

# DAGmap View

Vassilis Tsiaras and Ioannis G. Tollis

Institute of Computer Science, FORTH, and  
Department of Computer Science, University of Crete  
Heraklion, Greece  
`{tsiaras,tollis}@ics.forth.gr`

**Abstract.** DAGmap view is a program written in Java that draws directed acyclic graphs using space filling techniques. In DAGmap view the layout function and the hierarchy presentation function have been decomposed to improve the stability of the layout during navigation and zooming.

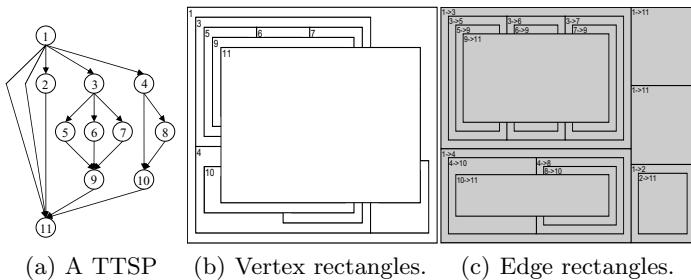
## 1 Description

Among the many alternative ways to visualize a tree, space filling visualizations, such as treemaps, have become very popular due to their efficiency, their scalability, and their easiness of navigation and user interaction [1]. Recently, we investigated space filling visualizations for hierarchies that are modeled by Directed Acyclic Graphs (DAG) and we defined the constraints for such a visualization [2].

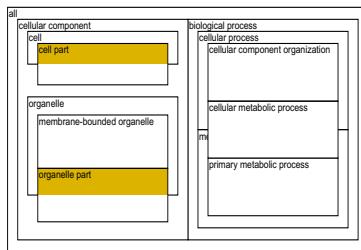
DAGmap view is a program written in Java that draws specific classes of DAGs, such as Two Terminal Series Parallel digraphs (TTSP), layered planar st-graphs and trees, using space filling techniques (Fig. 1). Additionally, it implements the vertex duplication heuristic and allows the user to specify a set of vertices to be duplicated. The tool has implemented many novel ideas such as drawing of vertices and of edges of a DAG, separate layout and hierarchy presentation functions, zooming without changing the size of the nesting borders and keeping the layout of the rectangles constant during zooming and navigation.

The two main functions of DAGmap are the layout and the hierarchy presentation functions. The layout function assigns rectangles to vertices and edges of a DAG  $G$  while the hierarchy presentation function illustrates the structure of a hierarchy using a number of techniques including the cushion, the nested, and the cascaded presentations. In DAGmap view we implemented nested presentation and we plan to implement cascaded and cushion presentations in the next release of the program.

In treemaps, nesting is trivial and is done along with the layout. The drawing rectangle  $R_u$  of a node  $u$  is shrunk and the resulting rectangle  $R'_u$  is located inside  $R_u$ . Then the border  $R_u \setminus R'_u$  is used for displaying information concerning  $u$  and  $R'_u$  is used for drawing the children of  $u$ . And this procedure is repeated recursively. In DAGmaps, layout and nesting should better be implemented as two separate functions. First, the layout function assigns rectangles to vertices



**Fig. 1.** Example of a TTSP digraph DAGmap drawing using the squarified layout



**Fig. 2.** In this example a subgraph of the Gene Ontology (GO) DAG is drawn. The color refers to the relationship between two GO terms. We use white color for *is\_a* relationship and light brown color for *part\_of* relationship. The term “cell part” *is\_a* “cellular component” and *part\_of* “cell”.

and/or to edges of a DAG  $G$ , in case that such an assignment is possible. Then the nesting function shrinks the rectangles proportionally to their longest path distance from the sources, in order to reveal the hierarchy structure. Decomposing the nesting from the layout greatly facilitates the implementation of the layout algorithm although slightly complicates the implementation of the nesting algorithm.

## References

1. Bederson, B.B., Shneiderman, B., Wattenberg, M.: Ordered and quantum treemaps: Making effective use of 2d space to display hierarchies. ACM Transactions on Graphics 21(4), 833–854 (2002)
2. Tsias, V., Triantafilou, S., Tollis, I.G.: Treemaps for directed acyclic graphs. In: Hong, S.-H., Nishizeki, T., Quan, W. (eds.) GD 2007. LNCS, vol. 4875, pp. 377–388. Springer, Heidelberg (2008)