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.

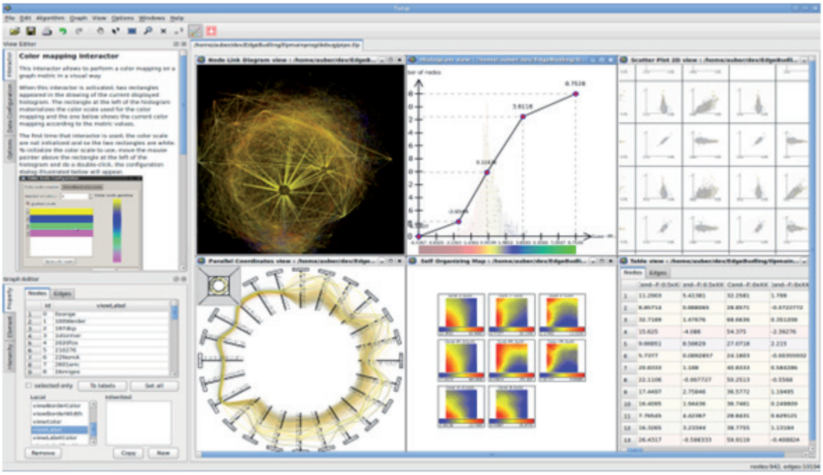

FIG. 1 – *Example working session using Tulip.*

One of Tulip's main feature is the possibility for a user to add one's own plug-ins to the whole framework. [1] The result of a plug-in will typically be stored in a pre-designed container called a *property*. Properties appear as a central paradigm to Tulip, allowing the framework to model and apply to a vast variety of application domains and application uses. Plug-ins can rely on each other through property invocation mechanisms. More sophisticated plug-in and interactors can be developed at the cost of dealing with QT.

Tulip also appears as a unique framework allowing the computation and visualization of multi-level clustering algorithms. Relying on OpenGL, graph hierarchies can be interactively explored where lower level graphs appears as they gain size.The most recent advances with Tulip showed how GPU programming could be used to improve performance when drawing graphs or when designing edges as high polynomials curves (Béziers).

Summary

La plate-forme de visualisation de graphes Tulip est maintenant riche de dix années d'expérience, tant du point de vue de son utilisation, de la conception de son architecture que de son développement logiciel. Conçue pour manipuler de très grands graphes dans un contexte interactif, elle adopte les concepts et pratiques de l'état de l'art du génie logiciel et implémente une large palette de représentations de graphes. Destinées à être utilisées avec des outils d'-analyse de données, son architecture permet à l'utilisateur d'intégrer ses propres composants d'analyse et de dessin.

[1] The actual methodology can currently be accessed on Tulip's website. See the URL `www.tulip-software.org`, to access the developer's manual, user's manual and tutorial. We provide C++ templates from which one can develop its own plug-in to draw graphs or compute specific graph statistics.