

# Information Visualization

## Exercises

Manuel Huber, Simon Nestler, Michael Schlegel

Lesson III  
Graphs and Trees

## Lesson III

### Graphs and trees

#### Exercise I

Drawing graphs with prefuse

#### Exercise II

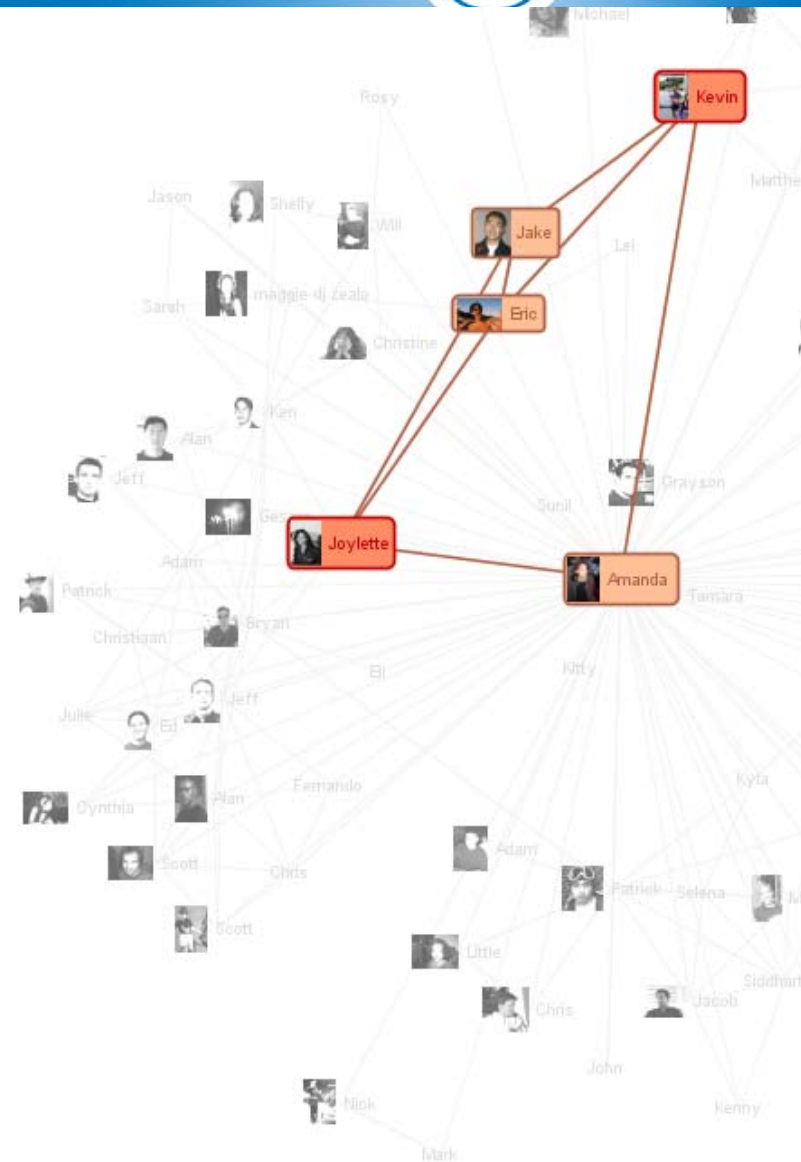
Layouting trees with prefuse

# Exercise I

## Drawing graphs with prefuse

### Nodes

- Tuple sub-interface that represents a node in a graph or tree structure.
- This interface supports both graph and tree methods, tree methods invoked on a node in a general graph typically default to operations on the graph's generated spanning tree.

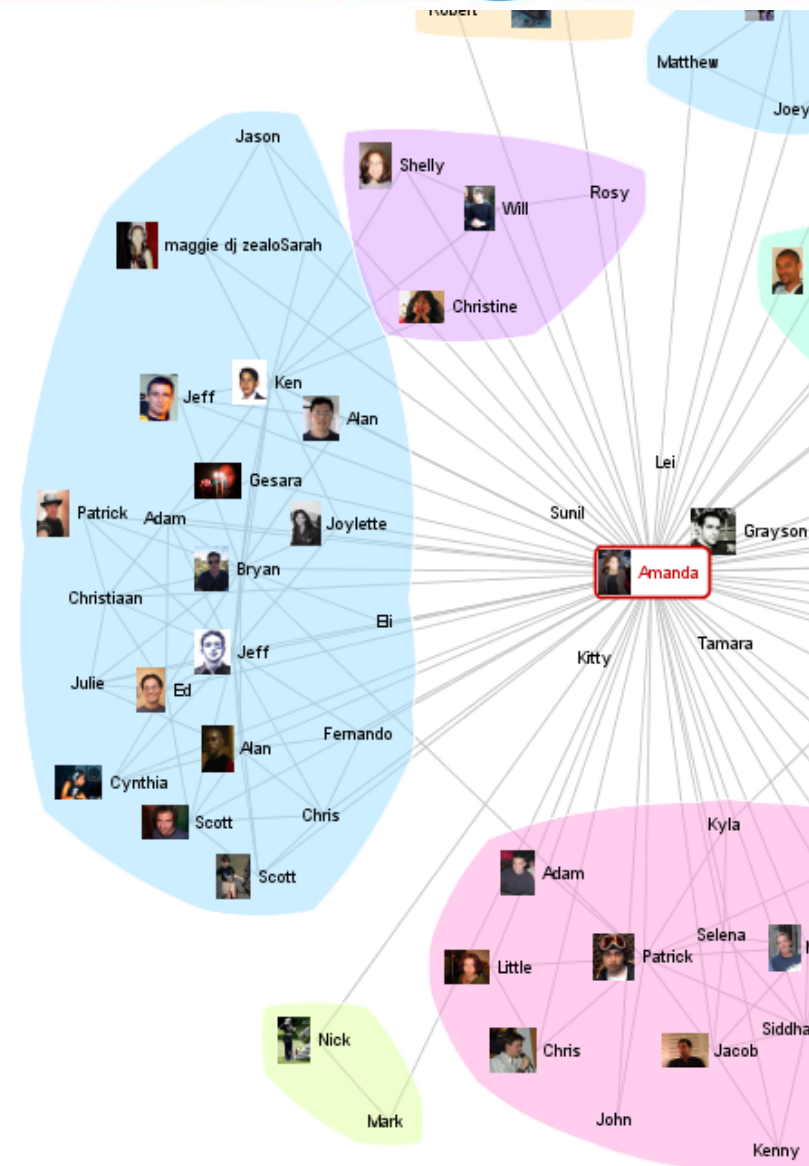


# Exercise I

## Drawing graphs with prefuse

### Edges:

- Tuple sub-interface that represents an edge in a graph structure.
- Each edge has both a source node and a target node.
- For directed edges, this distinction indicates the directionality of the edge.
- For undirected edges this distinction merely reflects the underlying storage of the nodes.



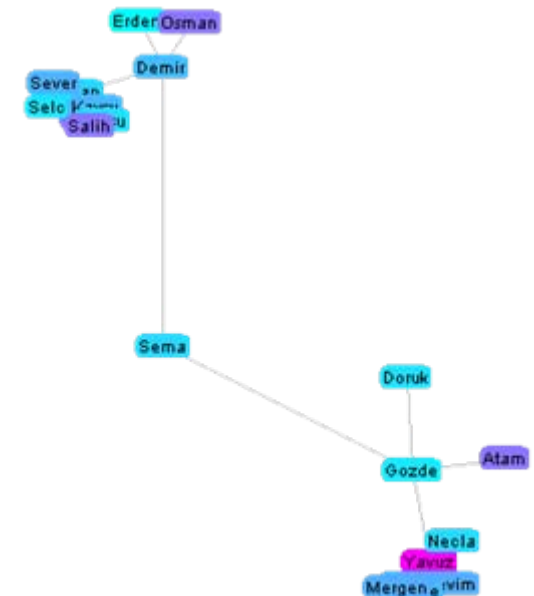
# Exercise II

## Layouting trees with prefuse

### BalloonTreeLayout

- Layout that computes a circular balloon-tree layout of a tree. This layout places children nodes radially around their parents, and is equivalent to a top-down flattened view of a ConeTree.

*(The Cone Tree is an attempt to fit a wide tree into a conventional display. This is accomplished by arranging all nodes at a level in a circular fashion. Thus, each level forms a cone. The 2D projection of such a tree is called a Cone Tree. When a node is selected, it moves to the front in a smooth animation taking about one second. This helps users to preserve the context during a transition between two views.)*

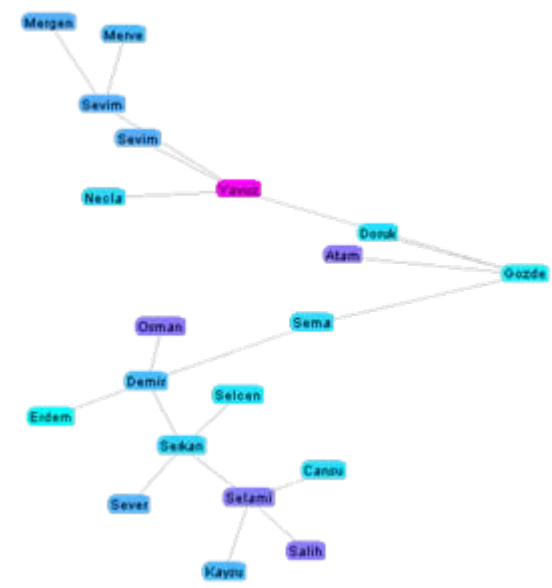
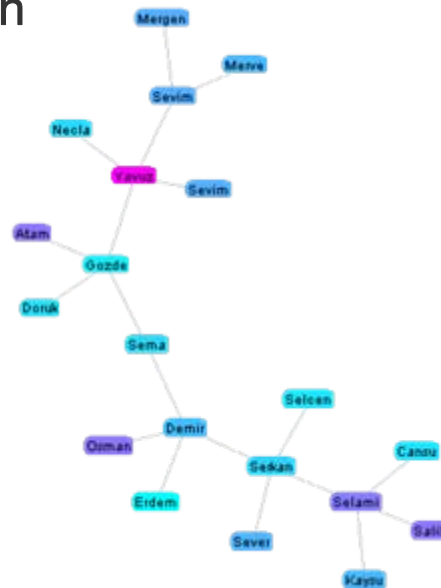


# Exercise II

## Layouting trees with prefuse

### ForceDirectedLayout

- Layout that positions graph items based on a physics simulation of forces
- By default, nodes repel each other, edges act as springs
- This algorithm can be run for multiple iterations



# Exercise II

## Layouting trees with prefuse

### NodeLinkTreeLayout

- Layout that places nodes using a tidy layout of a node-link tree diagram
- Algorithm lays out a rooted tree such that each depth level of the tree is on a shared line
- Orientation of the tree can be set such that the tree goes left-to-right (default), right-to-left, top-to-bottom, or bottom-to-top.

