Transparent Layering for Visualizing Dynamic Graphs Using the Flip Book Metaphor

Holger Stitz*, Samuel Gratzl, Stefan Luger, Nils Gehlenborg and Marc Streit

* Corresponding author: holger.stitz@jku.at

JKU JOHANNES KEPLER UNIVERSITY LINZ

HARVARD MEDICAL SCHOOL

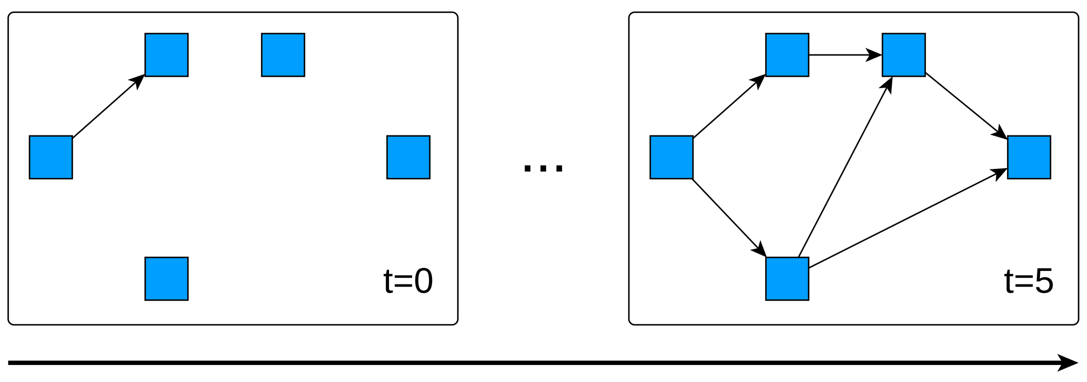
Visualizing dynamic graphs is challenging because changing node and edge attributes as well as topological alterations need to be encoded in the visual representation. However, existing approaches such as animation, juxtaposition, and superimposition do not scale well. In this poster we propose a novel layering approach for visualizing dynamic graphs where the graph for **each point in time is a single layer** and **parts of each layer are slightly shifted based on a degree-of-interest (DOI) function.** In contrast to 2.5D representations that also use layering, users cannot freely change the viewing perspective but are restricted to the top view, avoiding occlusion and distortion problems. We demonstrate the layering approach by applying the concept to two graph visualizations: a node-link diagram and a radial hierarchy visualization.

Visualizing Dynamic Graphs

Predominant visualization methods for dynamic graphs [1,2]:

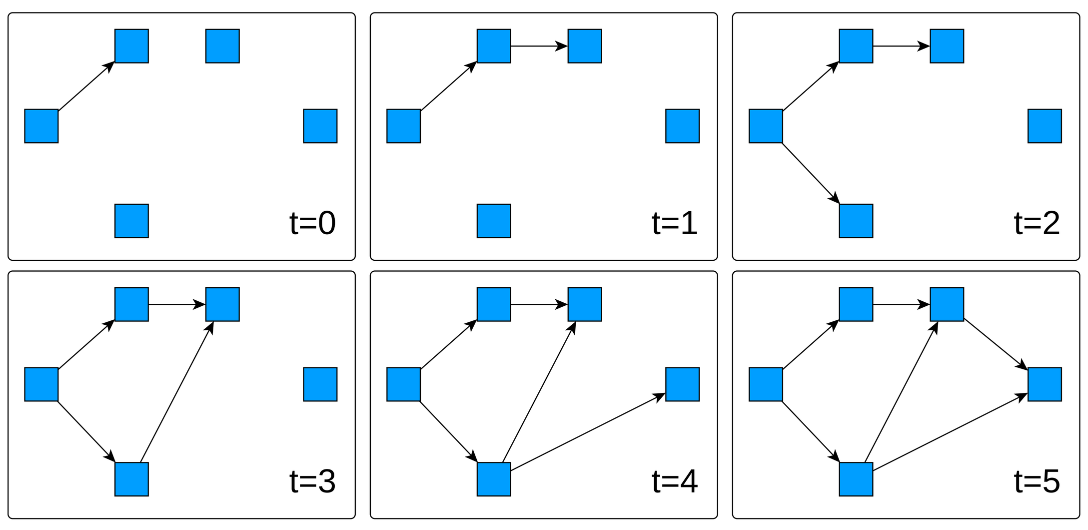
Animation

Animation maps time to time. In the flip book metaphor this corresponds to rapidly flipping the pages.



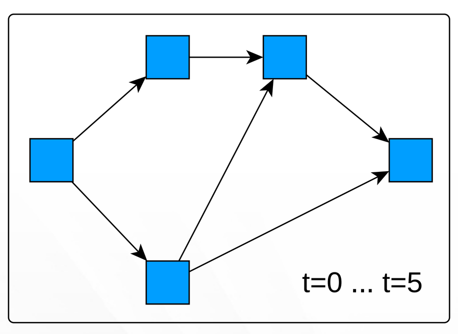
Juxtaposition

Juxtaposition represents the time steps using small multiples. In the flip book metaphor this corresponds to tearing out the pages and positioning them side-by-side.



Superimposition

Superimposition stacks the static graph representation of each time step as layers on top of each other. In the flip book metaphor this corresponds to the top view onto the book that has semi-transparent pages.



Layering Approach

We extend the superimposition approach where the individual semi-transparent layers—each representing the graph at a single time step—are stacked on top of each other. Time and attribute changes are encoded in position and brightness to ensure scalability by aggregating non-relevant subgraphs defined by a DOI function on all nodes and edges.

The DOI determines the relevance of an element for the user's task. We include user-driven components, such as results from filter and search operations, next to automatic components, expressing attribute and graph changes over time. Elements with a low DOI value are less relevant and therefore can be aggregated.

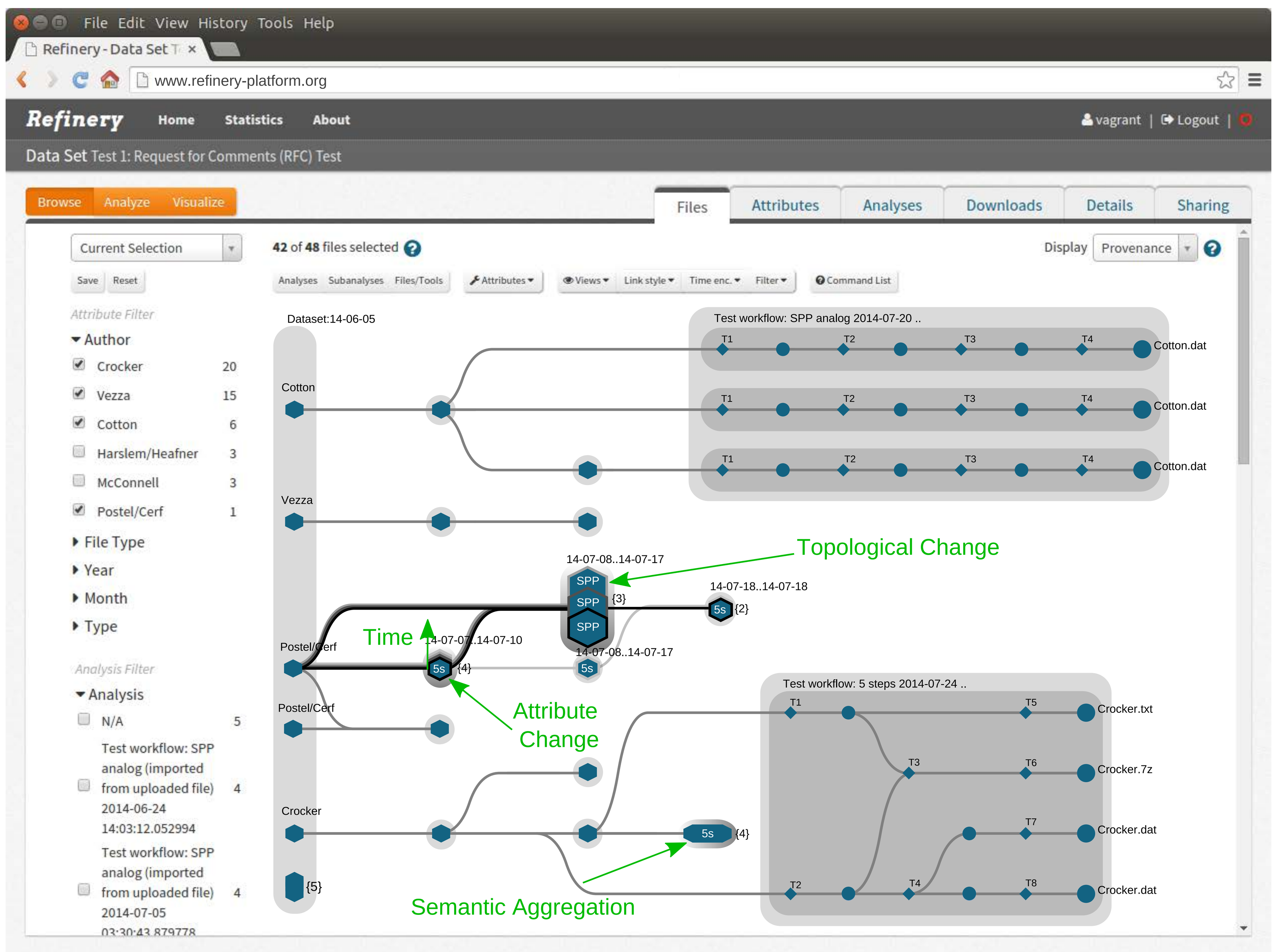


Figure 1: A mockup of a dynamic node-link diagram of multiple analysis workflows executed in Refinery [4] as provenance visualization.

Refinery is a flexible analysis platform designed for collaborative reproducible research on heterogeneous genomic datasets. One requirement is to provide access to results of comprehensive analysis results on large and complex datasets, as well as launching new workflows on previous results or re-running workflows with different parameters.

The provenance graph captures inputs, outputs, tools, and parameters of every workflow execution. In the graph time is encoded in the vertical position of the nodes and attribute changes encoded using brightness. Parts of the graph with a low DOI value are aggregated.

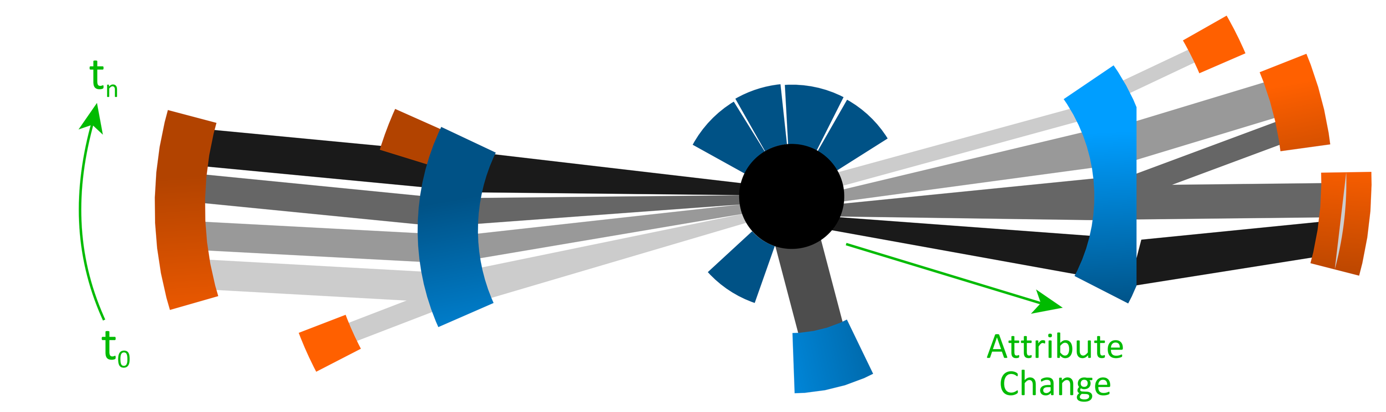


Figure 2: A datacenter infrastructure visualized as radial hierarchical network with physical servers and virtual machines as arc nodes and connections as edges. The time is encoded in brightness and attribute changes in node position.

Node-Link Diagram

Figure 1 visualizes a dynamic node-link diagram upon executed and chained Galaxy workflows in Refinery Platform. Individual time step layers are stacked slightly vertically shifted onto each other, encoding time in position. Time is redundantly encoded in brightness, such that old time steps fade to white. The combination of encoding time in position and brightness forms a black-to-white gradient on nodes and edges.

The DOI value drives the aggregation, such that only elements with small and similar DOI values are aggregated. The first aggregation level reduces the size of node glyphs and the contained data representations similar to a semantic zoom. The second level further reduces the glyph size. In addition, the DOI influences the shift of the layers, emphasizing the elements' change over time. While static elements appear as a single instance, highly changing ones have shifted layers.

In summary, applying our approach on a dynamic node-link diagram can be interpreted as using a rubber sheet [3] for each semi-transparent flip book page. Static parts of the diagram (which correspond to a low DOI) are aggregated by squeezing the rubber sheet locally, whereas highly changing ones (high DOI) are locally stretched.

Radial Hierarchy Visualization

In Figure 2 the black circle in the center represents the root node and arcs on each ring correspond to nodes in a certain hierarchy level. Edges between two levels are connections within the network. The width of the edges encode the used bandwidth. The node position encodes attribute changes and edge brightness combined with their relative position encode time. The more the connection between two nodes jitters, the higher the corresponding DOI, and the more the two nodes are separated. In the Figure 2 the horizontal arcs have varying connections over time, while the vertical arcs have stable connections that are attached to their parent node.

Discussion

Layout Algorithm: Algorithms that compute the layout for an individual time step need to consider the space required by the shift of other time steps. Also, within the same time step different DOI values and aggregation levels appear, which may also change during the analysis. Therefore, layout algorithms need to find the right balance between computing a scalable compact layout and preserving the user's mental map between DOI changes.

Topological Changes: The approach of stacking and shifting individual time steps of a dynamic graph works for small topological changes. However, large changes generate problems regarding the layout algorithm, as well as the precise tracking of changes among multiple time steps, since the most frequent time step is on top.

Conclusion

We introduce the flip book metaphor as a novel layering approach for visualizing dynamic graphs. We extend the superimposition approach by combining it with a flexible DOI function, which influences how non-relevant parts of the graph are aggregated. Changes in time are encoded in position and/or brightness.

[1] F. Beck et al. The state of the art in visualizing dynamic graphs. In Proceedings of the Eurographics Conference on Visualization (EuroVis '14) – State of The Art Reports, 2014.

[2] N. Kerracher et al. The design space of temporal graph visualisation. In Proceedings of the Eurographics Conference on Visualization (EuroVis '14), 2014.

[3] M. Sarkar et al. Stretching the rubber sheet: A metaphor for viewing large layouts on small screens. In Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '93), pages 81–91. ACM, 1993.

[4] http://www.refinery-platform.org

Acknowledgements

- This work was funded by the Austrian Research Promotion Agency (840232) and the US National Institutes of Health (K99 HG007583)
- Background image under Creative Commons by Horia Varlan

RISC Software GmbH · KT-NET · mowis · FFG