

VIEG, una herramienta para la Visualización de Información Estructurada mediante Grafos

Eduardo Fernández[†]

Universidad de la República

Julio Herrera y Reissig 565, piso 5

(11300) Montevideo, Uruguay

598 2 7114244 - 123

eduardof@fing.edu.uy

Marcelo Giorgi[†]

Universidad de la República

Julio Herrera y Reissig 565, piso 5

(11300) Montevideo, Uruguay

598 2 7114244 - 115

mgorgi@fing.edu.uy

Tomás Laurenzo[†]

Universidad de la República

Julio Herrera y Reissig 565, piso 5

(11300) Montevideo, Uruguay

598 2 7114244 - 110

laurenzo@fing.edu.uy

RESUMEN

Este artículo presenta la herramienta VIEG (Visualización de Información Estructurada mediante Grafos), la cual pretende facilitar la visualización, navegación e interpretación de información estructurada mediante su modelado como grafos. Para ello se parte de conceptos tomados de la literatura (tanto en temas gráficos como perceptuales) y de la reutilización de código proporcionado por diversas aplicaciones. VIEG es un producto del proyecto “Desarrollo de una herramienta para la visualización y exploración gráfica de información jerárquica” desarrollado por el Instituto de Computación de la Facultad de Ingeniería de la Universidad de la República, Uruguay.

Palabras clave

VIEG, Visualización de Información, VGJ, Disposición, Grafos, Foco y Contexto, Interacción Persona Ordenador, Software Libre.

1. INTRODUCCIÓN

1.1 La necesidad de visualizar la información

Diversos estudios, como el realizado por Larkin y Simon [19], ilustran sobre la utilidad de las técnicas de visualización de información. Estos muestran que la representación visual le facilita al usuario la utilización de complejas inferencias perceptuales ya implementadas en su corteza visual. El diseño adecuado de herramientas de visualización permite simplificar la búsqueda de información, mejorar las posibilidades de detección de patrones, aumentar los recursos accesibles por los usuarios, entre otros [5].

Se concede el permiso para la reproducción digital o impreso total o parcial de este trabajo sin contraprestación económica únicamente para la utilización personal o en clase. En ningún caso se podrán hacer o distribuir copias de para su explotación comercial. Todas las copias deben de llevar esta nota y la información completa de la primera página. Para cualquier otro uso, publicación, publicación en servidores, o listas de distribución de esta información necesitara de un permiso específico y/o el pago correspondiente.

Interacción '04 - 3-7 Mayo, 2004, Lleida, España.

1.2 Visualizando información estructurada.

Según Herman *et al.* [13], si la información a visualizar posee relaciones inherentes a sus elementos, los datos pueden ser representados por los nodos de un grafo con los arcos representando las relaciones. Por tanto, los grafos surgen como la representación estructural fundamental de esos datos. Si la información es de otro tipo, entonces las formas de representación son otras y uno de los objetivos de la visualización consiste en encontrar relaciones o patrones entre los datos. Así, al utilizar VisAD [28] para visualizar información atmosférica o generada¹ por modelos numéricos, se buscan isovalores, zonas de estabilidad o turbulencia, zonas con valores máximos o mínimos, entre otras. Estas relaciones no están explícitas en la información, sino que es el sistema el que favorece el agrupamiento o la diferenciación de los diferentes elementos que componen la información con la que se trabaja.

1.3 Algoritmos y herramientas para visualizar estructuras.

No son menores los problemas a resolver cuando se trata de dibujar grafos automáticamente. Por ejemplo, visualizar cientos de miles de nodos en un monitor de computadora de manera que el usuario pueda extraer información útil es un tema de investigación actual. Este y otros problemas han llevado al desarrollo de comunidades académicas que han generado formas innovadoras de visualizar información. Conetree [26], Treemap [17], algoritmos de posicionamiento radial [6], Information Cube [25], son algunos ejemplos de metodologías de visualización surgida de dichas comunidades. A su vez, han sido desarrolladas técnicas como zoom, paneo, fisheye [7], perspective wall [20], entre otras, las cuales al ser aplicadas a las metodologías anteriores permiten visualizar detalles del grafo sin perder, en algunos casos, la visión total del mismo (técnicas de foco + contexto). Por último,

[†] Instituto de Computación (INCO), Facultad de Ingeniería, Universidad de la República. <http://www.fing.edu.uy/inco>.

paquetes como el GraphViz [8], Pajek [4] y Tulip [2] implementan muchos de estos algoritmos y técnicas.

2. LA HERRAMIENTA

2.1 Descripción General

El proyecto pretende atacar la problemática descrita mediante la construcción de una herramienta que permita visualizar y explorar información almacenada en una base de datos relacional. Dada la heterogeneidad potencial en la información a visualizar por un sistema de este tipo, se adoptó como eje principal de diseño el brindar al usuario la mayor libertad posible, de manera que pueda definir su forma de visualización en función de las herramientas implementadas.

Al existir proyectos, en formato de software libre, que implementan bibliotecas y herramientas de visualización de grafos, se realizó un estudio con la finalidad de reutilizar código. En dicho trabajo, se analizaron bibliotecas como GraphViz [8], GRAPPA [3], Grip [11], GDtoolkit [9], Tulip [2] y VCG [27], VGJ [21], optándose finalmente por ésta última. VGJ (*Visualizing Graphs with Java*) es una aplicación desarrollada por McCreary *et al.* [21], de cuyas características destacamos:

- Está implementado en Java lo cual facilita la portabilidad y la conexión con bases de datos mediante JDBC [12].
- Provee un espacio tridimensional para la visualización del grafo, lo que permite implementar los algoritmos de disposición 3D de grafos.
- Implementa algunos algoritmos de disposición 2D de grafos, como CGD [22] y K&K [18] entre otros.
- Ofrece una interfaz gráfica.
- Utiliza una especificación textual de grafos (GML [14]) que permite interoperar con otras aplicaciones. Recientemente han aparecido nuevas bibliotecas, como Jung [15] y JGraphEd [16], que posiblemente se puedan utilizar en nuevas versiones de la herramienta.

Además de VGJ se utiliza AGD [1] la cual brinda una serie de algoritmos de sumo interés como, por ejemplo, los algoritmos de disposición de grafos planos. Ésta, al no ser distribuida en formato software libre, ve restringida su utilización a la ejecución del código objeto.

Sobre VGJ se codificó una nueva interfaz, la cual posee un entorno de múltiples ventanas, y donde se implementaron nuevas técnicas de visualización de los datos.

2.2 Funcionalidades proporcionadas por la herramienta

Dado que nuestra aplicación provee actualmente muchas funcionalidades, una descripción completa de la herramienta escapa a las posibilidades materiales de este artículo. A continuación, se brinda una descripción de sus funcionalidades más relevantes:

- *Interfaz MDI.* Se recodificó la interfaz original pasándose de la biblioteca AWT [30] utilizada por VGJ, a la SWING [10], sustituyendo el modelo de documento único (SDI) por otro de documentos múltiples (MDI). El modelo MDI permite al usuario manejar simultáneamente múltiples vistas y representaciones de un mismo grafo, las cuales pueden a su vez sincronizarse.
- *Algoritmos de disposición.* La herramienta hereda los algoritmos proporcionados por AGD (PureOrthogonal, Schnyder, Planarization, entre otros) y VGJ, proveyendo además Treemap y ConeTree junto con otros algoritmos de disposición para árboles.
- *Agrupamiento (clustering).* Si bien VGJ ofrece la funcionalidad de agrupar nodos, VIEG implementa distintas lógicas para realizar dicho agrupamiento, siendo posible concentrar nodos por igualdad en los campos de los datos representados, por cercanía espacial o por el grosor de los arcos incidentes.
- *Navegación y Foco + Contexto.* Para permitir la visualización de grandes volúmenes de información se han implementado las funcionalidades de zoom, paneo y fisheye [7].
- *Mapeo a variables visuales.* Es posible asignar a los nodos y arcos distintas propiedades visuales (posición, tamaño, forma, color, grosor, etc.) en función de sus datos asociados.
- *Scripting.* Por medio de Beanshell [23] se ofrece al usuario la posibilidad de interpretar en tiempo real scripts escritos en Java, permitiendo adaptar el funcionamiento de la aplicación así como crear nuevas funcionalidades.

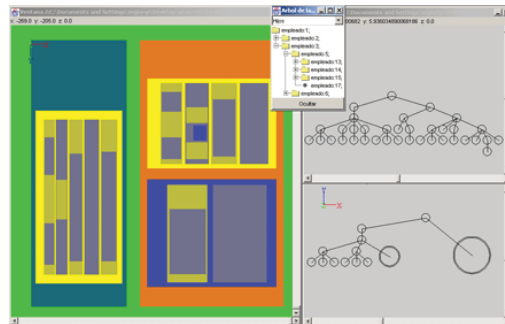


Figura 1. Distintas representaciones de un mismo conjunto de datos. De izq. a der. y de arr. a ab. TreeMap, TreeLayout (AGD), Treemap y TreeLayout con agrupamiento.

2.3 Mapeo a variables visuales

Los mapeos constituyen una de las principales herramientas brindadas por la aplicación para facilitar la interpretación de la información por parte del usuario.

De acuerdo con Stevens [29] los datos pueden clasificarse en cuatro tipos: nominal, ordinal, intervalo y ratio. A su vez, MacKinlay [24] ha mostrado cuán apropiados son distintos

esquemas visuales para comunicar datos de carácter nominal, ordinal y cuantitativo (intervalo y ratio). Por ejemplo, objetos de distinta forma son más aptos para representar datos de tipo nominal mientras que objetos de distinto largo resultan mejores para representar datos de tipo cuantitativo.

A raíz de estos trabajos, se implementaron una serie de técnicas (mapeos) que permiten modificar las propiedades visuales de nodos y arcos en función de sus datos asociados. Las variables visuales consideradas por la aplicación son el color, el tamaño, la posición, un rótulo asociado y el grosor, entre otras. Por tanto, las propiedades visuales de los grafos pueden ser variadas de forma tal que tengan valor semántico, facilitando al usuario la realización de las complejas inferencias perceptuales mencionadas anteriormente.

La aplicación permite que el usuario defina a qué tipo pertenece cada uno de los datos a mapear, logrando de esta manera definir la semántica que el usuario desea para la información visualizada.

Además es posible mencionar que el color de cada nodo o arco puede ser definido asociando algún dato a un valor dentro de un mapa de colores (eventualmente definido también por el usuario).

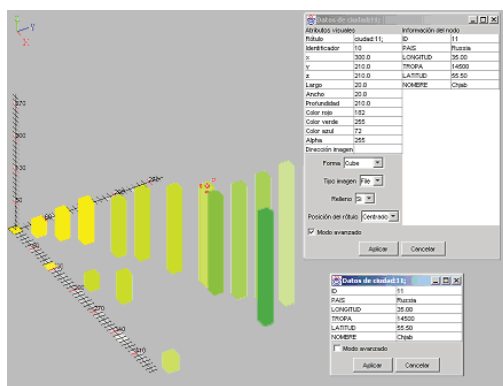


Figura 2. Visualización de los mismos datos de la figura 1, utilizando un mapeo 4D (posición, altura y color). Cada nodo del grafo se visualiza como una barra. Los diálogos muestran la información asociada al nodo seleccionado.

3. CONCLUSIONES

Este artículo presenta el estado actual de desarrollo de la herramienta *Visualización de Información Estructurada mediante Grafos* la cual es fruto del proyecto “Desarrollo de una herramienta para la visualización y exploración gráfica de información jerárquica” llevado a cabo en el Instituto de Computación de la Facultad de Ingeniería de la Universidad de la República, Uruguay. El proyecto comienza con una revisión de la bibliografía y las aplicaciones disponibles, para luego focalizarse en el desarrollo de la herramienta. Actualmente, VIEG brinda al usuario un conjunto importante de posibilidades de visualización de la información, facilitando su comprensión, especialmente

cuando ésta se presenta en grandes volúmenes de carácter multidimensional.

4. TRABAJOS FUTUROS

La primer versión de VIEG estará disponible para mediados del 2004, cuando se estudiarán sus posibles ampliaciones y mejoras. Una amplia gama de algoritmos no han sido implementados y muchos aspectos de la interfaz requieren proyectos específicos para su análisis e implementación. Se están buscando potenciales usuarios de la herramienta, en una primer etapa dentro del ámbito académico, con la finalidad de evaluar la propuesta y obtener nuevos requerimientos. A su vez, el código generado a lo largo del proyecto queda disponible, lo cual permitirá su reutilización en aplicaciones más específicas.

5. REFERENCIAS

- [1] Algorithms for Graph Drawing, <http://www.ads.tuwien.ac.at/AGD/index.html>. Última visita en diciembre de 2003.
- [2] Auber, D. Tulip an information visualization software for huge graphs, <http://www.tulip-software.org/>. Última visita en diciembre de 2003.
- [3] Barghouti, N., Mocenigo, J. y Lee, W. Grappa: A Graph Package in Java. En *Proceedings of the 5th International Symposium of Graph Drawing*. Springer-Verlag, Berlin, 1997, 336-343.
- [4] Batagelj, V., y Mrvar, A. Pajek - Analysis and Visualization of Large Networks. En Jünger, M., Mutzel, P., (Eds.) *Graph Drawing Software*. Springer, Berlin 2003. 77-103
- [5] Card, S. K., Mackinlay, J. D. y Shneiderman, B. *Readings in Information Visualization: Using Vision to Think*, Morgan-Kaufmann Publishers, San Mateo, CA, 1999, 16.
- [6] Eades, P. Drawing Free Trees. *Bulletin of the Institute for Combinatorics and its Applications*, 1992, 10-36.
- [7] Furnas, G. W. Generalized fisheye views. En *Proceedings of the CHI'86 Conference on Human Factors in Computing Systems and Graphic Interfaces*. Addison-Wesley, Boston, (Mayo 1986), 16-23
- [8] Gansner, E. R. Drawing graphs with GraphViz. <http://www.research.att.com/~north/graphviz/>. Última visita en diciembre de 2003.
- [9] GDTToolkit package. <http://www.inf.uniroma3.it/people/gdb/wp12/GDT.html>. Última visita en diciembre de 2003
- [10] Geary, D.M. *Graphic Java 2, Volume 2*, Swing, 3/e. Prentice Hall PTR, 1999.
- [11] GRIP, <http://tgrip.cs.arizona.edu/>. Última visita en diciembre de 2003.
- [12] George Reese: *Database Programming with JDBC and Java*, 1st Edition June 1997. ISBN 1-56592-270-0.

- [13] Herman, G. Melancon, and M. Marshall. Graph visualization and navigation in information visualisation: a survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1), 2000, 24-43.
- [14] Himsolt, M. *GML: A portable Graph File Format*. Technical report, University at Passau, 1997.
- [15] Java Universal Network/Graph Framework. <http://jung.sourceforge.net/>. Última visita en diciembre de 2003.
- [16] JGraphEd. <http://www.jharris.ca/JGraphEd/>. Última visita en diciembre de 2003.
- [17] Johnson, B., y Shneiderman, B. Treemaps: a Space-Filling Approach to the Visualization of Hierarchical Information Structures. En *Proceedings of the IEEE Visualization '91*, 284-291.
- [18] Kamada, T. & Kawai, S. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31, 1989, 7-15.
- [19] Larkin, J. H., y Simon, H. A. Why a diagram is (sometimes) worth 10,000 words. *Cognitive Science*, 11, 1987, 65-100.
- [20] Mackinlay, J.D., Robertson, G.G. y Card, S.K. The Perspective Wall: Detail and context smoothly integrated. En *Proceedings of CHI '91*, New Orleans, (Abril 1991), 173-179.
- [21] McCreary, C., y Barowski, L. VGJ: Visualizing graphs through java. En *Proceedings of the Graph Drawing'98*. Springer-Verlag, 1999, 454-455.
- [22] McCreary, C., Chapman, R. y Sheih, F. Using graph parsing for automatic graph drawing. *IEEE Trans. on System. Man and Cybernetics*. (Setiembre 1998).
- [23] Niemeyer, P. BeanShell User Manual, <http://www.beanshell.org/manual/contents.html>. Última visita en diciembre de 2003.
- [24] Nowell, L. T. *Graphical encoding for information visualization: using icon color, shape and size to convey nominal and quantitative data*. Ph.D. Thesis, Virginia Tech, 1997.
- [25] Rekimoto, J., y Green, M. The Information Cube: Using Transparency in 3D Information Visualization. En *Proceedings of the Third Annual Workshop on Information Technologies & Systems (WITS'93)*, 1993.
- [26] Robertson, G. G., Card, S. K., y Mackinley, J. D. Information Visualization using 3D Interactive Animations. *Communication of the ACM* 36, 4 (April 1993), 57-71.
- [27] Sander, G. *Graph layout through the VCG tool*. Technical report, FB14 Informatik, Universität des Saarlandes, Saarbrücken, 1994.
- [28] VisAD. <http://www.ssec.wisc.edu/~billh/visad.html>. Última visita en diciembre de 2003.
- [29] Ware, C., *Information Visualization. Perception for Design*, Morgan Kaufmann Publishers, 2000, 29.
- [30] Zukowski, J. *Java AWT reference 1.1*, 1st edition, O'Reilly, eds, 1997.

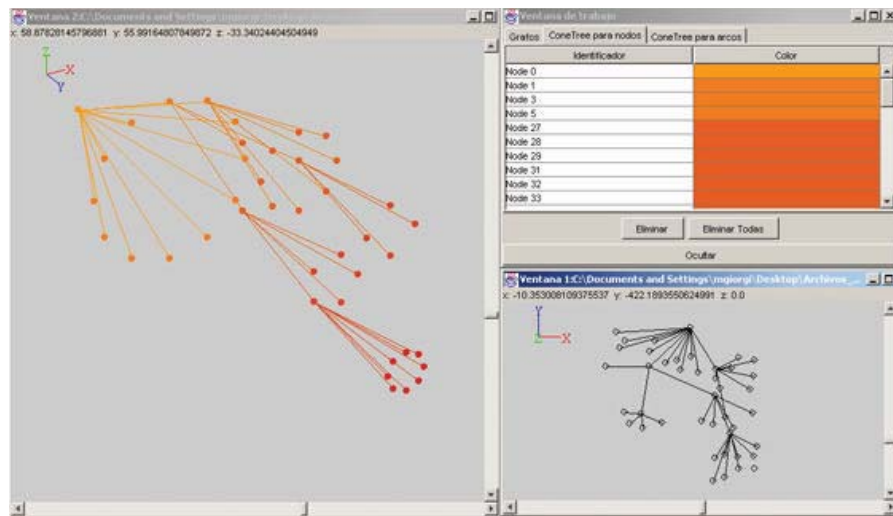


Figura 3. Grafo desplegado utilizando el algoritmo Conetree (izq.) junto con las referencias correspondientes a los colores utilizados (der. arr.) y otra representación del mismo grafo (der. ab.).