Flexible Graph Layout and Editing for Commercial Applications[*]

Arne Frick, Brendan Madden, and the Research and Development Staff

Tom Sawyer Software, 804 Hearst Ave, Berkeley, CA 94710,
{africk, bmadden}@tomsawyer.com
http://www.tomsawyer.com

1 Introduction

Tom Sawyer Software produces commercial-quality layout and diagramming component technology for use by corporate enterprises, software providers, and educational institutions. Software application developers utilize Tom Sawyer Software's technology to solve difficult modeling, complexity management, and diagram visualization problems. Our focus is the continued research into layout theory and how we can apply our technical expertise in solving our customers' problems. To date we have two comprehensive product families, the *Graph Layout Toolkit* (GLT) and the *Graph Editor Toolkit* (GET). Designed as software components to be embedded within customer applications and equipped with a well-documented API, they are available for a wide range of programming environments.

2 Products

2.1 Graph Layout Toolkit

The Graph Layout Toolkit [2] is a family of portable object positioning libraries designed for integration into Graphical User Interfaces (GUI). Currently available in more than 50 configurations, it creates more understandable graph drawings by choosing geometric positions for nodes and a suitable routing for edges. The GLT can be used with almost any GUI toolkit, windowing system, compiler, operating system, or database through C, C++, Java and ActiveX interfaces.

2.2 Graph Editor Toolkit

The Graph Editor Toolkit [1] provides a software framework that eases the addition of advanced graph visualization technology to GUI products. The Graph Editor Toolkit is built upon and linked to two framework systems, the Graph Layout Toolkit and the Microsoft Foundation Class (MFC) Library. It is currently available on Microsoft Windows platforms as DLLs and ActiveX controls, and a port to Unix is underway. Furthermore, we are working on a 100% Pure Java implementation of the GET.

[*] This work was supported in part by NIST, Advanced Technology Program grant number 70NANB5H1162.

S.H. Whitesides (Ed.): GD'98, LNCS 1547, pp. 446–447, 1998.

3 Commercial Applications

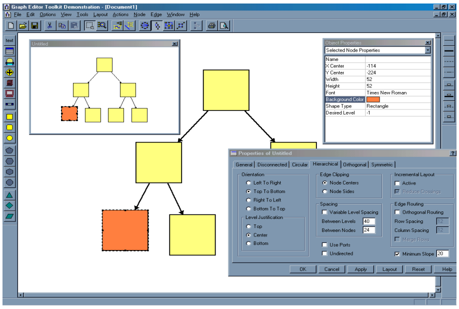
Graphs are commonly used in communicating relational information that arises in commercial applications ranging from database design to network topology diagrams. For example, many CASE tools use graphs to model the dependencies between modules in a large program. Further examples of diagrams which illustrate relational data are Data Flow Diagrams, PERT charts, various UML diagram types, and models abstracting complex physical networks such as large corporate communication networks.

4 Software Architecture

Table 1 shows the 4-tier architecture of Tom Sawyer Software's technology. The diagramming layer supports four primary layout styles with specific features available within each library. Other diagramming features include ports in hierarchical and orthogonal layout, incremental layout, and nesting functions. The abstraction layer provides folding, hiding, and collapse/expand functionality to simplify drawings. For applications requiring a GUI layer, the GET provides components to create a presentation layer to access the underlying model, diagramming and abstraction layers (cf. Fig. 2), as well as overview windows, property inspectors and high-level event handling.

Foundation	Functionality
Presentation	Zoom, scroll, menus, icons, inspectors
Abstraction	Navigation, folding, hiding, nesting
Diagramming	Sizes, routing, labels, layout, pasting
Model	Graph, nodes, edges
user database	

Fig. 1. Technology overview.



Selected Node Properties

Name	
X Center	-114
Y Center	-224
Width	52
Height	52
Font	Times New Roman
Background Color	🟧
Shape Type	Rectangle
Desired Level	-1

Orientation
○ Left To Right
● Top To Bottom
○ Right To Left
○ Bottom To Top

Level Justification
○ Top
● Center
○ Bottom

Edge Clipping
● Node Centers
○ Node Sides

Spacing
□ Variable Level Spacing
Between Levels 40
Between Nodes 24

□ Use Ports
□ Undirected

Incremental Layout
□ Active
☑ Reduce Crossings

Edge Routing
□ Orthogonal Routing
Row Spacing 12
Column Spacing 12
□ Merge Rows

☑ Minimum Slope 20

Fig. 2. Examples of presentation level functionality: property inspector and overview windows.

References

1. U. Doğrusöz *et. al.* Toolkits for development of software diagramming applications. In *Proceedings of SEKE'98*. Knowledge Engineering Institute, 1998.
2. B. Madden *et. al.* Portable graph layout and editing (system demonstration). In *Proceedings of Graph Drawing'95*, pages 385–395. Springer Verlag, 1996.