Features planned in future releases

Some features under development for the next major release are listed here.

1. *Mode-less graph editing.* In place of the GDR-like mechanism, where panel buttons are used to determine how the graph editor responds to mouse actions – a click might create a node, initiate an edge or select an object – the Ctrl and Shift keys can determine the "mode". So, for example, Ctrl-Click would create a node and Shift-Click would initiate an edge.

2. *Scrolling in the graph window.* Currently, if the graph is too large to fit in the current window, the only recourse is to do force-directed layout.

3. *Scaling of graph within window.* A possible alternative or option to the previous item is making the node positions adjust so that they always fit into the current window. How best to handle the semantics is not clear: you don't necessarily want the editor to change x and y attributes in the GraphML whenever user resizes the window. It may make sense to establish a grid based on maximum horizontal and vertical dimensions when the graph is first loaded.

4. *Mapping attributes to actions.* In order to make animations more accessible to visually impaired users, there should be a mechanism that, under user control, specifies how Boolean attributes such as marking or highlighting are "displayed". Currently, the thickness of highlighted node borders and edges can be controlled in the Preferences panel. A more sophisticated mapping mechanism that incorporates sound as well as visuals is needed. The ultimate approach would allow mappings for arbitrary attributes defined by user or programmer.

5. *Mouse-based selection during animation.* Currently, an animation can ask the user for the id of a node or the id's of endpoints of an edge. It would be nice if the user could click on the node or edge instead.

6. *Inflection points on edges.* For animation of automata it's important to have curved edges if there is a transition going from state q to state r and another from r to q; other applications may need this as well. A single inflection point, carefully placed, and present only if there are parallel edges, could accomplish this.

7. *More preferences.* Font sizes for labels and weights, thickness for colors distinct from highlights, distinct highlight colors for Galant types versus functions.

8. *Selection of multiple nodes and/or edges.* It might prove useful to move a collection of nodes instead of just a single node or give a collection of nodes or edges the same color, label or weight. This is mostly for edit mode.

9. *Skip to end of algorithm.* Allow user to skip to the end of algorithm execution by pressing a single button/key. This is useful for algorithms that display the optimum solution at the end.

10. *Input utility for sorting.* Create a function that reads a sequence of numbers from a file (prompts user on command line – file browser would require unnecessary sophistication) and converts it to a set of evenly spaced (horizontal) nodes in the current window with the appropriate weights.