Visual Layout of Graph-Like Models

Tarek Sharbak

MhdTarek.Sharbak@uantwerpen.be

Introduction

- Visual formalisms are used to create models of problems
- Visual formalisms use graphical icons and arrows to represent the entities of the model
- Good models should be readable at a glance
- Good layout techniques are required in visual modeling tools
- Tools that support multi-formalisms should have dynamic layout behavior

Graph Basics

- Models are simply graphs
- Different formalisms add different constraints on these graphs
- Graphs can be of different types, graphs, digraphs, Bipartite-graph, etc…
- Graphs might be constrained by number of indegrees, outdegrees, or number of vertices allowed

Visual Aesthetics

- Measure the quality of the graph
- Graph Area
  - The minimal area that the graph occupies the better
- Vertex Placement
  - Multiple vertices should not overlap
- Edge Crossings
  - Edge crossings should be minimized
- Edge Bends
  - Edge bends makes following edges harder

Visual Aesthetics (cont.)

- Direction of flow
  - Easy follow of the direction of the graph and easier to identify source and sink
- Edge Length
  - Short edges are obviously easier to follow for humans
- Mental Map
  - The drawing technique should maintain the mental map of the graph
- Vertex Connections
  - Big angles between connected edges, makes them more distinguishable

Graph Drawing Techniques

- A good graph drawing technique is one that optimize between as much visual aesthetics as possible
- Some visual aesthetics contradict with others in principle
- Different techniques are more suitable in different situations

Graph Drawing Techniques (cont.)

- Layered
  - Covers wide range of aesthetics and relatively easy to implement
  - Graphs should be digraphs and acyclic
  - Layer assignment, crossing minimization, and horizontal placement
- Force-directed
  - Based on virtual physics models
  - Simulation of graphs as physical objects, yields a good layout
  - Vertices as molecules and edges as virtual forces

Graph Drawing Techniques (cont.)

- Orthogonal
  - Drawn as a grid where vertices and edges are assigned integer numbers as coordinates
  - Connected by horizontal and vertical lines
  - Optimize a wide range of visual aesthetics
- Linear Constraints
  - Commonly used to layout windows in user interfaces
  - Provide a declarative approach to layout
  - A complex linear solver evaluates the constraints

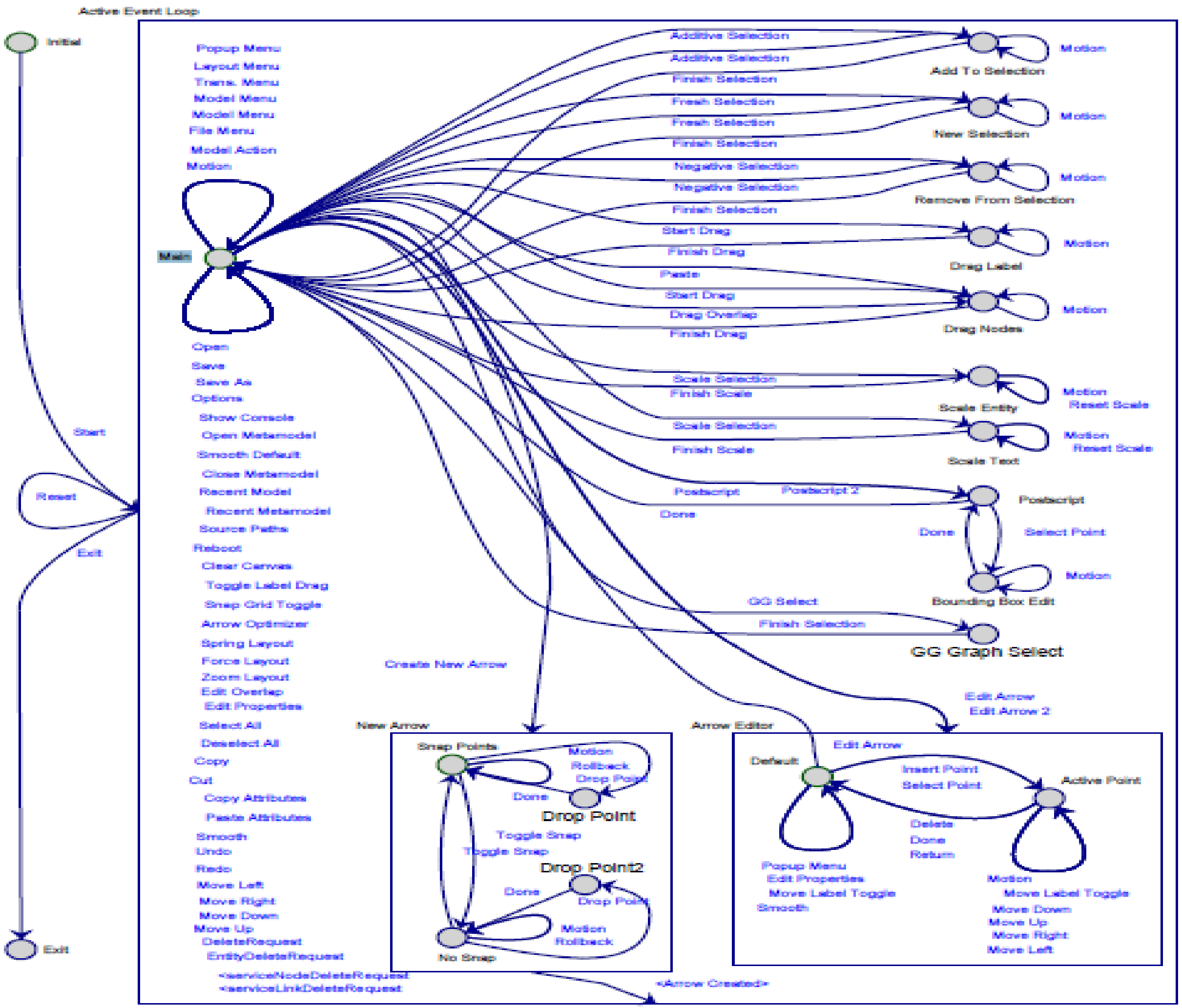
Graph Drawing Implementations AToM^3

- Models are as useful as how readable they are
- AToM^3 is a tool that supports multi formalisms
- Highly extensible
- Drawing algorithms in AToM^3 work through an abstraction level
- Isolate the layout techniques from the internal data structure of AToM3
- Implemented layout techniques
  - Layered.
  - Spring-embedder (Force-directed).
  - Force-transfer.
  - Tree-like and circle.
  - Linear Constraints.

Formalism-Specific UI and Layout Behavior

- Support for multiple formalisms requires more robust and dynamic layout behavior algorithms
- The use of a generic user-interface behavioral model in statecharts
- Can be extended by formalism-specific layout behavior
- The reactive behavior defines how a certain model reacts to a sequence of input events, like mouse or keyboard clicks

Example of layout behavior in statecharts



Formalism-Specific UI and Layout Behavior (cont.)

- Generic UI Behavior
  - Created using statecharts
  - Easily modifiable
  - Isolated from other specific layout behavior models
- Formalism-specific Behavior
  - Extending the layout behavior of the part that require special handling for the formalism
  - Formalisms use virtual entities to define the scope of the formalism
  - Locks can lock the event loop and effectively direct all input to the specific layout behavior
- Pre/post UI observers
  - Common with delete and select events
  - Observers only observe and their sole purpose is to direct the input to the correct layout, they do not handle the input or conflict may occur

Reference

- This work is based on the following thesis,
- Dubé, Denis. "Graph Layout for Domain-Specific Modeling." (2006): 107.

Work to be done

- Implement a formalism-specific layout behavior for the RPG formalism
- Model the behavior using statecharts
- Consider hierarchy models for scenes with zoom in, zoom out functionality
- Moving all of the connected tiles when after the user moves one