

VGJ: Visualizing Graphs Through Java

Carolyn McCreary¹ and Larry Barowski²

¹ Tufts University, Medford, MA 02155, USA

² Auburn University, Auburn, AL 36849

1 Introduction

VGJ is an automated system capable of converting a textual description or a drawing of a graph into a well organized and readable layout of the graph. VGJ (visualizing graphs with Java), includes a graph editor and a set of algorithms that will automatically layout and draw a graph. The graph drawing package can be accessed through the web at: www.tufts.edu/mccreary/graph_drawing.html.

2 Graph Editing Capabilities

The user interface of VGJ is very intuitive and contains both the typical features of tools that support drawing graphs as well as some additional features.. With a click of the mouse, a user can create nodes and edges, select single or multiple nodes and edges, and move selected objects. Initially the graph is directed and the edges are drawn with an arrow to indicate direction. The user can set the graph type to undirected by selecting an option in the properties menu. A dialog box allows the user to specify a node's (i) shape (oval or rectangle), (ii) label, (iii) label position, (iv) exact node position and (v) node dimensions. The label can be placed in one of three ways: below the node, centered in the node or autosized. If the node is autosized, the label is placed inside the node and the node is sized to encompass the label. Associated with selected nodes are handles that allow them to be dragged, scaled proportionally or scaled in either dimension. Nodes can be combined and displayed as a single node through the "group control" menu. Groups can be created in stages and separated in reverse stages.

A dialogue box with edge attributes is associated with each edge. The attributes include edge label, line style, bend points and data. Edge labels are drawn parallel to the edge so that edge identification is clear. Bends can be inserted by specifying the x- and y-coordinates of each of the bend points.

Corresponding to every visual graph is a textual representation in GML (Graph Modeling Language)[1]. The user can also convert the VGJ drawn graph to PostScript format.

3 Layout Algorithms

Currently VGJ offers three layout algorithms: Tree, to layout rooted trees; CGD, for directed graphs; and spring, for undirected graphs. There is also an algorithm to test a graph for biconnectivity or make it biconnected by adding edges.

Tree and Undirected Graph Layout: The tree algorithm implementation is that of Walker [4]. Trees are drawn so that nodes at same level lie on a straight line; parents are centered over their children; there is vertical symmetry; isomorphic subtrees are drawn identically.

The undirected graph algorithm is that of Kamada and Kawai[2]. Their algorithm defines ‘energy’ between pairs of graph points and works to minimize the total energy of the graph.

Directed Graph Layout: CGD, clan-based graph drawing, produces a layout for directed graphs[3]. The goals of the layout are to (1) follow the direction of the arcs so that ancestor nodes always lie above their descendants; (2) balance the nodes horizontally within each level; (3) have few edge crossings; (4) have few edge bends. The node layout is determined by the combination of (1) parsing of the graph into logically cohesive subgraphs and (2) defining layout attributes to apply to the resulting parse tree. The parse is based on a simple graph grammar, and the attributes that are now programmed into CGD produce a layout whose nodes are balanced both vertically and horizontally. Its parser is the first we know about that decomposes directed graphs into a tree of subgraphs (clans).

CGD defines an attribute grammar for the parse tree, and computes node layout through the attributes. Graph parsing is an improvement over the hierarchical approach because it discovers clans, structures which have two-dimensional affinity rather than layers which have only one-dimensional similarity. The use of graph parsing distinguishes CGD from all other general directed graph drawing schemes.

CGD’s drawings are unique in several ways: (1) The node layout is balanced both vertically and horizontally. (2) Nodes within a clan, a subgraph of nodes that have a common relationship with the rest of the nodes in the graph, are placed close to each other in the drawing. (3) Nodes are grouped according to a two-dimensional affinity rather than a single dimension such as level. (4) The layout can easily display nodes of varying sizes because space is reserved for each node in the bounding box attribute of the parse tree.

References

1. Himsolt, M.: GML: the Graph Modeling Language. on the internet at: <http://www.uni-passau.de/Graphlet/GML/>
2. Kamada, T., Kawai, S.: An Algorithm for Drawing General Undirected Graphs. *Information Processing Letters* **31** (1989) 7–15
3. McCreary, C., Chapman, R., and Shieh, Fwu-Shan Using Graph Parsing for Automatic Graph Drawing *IEEE Trans. on Systems, Man and Cybernetics*, Sept. 1998.
4. Walker, J. Q.: A Node-positioning Algorithm for General Trees. *Software Practice and Experience* **20**, 7 (1990) 685–705