

Tulip: a Scalable Graph Visualization Framework

David Auber, Patrick Mary, Morgan Mathiaut, Jonathan Dubois, Antoine Lambert, Dan Archambault, Romain Bourqui, Bruno Pinaud, Maylis Delest, Guy Melançon ^{*,**}

^{*}CNRS UMR 5800 LaBRI, Campus Université Bordeaux I
@labri.fr

<http://www.labri.fr>

^{**}INRIA Bordeaux Sud-Ouest, Campus Université Bordeaux I
@inria.fr

<http://www.inria.fr/bordeaux>

Résumé. The Graph Visualization Framework Tulip now enjoys 10 years of user experience, and has matured its architecture and development cycle. Originally designed to interactively navigate large graphs, the framework integrates state-of-the-art software engineering concepts and good practices. It offers a large panel of graphical representations (traditional graph drawing as well as alternate representations). Tulip is most useful in a data mining and knowledge discovery context, allowing users to easily add their own data analysis and computing routines through its plug-in architecture.

Most research efforts in data mining and knowledge extraction and representation require experimentation and validation. To this end, our group develops the Tulip Graph Visualization Framework. Tulip offers a C++ plug-in mechanism easing the development and addition of new algorithms (computing graph statistics, graph drawing or graph clustering). This actually is a main feature of the Tulip platform.

Tulip implements optimized and efficient data structures, data management and filtering/inheritance mechanisms. Its rendering engine relies on OpenGL, while its GUI rests on Trolltech's QT library. The main interaction paradigm offered by Tulip is the computation and direct manipulation of graph hierarchies, making it unique among all available graph visualization platforms.

Tulip can be used in a typical scenario where the goal is to discover trends in data. The exploration can be driven by the user using available plug-ins, computing statistics on a graph, designing colormaps or assigning sizes to nodes, for instance. Fig. 1 illustrates a typical scenario when performing data exploration and analysis using Tulip. While the left pane gives access to several graph properties and tuning parameters, the right panes shows several views on the data : node-link graph drawing (top left) ; parallel coordinates (bottom left, here shown circularly) ; self-organizing maps and histograms, scatterplot matrix (top right) and standard tabular sheet (bottom right). Tulip is incredibly efficient at synchronizing views and keeping properties coherent between views. Applying colormaps through histograms or selecting nodes through self-organizing maps immediately transfers to all other views.