Features planned in future releases

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

1. Eventually lists, queues, stacks and priority queues of nodes and edges will all be concrete and all graph, node and edge methods that return containers will return one of these. For example, `getNodes()` will return a `NodeList` instead of a `List<Node>`.

2. *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.

3. *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.

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. *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.

6. *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.