WilmaScope – An Interactive 3D Graph Visualisation System

Tim Dwyer and Peter Eckersley

Department of Computer Science, University of Melbourne,
Parkville, Victoria 3052, Australia,
dwyer@cs.usyd.edu.au,pde@cs.mu.oz.au,
http://www.wilmascope.org

Abstract. This is a brief description of the WilmaScope interactive 3D graph visualisation system. Wilma features clustering of related groups of nodes, a GUI for editing graphs and adjusting the force layout parameters and a CORBA interface for creating and interacting with graphs remotely from other programs. It has also been used to construct 3D UML Class and Object models as part of a usability study. Wilma is freely available under the terms of the GNU Lesser General Public License[1].

Wilmascope is implemented in Java, using the Java3D libraries to provide a portable 3D graph visualisation system. It has been tested under Linux, Solaris and Windows NT platforms. It includes a GUI for interactively creating and modifying graph structures, and tuning the parameters of the force directed layout engine, see Figure 1. When a graph element is added to a graph Wilma animates the layout process until the graph returns to a balanced state. The basic engine is flexible enough to be used in many different types of graphs. Users are able to zoom, rotate and laterally scroll to "fly through" 3D graph structures.

A usability study was conducted on a 3D UML Class modelling tool based on the Wilma engine with very encouraging results[1].

Wilma also includes an API which allows other programs to generate complex graphs which can be displayed in real-time. The API is accessible via a CORBA interface, and we have implemented a number of clients in Python[2] to create example graphs. The CORBA interface features callbacks allowing remote programs to not only generate graphs but also to interact with them via a point and click interface. For example, a Wilma client could provide a bridge to a web browser such as Netscape, graphing the history of visited pages and when a node in the history graph is clicked, reloading that page into the browser.

Wilma supports a powerful hierarchical graph model, supporting nested clusters such that each cluster has its own force parameters. Force parameters for a cluster or the root graph may be adjusted in real time and the affects on the

[1] http://www.gnu.org/copyleft/lesser.html
[2] A high-level, interpreted, object-oriented language; http://www.python.org

P. Mutzel, M. Jünger, and S. Leipert (Eds.): GD 2001, LNCS 2265, pp. 442–443, 2002.

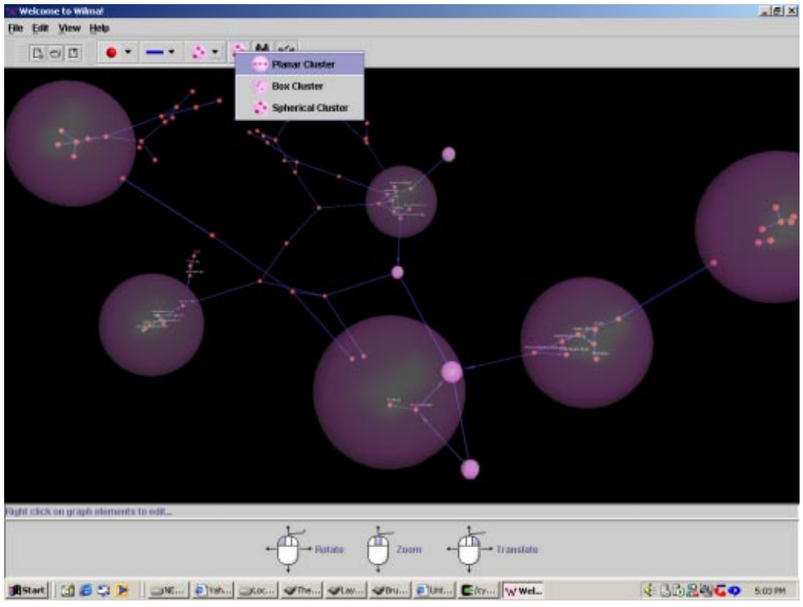

Fig. 1. A screenshot of Wilma in action.

layout are visible instantly. Well clustered graphs will also be arranged faster since internode forces need only be calculated for each cluster. Clusters may also be collapsed or expanded by the user to elide or show detail providing an efficient way to navigate large nested graphs.

Within clusters it is also possible to define constraint systems such as confining nodes in the cluster to a plane. An attempt is made to satisfy the constraint through the use of a force attracting nodes to the ideal location. An equal force is exerted back on the constraint system causing the whole cluster to rotate such that the constraints may be satisfied within the cluster while forces external to the cluster affect the cluster's orientation. This can be used to obtain sophisticated 3D layouts such as Cone Trees within the force directed system.

References

1. T. Dwyer. Three dimensional uml using force directed layout. Technical report, Department of Computer Science, The University of Melbourne, Parkville, Australia, 3052, 2000.