

El sistema SVED en la enseñanza semi-presencial

Norma Moroni – Perla Señas
Laboratorio De Investigación y Desarrollo en Informática y Educación (LIDInE)
Instituto de Investigación en Ciencias y Tecnología Informática (IICTI)
Departamento de Ciencias de la Computación
Universidad Nacional del Sur - Bahía Blanca
Argentina
nem@cs.uns.edu.ar - ips@cs.uns.edu.ar

Resumen

En este artículo se presenta un Sistema de Visualización de Algoritmos y Estructuras de Datos, SVED, en el marco de la enseñanza de la programación con metodología semi-presencial. Consiste en la integración de nuevas tecnologías con otras no tan novedosas, con la finalidad de favorecer el aprendizaje en la modalidad semi-presencial. Se trata de contribuir favorablemente ante la falta de relación cara a cara en todo tiempo, entre el alumno y el docente y entre el alumno y sus pares. La modalidad de enseñanza dispuesta se basa en tecnología computacional, tal como tecnologías de la comunicación, coursewares, sitios Web, grupos de discusión. Si bien el aprendizaje a través de la Web no sustituye enteramente al presencial, puede servir como una alternativa razonable cuando aquel no está disponible. Por otra parte, es la vía de comunicación más usada y aceptada por muchos estudiantes y es la que más se adapta tanto como canal de consulta o como herramienta de estudio [10]. Cuando el uso de la Web es voluntario, los estudiantes no aprovechan totalmente las ventajas de ésta. Por este motivo, cuando los alumnos se ven forzados a usarla, dentro del marco de trabajo establecido, se debe ofrecer y debe estar a disposición de sus necesidades toda la información y la comunicación posible.

La inclusión del Sistema de Visualización de Algoritmos y Estructuras de Datos, SVED, contribuye a incrementar el potencial pedagógico de la enseñanza de la programación, dentro de este marco propuesto.

Palabras claves: visualización de software, educación semipresencial, tecnología computacional.

Introducción

Está claro que la modalidad de enseñanza a distancia o semi-presencial se impone en muchas situaciones de aprendizaje. Con los recursos computacionales actuales, entre los que podemos mencionar los sitios Web, el correo electrónico, los coursewares, los grupos de discusión, las conferencias electrónicas, la comunicación entre el alumno y su maestro o con los otros alumnos, se puede realizar de manera razonablemente satisfactoria. No obstante, la comunicación cara a cara no se puede reemplazar totalmente con los recursos mencionados. De allí que la modalidad de enseñanza semi-presencial cubre en parte este aspecto aunque requiere un alto nivel de habilidad y voluntad de dedicación al estudio.

Esta metodología de enseñanza impone un sistema logístico complejo que está basado en la idea de llevar la universidad al hogar para lograr que los alumnos, quienes no pueden asistir a clase, tengan también la posibilidad de realizar un estudio estructurado que los conduzca a la meta establecida [10].

El marco de trabajo propuesto integra nuevas tecnologías con otras que no lo son tanto, con la finalidad de favorecer el aprendizaje en la modalidad semi-presencial. El Sistema de Visualización de Algoritmos y de Estructuras de Datos realiza un aporte importante permitiendo que los alumnos dispongan de él en el momento en que lo necesitan y puedan descubrir a través de él la conducta de los algoritmos que les fueron explicados en clase o que leen en los textos que tienen a disposición en la Web. Cualquiera de estas dos modalidades de recibir la información (y por supuesto la segunda es totalmente estática) carecen de la dinámica que ofrece la animación de algoritmos para mostrar el comportamiento de los programas.

Organización del marco de trabajo

Cada curso debe contar con una estructura de cátedra formada por un profesor y un grupo de tutores. El profesor es el responsable de la organización del curso, del material bibliográfico que se debe enviar a los alumnos, de las prácticas que ellos deben realizar y de los exámenes, como así también de los horarios de encuentro y de comunicación a través de la red. El material bibliográfico debe estar a disposición de los alumnos antes de comenzar el curso. Los tutores constituyen el personal que se comunica con los alumnos electrónicamente, ya sea vía e-mail, a través de conferencias electrónicas, por medio de foros de discusión, usando la capacidad de chat, etc, [5]. Esta comunicación debe realizarse en forma organizada de manera tal que no sea caótica. En principio, se pueden organizar grupos de alumnos que se comunican con un determinado tutor, y éste establece los horarios y la frecuencia de comunicación. Por supuesto, los alumnos entre sí pueden establecer sus propios encuentros. Es importante que estas actividades de estudio se realicen independientemente del tiempo y del espacio. De esta manera se pretende crear un campo virtual de estudio que supla en la mayor cantidad de aspectos posibles al campo real [5].

Cada curso tiene su propio sitio Web que provee, además de las posibilidades de comunicación indicadas, el material de aprendizaje interactivo y el software que los alumnos pueden usar. SVED es un sistema de software orientado a la exhibición de la conducta de un programa en lenguaje procedural.

Sistema SVED

El sistema SVED [14] tiene como finalidad ayudar a los estudiantes de Programación a interpretar la conducta de los algoritmos subyacentes en los programas que hacen uso de las mismas. El sistema ha sido pensado para trabajar sobre estructuras de Hash abierto, árboles binarios de búsqueda, AVL, árboles 2-3 y árboles parcialmente ordenados.

Las siguientes características identifican a SVED:

Establece diferenciación de código cromático entre los elementos pasivos y los dinámicos de la estructura.

La representación de los nodos de las estructuras se realiza por medio de círculos cuyos tamaños están en relación de proporcionalidad directa con el valor del dato que representa.

Permite el ingreso de datos específicos o de la cardinalidad del conjunto a estudiar.

La comparación entre nodos se realiza por superposición de sus respectivas representaciones gráficas.

El recorrido que se realiza en la búsqueda de un elemento determinado se marca por medio de flechas.

El éxito en la búsqueda se señala de manera especial.

La exhibición del comportamiento del algoritmo se puede realizar en forma estática o dinámica, y ésta última con animación continua o discreta.

La representación de un subárbol se indica por medio de un triángulo semi-transparente que lo abarca.

La representación del árbol original y la del árbol resultado de la búsqueda, son similares.

La animación de código, destaca la línea del programa que está siendo ejecutada en cada momento.

Ejemplo:

La secuencia de cuadros en la fig 1 muestra las instantáneas más relevantes de la visualización del comportamiento del algoritmo para eliminar un elemento de un árbol binario de búsqueda cuando el nodo que contiene dicho elemento tiene dos hijos no nulos.

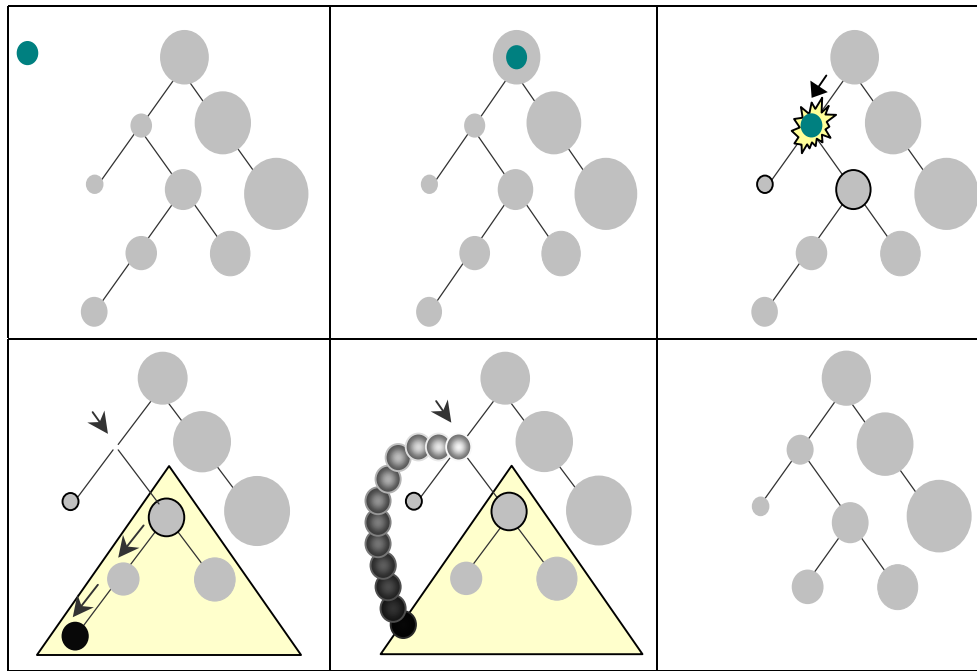


figura 1

Aporte de SVED a la enseñanza semi-presencial

El sistema propone el algoritmo del método que resuelve el problema que el alumno debe comprender. El alumno sólo incorpora los datos de entrada. Esta característica del sistema SVED permite conducir al alumno en la comprensión de la conducta de los algoritmos sin que se pierda en la elaboración del programa respectivo, y sin que fallas del mismo, por mínimas que sean, lo perturben o lo distraigan del objetivo propuesto. Por otra parte, ayuda a comprender los conceptos que el docente ha impartido durante la clase presencial y permite que cada alumno vaya a su propio ritmo, según su capacidad intelectual y sus limitaciones en el tiempo de estudio.

Además el sistema presenta los beneficios educativos [9] propios de las animaciones de software como el desarrollo de destrezas realizando prácticas adicionales ya que el alumno tiene a disposición un sistema on-line, la exploración de las características del programa en estudio por medio de la interacción, el progreso del aprendizaje individual incrementando la capacidad de predicción de nuevas situaciones [4].

Conclusión

SVED se adapta perfectamente a la modalidad de enseñanza semi-presencial, ya que este sistema refuerza los conocimientos presentados por el docente en clase y minimiza las posibilidades de errores que puedan surgir de la escritura de un programa. Es importante que el estudiante interactúe activamente con el sistema [15], el mismo le permite la entrada de distintos conjuntos de datos, la elección del tipo de visualización que prefiere, la interrupción de la visualización, la continuación o el reinicio de la misma. Por otra parte prepara al alumno para que comience a escribir sus propios programas, ya que el sistema realiza una visualización del código del programa y destaca la línea en ejecución, mostrando al mismo tiempo los efectos que esa ejecución produce en el comportamiento del algoritmo subyacente [12]. Esta actividad conduce favorablemente al alumno a escribir sus propios programas ya que le permite adquirir habilidad en la predicción de acciones futuras.

Bibliografía

- [1] Bailey, R. *Human Performance Engineering*. Prentice Hall. 1996.
- [2] Basik, J- Tamassia,R- Reiss, S- van Dam, A. *SV in teaching at Brown University*. 1990.
- [3] Brown- Sedgewick. *A system for Algorithm Animation*. ACM Computer Graphics.1985.
- [4] Byrne, M.- Catrambone, R.- Stasko, J. *Evaluating Animations as Student Aids in Learning Computer Algorithms*. Computer & Education. December 1999.
- [5] Gal-Ezer Judith- Lupo D. *Integrating internet tools into traditional CS distance education: students' attitudes*. Computers & Education. May 2002.
- [6] Gomez Henriquez. *Sistematización y técnicas de Visualización de Programas Concurrentes*. 1993.
- [7] Horton W. *Designing Web-Based training: How to teach Anyone Anything Anywhere Anytime*. John Willey. 2000.
- [8] Jeffery, C. *Program Monitoring and Visualization*. Springer-Verlag. 1999.
- [9] Lawrence, A- Brade, A.- Stasko, J. *Empirically Evaluating the Use of Animations to Teach Algorithms*. Georgia Institute of Technology. 1999.
- [10] Lupo, D- Erlich, Z. *Computer literacy and applications via distance e-learning*. Computer & Education, May 2001.
- [11] Merrill, P., at all . *Computers in education*. Allyn & Bacon. 1996.
- [12] Moroni, N. *Entornos para el Aprendizaje de la Programación*. WICC. 2001.
- [13] Moroni N - Señas P. *La Visualización de Algoritmos como Recurso para la Enseñanza de la Programación*. WICC. 2002.
- [14] Moroni, N- Señas P. *SVED: sistema de visualización de algoritmos*. CACIC 2002. Octubre 2002.
- [15] Naps, T. *Algorithm Visualization in Computer Science Laboratories*. SIGCSE. 1990
- [16] Price, B.- Beacker, R. – Small, I. *An Introduction to Software Visualisation*. MIT Press. 1998.
- [17] Stasko, J.- Domingue, J.- Brown, M.- Price, B. *Software Visualization: Programming as a Multimedia Experience*. MIT Press, 1998.
- [18] Stasko, J- Lawrence, A. *Empirically Assessing Algorithm Animations as Learning Aids*.1997.
- [19] Stasko, J. *Tango: A framework and system for algorithm animation*. IEEE Computer. 1990.