. Por tanto, ellos proponen un nuevo modelo denominado Intervención-Bayesian Knowledge Tracing (Intervention-BKT), que supera al BKT convencional. Con este nuevo modelo, es posible: (1) incorporar diferentes tipos de intervenciones de instrucción en el modelo del estudiante, y (2) puede separar sus efectos sobre el desempeño del estudiante formando un conjunto separado de parámetros para cada tipo de intervención.

4.2. Item Response Theory (IRT)

La Teoría de Respuesta por Elemento (Item Response Theory, IRT) [4, 5] es una herramienta para diseñar, analizar, y valorar pruebas, cuestionarios, e instrumentos similares que miden capacidades. Una de sus aplicaciones principales es la realización de pruebas adaptables mediante computadoras, donde la selección de las preguntas de la prueba se hace dinámicamente en función de las respuestas del estudiante.

IRT considera las pruebas donde cada ítem (pregunta, aspecto) tiene varias respuestas posibles. Los modelos de IRT permiten relacionar una habilidad latente del estudiante y la probabilidad de que su respuesta sea correcta. IRT se centra en las pruebas con respuestas correctas e incorrectas. IRT asume que las habilidades se mantienen constantes (no cambian) durante la prueba.

La suposición principal de IRT es que cada prueba mide una habilidad latente específica, y los modelos relacionan dicha capacidad con la probabilidad de que se responda correctamente un ítem (pregunta, inciso) de prueba. Esta relación se expresa mediante una función de respuesta logística en forma de S.

El modelo más utilizado es un modelo logístico con los parámetros siguientes:

- b es la dificultad básica de un ítem (pregunta, inciso),
- a es el factor de discriminación y
- c es un parámetro que describe la probabilidad de responder el elemento correctamente por simple adivinación

$$P_{a,b,c,\theta} = c + (1 - c) \frac{e^{a(\theta-b)}}{1 + e^{a(\theta-b)}}$$

Para aplicar estos modelos, es necesario estimar los valores de sus parámetros. Puesto que no conocemos ninguno de los parámetros (a ; b ; c), ni las habilidades de la persona (A), necesitamos estimarlos al mismo tiempo. Esto se hace generalmente mediante la estimación de máxima verosimilitud conjunta, que se realiza repitiendo dos pasos:

- estimar las habilidades a partir de los parámetros de los ítems y
- estimar los parámetros de los ítems a partir de las habilidades.

Estos pasos se repiten hasta que los valores de los parámetros convergen [11]. El proceso de estimación tiene una desventaja importante: exige una gran recolección de datos antes de la prueba (por ejemplo, 500 estudiantes y más).

BOTBLOQ: Ecosistema integral para el diseño, fabricación y programación de robots DIY

El enfoque planteado por Jarušek [2, 3, 25] plantea que el modelado de estudiantes en los STI se ocupa principalmente de modelar la corrección de las respuestas de los estudiantes. A medida que las actividades de resolución de problemas en forma interactiva se vuelven cada vez más comunes, es útil analizar también la información sobre el tiempo requerido para responder asociada con la resolución de los problemas. Se argumenta que el análisis del tiempo es natural para ciertos tipos de problemas educativos, por lo que se plantea un modelo simple de resolución de problemas que asume una relación lineal entre una habilidad latente de resolución de problemas y el logaritmo del tiempo para resolver el problema. El modelo está estrechamente relacionado con la Teoría de Respuesta por Elemento.

Una dirección interesante para futuras investigaciones sería analizar la relación entre la habilidad del estudiante según lo detectado por el modelo propuesto con otras medidas de la capacidad de resolución de problemas, por ejemplo, pruebas o modelos estandarizados usando problemas que combinan la corrección de respuestas y la información del tiempo requerido [25]

La resolución de problemas es un componente importante de la educación. Para que las actividades de resolución de problemas sean atractivas, es importante confrontar a los estudiantes con problemas de dificultad adecuada, ni demasiado fáciles ni demasiado difíciles. Dado que los estudiantes varían en sus habilidades, es crucial que las recomendaciones sobre los problemas sean adaptables individualmente. [3]

Nuestro modelo está directamente inspirado por el IRT, pero tiene una diferencia importante. El IRT se centra en las pruebas con respuestas correctas e incorrectas, mientras que con nuestro modelo podemos estudiar la resolución del problema y medir el tiempo requerido para resolverlo [3].

4.3. Multi-Armed Bandid (MAB) Problem

Los Sistemas Tutoriales Inteligentes (ITS) intentan recomendar los mejores objetos de aprendizaje disponibles para los estudiantes. Pero también quieren que los estudiantes prueben nuevos objetos para que el sistema pueda evaluarlos y poder hacer una mejor recomendación en el futuro. Esta situación puede considerarse como un problema de exploración vs explotación. Las posibles soluciones a este problema se analizan en [7], utilizando algoritmos MAB (Multi-Armed Bandit).

Los problemas MAB constituyen un ejemplo de problemas básicos de decisión secuencial que requieren un balance entre exploración y explotación, donde se requiere escoger entre permanecer con la opción más exitosa en el pasado y explorar nuevas opciones que podrían dar mejores resultados en el futuro. El término 'bandido' se refiere al término coloquial con que se reconocen las máquinas tragaperras ("one-armed bandit" en argot estadounidense) [8] donde el jugador debe equilibrar la explotación de las acciones que le dieron buenos resultados previamente y la exploración de nuevas acciones que podrían darle mayores ganancias en el futuro.

Un problema de bandido con múltiples brazos o, simplemente, un problema de bandido es un problema de asignación secuencial definido por un conjunto de acciones. En cada instante de tiempo, se asigna un recurso unitario a una acción y se obtiene algún resultado observable. El objetivo es maximizar el rendimiento total obtenido en una secuencia de asignaciones.

Existen tres planteamientos fundamentales del problema del bandido en dependencia del tipo de proceso de recompensa que se asuma [6], cada una de las cuales emplea una estrategia de juego diferente [9]:

1. Estocástico: emplea el algoritmo UCB (Upper Confidence Bound) [10]
2. Antagónico o contradictorio: emplea el algoritmo Exp3 aleatorio [11]
3. Tipo Markov: emplea los índices Gittins [9]

Para analizar el comportamiento del agente que implementa una estrategia de bandidos, podemos comparar su rendimiento con el de una estrategia óptima que consistentemente elige el mejor brazo en los primeros n pasos, cualquiera sea n .

Asumiendo que se cuenta con K brazos ($K \geq 2$) y secuencias de recompensas no conocidas $X(i,1), X(i,2), \dots$ asociadas a cada brazo i ($i = 1, \dots, K$) se evalúa el comportamiento de los agentes que en cada paso $t = 1, 2, \dots$ selecciona un brazo $I(t)$ y recibe la recompensa asociada $X(I(t), t)$. El arrepentimiento después de n jugadas $I(1), \dots, I(n)$ se define como:

$$R_n = \max_{i=1,\dots,K} \sum_{t=1}^n X_{i,t} - \sum_{t=1}^n X_{I_t,t} .$$

4.4. Interaction Networks

La mayoría de los STI apoyan el aprendizaje de los estudiantes indicándoles qué problemas seleccionar y dándoles retroalimentación que los ayuda a afirmar o corregir el trabajo del estudiante después de cada paso. Este apoyo individualizado se logra mediante:

- Una traza del conocimiento demostrado por los estudiantes basada en un modelo de dominio para medir el conocimiento del estudiante en base a un conjunto de problemas definidos por el tutor y
- Una traza del modelo paso a paso reconociendo y diagnosticando el trabajo que muestran los estudiantes durante la toma de decisiones de cada problema.

Estos modelos de dominio normalmente se instrumentan como sistemas expertos que modelan cómo se resuelven los problemas en el dominio dado. Sin embargo, en [12] se plantea el uso de redes de interacción para representar modelos de dominios basados en datos para problemas complejos. Para estos problemas el solucionador tiene que superar barreras entre un estado del problema dado y un estado de objetivo utilizando actividades con múltiples pasos e interacciones complejas entre las destrezas y habilidades empleadas por el estudiante y las necesidades cambiantes del espacio de estados.

Los métodos planteados en [12] construyen modelos de dominio a partir de los datos de rastreo de las interacciones entre humanos y computadoras, aprovechando la interfaz de resolución de problemas, así como los registros del comportamiento de los estudiantes para construir la traza del modelo y la traza del conocimiento.

Cada problema se modela mediante una red de interacción en forma de grafo con un modelo para el problema concreto, la cual recolecta los datos durante la resolución del problema de forma interactiva. Esto facilita la creación de sistemas de tutoría inteligente (ITSs) a partir de software de instrucción sin la necesidad de desarrollar modelos de dominio formal. Las redes de interacción se utilizan para marcar los pasos del problema como correctos o incorrectos llevando el control de la capacidad de cada estudiante para reconocer cuándo usar acciones específicas del tutor y asignar problemas apropiados al nivel de cada estudiante. La combinación de estos métodos permite a los desarrolladores de tutores individualizar la selección de problemas y proporcionar sugerencias para el paso siguiente y ejemplos desarrollados que pueden mejorar el aprendizaje de los estudiantes.

BOTBLOQ: Ecosistema integral para el diseño, fabricación y programación de robots DIY

Resulta fácil aplicar estos métodos a problemas de lógica y álgebra, que utilizan un conjunto de reglas definidas, tienen múltiples soluciones correctas y tienen múltiples caminos a esas soluciones. Se ha comprobado además que es posible representar problemas complejos como escribir programas valiéndose de las redes de interacción basadas en datos

En entornos de resolución de problemas, las acciones se caracterizan por aplicaciones de reglas, asignación de valores a variables y vinculación de elementos de un problema de palabra a nombres de variables, por solo mencionar algunos.

Las redes de interacción pueden construirse usando cualquier registro del sistema que pueda ser mapeado en estado, acción y tuplas de estado resultante. Hoy en día, la mayoría de los registros del entorno de aprendizaje registran información suficiente para reproducir la secuencia de comportamientos de los alumnos registrados, lo que hace muy probable que la mayoría de los tutores interactivos puedan asociarse a estas secuencias.

En el enfoque clásico BKT (Bayesian Knowledge Tracing), los modelos de estudiantes emplean la probabilidad de que los estudiantes hayan aprendido cada componente de conocimiento (KC), actualizando estas puntuaciones cada vez que el estudiante interactúa con un elemento relacionado con el KC [14]. En el enfoque planteado en [12] denominado DKT (Data-driven Knowledge Tracing) no hay KCs definidos por expertos, por lo que se definen KCs para cada acción (acción-KC), actualizándolos en base a las acciones de los estudiantes. Esto resulta en un modelo que representa si los estudiantes saben cómo y cuándo aplicar acciones [13].

Una vez que se ha construido el sistema DKT, este se usa para crear perfiles basados en datos de estrategias exitosas para completar un tutor de resolución de problemas y para la selección de problemas basados en datos (DDPS). Para ello, se identifican ejemplos que hayan completado el tutor dado y se utilizan sus datos para construir modelos de rendimiento por objetivo. Al final de cada conjunto de problemas, se calcula el conjunto de puntuaciones como sumas ponderadas de acción-KC, comparando el conjunto de puntaje obtenido con los modelos para decidir qué problemas el estudiante debe resolver a continuación. Inicialmente, se utilizan pesos establecidos por expertos para las puntuaciones de acción-KC, y una vez que se tienen datos recopilados, se agrupan los estudiantes que se escogen de modelos para crear pesos de puntuación en base a los datos

4.5. Redes de Petri

Las Redes de Petri o Petri Nets (PN) constituyen un lenguaje formal gráfico apropiado para modelar sistemas concurrentes que permiten compartir recursos. El lenguaje es una generalización de la teoría de los autómatas de manera que se puedan reflejar

BOTBLOQ: Ecosistema integral para el diseño, fabricación y programación de robots DIY

sucesos concurrentes. Las PN constituyen un tipo particular de grafo dirigido con un estado inicial, en el cual se representa el flujo de información mediante un flujo de marcadores, que representa condiciones en el gráfico. El grafo es bipartito y ponderado y consiste de dos tipos de vértices, llamados condiciones y transiciones, de forma que los arcos son de una condición a una transición o de una transición a una condición.

En los Sistemas Hipermedia Educativos Adaptativos (Adaptive Educational Hypermedia Systems AEHS) se espera que la presentación del contenido de aprendizaje sea apropiadamente recuperada de los repositorios de objetos de aprendizaje y adaptada dinámicamente a las necesidades de cada alumno. Cada alumno tiene un perfil, sujeto a cambios continuos. Los componentes básicos del perfil del alumno incluyen sus características cognitivas, antecedentes de conocimiento, experiencia previa y situación emocional actual.

En [15] se plantea la arquitectura de un AEHS que emplea un motor de flujo de trabajo basado en redes de Petri para la ejecución de flujos de cursos de aprendizaje en un entorno de red. El sistema toma en cuenta las características cognitivas del alumno y la evaluación de su progreso, para adecuar el flujo de la enseñanza a sus necesidades en función de dos variables: el Estilo de Aprendizaje (LS) del alumno y su progreso en el aprendizaje.

Para estimar el LS de un usuario, Kolb [16] plantea realizar una encuesta a los estudiantes, con la cual el educador selecciona el LS que expresa mejor las preferencias cognitivas del alumno.

La PN propuesta tiene componentes asignados a: (1) varios metadatos de Objetos de Aprendizaje, (2) un modelo de cuatro niveles que agrupa las preferencias de aprendizaje de los estudiantes en procesos de aprendizaje basados en el aprendizaje experiencial y (3) la evaluación del desempeño del alumno en cada nivel del proceso de aprendizaje. El sistema permite al alumno navegar a través de una amplia gama de material de aprendizaje.

4.6. PIGEON algorithm

En [17] se utiliza un algoritmo de aprendizaje automático denominado PIGEON [18], que ha sido utilizado con éxito [19] para enseñar a un agente cómo realizar una tarea o llevar a cabo una misión a través de una combinación de observación no perceptible de un intérprete humano (un actor) y un autoaprendizaje exploratorio.

PIGEON proviene de “Particle swarm Intelligence and Genetic programming for the Evolution and Optimization of Neural networks” (inteligencia mediante enjambre de partículas y programación genética para la evolución y optimización de redes

BOTBLOQ: Ecosistema integral para el diseño, fabricación y programación de robots DIY

neuronales). PIGEON combina neuro evolución mediante el sistema NEAT [20] con la optimización mediante enjambre de partículas (Particle Swarm Optimization PSO) [21] para realizar un rápido aprendizaje. NEAT (Neuro-Evolution of Augmenting Topologies) [20] es un algoritmo híbrido que combina redes neuronales artificiales y algoritmos genéticos (AG). La salida de NEAT es una red neural entrenada.

En PIGEON, NEAT complejiza la arquitectura de una población de redes neuronales de manera evolutiva, y luego utiliza PSO para optimizar los pesos de las redes neuronales. PSO es un algoritmo de optimización que capacita a una población de individuos de forma similar a los algoritmos genéticos. Se ha demostrado que aplicando PSO directamente a las redes neuronales es capaz de entrenarlas (i.e. establecer los pesos) con rapidez.

PIGEON utiliza NEAT para proporcionar la estructura del agente mediante una red neuronal y permite a NEAT mejorar esa estructura complejizando la estructura de la red neuronal de los agentes emergentes para satisfacer los requisitos de la función de aptitud. NEAT es esencialmente un algoritmo genético, por lo que creará una población de agentes de red neuronal. Estas redes neuronales de agentes compiten entre sí para obtener el mejor desempeño. La función de aptitud utilizada por NEAT se basa en el factor de similitud o en la clasificación de desempeño, dependiendo de la fase de entrenamiento del agente. Durante el aprendizaje observacional, la función de fitness del factor de similitud utilizada fue la similitud de la conducta de los agentes con la del actor humano. En la fase de aprendizaje experiencial, las recompensas dependían de la calificación de rendimiento del comportamiento del agente contra algún estándar absoluto. Las redes neuronales del agente que están siendo desarrolladas por PIGEON buscan maximizar su aptitud y / o recompensas.

La salida de PIGEON es un agente basado en redes neuronales que es capaz de mostrar una habilidad en el desempeño de una tarea.

Referencias

1. R Pelánek, P Jarušek, Student modeling based on problem solving times, International Journal of Artificial Intelligence in Education, December 2015, Volume 25, Issue 4, pp 493–519
2. P Jarušek, Modeling problem solving times in tutoring systems, PhD Thesis, Masarykova univerzita, 2013
3. P Jarušek, R Pelánek, Modeling and predicting students problem solving times, M. Bieliková et al. (Eds.): SOFSEM 2012, LNCS 7147, pp. 637–648, 2012. Springer-Verlag Berlin Heidelberg 2012
4. Baker, F. (2001). The basics of item response theory. University of Wisconsin.
5. De Ayala, R. (2008). The theory and practice of item response theory. The Guilford Press.
6. Sébastien Bubeck and Nicolò Cesa-Bianchi, “Regret Analysis of Stochastic and Nonstochastic Multi-Armed Bandit Problems”, To appear in Foundations and Trends in Machine Learning, 2012 [arXiv:1204.5721](https://arxiv.org/abs/1204.5721)
7. Minh-Quan Nguyen, Multi-Armed Bandit Problem and Its Applications in Intelligent Tutoring Systems, Master Thesis, Ecole Polytechnique Paris, 2014
8. A. Mahajan and D. Teneketzis. Multi-armed bandit problems. In Foundations and Applications of Sensor Management, pages 121–151. Springer, 2008.
9. J. Gittins, K. Glazebrook, and R. Weber. Multi-Armed Bandit Allocation Indices (2nd edition). John Wiley and Sons, 2011.
10. TL Lai and H Robbins. Asymptotically efficient adaptive allocation rules. Advances in applied mathematics, 22:4–22, 1985.
11. Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund Robert E. Schapire, The non-stochastic multi-armed bandit problem, SIAM Journal on Computing, 2002
12. Tiffany Barnes, Behrooz Mostafavi, and Michael J. Eagle, CHAPTER 12 – Data-Driven Domain Models for Problem Solving, in “Design Recommendations for Intelligent Tutoring Systems: Volume 4 - Domain Modeling” by Bob Sottolare, US Army Research Laboratory, page 137, July 2016
13. Eagle, M. & Barnes, T. (2012). Data-driven methods for assessing skill-opportunity recognition in open procedural problem solving environments. In 11th International Conference on Intelligent Tutoring System (pp. 615–617). Chania, Crete, Greece.
14. Desmarais, M. & Baker, R. (2011). A review of recent advances in learner and skill modeling in intelligent learning environments. User Modeling and User-Adapted Interaction, 22(1–2), 9–38.
15. Georgios Tsoulouhas, Dimitrios Georgiou and Alexandros Karakos, Adaptive Content Presentation in Asynchronous Learning Environments, iJET – Volume 7, Issue 2, June 2012
16. D. A. Kolb, “Learning Style Inventory - version 3: Technical Specifications,” TRG Hay/McBer, Training Resources Group, 1999.
17. Gary Stein, Member, Avelino J. Gonzalez, and Clayton Barham, Machines that learn and teach seamlessly, Ieee Transactions On Learning Technologies, VOL. 6, NO. 4, October-December 2013

BOTBLOQ: Ecosistema integral para el diseño, fabricación y programación de robots DIY

18. G. Stein, "FALCONET: Force-Feedback Approach for Learning from Coaching and Observation Using Natural and Experiential Training," doctoral dissertation, Computer Eng., Univ. of Central Florida, Aug. 2009.
19. G. Stein and A.J. Gonzalez, "Building High-Performing Human-Like Tactical Agents through Observation and Experience," IEEE Trans. Systems, Man and Cybernetics, Part B, vol. 41, no. 3, pp. 792-804, June 2011.
20. K. Stanley and R. Miikkulainen, "Evolving Neural Networks through Augmenting Topologies," Evolutionary Computation, vol. 10, no. 2, pp. 99-127, 2002.
21. J. Kennedy and R. Eberhart, "Particle Swarm Optimization," Proc. IEEE Int'l Conf. Neural Networks, 1995.
22. Chen Lin and Min Chi, Intervention-BKT: Incorporating Instructional Interventions into Bayesian Knowledge Tracing, Springer International Publishing Switzerland 2016. A. Micarelli et al. (Eds.): ITS 2016, LNCS 9684, pp. 208–218, 2016. DOI: 10.1007/978-3-319-39583-8 20
23. Corbett, A. T.; Anderson, J. R. (1995). "Knowledge tracing: Modeling the acquisition of procedural knowledge". User Modeling and User-Adapted Interaction. **4** (4): 253–278.
24. K. Vanlehn. The behavior of tutoring systems. International Journal of Artificial Intelligence in Education, 16(3):227–265, 2006.
25. Pelánek, R. & Jarušek, P. Int J Artif Intell Educ (2015) 25: 493. doi:10.1007/s40593-015-0048-x
26. Dian Schaffhauser (2014-07-14). "D2L Unveils Revamped Platform 'Brightspace' with Adaptive Learning". THE Journal.
27. Henderson, J. "Smart Sparrow targets Next Generation Learning after securing first Round of Funding". NewSouth Innovations, 2012
28. Doignon, Jean-Paul; Falmagne, Jean-Claude (1999), "A Practical Application: The Aleks System", Knowledge Spaces, Berlin, Heidelberg, and New York: Springer-Verlag, pp. 9–10.
29. Upbin, Bruce (October 31, 2011). "Pearson, Knewton Team Up To Personalize College". Forbes. Retrieved 2011-11-02.
30. Weber, Matt. "Harvard EdCast: Adaptive Ed Tech". Harvard Graduate School of Education. Retrieved 3 July 2013.
31. Bloom, Benjamin S. (1984). The 2-Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring. Educational Researcher, 13(6), 4-16. <http://dx.doi.org/10.3102/0013189X013006004>
32. Elsom-Cook, Mark. (1987). Intelligent Computer-aided Instruction Research at the Open University. Technical Report No: 63. Computer-Assisted Learning Research Group. The Open University: Milton Keynes.

BOTBLOQ: Ecosistema integral para el diseño, fabricación y programación de robots DIY

33. Sonwalkar, Nishikant. (2005). Adaptive Learning Technologies: From One-size-fits-all to Individualization. Educause Center for Applied Research, 7, 1-11.
34. Corbett, A., & Anderson, J. R. (1995). Knowledge tracing: Modeling the acquisition of procedural knowledge. User Modeling and User-Adapted Interaction, 4, 253-278.
35. Juan Carlos Fernández Rodríguez, José Javier Rainer Granados, Fernando Miralles Muñoz, Engineering education through eLearning technology in Spain, IJIMAI, ISSN-e 1989-1660, Vol. 2, Nº. 1, 2013, págs. 46-50
36. Rosado, Muñoz-Marí and Magdalena. "Herramienta eLearning para la programación de robots mediante entorno web". Revista d'Innovació Educativa, 1, 2008, pp. 45-48.
37. H. Vargas, J. Sánchez, C.A. Jara, F.A. Candelas, O. Reinoso and J.L. Díez. "Docencia en automática. Aplicación de las TIC a la realización de actividades prácticas a través de Internet". Revista Iberoamericana de Automática e Informática Industrial, (7) 1, 2010, pp. 35 - 45.
38. M. C. Desmarais and R. S. J. de Baker. A review of recent advances in learner and skill modeling in intelligent learning environments. User Model. User-Adapt. Interact., 22(1-2):9 - 38, 2012.
39. P. Jarusek and R. Pelanek. A web-based problem solving tool for introductory computer science. In Proc. of Innovation and technology in computer science education, pages 371 - 371. ACM, 2012.
40. P. Jarusek and R. Pelanek. Problem response theory and its application for tutoring. In Educational Data Mining, pages 374 - 375, 2011.
41. B. Martin, A. Mitrovic, K. R. Koedinger, and S. Mathan. Evaluating and improving adaptive educational systems with learning curves. User Modeling and User-Adapted Interaction, 21(3):249 - 283, 2011.
42. A. Newell and P. Rosenbloom. Mechanisms of skill acquisition and the law of practice. Cognitive skills and their acquisition, pages 1 - 55, 1981.
43. A. Corbett and J. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. User modeling and user-adapted interaction, 4(4):253 - 278, 1994.
44. P. Jarusek and R. Pelanek, Analysis of a Simple Model of Problem Solving Times, S.A. Cerri et al. (Eds.): ITS 2012, LNCS 7315, pp. 379–388, 2012. Springer-Verlag Berlin Heidelberg 2012
45. Rasch, G. (1960). Probabilistic models for some intelligence and attainment tests. Copenhagen: Danish Institute for Educational Research.
46. M. Fjeld and B. M. Voegtli, "Augmented chemistry: An interactive educational workbench," 2002, p. 259.
47. O. Bimber and R. Raskar, Spatial augmented reality: Merging real and virtual worlds: AK Peters Ltd, 2005.
48. M. Zaeh and W. Vogl, "Interactive laser-projection for programming industrial robots," in Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality, 2006, pp. 125-128.
49. Billard, A., Calinon, S., Dillmann, R., and Schaal, S. (2008). "Robot programming by demonstration," in Handbook of Robotics, eds B. Siciliano and O. Khatib (Berlin/Heidelberg: Springer), 1371–1394.

BOTBLOQ: Ecosistema integral para el diseño, fabricación y programación de robots DIY

50. Gary Stein, Avelino J. Gonzalez, and Clayton Barham, Machines that Learn and Teach Seamlessly, IEEE Transactions on Learning Technologies, VOL. 6, NO. 4, October - December 2013
51. Bennie G. Lewis Jr. Human Robot Interaction for Multi-Robot Systems, University of Central Florida, Orlando, Florida, 2014
52. H. Cheng, Z. Sun, and P. Zhang. Imirok: Real-time imitative robotic arm control for home robot applications. In IEEE International Conference on Pervasive Computing (Work in Progress), pages 360–363, 2011.
53. S. Pieska, J. Kaarela, and O. Saukko. Toward easier human-robot interaction to help inexperience operators in smes. In IEEE International Conference on Cognitive Infocommunications, pages 333–338, Dec 2012.
54. Gary Stein and Avelino J. Gonzalez, Building and Improving Tactical Agents in Real Time through a Haptic-Based Interface, J. Intell. Syst. 2015; 24(4): 383–403
55. D. H. Grollman and A. G. Billard, Robot learning from failed demonstrations, Int. J. Soc. Robotic. 4 (2012), 331–342.
56. J. Aleotti, S. Caselli and M. Reggiani, Evaluation of virtual fixtures for a robot programming by demonstration interface, IEEE T. Syst. Man Cyb. Part A. 35 (2005), 536–545.
57. S. Eguchi, K. Sugiyama and Y. Wada, Robot task learning using haptic interface in virtual space, in: SICE 2004 Annual Conference, vol. 2, 2004.
58. S. Calinon and A. G. Billard, “What is the teacher’s role in robot programming by demonstration? Toward benchmarks for improved learning”, Interaction Studies 8 (2007), 441–464.
59. A. Kucukyilmaz, T. M. Sezgin and C. Basdogan, Intention recognition for dynamic role exchange in haptic collaboration, IEEE Trans. Haptics 6 (2013), 58–68.
60. Alexander Henning, Brendan McLeod, Fredric Silberberg, FRC Control System Usability Analysis, Worcester Polytechnic Institute, 2013
61. https://www.tel-thesaurus.net/wiki/index.php/Intelligent_tutoring_system
62. Intelligent Tutoring Systems, EDUCAUSE Learning Initiative (ELI), Collection ELI 7 Things You Should Know 2013
63. Joseph Psotha, Sharon A. Mutter (1988). *Intelligent Tutoring Systems: Lessons Learned*. Lawrence Erlbaum Associates. ISBN 0-8058-0192-8. https://en.wikipedia.org/wiki/Intelligent_tutoring_system
64. Anderson, J., Boyle, C., Farrell, R., & Reiser, B. (1987). Cognitive principles in the design of computer tutors. In P. Morris (Ed.), *Modeling cognition*. NY: John Wiley.
65. Corbett A. T., Koedinger, K. R., & Anderson, J. R. (1997). Intelligent tutoring systems. In M. G. Helander, T. K. Landauer, & P. V. Prabhu (Eds.), *Handbook of human-computer interaction* (pp. 849–874). Amsterdam: Elsevier.
66. Corbett A. T., Koedinger, K. R., & Anderson, J. R. (1997). Intelligent tutoring systems. In M. G. Helander, T. K. Landauer, & P. V. Prabhu (Eds.), *Handbook of human-computer interaction* (pp. 849–874). Amsterdam: Elsevier.
67. https://www.tel-thesaurus.net/wiki/index.php/Intelligent_tutoring_system

BOTBLOQ: Ecosistema integral para el diseño, fabricación y programación de robots DIY

68. <http://edutechwiki.unige.ch/mediawiki/index.php?oldid=30245>
69. Burns H.& Capps C. (1988) Foundation of intelligent tutoring systems: an introduction. In: Polson M. C., Richardson J. J. (eds.), Foundations of Intelligent tutoring systems (pp.1-19). Hillsdale, NJ: Laurence Erlbaum.
70. Nkambou R., Bourdeau J., Mizoguchi R. (eds.) (2010). Advances in Intelligent tutoring systems. Springer Verlag.
71. Self J. (1999) The defining characteristics of intelligent tutoring systems research: ITSs care, precisely. International Journal of Artificial Intelligence in Education, 10, 350-364
72. Sleeman D., Brown J. S. (eds) (1982) Intelligent tutoring systems. London: Academic Press.
73. Wenger E. (1987) Artificial Intelligence and Tutoring Systems. Los Altos, CA: Kaufman Publishers.
74. Woolf B. (2009) Building Intelligent Interactive Tutors: Student-centered Strategies for Revolutionizing E-learning. Burlington, MA, Morgan Kaufmann.