Software Matrix Layout

Matrix visualizations of graphs are attracting widespread interest in fields such as social networks, security visualization, bio-chemistry, or software engineering. Analysts of social networks deal with a large amount of connections between participants of social networks. They are particularly interested in questions like, which are the groups, how is the structure of the group, is there a main person or is the group activity balanced, or how are groups linked. In security analysis, attacks are in general depicted using attack graphs. Main questions in this area are, which are the attack paths, or how to secure the structure. In bio-chemical networks, the structure is normally displayed using node-link-diagrams. The networks themselves are quite involved, and are constantly changing and growing due to new findings. Questions in this area include, what is the structure of the graph, or how are different parts are related. In software engineering, matrix visualization plays an important rôle in showing the static structure of the software system. This static structure is used for a large variety of tasks, e.g., system design, detailed design, or reverse engineering. Especially, supporting reverse engineering of legacy systems is an important task, as many systems need to be extended even after the main developers are no longer available. As the matrix visualization provides a scalable and space efficient visualization of software architectures, it is particularly well suited to support these tasks.

Overall, the design space of matrix visualizations of graphs is as rich as the one of node-link-diagrams. Although some substantial work has been done to overcome major drawbacks of matrix visualizations of graphs, more work as been spent on improving the readability of node-link-diagrams than on improving the usability of matrix visualizations. In both cases, a major problem to be solved is scalability. Although quite some graphs are small, the size of graph-based data is growing tremendously. As screen real estate is limited, graph compression techniques are mandatory to handle the huge graphs. Interaction is one way to link compressed views to detailed views, to link overview and detailed information.

This, a major current focus in matrix visualization is to find the best scalable and space efficient visualization. We are especially interested in the questions, how to extend the interaction with matrix visualizations and how to combine additional visualizations with it. Additionally, we address scalability and space issues to facilitate the extraction of hidden information and to support the search for patterns in large scale data. Both interaction mechanisms and additional visualization are built to optimally support the analysis of large graphs, while supporting the previously mentioned tasks.