

ENTORNOS PARA EL APRENDIZAJE DE LA PROGRAMACIÓN

Norma Moroni

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

nem@cs.uns.edu.ar

1. Introducción.

La Informática cuenta con un aspecto muy importante que es la programación de la computadora. Programar significa analizar, resolver, plantear alternativas a un determinado problema y definir cuáles son las órdenes o instrucciones que se debe dar a una computadora para que halle la solución al mismo. Un programa es la resolución formal de un problema y la obtención de su solución, siendo esta última operatoria, ejecutable, susceptible de ser puesta a prueba, verificable. El alumno programador es un usuario activo y se convierte en el ingeniero de mantenimiento de su propio producto.

2. La Computadora y la Enseñanza de la Programación.

La computadora es una herramienta útil para el aprendizaje de programación de la misma manera que lo es para la enseñanza de distintas asignaturas. Para ello se debe contar con entornos de aprendizaje que conduzcan e induzcan tal proceso educativo. La calidad de dicho entorno está ligada íntimamente con su capacidad de logro de un proceso efectivo de enseñanza y de aprendizaje. Un entorno debe contemplar el nivel cognitivo, la capacidad de percepción, de respuesta, de abstracción de la audiencia a la que va dirigido, como así también, la capacidad de interacción hombre-máquina, la velocidad de presentación, la estandarización de colores y sonidos, etc. Para ello se debe disponer de sistemas específicos, amigables con el usuario, que no actúen como elementos de dispersión en la elaboración y en la interpretación de la conducta del software propiamente dicho.

La programación suscita interés psicopedagógico debido a sus efectos en las capacidades cognitivas, desarrollando una actividad intelectual que permite establecer planificaciones y estrategias, construir algoritmos, estructurar instrucciones, analizar y comprender los programas propios o escritos por otros, aprender la sintaxis de los distintos lenguajes de programación, comparar estrategias entre los programadores expertos y principiantes.

Si bien los entornos de los lenguajes de programación ofrecen capacidades cada vez más útiles al programador, como posibilidad de indentación, depuradores, estos no son aún lo suficientemente amigables con el usuario como es deseable. Se ha pensado en el desarrollo de entornos específicos que ayuden y conduzcan efectivamente a los estudiantes en el proceso de aprendizaje de la programación de manera más gráfica, diversa y comprensible.

3. Visualización de software

La visualización de software es un campo particular y el más contemporáneo de visualización por computadora. Comprende la visualización de algoritmos y de programas. La primera consiste en la visualización de abstracciones de alto nivel que describen el software, mientras que la segunda se refiere al código real de programa y a las estructuras de datos. Ambas pueden darse en forma estática o dinámica. La visualización estática de algoritmo está representada generalmente por medio de organigramas mientras que la dinámica es lo que se llama animación de algoritmos y exhibe el comportamiento de los mismos durante la ejecución. La visualización estática de código incluye algún tipo de mejoramiento de la impresión como indentado o estructura del programa mientras que una representación dinámica del mismo puede destacar las líneas de código a medida que ellas están siendo ejecutadas. La visualización estática de datos representa la estructura que integran los datos mientras que una representación dinámica de la misma muestra cómo los valores de los datos cambian mientras el programa corre [Sta98]. La Visualización de Software hace uso de diseño gráfico, animación, sonido, vídeo con moderna comunicación hombre máquina y tecnología

de gráfica de computadora para facilitar la comprensión humana y el uso efectivo de software de computadora [Bro95].

Frente a la posibilidad de enseñar a programar que nos ofrece la computadora se describieron e implementaron distintos entornos que se corresponden con este objetivo.

4. Trabajos realizados

4.1. Entorno Cubik

En la búsqueda de recursos metodológicos para la enseñanza de la programación se destaca la importancia de disponer de un entorno interactivo adecuado que permita al alumno el desarrollo de los algoritmos. El entorno Cubik implementado se basa en el paradigma imperativo y permite el trabajo en programación estructurada y modular, de acuerdo con las características de los cursos en los cuales se aplica [Gar96]. Cuenta con un editor interactivo de algoritmos y con un traductor automático de algoritmos a programas. Este entorno controla con naturalidad la edición evitando errores sintácticos y semánticos y posibilita la invocación de otros algoritmos como primitivas promoviendo el concepto de modularidad. Además, el traductor automático a programas permite la aplicación de un método global para el aprendizaje del lenguaje de programación usado.

El uso del editor interactivo de algoritmos consigue que la construcción de algoritmos resulte una tarea atractiva y motivadora, que el autor no necesite memorizar esquemas ni reglas fijas y que su uso no se abandone aunque haya aprendido el lenguaje de programación. El uso de esta metodología frente a la tradicional presenta beneficios importantes sobre los tiempos de aprendizaje y sobre la complejidad de los problemas que se pueden manejar inicialmente.

No obstante, las ventajas observadas con el uso del editor de algoritmos, se pudo observar que los alumnos persisten en sus inconvenientes para la depuración y puesta a punto de los algoritmos y programas. Para mejorar esta situación se propuso la ampliación del entorno original de Cubik de manera tal que permita la visualización de los algoritmos y programas propuestos.

4.2. Entorno de Verificación

Con la finalidad de favorecer la comprensión de la conducta de un programa se diseñó el Entorno de Verificación, para anexar a Cubik, que posibilita la realización de Trazas de Ejecución, de Esquemas de Ejecución de Algoritmos y de Programas en Lenguaje Pascal [Señ98]. El Entorno de Verificación permite visualizar la conducta del algoritmo y del programa durante la ejecución, mediante la exhibición de los sucesivos estados de la computación. Se permite cierta forma de interacción como lo es la detención de la representación, con la posibilidad de continuar o de abortar la misma en cualquier momento.

5. Investigación Actual

Se está trabajando actualmente en el desarrollo de un sistema de visualización de algoritmos y programas que permita además, representar de manera más abstracta los resultados obtenidos en la ejecución de los mismos. Esto con la finalidad de transformar la información en una más significativa, desde la cual se pueda ganar en comprensión. Existe una variedad enorme de aportes sensitivos que pueden favorecer la formación de un cuadro mental, para ello todo lo informado a través de la visualización debe tener en cuenta aspectos de la percepción y el conocimiento humano, con el fin de satisfacer las necesidades de la persona que observa la visualización [Sta98].

El sistema de visualización de software a desarrollar debe facilitar la exhibición de:

- Texto del algoritmo o programa.
- Estructura estática del algoritmo o programa.
- Estructura estática de los datos.
- Control de flujo del algoritmo o programa (en ejecución).
- Estructura dinámica de los datos y modificación de valores (en ejecución)
- Explicación textual y/u oral de la propuesta.
- Errores en forma visual o auditiva.

Con la posibilidad de interacción antes y durante la ejecución del algoritmo o programa facilitando:

- La modificación de los valores de los datos.

- La modificación de las acciones.
- La elección de los datos a representar.
- La elección de la forma de representación.
- La interrupción de la exhibición temporal o definitivamente.
- La recuperación de una exhibición anteriormente abandonada.
- La eliminación, minimización y recuperación de ventanas.

6. Aspectos Pedagógicos de la Visualización de Algoritmos

Se considera que la Visualización de Software y especialmente la Visualización de Algoritmos, contribuye favorablemente en la comprensión de los mismos y que ese aporte es realmente efectivo. Se pretende que conduzca al alumno tanto en la tarea de comprensión del software como en la de diseño, depuración, prueba y mantenimiento del mismo. Por otra parte, estos entornos ayudan a los estudiantes a adquirir el conocimiento de acuerdo a sus posibilidades de aprender según lo que sus propias capacidades le permiten. La Visualización de Algoritmos presenta un medio audio-visual interactivo multimedial cuyas bondades en el campo de la psicopedagogía han sido comprobadas [Keh99]. Los sistemas no sólo deben estar diseñados por especialistas en computación sino por psicopedagogos que permitan establecer un equilibrio entre el exhibicionismo espectacular que presentan algunas visualizaciones y lo que el ser humano puede percibir efectivamente de ellas. La animación de algoritmos y la visualización de programas colaboran tanto con los estudiantes en la comprensión de los conceptos de software como con los docentes en su tarea de enseñar dichos conceptos.

7. Bibliografía

- [Ale91] Alessi, S and Trollip, S. *Computer-Based Instruction*. Prentice Hall. 1991.
- [Bro95] Brown, M.- Sedgewick, R. *A system for Algorithms Animation*. IEEE Software. 1995.
- [Cha91] Chavey, D.- College, B. *A Structured Laboratory Component for the Introductory Programming Course*. SIGCSE BULLETIN ACM. 1991.
- [Dib99] Di Batista, P.- Eades, G.- Tamassia, R.- Tollis, I. *Graph Drawing algorithms for the visualization of graphs*. Prentice Hall. 1999.
- [Gar96] García, J.- Señas, P.- Moroni, N. 'Cubik: Una Herramienta de Apoyo a la Enseñanza de la Programación'. IV Ateneo de Profesores Universitarios de Computación. San Luis. 1996.
- [Gei94] Geitz, R. *Concepts in the classroom, programming in the lab*. ACM SIGCSE BULLETIN. 1994.
- [Gri95] Grinstein, G.- Levkowits, H. *Perceptual Issues in Visualization*. Springer-Verlag. 1995.
- [Mer96] Merrill, P. - Hammons, K. – Vincent, B. *Computers in education*. Allyn and Bacon. 1996.
- [Nap98] Naps, T.- Bressier, E. *A multi-windowed environment for simultaneous visualization of relater algorithms on the World Wide Web*. SIGCSE. 1998.
- [Rue94] Rueda, S.- Castro, S.- Zanconi, M. *Resolución de Problemas y Algoritmos: notas de curso*. 1994.
- [San98] Sangwan, R.- Korsh, J.- La Foliette, P. *A System for Program Visualization en the Classroom*. SIGCSE. 1998.
- [Señ98] Señas, P.- Moroni, N. *Herramientas no convencionales para el aprendizaje de la programación*. CACIC'98. 1998.
- [Sta98] Stasko, J.- Domingue, J.- Brown, M.- Price, B. *Software Visualization: Programming as a Multimedia Experience*. MIT. 1998.
- [Keh99] Kehoe, C- Stasko, J.- Taylor, A. *Rethinking the Evaluation of Algorithms Animations as Learning Aids: As Observational Study*. Atlanta. 1999.