

El rol de la simulación en la teoría constructivista

Sonia Rueda

Departamento de Ciencias de la Computación
Universidad Nacional del Sur
(8000) Bahía Blanca
srueda@criba.edu.ar

Resumen

La filosofía educativa basada en el constructivismo establece que los niños aprenden mejor cuando construyen cosas, considerando que la manera más motivadora de aprender acerca de un fenómeno o proceso es participando activamente en su creación. La computadora brinda en este sentido facilidades únicas por su dinamismo, interactividad y flexibilidad.

La simulación en particular es una herramienta muy poderosa y puede llegar a transformar el modo en que los niños piensan y aprenden. Una simulación simbólica es un micromundo de objetos individuales interactuando, controlado por una computadora. Una simulación permite expresar ideas dinámicas en un medio interactivo.

En la actualidad existe una gran variedad de programas de simulación para niños, en general muy atractivos y motivadores. Estos simuladores permiten que los niños construyan mundos únicos, con la sensación de que controlan su creación. Sin embargo la mayoría de los programas de simulación actuales son de propósito específico. La programación es en este sentido una alternativa más flexible.

La programación convencional requiere el aprendizaje de un lenguaje. Un lenguaje de programación exige el uso de una sintaxis muy rigurosa e inflexible que permanentemente distrae al niño de su tarea principal. El ambiente de trabajo debería favorecer el proceso creativo y no entorpecerlo. Debería utilizarse entonces alguna alternativa mucho más simple y natural, pero al mismo tiempo rica y expresiva. Las interfaces gráficas han resultado muy atractivas ya que son fáciles de usar y muy intuitivas.

El lenguaje de programación no es la única razón que dificulta esta actividad. Escribir un programa requiere llevar a cabo un proceso de abstracción cuya complejidad depende del problema. Es preciso modificar entonces el modo de pensar acerca de los programas usando una metáfora que tome elementos del mundo real para construir simulaciones simbólicas.

La programación basada en agentes permite construir mundos de objetos autónomos interactivos. Los sistemas de agentes toman elementos de los agentes vivos del mundo real para construir mundos virtuales.

En este artículo se analiza el rol de la simulación simbólica en la teoría constructivista, se describen algunos programas de simulación de propósito específico que favorecen el aprendizaje exploratorio no estructurado y se propone a la programación basada en agentes como metáfora de programación para la construcción de simulaciones simbólicas.

El rol de la simulación en la teoría constructivista

Introducción

La simulación simbólica es una herramienta muy valiosa en educación porque puede llegar a transformar el modo en que los niños piensan y aprenden. A través de una simulación un niño puede construir un mundo de objetos autónomos interactuando, como un juego de video o un simulador del mundo submarino [Travers].

En la actualidad existe una gran variedad de programas de simulación para niños, en general muy atractivos y motivadores. Estos simuladores permiten que los niños construyan mundos únicos, con la sensación de que controlan su creación. Sin embargo la mayoría de los programas de simulación actuales son de propósito específico. La programación es en este sentido una alternativa más flexible.

La programación convencional requiere el aprendizaje de un lenguaje. Un lenguaje de programación exige el uso de una sintaxis muy rigurosa e inflexible que permanentemente distrae al niño de su tarea principal. El ambiente de trabajo debería favorecer el proceso creativo y no entorpecerlo. Debería utilizarse entonces alguna alternativa mucho más simple y natural, pero al mismo tiempo rica y expresiva. Las interfaces gráficas han resultado muy atractivas ya que son fáciles de usar y muy intuitivas.

El lenguaje de programación no es la única razón que dificulta la creación de una simulación. Escribir un programa requiere llevar a cabo un proceso de abstracción cuya complejidad depende del problema. Es preciso entonces modificar el modo de pensar acerca de los programas usando una metáfora que tome elementos del mundo real para construir simulaciones simbólicas.

La programación basada en agentes permite construir mundos de objetos autónomos interactivos. Los sistemas de agentes toman elementos de los agentes vivos del mundo real para construir mundos virtuales.

En este artículo se analiza el rol de la simulación simbólica en la teoría constructivista, se describen algunos programas de simulación de propósito específico que favorecen el aprendizaje exploratorio no estructurado y se propone a la programación basada en agentes como metáfora de programación para la construcción de simulaciones simbólicas.

Teorías Educativas

En la historia de la epistemología la visión del conocimiento se ha movido desde un enfoque esencialmente pasivo hacia otro más activo. En esta evolución el objetivismo y el constructivismo representan extremos opuestos [Murphy].

En las primeras teorías cada objeto de estudio tenía un significado intrínseco, independiente del sujeto que lo estudiaba. En un enfoque objetivista el mundo real tiene una existencia propia y el conocimiento es correcto si refleja esa realidad fielmente. Bajo este enfoque surge la Teoría Conductista del aprendizaje.

En contraste, la visión constructivista argumenta que ni el conocimiento ni la realidad tienen un significado o valor absoluto o al menos no hay forma de conocer esa realidad con exactitud. El sujeto interpreta y construye su propia realidad en base a su experiencia y a su interacción con el entorno. Bajo esta teoría el conocimiento no es una copia de la realidad, sino una construcción que el ser humano realiza a partir de elementos ya construidos anteriormente [Carretero].

Los investigadores han desarrollado variaciones del constructivismo que han hecho evolucionar la teoría en direcciones diferentes. Mientras que las diferencias entre constructivismo y objetivismo están claramente delineadas, es más difícil establecer los límites precisos de cada uno de los enfoques constructivistas.

Cada una de estas teorías epistemológicas ha dado lugar a una teoría del aprendizaje. Desarrollar una Teoría Constructivista del Aprendizaje representa un desafío porque requiere sintetizar un espectro muy amplio de conceptos, a veces dispares entre sí. A pesar de estas diferencias es posible desarrollar una epistemología y una teoría del aprendizaje constructivista que exprese los principios fundamentales y las características comunes a todas las variaciones.

La Teoría Conductista

La teoría conductista parte del principio teórico, muy extendido, según el cuál el individuo trae al nacer un potencial genético de aptitudes preformadas cuyo desarrollo será natural e independiente de los factores sociales que lo rodean [Carretero].

La escuela, en este enfoque, tiene como función principal transmitir a los niños la mayor cantidad de conocimientos y conceptos acumulados por las generaciones anteriores. El niño retendrá más o menos información dependiendo de sus capacidades innatas, en particular de su atención y memoria. El aprendizaje consiste en un proceso de asimilación de conocimientos que, se espera, reflejan la realidad tal cual es. Hay poco espacio para preguntas, pensamientos independientes o interacción entre alumnos. En este enfoque un buen maestro es un conferenciante claro y ameno, con un trato afectuoso pero severo hacia los alumnos.

El primer riesgo de esta teoría es formar niños que realicen correctamente los ejercicios escolares de manera mecánica, sin haber desarrollado las bases intelectuales que les permitan comprenderlos. Los niños tienden a desconectar los conocimientos adquiridos formalmente en la escuela de los problemas que enfrentan cotidianamente. Existe entonces una disociación entre el saber formal y el informal.

Esta disociación suele aumentar en los grados superiores, los contenidos se hacen más académicos y las actividades más formales. La mayoría de los niños pierde el interés y esta situación se acentúa en la adolescencia. Aunque su capacidad cognitiva es mayor, su rendimiento suele disminuir debido a su falta de motivación. No existe conexión entre sus actividades cotidianas y las propuestas por la escuela, en general demasiado formales y desvinculadas con sus intereses [Sastre y Moreno].

Este riesgo, aunque importante, afecta únicamente a la etapa escolar. Qué sucede más adelante? Si alguna utilidad tiene el desarrollar la capacidad de pensar en los individuos, no es para que puedan reproducir ciegamente los conocimientos que la humanidad ha ido acumulando a lo largo de los siglos, sino para que sean capaces de crear otros nuevos. Es evidente que este objetivo requiere que la escuela no se limite a transmitir información sino que proporcione los instrumentos necesarios para el desarrollo intelectual.

La Teoría Constructivista

La teoría constructivista considera que cada individuo no es un producto generado por el ambiente que lo rodea ni un simple resultado de sus disposiciones internas, sino una **construcción** propia que se produce día a día a partir de la interacción de estos dos factores.

Todo individuo nace en el seno de una sociedad, su conducta, su sistema de valores e incluso su forma intelectual de organizar la realidad depende en parte de la cultura de esa sociedad.

Inteligencia y cultura son dos aspectos de un mismo fenómeno y existe un equilibrio dinámico entre la inteligencia de un individuo y el sistema cultural de su entorno. Según Piaget la inteligencia puede pensarse como un caso particular de adaptación al medio, a través de la construcción del sistema de respuestas inteligentes que un individuo elabora a lo largo de su historia personal [Sastre y Moreno].

Según Lévi-Strauss “Cada niño trae consigo al nacer la suma total de habilidades potenciales, pero sólo retendrá y desarrollará algunas dependiendo de la cultura y el período histórico en que esté inserto”.

El rol de la escuela consiste en estimular sus potencialidades innatas para alcanzar un mejor desarrollo personal. Así, el nivel evolutivo de cada niño dependerá de la interacción entre su potencial genético y el aprendizaje escolar. En este enfoque el maestro organiza y plantea situaciones problemáticas que estimulan el ejercicio autónomo y crítico. El maestro es un guía que inserta elementos constructivos en el entorno, plantea inquietudes, propone y coordina actividades, incentiva la reflexión y la discusión de ideas.

Cada alumno interpreta y construye la realidad en base a su experiencia y a su interacción con el entorno. El aprendizaje requiere la construcción de estructuras conceptuales a través de la reflexión y la abstracción. El énfasis en la resolución de un problema está dado en el proceso, no en la solución objetiva. El aprendizaje es un proceso durante el cual se construyen representaciones significativas.

La realidad puede enfocarse además bajo diferentes perspectivas y dar lugar a diferentes representaciones, con lo cual la noción de correcto o incorrecto puede quedar desdibujada. La multiplicidad es un concepto central en el constructivismo.

El constructivismo brinda una sólida base conceptual sobre la que se han diseñado e implementado una gran cantidad de ambientes de aprendizaje. Aun así, la línea entre la filosofía educativa y la práctica es tenue. Pasar de la teoría a la práctica es, como en cualquier dominio, un desafío interesante.

La tecnología actual brinda medios muy ricos para la aplicación de los principios del constructivismo. Los ambientes basados en esta tecnología han demostrado ser recursos muy apropiados para pasar de la teoría a la práctica.

Limitaciones de la Teoría Constructivista

La enseñanza por descubrimiento demanda mayor cantidad de tiempo que la enseñanza expositiva. Uno de los mayores inconvenientes de la teoría constructivista es entonces que, al evitar los elementos expositivos, resulta indispensable reducir la amplitud de contenidos curriculares.

Otro de los inconvenientes es que aprender no se reduce a comprender y no todo el conocimiento que adquiere un individuo se construye a partir de otros conocimientos previos. Existen objetivos que pueden alcanzarse con mayor eficacia a través de otros tipos de aprendizaje. Por ejemplo para aprender a escribir a máquina, memorizar los símbolos químicos o los nombres de los ríos el aprendizaje por repetición resulta más efectivo. Estos conocimientos adquiridos de manera no significativa, resultarán indispensables para adquirir otros de manera significativa. Si bien es cierto que existe mayor probabilidad de retener aquello que puede ligarse con nuestros conocimientos previos, que se retengan no significa que puedan aplicarse adecuadamente en diferentes situaciones.

Las ideas previas o conceptos espontáneos son conocimientos que permiten interpretar nuevas experiencias, a veces de manera incompleta. Otro de los problemas es entonces que dentro de las ideas previas a partir de las cuales se construyen otras nuevas, existen seguramente algunas incorrectas y difíciles de cambiar porque forman parte de su propia concepción y por lo tanto tienen mucho valor para el individuo.

Cuanto más arraigado esté un concepto será más fácil aplicarlo para construir otros, pero también resultará más difícil cambiarlo si es erróneo. Si la enseñanza se plantea como un conjunto de acciones dirigidas a favorecer la construcción de nuevos conocimientos, es fundamental analizar cuidadosamente los conceptos previos de los alumnos.

El rol de la computadora en la educación

La **computadora** es una herramienta valiosa para cualquiera de los dos enfoques.

Una alternativa es usar la computadora como un recurso de aplicación y refuerzo de las actividades del aula. Actualmente hay una enorme cantidad de software que propone actividades escolares, el niño las resuelve, la computadora analiza la respuesta. Muchos de estos productos no aportan nada demasiado nuevo salvo el aprovechar la atracción natural de los niños por las computadoras. Existe también software didáctico con explicaciones atractivas y estimulantes. Sin embargo, como sucede ante un programa de televisión, el niño tiene un rol receptor y se mantiene en un plano pasivo. Ambas alternativas son conductistas.

Como hemos visto, la filosofía educativa basada en el constructivismo considera que los niños aprenden mejor cuando contruyen cosas, ya que la manera más motivadora para aprender acerca de un fenómeno o proceso es participando activamente en su creación. La computadora brinda en este sentido facilidades únicas por su dinamismo, interactividad y flexibilidad.

La computadora es una especie de dispositivo universal en el sentido que puede soportar el diseño de proyectos de música, arte, ciencia, matemática [Resnick y Rusk]. Un grupo de niños puede usar la computadora para controlar un robot, otros pueden crear una animación gráfica y otros componer música. Aunque sus intereses son evidentemente muy diferentes, todos están desarrollando algunas habilidades compartidas. Ninguna de estas actividades enseña conceptos directamente, pero permite retomar y reforzar conceptos aprendidos en el aula, como parte de un proyecto integrado que conecta los intereses de los niños con los contenidos propuestos por la escuela.

Proponer una caracterización para un ambiente de aprendizaje constructivista representa un gran desafío porque requiere la sintetización de un gran espectro de conceptos dispares.

Ambientes de aprendizaje constructivistas

Una gran cantidad de desarrollos de software han aplicado el constructivismo para el diseño de ambientes de aprendizaje. El mayor desafío para estos ambientes es favorecer el desarrollo de proyectos y actividades que permitan realizar la transición entre la filosofía constructivista y la práctica [Murphy].

La siguiente lista enuncia algunas de las características que debería reunir un ambiente de aprendizaje para soportar los principios del constructivismo. Esta lista puede servir como instrumento para diseñar ambientes de software constructivistas o como para evaluar el soporte de uno ya desarrollado.

- Crear ambientes que reflejen una porción del mundo real, relevante para el aprendizaje.

- Enmarcar el aprendizaje en ambientes que desarrollen el razonamiento y la capacidad para resolver problemas.
- Favorecer el desarrollo de estrategias útiles para la resolución de problemas.
- Brindar mecanismos sencillos para representar conceptos y relaciones complejas.
- Desarrollar experiencia en el proceso de construcción de conocimientos.
- Enfatizar la construcción sobre la reproducción de conocimientos.
- Permitir que el alumno armonice los objetivos escolares propuestos por el maestro con sus propios intereses. El alumno es el protagonista del aprendizaje y puede insertar elementos que le resultan atractivos o cercanos a su realidad cotidiana.
- Permitir la construcción de múltiples perspectivas, interpretaciones y representaciones para un mismo concepto.
- Evaluar a través de herramientas de autocorrección que favorezcan el análisis y la reflexión. Tratar al error como un mecanismo de realimentación que mejore la comprensión del problema.
- Alentar la construcción colaborativa a partir del trabajo social.
- Permitir que los alumnos ejecuten tareas que escapen a sus habilidades adquiridas.
- Favorecer la exploración y el trabajo independiente

Los ambientes computacionales que soportan estos principios resultan herramientas extraordinariamente poderosas y pueden llegar a transformar el modo en que los niños piensan y aprenden.

Por supuesto el ambiente de desarrollo por sí mismo no asegura el éxito. La mejor herramienta de software puede quedar totalmente desvirtuada si el maestro fija un objetivo con total precisión, le indica a los niños como alcanzarlo y luego ellos siguen la receta rigurosamente. Cada proyecto debe tener una estructura básica dentro de la cual el niño pueda moverse libremente.

En los últimos años un número creciente de investigadores y educadores han encontrado en la simulación simbólica una herramienta poderosa para el aprendizaje. En la actualidad existe una gran variedad de simuladores que reúnen muchas de las características enunciadas.

La simulación

Una simulación simbólica es un micromundo de objetos individuales interactuando, controlado por una computadora. La simulación es una poderosa herramienta en educación porque permite crear mundos virtuales que favorecen el aprendizaje exploratorio. Los niños construyen mundos únicos, con la sensación de que controlan su creación. Las leyes de la física, los patrones del comportamiento animal, dejan de ser teorías abstractas y se transforman en modelos concretos y visibles. Mientras observan y modifican el mundo creado por ellos mismos, aprenden acerca del mundo que intentan modelar.

Algunos de los beneficios más significativos de los simuladores son:

- Favorecen una actitud participativa, dándole a los niños una sensación de control y responsabilidad sobre el proceso modelado.
- Desarrollan la capacidad de resolver problemas en forma creativa, evitando la tradicional dicotomía entre el éxito/fracaso de muchas de las actividades convencionales en matemáticas y ciencia.
- Permiten reconciliar los objetivos de la escuela con los intereses de los niños, permitiendo vincular las actividades cotidianas con las actividades de la escuela.
- Favorecen el desarrollo de actividades interdisciplinarias creando asociaciones entre conceptos matemáticos, sociales y artísticos.

- Brindan un contexto para la reflexión y la discusión, resaltando la multiplicidad de estrategias y soluciones y permitiendo alcanzar un alto nivel de comprensión para los temas tratados.

El desarrollo de un proyecto atraviesa diferentes etapas. Si la herramienta es nueva es probable que la etapa inicial sea exploratoria, el niño juega escogiendo opciones y analizando el efecto producido por cada una. En la siguiente etapa resulta útil usar algún producto similar al que se intenta desarrollar y probablemente intentar reproducirlo. En general antes de que sea reproducido completamente el niño introduce variaciones y desarrolla sus propias ideas.

Los proyectos de simulación pueden utilizarse para desarrollar la capacidad de trabajo en grupo. El maestro puede proponer el mismo o diferentes proyectos a varios grupos de niños. Cada grupo parte de la especificación inicial pero luego los modifica y enriquece con sus propias ideas e intereses. Algunos de los grupos pueden decidir luego unirse y combinar sus proyectos en otro más ambicioso. El trabajo es colaborativo pero el rol de cada grupo no está definido inicialmente, la participación surge en forma natural y dinámica, guiada por los intereses y necesidades de los participantes.

En la actualidad existen algunos simuladores muy interesantes a partir de los cuales es posible diseñar proyectos muy enriquecedores y atractivos.

El programa “Pinball construction set“ permite construir juegos de pinball “arrastrando y dejando caer” las piezas del juego sobre un tablero. Las piezas comienzan a funcionar tan pronto como “caen” sobre el tablero. La manipulación directa es muy sencilla e intuitiva, sin embargo el dominio de aplicación del programa es muy específico y eso lo hace poco versátil.

El programa “Destination: Ocean” permite que los niños creen películas animadas con escenarios y personajes del fondo del mar. Pueden reproducir la migración de las ballenas, el comportamiento de los tiburones o inventar historias y aventuras totalmente fantásticas. Este programa forma parte de un paquete de aplicaciones cada una de ellas orientada a un tema diferente: el barrio, los castillos, las pirámides, el espacio y el bosque. Todos estos programas desarrollan la creatividad, la capacidad de diseñar y escribir historias y permiten incorporar componentes multimediales en el proceso.

“The incredible machine” simula un laboratorio de física en donde los niños pueden resolver algunos puzzles propuestos o diseñar ellos mismos máquinas increíbles. En el contexto del programa una máquina está constituida por una serie de artefactos que reaccionan y trabajan en conjunto y provocan alguna clase de efecto final. Por ejemplo una pelota puede caer debido a la fuerza de la gravedad sobre una linterna que, al encenderse, provoca un haz de luz que se refracta a través de una lupa y enciende la mecha de un cañón. En la etapa exploratoria los niños resuelven algunos de los puzzles propuestos y se familiarizan con el funcionamiento de cada una de las piezas provistas. En esta etapa los niños se enfrentan con algunas nociones básicas de física como gravedad, presión atmosférica, fricción, etc. En la siguiente etapa el desafío es mayor, ellos mismos deben diseñar y construir una máquina que produzca algún tipo de efecto. Una vez implementada pueden remover algunas de las piezas y proponer el nuevo puzzle a sus compañeros.

“SimCity” es un programa de simulación que permite diseñar y construir ciudades completas. El problema de SimCity es que algunos parámetros importantes están fijos y eso restringe la posibilidad de modelar algunas situaciones interesantes. Esta inflexibilidad impone fuertes restricciones sobre los proyectos que pueden desarrollarse usando este programa.

Todos estos programas son muy interesantes y atractivos, pero también son muy específicos y relativamente poco versátiles. La mayoría de los programas de simulación actuales solo permiten que los usuarios modifiquen algunos parámetros, pero los fundamentos básicos están dados y no pueden cambiarse. Esta rigidez reduce la posibilidad de expresar nuevas ideas y limita su utilidad pedagógica. La forma más flexible de construir una simulación es programándola.

Programación de Simulaciones

Los ambientes y lenguajes de programación convencionales permiten desarrollar mundos virtuales, sin embargo no resultan adecuados para esta tarea. Un lenguaje de programación convencional exige el uso de una sintaxis muy rigurosa e inflexible que permanentemente distrae al niño de su tarea principal. A la mayoría de los ellos les encanta manipularla, tener control sobre ella pero no les resulta natural el modo de indicarle sus deseos. El ambiente de trabajo debería favorecer el proceso creativo y no entorpecerlo. Debería utilizarse entonces alguna alternativa mucho más simple y natural, pero al mismo tiempo rica y expresiva.

El lenguaje de programación no es la única razón que dificulta esta actividad. Escribir un programa, cualquier programa, requiere llevar a cabo un proceso de abstracción cuya complejidad depende el problema. La concepción de un programa como una secuencia de instrucciones, en la que se basa el paradigma procedural, requiere una considerable capacidad de contorsión mental. Aún el paradigma de orientación a objetos, basado en el pasaje de mensajes, exige un fuerte nivel de abstracción. En particular, ambas metodologías resultan demasiado complejos para los niños.

No podemos eliminar la complejidad inherente al problema de construir un mundo virtual, pero si podemos buscar herramientas lo suficientemente expresivas como para no complicar la tarea aún más. La construcción de mundos dinámicos requiere entonces de ambientes y lenguajes de programación que brinden una nueva forma de pensar acerca de los programas, pero también de una metáfora que resulte natural para la construcción de simulaciones simbólicas. Una alternativa es modificar totalmente el modo de pensar acerca de los programas usando una metáfora que tome elementos del mundo real para construir simulaciones simbólicas.

Sistemas Animados

El término *sistemas animados* ha sido introducido en la bibliografía para referirse a simulaciones de mundos dinámicos interactivos, constituidos por objetos autónomos que interactúan entre sí. Los objetos de un sistema animado deben ser:

- tangibles, esto es, concretos desde los sentidos, y manipulables directamente
- reactivos, es decir reaccionan motivados por estímulos el usuario u otros objetos
- incrementales, deben poder ser modificados para explorar nuevas posibilidades
- flexibles, el sistema debe permitir que las nuevas habilidades se construyan a partir de las anteriores

Los sistemas animados permiten que los niños construyan mundos únicos, con la sensación de que controlan su creación. Las leyes de la física, los patrones del comportamiento animal, dejan de ser teorías abstractas y se transforman en realidades tangibles. Mientras observan y modifican el mundo creado por ellos mismos, aprenden acerca del mundo que intentan modelar.

Programación Basada en Agentes

Una alternativa que permite construir mundos de objetos autónomos interactivos es la **programación basada en agentes**. Esta alternativa resulta atractiva ya que es natural construir sistemas animados basándose en una metáfora que toma elementos de los agentes vivos del mundo real para construir un mundo virtual.

Un agente es una componente del programa diseñada para ser vista como objeto autónomo e interactivo. Un agente sugiere una variedad de atributos tales como sentido de propósito, autonomía, capacidad de reacción ante estímulos, etc. Los agentes son autónomos en el sentido de que puede iniciar acciones por sí mismos en función de su meta. Un agente es un mecanismo simple. Sin embargo, a partir de la combinación de agentes simples es posible obtener estructuras más elaboradas capaces de modelar un sistema complejo.

Interfaces Gráficas

La programación basada en agentes brinda las herramientas y los fundamentos conceptuales necesarios para construir sistemas animados. Ahora bien, ¿Cómo puede un niño describir el comportamiento de un agente?

Los niños se familiarizan con sorprendente facilidad con aplicaciones desarrolladas siguiendo los principios de **interfaces gráficas para usuarios**. Las interfaces gráficas han resultado muy exitosas para los usuarios finales y para los niños en particular. Cada concepto tiene una representación visual que el usuario puede manipular directamente.

Un agente es una entidad de software persistente dedicada a un propósito específico. Los agentes tienen representaciones gráficas tangibles y concretas que se crean y modifican por manipulación directa. Cada agente está asociado a un ícono, tiene un nombre, un comportamiento y una serie de atributos con valores.

Los agentes se mueven e interactúan dentro de un escenario que representa el “mundo” de la simulación. La representación gráfica del escenario permite que el usuario visualice el estado completo del mundo virtual.

Una forma de especificar el comportamiento de los agentes es a través de **reglas gráficas**. Una regla gráfica es una transformación de una porción del escenario de un estado en otro. Una regla consta de dos partes: **antes** y **después**. Cada parte incluye una escena simple que puede ocurrir durante la ejecución de la simulación. Una regla se aplica si su parte **antes** es igual a alguna porción del escenario en un momento dado. Cuando una regla puede aplicarse el escenario se transforma al reemplazarse la porción que dio lugar a la coincidencia con la parte **después** de la regla.

Las reglas pueden incluir algunas condiciones basadas en los valores de los atributos del agente y que restringen su aplicabilidad. El lenguaje para especificar estas condiciones puede variar desde un lenguaje de programación hasta un subconjunto bien definido de lenguaje natural.

La característica gráfica de las reglas, aún cuando esté enriquecida por condiciones, limita su utilidad para expresar semántica y generalización. Es decir, una regla gráfica no expresa como se realiza efectivamente la transformación y es difícil de extender a otras situaciones más generales. El problema de la generalización puede resolverse a través de la herencia basada en prototipos. La semántica puede atacarse usando programación por demostración.

Herencia basada en Prototipos

La herencia basada en prototipos es simple y concreta. Cada nuevo agente se crea "clonando" un agente ya existente usado como prototipo. Los clones heredan propiedades y comportamiento del prototipo. Cuando se especifica el comportamiento de un agente, se determina el nivel de generalización para la regla.

La herencia basada en prototipos está fuertemente ligada a la programación orientada a objetos, pero en esta forma de herencia no hay distinción entre clases e instancias: toda entidad es un objeto que puede utilizarse como prototipo para crear otros objetos. La herencia basada en prototipos puede pensarse como una forma de creación basada en ejemplos. Hay varias investigaciones que indican que la representación mental de categorías a partir de prototipos es más natural que la jerarquía de clases, pero todavía es una cuestión abierta.

LiveWorld es un ambiente diseñado para la construcción de sistemas animados que usa herencia basada en prototipos para soportar creación y modificación de agentes. LiveWorld consiste en un sistema de objetos y una interface integrada a través de la cual se manipulan las componentes del mundo animado. La interface permite visualizar el estado completo del mundo virtual. La herencia es dinámica de modo tal que los cambios en un prototipo afecta a sus clones. Las propiedades de los clones también pueden ser modificadas, pero estos cambios no afectarán al prototipo [Travers].

Programación basada en la demostración

La programación basada en demostración es una técnica mediante la cual el usuario pone al sistema en modo "registro" y luego continúa operando normalmente. El sistema registra las acciones del usuario generando un programa en forma transparente al usuario.

Las reglas gráficas permiten que los niños expresen sus intenciones respecto al comportamiento esperado de los personajes de la simulación. La programación por demostración permite tomar una regla gráfica especificando una transformación y representarla como un programa escrito en un lenguaje convencional.

KidSim es una herramienta para que los niños construyan simulaciones programando su comportamiento. KidSim combina dos ideas poderosas: reglas de escritura gráficas y programación por demostración. Los agentes se crean y modifican por manipulación directa. Cada agente tiene asociado un conjunto de reglas gráficas que se transforman en programas convencionales usando programación por demostración [Cypher-Smith].

Programación Incremental

Un ambiente de simulación debería ser al mismo tiempo simple y poderoso, esto es, brindar varios niveles de uso. Los niveles más bajos tienen que ser simples, muy fáciles de aprender y de usar. Los niveles más altos pueden requerir mayor nivel de entrenamiento y permitir la simulación de mundos más complejos. El usuario puede aprender y adoptar las características del sistema progresivamente.

El ambiente puede brindar algunos agentes y micromundos predefinidos con los cuales los niños pueden empezar a jugar rápidamente. Estos agentes pueden modificarse o combinarse incrementalmente para definir otros nuevos. Cuando el niño se siente suficientemente familiarizado con el entorno puede empezar a crear simulaciones completamente nuevas. Pedagógicamente esta es una característica muy interesante porque estimula el pensamiento y favorece el aprendizaje. Los niños comienzan modificando personajes predefinidos, pero terminan creando otros completamente nuevos.

El ambiente debería permitir además construir, modificar y deshacer el trabajo, total o parcialmente. Esta posibilidad de avanzar y retroceder tantas veces como sea necesario es muy importante porque favorece la experimentación y libera al niño de la preocupación de equivocarse.

El desarrollo de un programa en un lenguaje convencional atraviesa diferentes etapas: diseño, construcción, verificación y modificación. En la programación incremental el proceso de desarrollo es un ciclo idea-ejecución-evaluación, en el que el usuario no percibe el paso de una fase a la siguiente.

Desde el punto de vista de la interfase, en la programación convencional la transición entre la construcción y la verificación es bien tangible. La programación incremental intenta reducir o eliminar esta transición.

Un ambiente de programación para novicios debe ser dinámico, esto es, debe soportar experimentación e improvisación. Ambas características favorecen el desarrollo incremental. La programación basada en prototipos también favorece la programación incremental, porque el usuario puede crear objetos concretos sin necesidad de reconocer sus propiedades abstractas explícitamente y modificarlos sin necesidad de rediseñar la jerarquía de clases.

Conclusiones

Cuando un niño está interesado en lo que está haciendo, la dinámica del aprendizaje cambia. Al no estar forzado a trabajar sino motivado a hacerlo, puede lograr un mejor nivel de comprensión y establecer conexiones más ricas con sus conocimientos previos. El desafío consiste en encontrar el modo de ayudar a los chicos a establecer relaciones y desarrollar ideas a partir de sus propios intereses.

El constructivismo brinda una sólida base conceptual sobre la que se han diseñado e implementado una gran cantidad de ambientes de aprendizaje atractivos y motivadores. La tecnología actual brinda medios muy ricos para la aplicación de los principios del constructivismo. Los ambientes basados en esta tecnología han demostrado ser recursos muy apropiados para pasar de la teoría a la práctica. Los simuladores son herramientas particularmente atractivas porque permiten conectar los intereses de los chicos con los contenidos curriculares de la escuela.

Bibliografía

Carretero Mario, "Constructivismo y Educación". Aique, 1993.

Cypher Allen and Smith David, "KidSim: End User Programming of Simulations". CHI'95.

Minsky M. "Society of Mind". New York, Simon & Schuster. 1987

Murphy E. "Constructivism: From Philosophy to Practice" 1997
<http://www.stemnet.nf.ca/elmurphy/emurphy/cle.html>

Papert S. "Mindstorms: Children, Computers and Powerful Ideas". New York, Basic Books. 1980.

Resnick M. y Rusk N. "The Computer Clubhouse: Preparing for the Life in a Digital World" . 1997 http://el.www.media.mit.edu/grou.rs/mres/Comp_club/Clubhouse.html

Sastre G. y Moreno M. "Descubrimiento y Construcción de conocimientos". España, Gedisa. 1985.

Shoham Y. "Agent Oriented Programming". Artificial Intelligence, 1993.

Travers Michael, "Programming with Agents: New metaphors for thinking about computation". Ph.D. Thesis, Massachussets Institute of Technology, Bostón, USA. 1996.