

Visualización de Algoritmos y Programas para el Aprendizaje de Estructuras Arbóreas*

Fernando González - Norma Moroni - Perla Señas
Laboratorio De Investigación y Desarrollo en Informática y Educación (LIDInE)
Instituto de Investigación en Ciencias y Tecnología Informática (IICyTI)
Departamento de Ciencias de la Computación. Universidad Nacional del Sur
Bahía Blanca. Argentina
fggonzal@yahoo.com.ar - nem@cs.uns.edu.ar - ips@cs.uns.edu.ar
T. (0291) 4595101 int 2614 - fax (0291)4595136

Resumen

Este artículo presenta el Diseño de un Software de Visualización Árboles Binarios de Búsqueda y de árboles AVL. Está específicamente basado en la animación de algoritmos y visualización de programas. El software exhibe la conducta del programa representando la estructura de los árboles tanto en forma abstracta [1] como por medio de celdas y enlaces, y mostrando el paralelismo entre ambas representaciones. Además, incorpora el algoritmo en pseudo-código señalando las acciones que se llevan a cabo en cada momento que se realiza la manipulación de la estructura. Para su diseño se han tenido en cuenta distintas técnicas de animación, de color y de sonido [2].

El objetivo del software consiste en brindar una herramienta que sirva como material de aprendizaje básico del tema, de consulta y de refuerzo. El software tiene como finalidad la combinación de nuevas tecnologías informáticas, para favorecer el aprendizaje individual y personalizado y para ser parte de una metodología presencial, semi-presencial y a distancia. El software diseñado será integrado a la Web [3], junto con otros que conformarán el curso de Estructuras de Datos y Algoritmos, correspondiente al segundo año de las carreras de Computación que dicta el Dpto. de Ciencias e Ingeniería de la Computación de la Universidad Nacional del Sur.

Palabras claves: visualización de software, Arbol Binario de Búsqueda, Árbol AVL

1. Introducción

La *Visualización de Software* es el uso de representaciones de software interactivas y multisensoriales, para facilitar la comprensión humana de la conducta y la interpretación efectiva del mismo [4]. La enseñanza de la programación y la manipulación de estructuras de datos encuentran, con la visualización de software, un mecanismo de enseñanza y aprendizaje que ayuda fundamentalmente a la lectura comprensiva de algoritmos [5]. Considerando la inclusión de visualizaciones explicativas de procedimientos de actualización de estructuras de datos, se genera una poderosa herramienta que extiende el aprendizaje de la programación, con medios de resolución de problemas y pautas de estudio, para el dominio y la manipulación de nuevas estructuras de datos.

Esta nueva herramienta realiza un aporte importante permitiendo que los alumnos dispongan de él en el momento en que lo necesiten y puedan descubrir la conducta de los algoritmos que les fueron explicados en clase o que encuentra en los textos. El Software de Visualización de Algoritmos y Programas propuesto define el concepto de Árbol Binario de Búsqueda y de AVL, y muestra el manejo de las estructuras describiendo las operaciones de generación, búsqueda y actualización.

2. Características de la Herramienta

La herramienta cuenta con las siguientes características:

* El presente trabajo fue financiado por la Universidad Nacional del Sur, Bahía Blanca, Argentina.

- Presentación de un entorno amigable con el usuario que atrapa fácilmente la atención del observador.
- Interacción con el usuario en forma limitada, evitando la inclusión de elementos que puedan alterar la correctitud de la exhibición y que conduzcan a errores de conceptos. La interacción permitida brinda una relación temporal individual, de manera que cada observador pueda establecer su propio ritmo de aprendizaje. La herramienta permite la repetición de fragmentos de la animación y el reinicio de pausas específicas, utilizadas para congelar contenido explicativo.
- Claridad en los conceptos introducidos respetando las características tanto explicativas como de implementación, impartidas en las clases teóricas y en los libros de texto. Ver figura 1.
- Incremento gradual de la complejidad de los contenidos. Ver figura 2.
- Cubrimiento completo de casos particulares que permiten disminuir la desventaja del planteo rígido de los contenidos que imposibilitan la observación de reacciones ante casos puntuales. Esta propuesta responde al propósito de evitar errores por parte del observador al manejar temas desconocidos. Ver figura 3.
- Uso de técnicas de Visualización de Algoritmos y Programas, técnicas de animación, de color, y de la percepción humana tanto en el campo visual como auditivo.
- Acceso a la herramienta a través de la Web dado que el entorno de desarrollo tiene soporte Web de utilización masiva, ampliamente difundido.
- Alto grado de detalle para indicar claramente la evolución de las animaciones.

3. Conceptos y técnicas consideradas para el desarrollo

Con el fin de satisfacer las necesidades de la persona que interactúa con la representación resultante de la *Visualización*, todo lo informado a través de la misma debe tener en cuenta aspectos de la percepción y de la adquisición del conocimiento humano [6] [7].

- **Percepción**

Las complejas y poderosas capacidades del sistema visual humano pueden ser aprovechadas para facilitar la comprensión de sistemas informáticos de mediana y alta complejidad. Como la información contenida en las visualizaciones debe pasar a través del sistema de percepción, la atención cuidadosa a las características del mismo mejora asombrosamente la efectividad de las visualizaciones. Existen diversas técnicas que favorecen la percepción de acuerdo a la información a transmitir. A continuación se nombran algunas de las más utilizadas:

Uso de Escenarios Familiares. Las visualizaciones son más efectivas cuando su entorno, disposición e iluminación están asociados con la experiencia perceptual común. En el sistema en desarrollo, las burbujas diseñadas, presentan iluminación superior, levemente desplazada a la izquierda, contemplando el hecho que los humanos están acostumbrados a fuentes de iluminación que proceden desde arriba de los objetos en una escena. Además por medio de sombras, se logra una sensación de tridimensionalidad en las celdas que contienen los elementos del árbol.

Énfasis sobre el interés. Se ha tenido en cuenta la importancia de considerar que las características más llamativas de la imagen sean también las más importantes de la escena. Para ello se han dispuesto colores brillantes, movimientos, cambios, límites sobresalientes o altamente saturados. Los arcos en la representación de celdas y enlaces se encuentran representados por medio de flechas que modifican sus formas, sus orientaciones y cuyos colores cambian a medida que se va desarrollando la animación.

Minimización de las ilusiones. El tamaño, la profundidad y el color percibido de un objeto pueden ser influenciados tanto por su color como por el color de los objetos que lo rodean. Se presentan en forma más destacada aquellos objetos en los cuales se necesita captar la atención.

Control de nivel de detalle. Se contempla el hecho que a medida que la complejidad de las animaciones crece, se disminuye el nivel de detalle con el fin de priorizar los conceptos importantes a transmitir. Con ese propósito, en las animaciones donde se representa en paralelo el código y el árbol con celdas y enlaces se elimina la representación de los subárboles, incluida en animaciones de etapas anteriores. Ver figura 4.

Metáforas. Una metáfora visual familiar o fácilmente inferida para la conducta presentada puede disminuir la carga cognitiva impuesta sobre el usuario e incrementar la velocidad de comprensión. Aunque algunas metáforas son dibujadas naturalmente desde un dominio de aplicación específico, otras lo son desde la naturaleza o desde símbolos no técnicos encontrados en la vida diaria.

- **Animación de algoritmos**

Instantáneas y signos de estado. La animación muestra cambios en el estado de la estructura de datos cambiando su representación gráfica sobre la pantalla. Además, los signos de estado vinculan distintas películas representando objetos específicos de la misma manera en distintas animaciones [8]. Se observan en las vistas donde aparece la pila de activación de las llamadas recursivas sobre el mismo algoritmo.

Animaciones incrementales, continuas o discretas. La representación gráfica de un cambio en una estructura de datos puede darse en forma incremental, continua o discreta. Tanto los cambios incrementales como los continuos presentan la información progresivamente donde las modificaciones deben ser suaves a medida que los eventos ocurren. La diferencia que se destaca entre ambas modalidades es que la incremental permite interactuar con la visualización en determinadas situaciones y comprender la secuencia de cambios realizados con la posibilidad de rever el último incremento, reiniciar la animación o continuar con ella, teniendo conciencia de ello. Los cambios incrementales y continuos son más útiles para conjuntos de datos pequeños donde la complejidad de las modificaciones sobre la estructura, requieren la comprensión y el cambio producido. Las animaciones suaves y continuas ayudan a describir las operaciones individuales de un proceso y permiten al usuario percibir y comprender fácilmente los cambios. La modalidad discreta ofrece instantáneas de posiciones previas a la operación y posteriores a la misma sin reflejar el camino recorrido. Se adecuan a sistemas con gran cantidad de elementos. El software en desarrollo presenta animación incremental. Un arco entre nodos es representado como una flecha que es animada lentamente sin parar desde su referente anterior y eventualmente apuntando a su futuro referente. Además, se introduce una serie de cuadros intermedios para actuar como una forma de interpolación entre los estados iniciales y finales.

Complejidad de los algoritmos. En los algoritmos que contienen un alto grado de detalle, se reemplazan los grupos de acciones por una acción general que los encapsulan. Cuando se reproduce la ejecución de dicha acción se despliega el grupo de acciones encapsulado mostrando todos los detalles ocultos. Esta técnica también fue usada para evitar el scroll en la vista del código, que a veces no resulta tan adecuado.

- **Color**

El color tiene el potencial de comunicar una enorme cantidad de información eficientemente. Se usa el color para el estado del algoritmo, mejorando y complementando las técnicas gráficas descriptas. Permite distinguir la nueva información incorporada a la vista y resalta los objetos importantes en la evolución de la animación. En el uso de vistas múltiples, cuando un grupo de animaciones genera un entorno y una gráfica integrada, se ilustran ciertas características con los mismos colores en todas las vistas. Presenta con diferentes colores a los elementos que están activos de los que se encuentran pasivos en un determinado momento.

- **Sonido**

La inclusión de sonido en las visualizaciones de algoritmos está ligado al alto grado de capacidad del ser humano para detectar y recordar patrones de sonidos. En las visualizaciones de algoritmos el sonido se incluye para refuerzo visual, para indicar patrones de comunicación, para detectar reemplazos visuales y para señalar condiciones excepcionales, aumentando el nivel de expresividad. Por otra parte el sonido, ayuda en la interpretación de la animación que se lleva a cabo y permite una conducta más pasiva por parte del usuario ya que se pueden detectar eventos excepcionales, sin necesidad de estar abocado a la observación de los mismos. El software en desarrollo establece un código auditivo donde no hay sobrecarga en los sonidos. El sonido para indicar que un objeto es hermano izquierdo de otro, es distinto al que indica que es hermano derecho. Como el sonido es intrínsecamente dependiente del tiempo, no es sorprendente que sea muy efectivo para mostrar fenómenos dinámicos tal como ejecuciones de algoritmos. Pero, el sonido es un medio más difícil para dominar, con muchos parámetros, frecuencia, volumen, velocidad de ataque y decadencia, duración, timbre, reverberación, brillantez, estéreo [9].

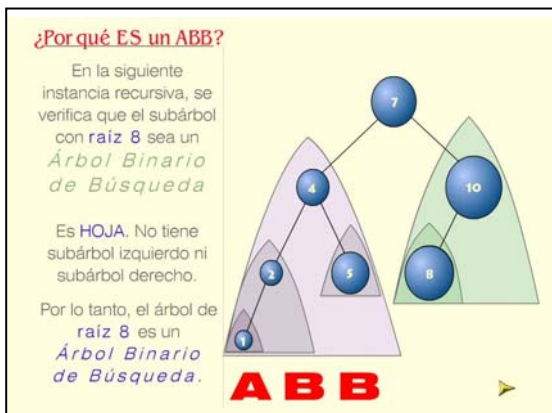


Figura 1

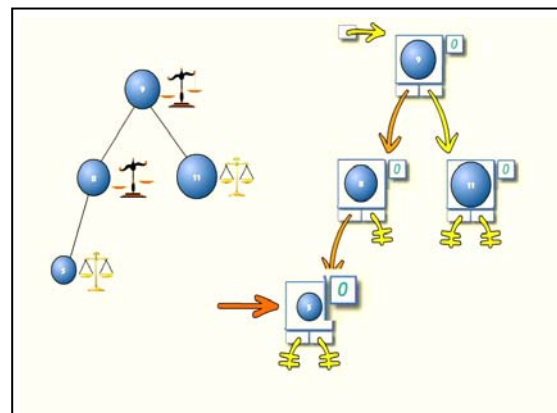


Figura 2

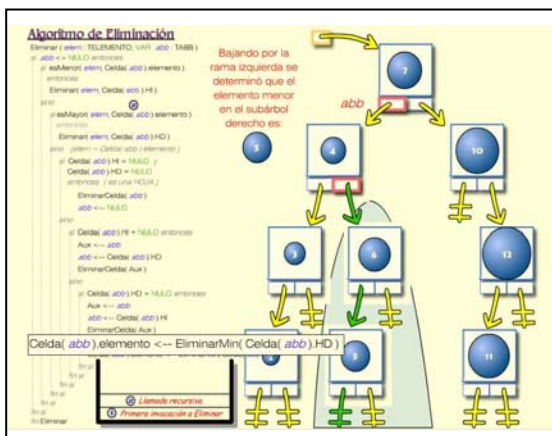


Figura 3

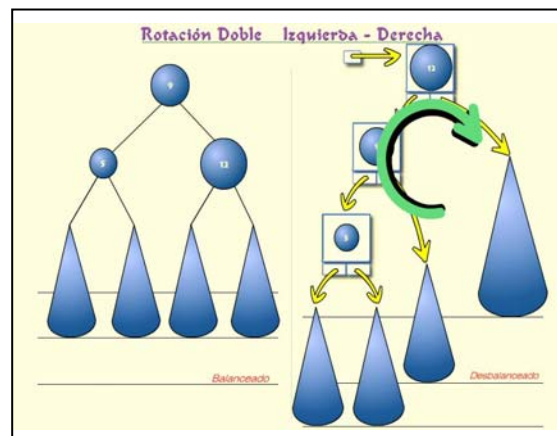


Figura 4

4. Conclusión

La visualización de software favorece sensitivamente la formación de un cuadro mental del proceso que se está llevando a cabo. Con tal propósito, la visualización debe identificar estructuras, características, anomalías y relaciones entre los datos objeto de la misma, como así también, presentar un “pantallazo cualitativo” cuando se trata de conjuntos grandes y complejos de información, y detectar las zonas de interés que merecen un análisis cualitativo focalizado.

La *Visualización de Software* que permite interacción con representaciones gráficas de algoritmos o estructuras de datos que cambian dinámicamente constituye, sin duda, una posibilidad de exploración de gran valor en la enseñanza de la programación.

La característica principal del software presentado es la animación incremental que prima en todas las exhibiciones, como así también el color, y el sonido que acompañan tanto a la animación del algoritmo como a la visualización dinámica del programa. Existe un código cromático y sonoro que se mantiene en todas las exhibiciones.

Es una herramienta con posibilidad de ayudar al alumno de un curso de Estructuras de Datos y Algoritmos a aprender los conceptos básicos del tema, a consultar frente a dudas surgidas durante el estudio y a reforzar los conocimientos adquiridos. Este beneficio lo ofrece tanto en la etapa de comprensión de la conducta del algoritmo como en las etapas posteriores de representación del árbol con celdas y enlaces y en la descripción del algoritmo. Además, la animación incremental permite predecir acciones futuras cuando el usuario ha comprendido el proceso subyacente [10].

El Software de Visualización Árboles Binarios de Búsqueda y de árboles AVL está basado fundamentalmente en la *Animación de Algoritmos* y en la *Visualización de Programas* presentando, para estos últimos, características tanto de una exhibición estática como dinámica. El software está desarrollado en Macromedia Flash e incorporado a la Web por medio de Dream Weaver, ambos software con gran compatibilidad con Microsoft.

5. Bibliografía

- [1] Moroni, N- Señas P. *SVED: sistema de visualización de algoritmos*. CACIC 2002. Octubre 2002.
- [2] Naps, T. *Algorithm Visualization in Computer Science Laboratories*. SIGCSE. 1990.
- [3] Lupo, D- Erlich, Z. *Computer literacy and applications via distance e-learning*. Computer & Education, May 2001.
- [4] Stasko, J.- Domingue, J.- Brown, M.- Price, B. *Software Visualization: Programming as a Multimedia Experience*. MIT Press, 1998.
- [5] Lawrence, A- Brade, A.- Stasko, J. *Empirically Evaluating the Use of Animations to Teach Algorithms*. Georgia Institute of Technology. 1999.
- [6] Gomez Henriquez. *Sistematización y técnicas de Visualización de Programas*. 1993.
- [7] Landreth C. y Rheingans P. *Preceptual Principles for Effective Visualizations*. USEPA. Visualization Center. 1995
- [8] Stasko J., Domingue J., Brown M.H. y Price B.A. *Software Visualization. Programming as a Virtual Experience*. The MIT Press. Cambridge. Massachusetts. 1998.
- [9] Gal-Ezer Judith- Lupo D. *Integrating internet tools into traditional CS distance education: students' attitudes*. Computers & Education. May 2002.
- [10] Merrill, P. at all. *Computers in education*. Allyn & Bacon. 1996.