Relational Data

Learning Objectives

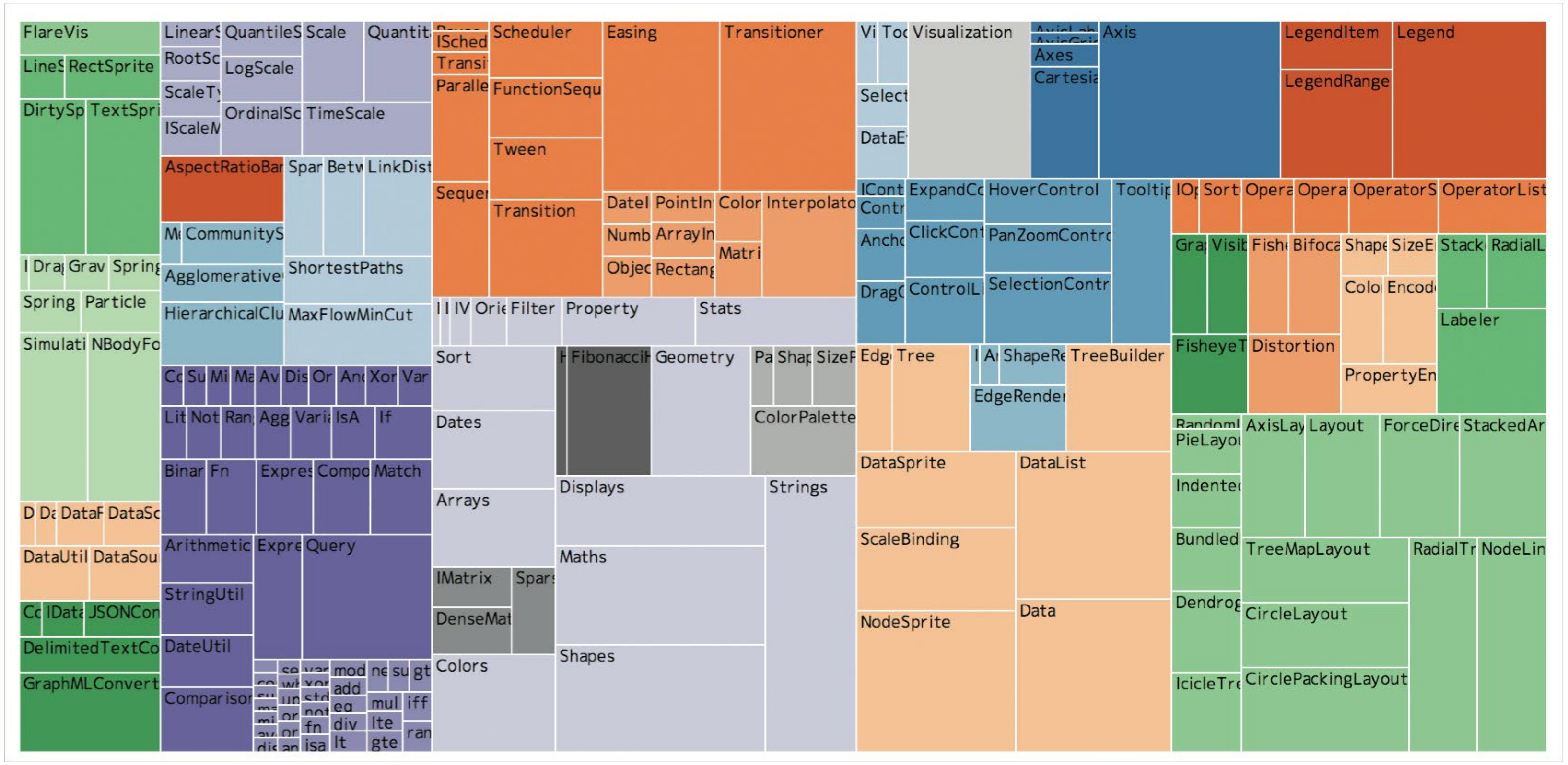
- Methods to Visualising relationships
- Relationships are often described in the form of Graph
- Graphs are often visualised to show
  - Hierarchical structure
  - Relationships

Visualising Hierarchical Structure

- Many methods show hierarchy in the form of grouping

Visualising Hierarchical Structure – Method I

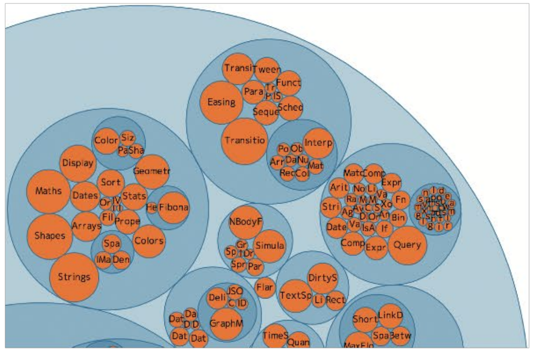
- Tree Map
  - Line /Area : size/ location/ nested



※『d3.js garally』(http://bl.ocks.org/mbostock/4063582)

Visualising Hierarchical Structure – Method II

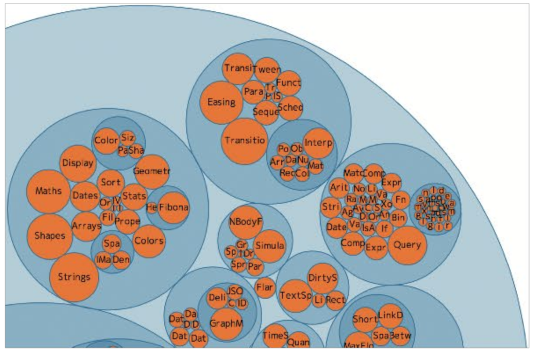
- Partition Diagram
  - Line/ Area : describe data with size/ nested relationships



※いずれも『d3.js garally』(https://github.com/mbostock/d3/wiki/Gallery)

Visualising Hierarchical Structure – Method III

- Sankey Chart
  - Point/ Line/ Area : describe nested information



※いずれも『d3.js garally』(https://github.com/mbostock/d3/wiki/Gallery)

Relationships

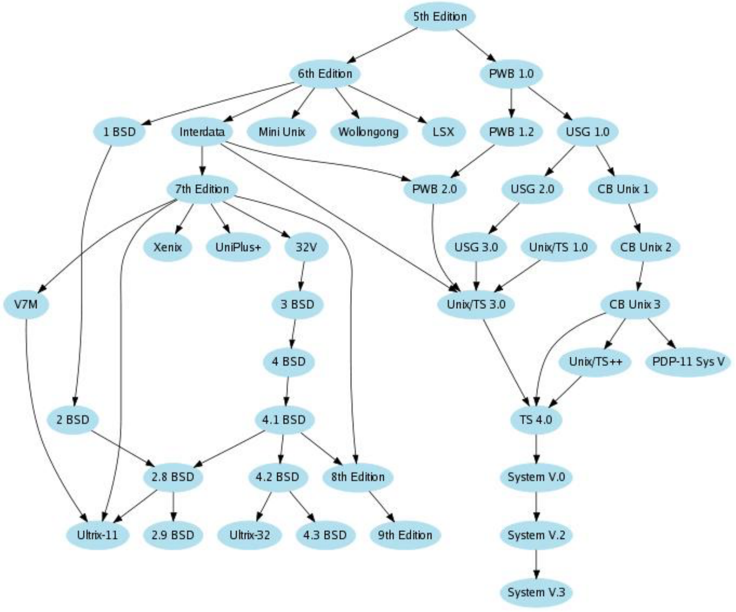
- Entities and relationships between them
- Graph - mathematical and abstract concept
- flowchart, organisational chart, UML, etc – examples

- Any other examples of data containing relationships?

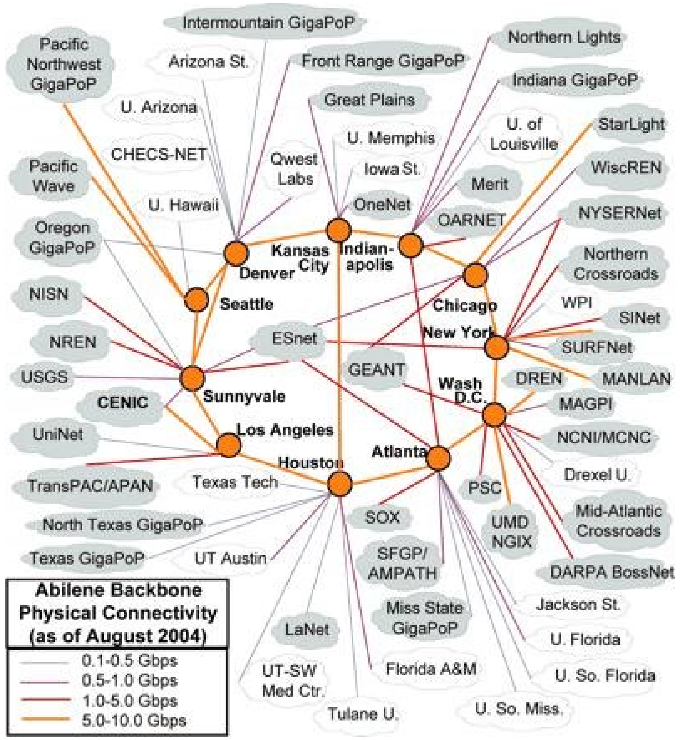
Visualising Relationships

- Many methods show relationships in the form of a graph

Visualising Relationships – Examples (I)

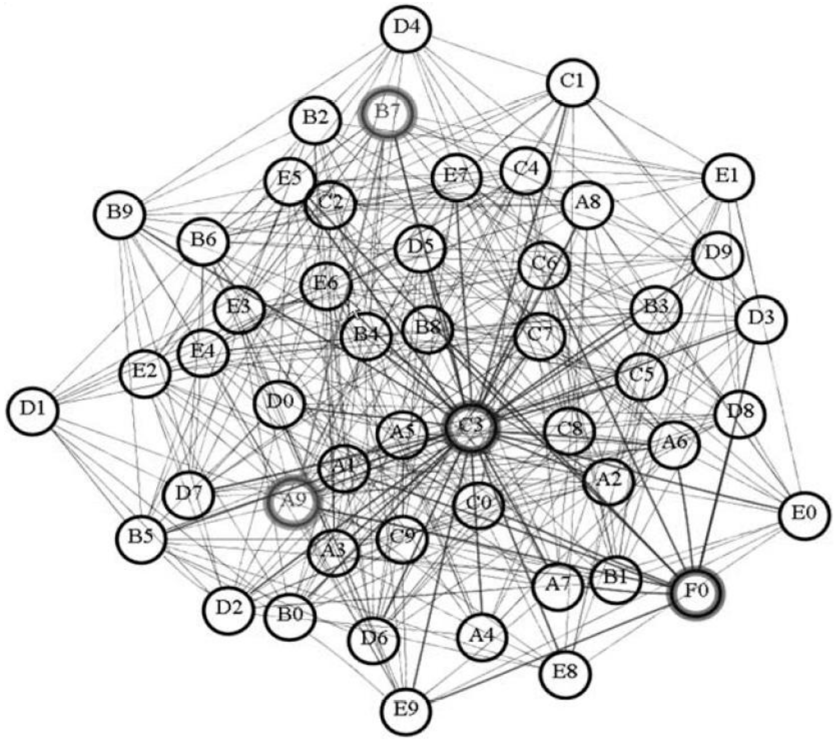


Ellson, J. and Gansner, E., "Graphviz - graph visualization software," 2004, URL: http://www.graphviz.org.

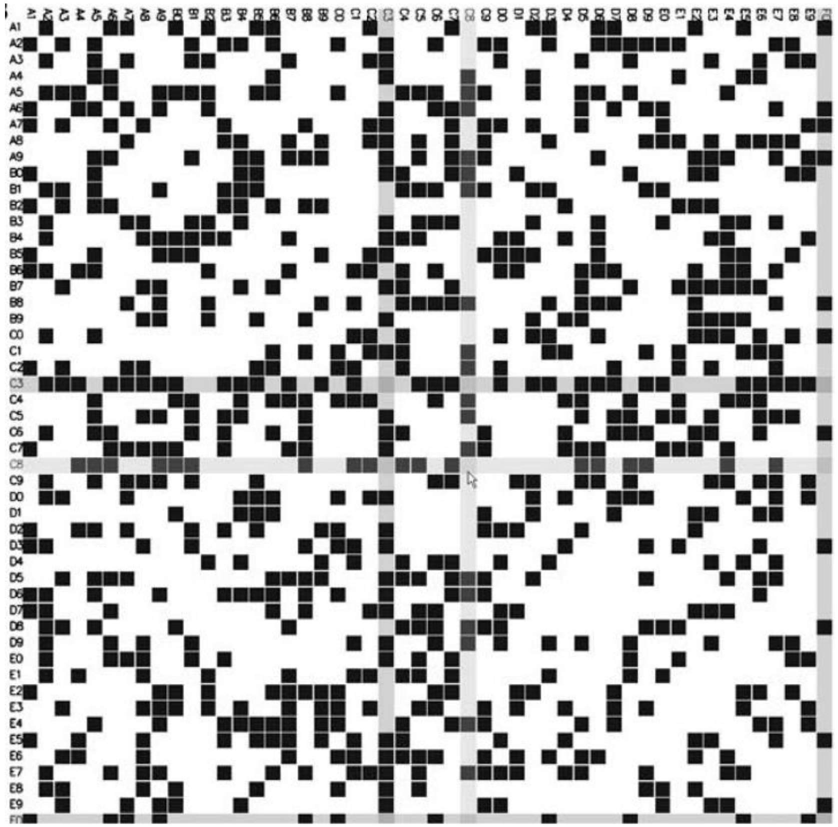


Doyle, J. C., Alderson, D. L., Li, L., Low, S., Roughan, M., Shalunov, S., Tanaka, R., and Willinger, W., "The "robust yet fragile" nature of the internet," Proceedings of the National Academy of Science of the United States of America, Vol. 102, 2005, pp. 14497 – 14502.

Visualising Relationships – Examples (II)



(a) Node-Link Diagram



(b) The Matrix View

Ghoniem,M.,Fekete,J.-D.,andCastagliola,P.,“Onthereadabilityofgraphsusingnode-link and matrix-based representations: A controlled experiment and statistical analysis,” Information Visualization, Vol. 4, 2005, pp. 114 – 135.

Graph Drawing

Components of a Graph

- V, E
  - V: Nodes
  - E : Edges (binary relationships between nodes)

W.T. Tutte (1917 - 2002)

- Code breaker at Bletchley park
- Pioneer of matroid theory
- Pioneer of graph theory
- Inventor of the first graph drawing algorithm
- William T. Tutte, “How to draw a graph.”, Proc. London Math. Society, 13(52): 743 - 768, 1963

How to construct a graph? – Part I

First of all, get a data

Example

- Nodes: Alice, Andrea, Annie, Amelia, Bob, Brian, Bernard, Boyle
- Edges:

Bob is connected to Alice
Bob is connected to Andrea
Bob is connected to America
Brian is connected to Alice
Brian is connected to Andrea
Brian is connected to Amelia

Boyle is connected to Alice
Boyle is connected to Andrea
Boyle is connected to Annie
Bernhard is connected to Alice
Bernhard is connected to Andrea
Bernhard is connected to Annie

How to construct a graph? – Part II

Graph

- Nodes and Edges

A picture of a graph, that maps

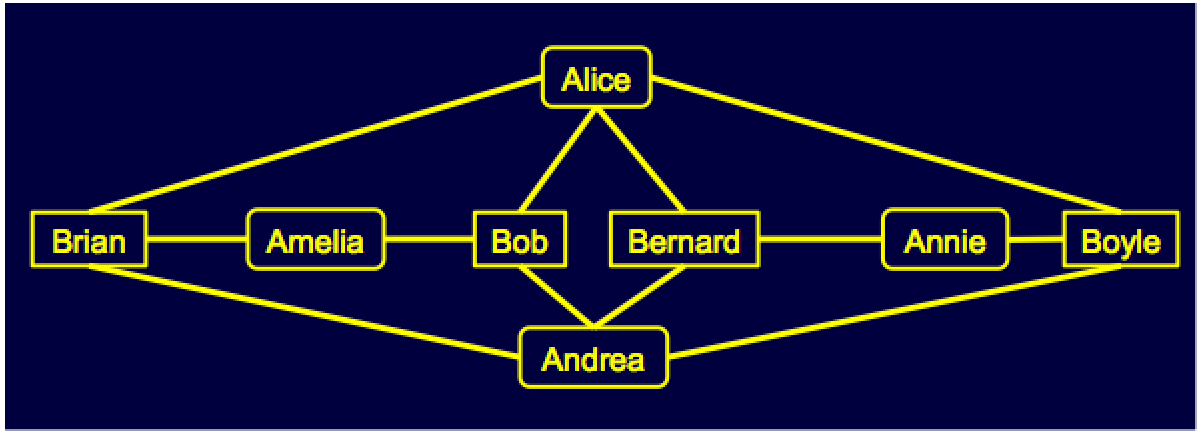
- a location for each node, and
- a curve/line to each edge

Example I

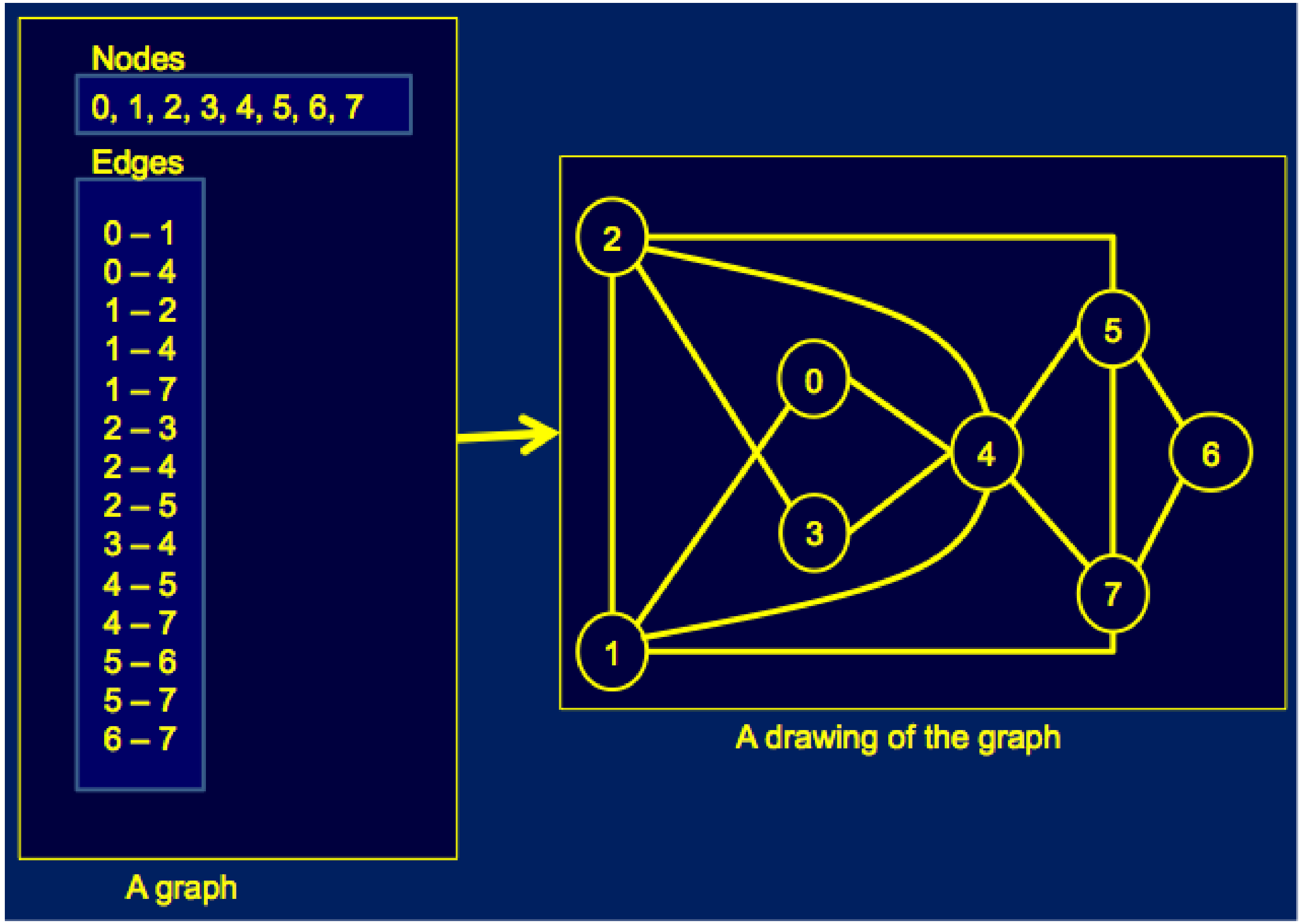
- Nodes : Alice, Andrea, Annie, Amelia, Bob, Brian, Bernard, Boyle
- Edges :

Bob is connected to Alice
Bob is connected to Andrea
Bob is connected to America
Brian is connected to Alice
Brian is connected to Andrea
Brian is connected to Amelia

Boyle is connected to Alice
Boyle is connected to Andrea
Boyle is connected to Annie
Bernhard is connected to Alice
Bernhard is connected to Andrea
Bernhard is connected to Annie



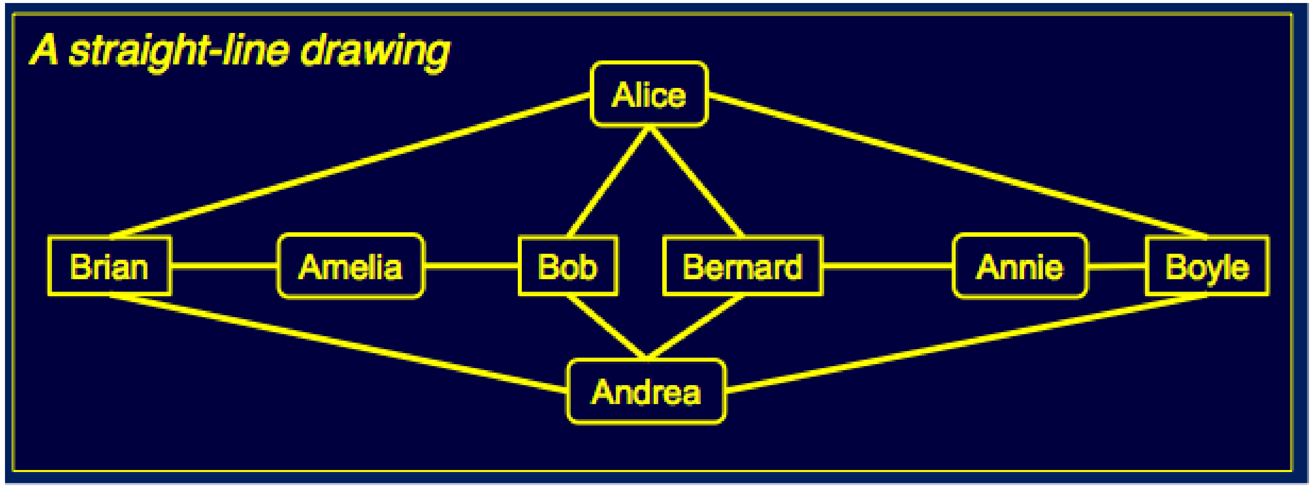
Example II

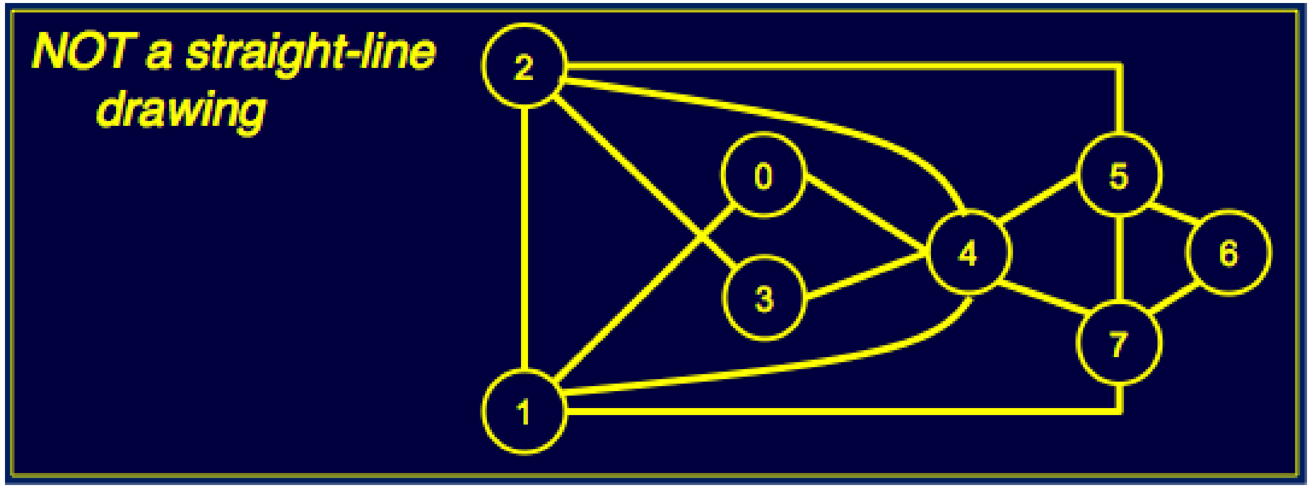


Nodes
0, 1, 2, 3, 4, 5, 6, 7
Edges
0 – 1
0 – 4
1 – 2
1 – 4
1 – 7
2 – 3
2 – 4
2 – 5
3 – 4
4 – 5
4 – 7
5 – 6
5 – 7
6 – 7
A graph
2
0
5
4
6
3
7
1
A drawing of the graph

Straight-line drawing

- A graph drawing is a straight-line drawing if every edge is a straight line segment





Appearance of Graph

Planar

- A graph is planar if it can be drawn without edge crossings

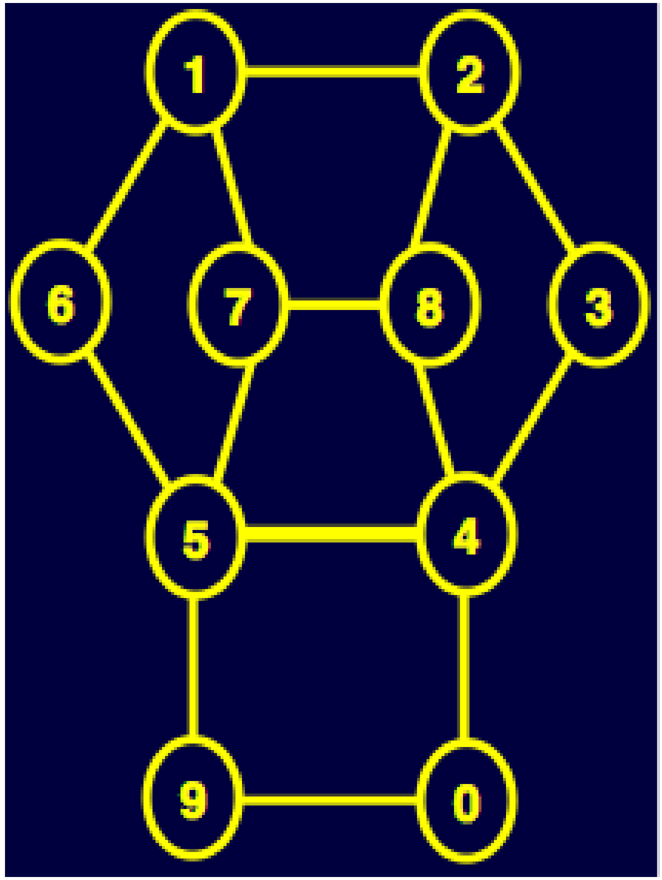
Example of Planar Graph

Nodes:

- 0,1,2,3,4,5,6,7,8,9

Edges

- 0 – 4
- 0 – 9
- 1 – 2
- 1 – 6
- 1 – 7
- 2 – 3
- 2 – 8
- 3 – 4
- 4 – 5
- 4 – 8
- 5 – 6
- 5 – 7
- 7 – 8



Non-planar

- There is at least one edge crossing in a graph

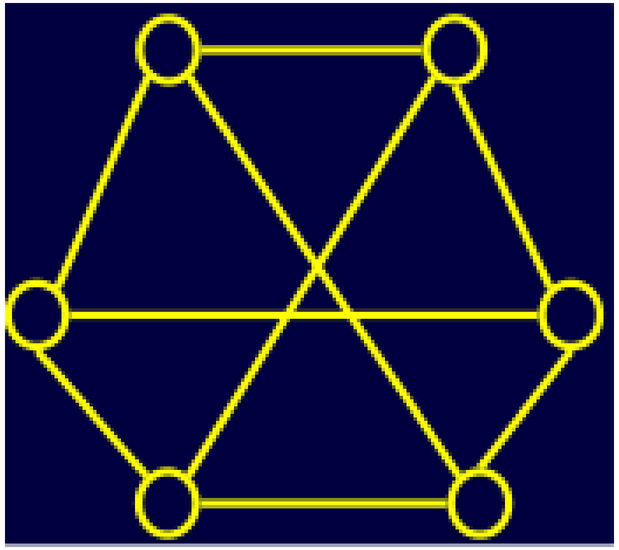
Example of Non-Planar Graph

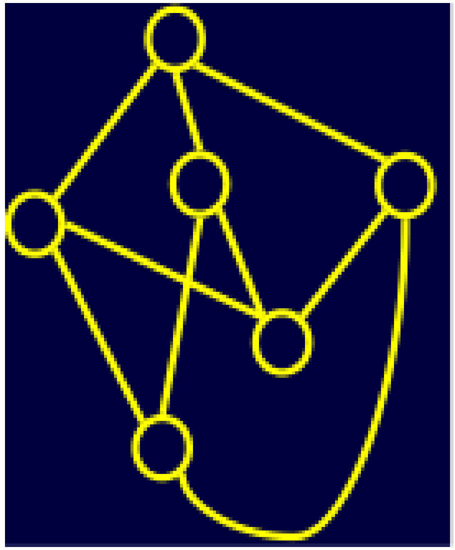
Nodes:

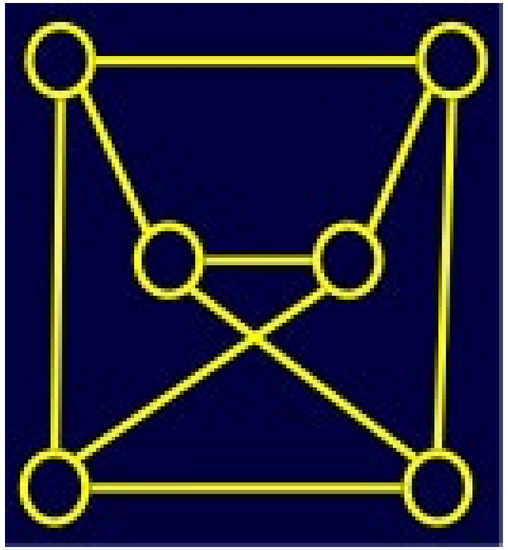
- 0,1,2,3,4,5

Edges

- 0 – 1
- 0 – 3
- 0 – 5
- 1 – 2
- 1 – 4
- 2 – 3
- 2 – 5
- 3 – 4
- 4 – 5



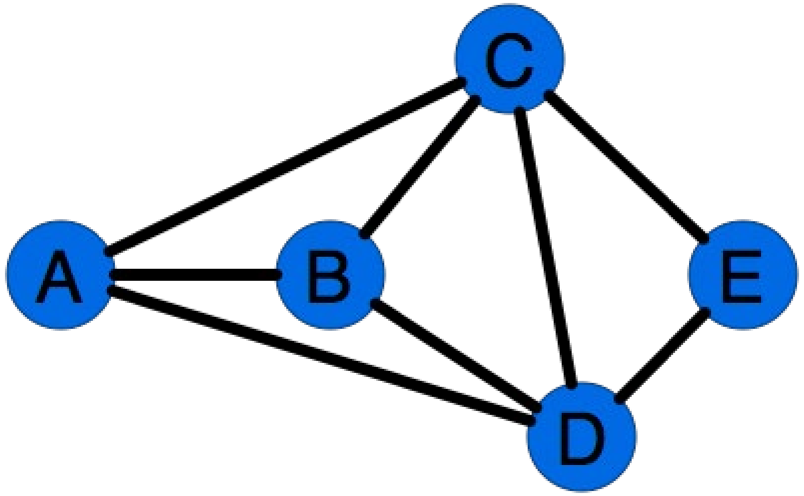




“Good” Graph Drawing?

- easy to understand
- easy to remember
- beautiful…

A - B, C, D
B - A, C, D
C - A, B, D, E
D - A, B, C, E
E - C, D

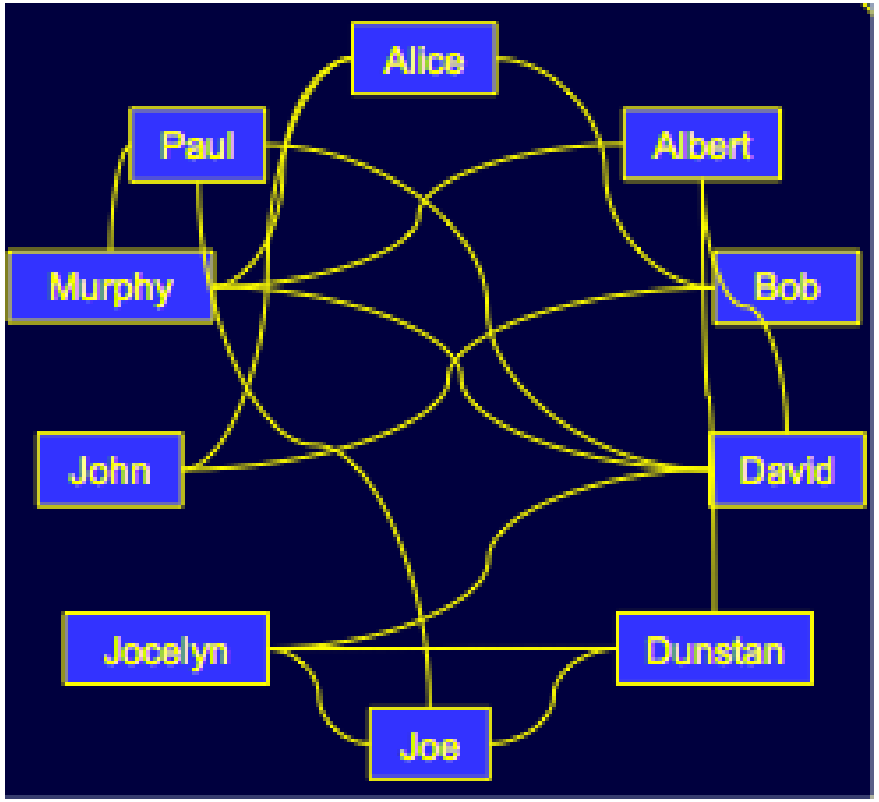


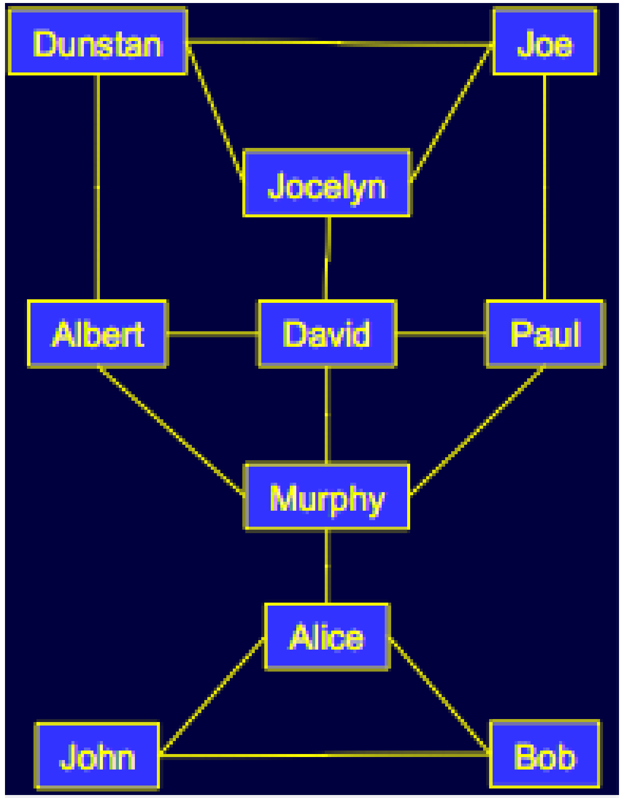
Appearance of Edges

~1979 Intuition

Sugiyama et al. 1979, Batini et al. 1982, etc

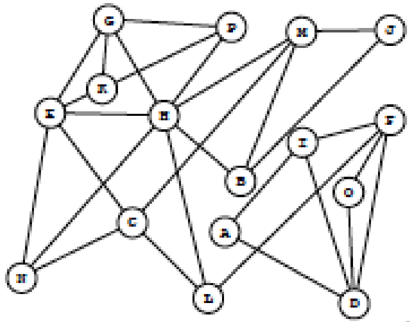
- Planar straight-line drawings make good pictures

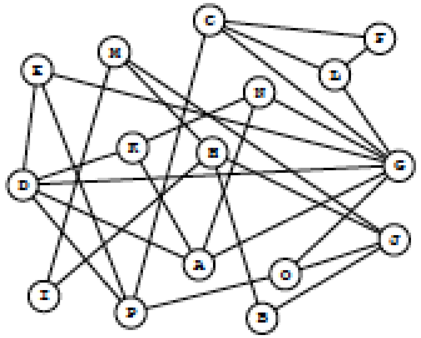


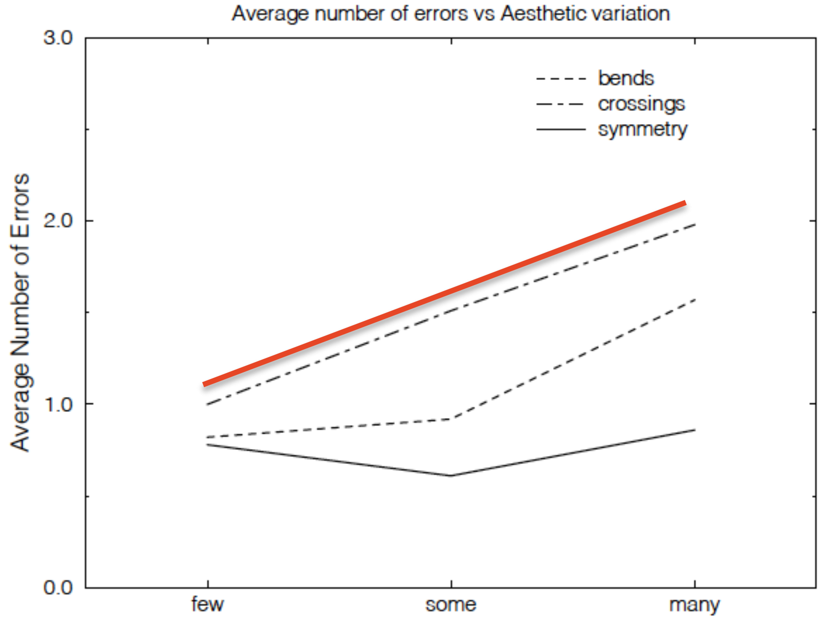


Findings of Purchase et al. 1997

- Significant correlation between edge crossings and human understanding
  - More edge crossings means more human errors in understanding







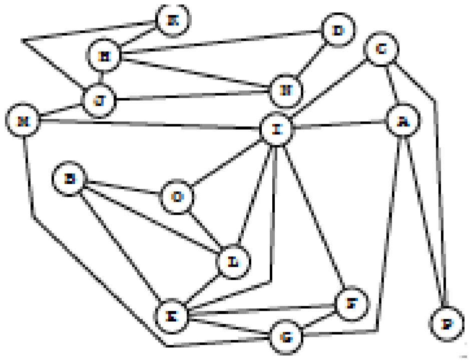
bends	6	18	30
crossings	6	24	42
symmetry	4.57	25.7	51.0

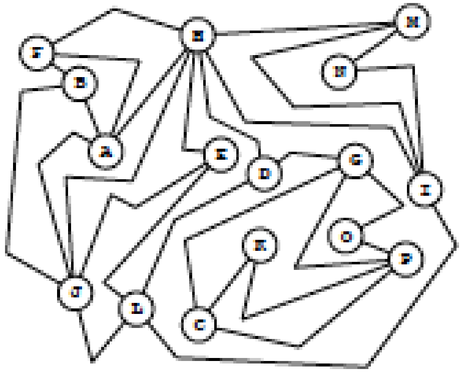
Aesthetic Variation

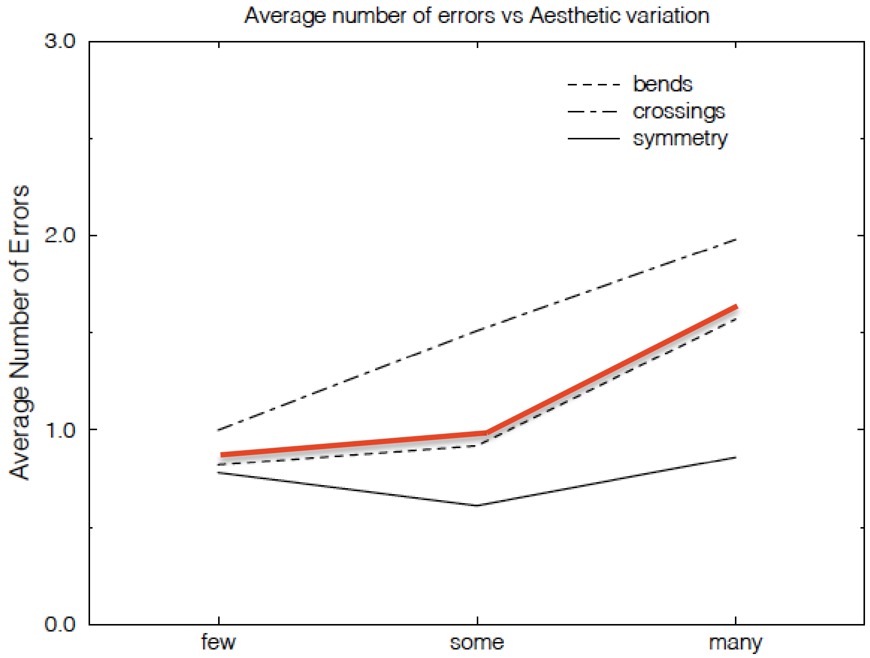
Fig. 3. Results for the dense graph

Findings of Purchase et al. 1997 (cont.)

- Significant correlation between straightness of edges and human understanding
- More bends means more human errors in understanding







bends	6	18	30
crossings	6	24	42
symmetry	4.57	25.7	51.0

Aesthetic Variation

Fig. 3. Results for the dense graph

Aesthetic Rules in Graph Drawing

Rules in Graph Drawing

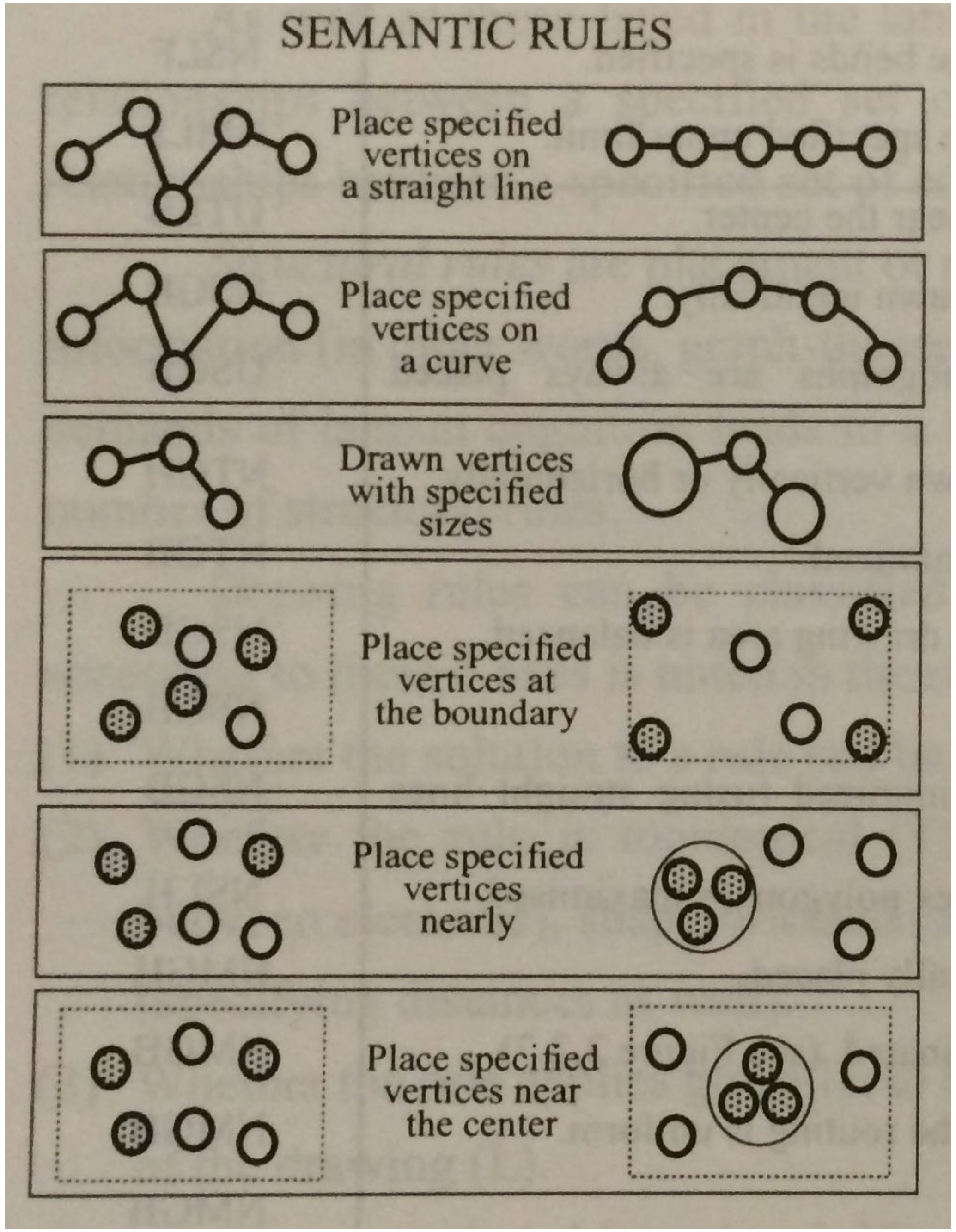
- Semantic rules : derived from the meaning of vertices and edges,
  - importance of a vertex, strength of relationships
- Structural rules : derived from graph-theoretic features

Classification Criteria of Rules

- Whether the solution to a rule can be obtained uniquely(U) or not (N)
- Whether the rule is
    - topological (T) (specifying only the placement relationship between elements),
    - shape-oriented (S) (specifying the direction also) or
    - metric (M) (specifying distances as well)
- Whether the rule applies globally, to the whole drawing (G) or locally, only to a part of the drawing (L)
- Whether the rule is hierarchical (H), or flat (F), or both (B)

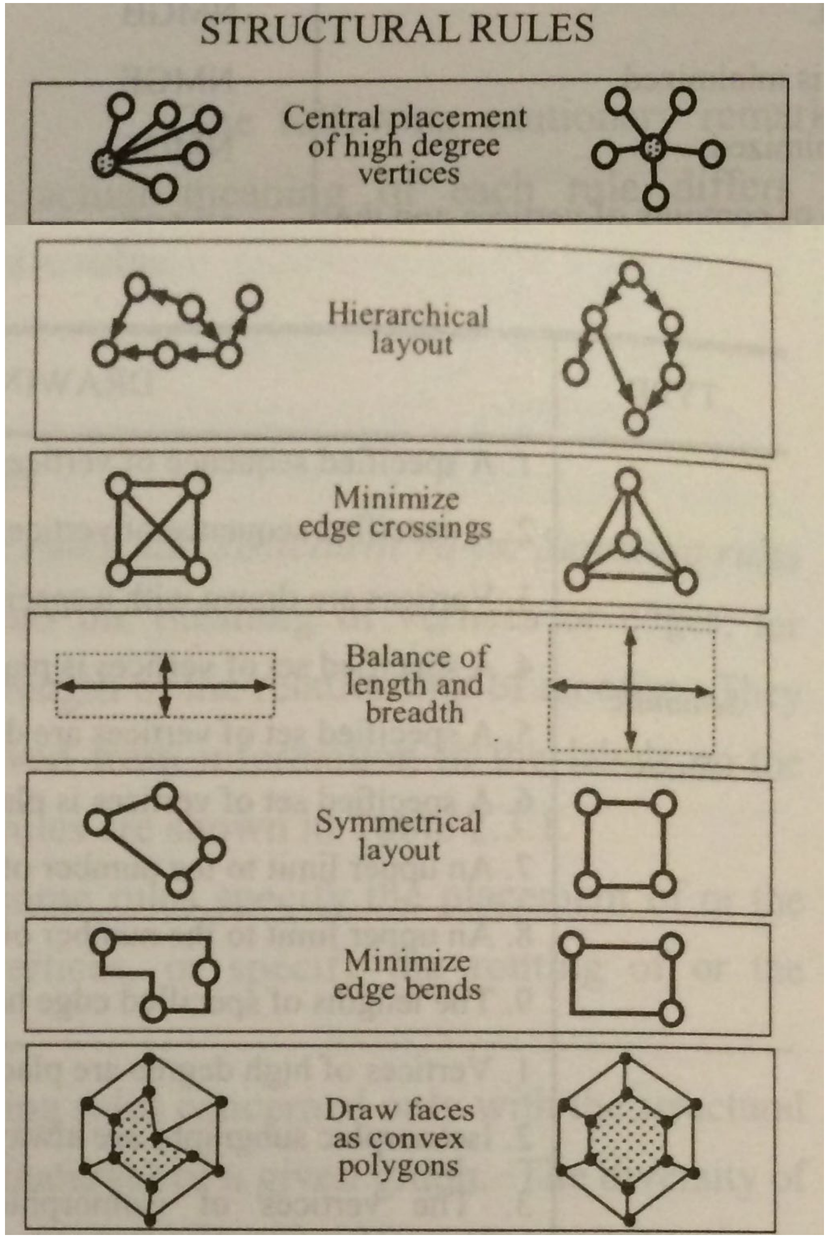
Semantic Rules

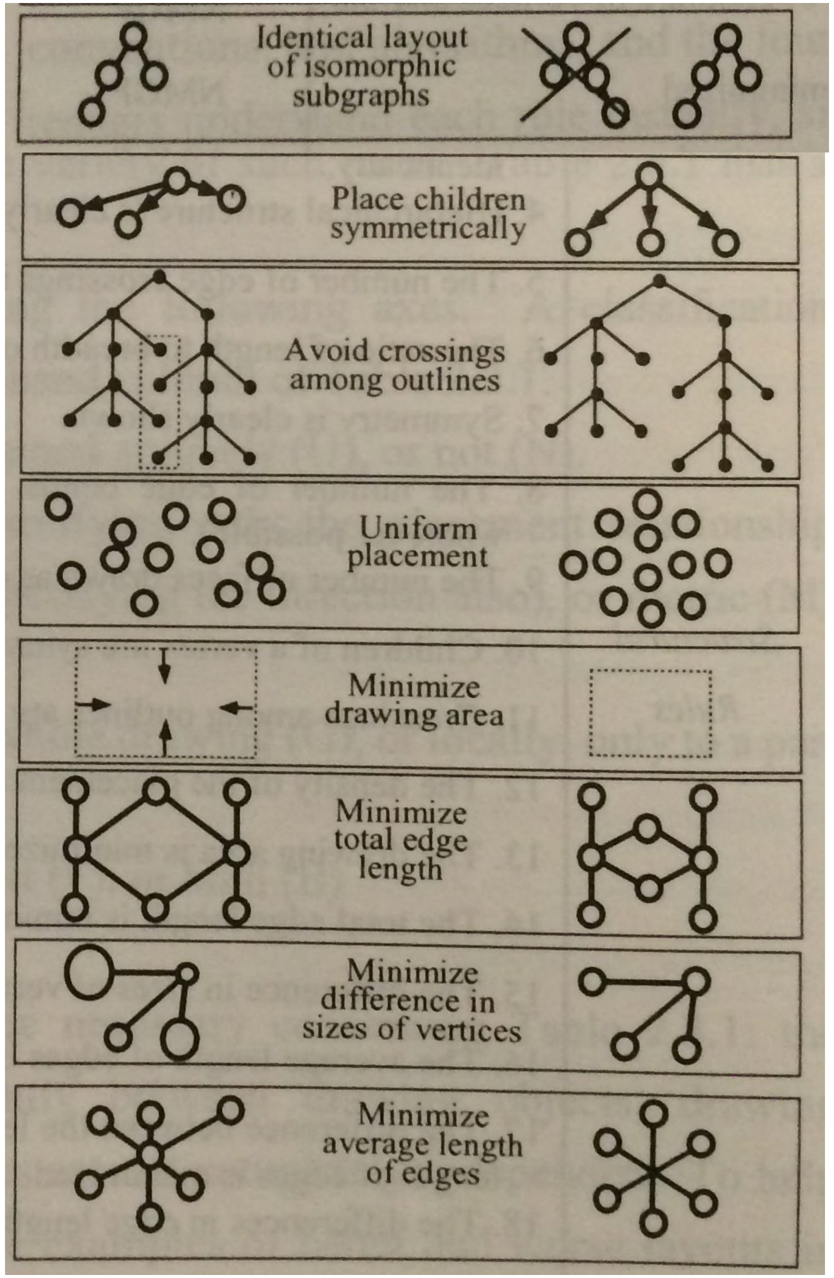
Drawing rules	Classification
A specified sequence of vertices is placed on a straight line.	USLB
A specified sequence of vertices is placed on a specified curve.	USLB
Vertices are drawn with a specific size.	UMLB
A specified set of vertices is placed at the boundary of the drawing.	NTLB
A specified set of vertices are drawn near to each other.	NTLB
A specified set of vertices is placed near the center.	NTLB
An upper limit to the number of edge crossings is specified.	NTLB
An upper limit to the number of edge bends is specified.	NSLF
The lengths of specified edge have a specified upper limit.	NMLF


SEMANTIC RULES
Place specified vertices on a straight line
Place specified vertices on a curve
Drawn vertices with specified sizes
Place specified vertices at the boundary
Place specified vertices nearly
Place specified vertices near the center

Structural Rules

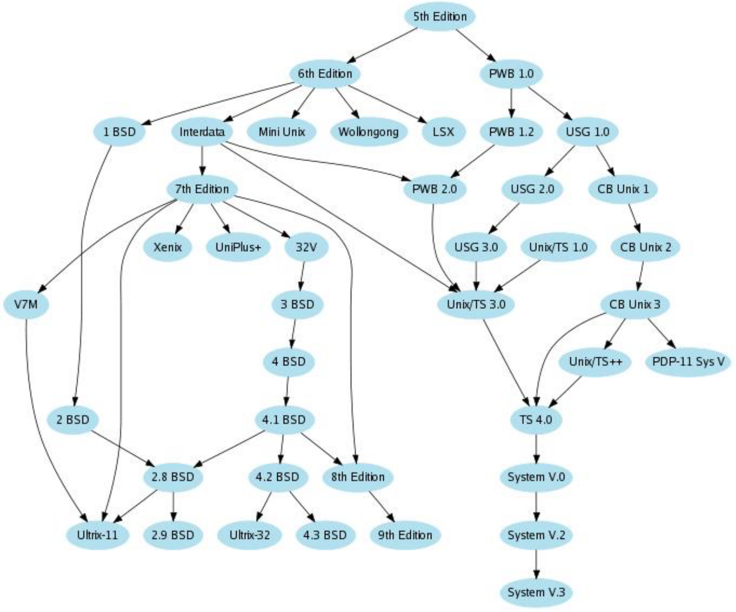
Drawing rules	Classification
Vertices of high degree are placed near the centre.	UTLB
Isomorphic subgraphs are always drawn identically.	USGB
The vertices of isomorphic subgraphs are always placed identically.	USGB
Hierarchical structure is clearly shown vertically or horizontally.	NTGH
The number of edge crossings is minimized.	NTGB
The ratio of length to breadth of the drawing area is balanced.	NSGB
Symmetry is clearly shown.	NSGB
The number of edge bends is minimised (using straight lines wherever possible)	NSGB
The number of faces drawn as convex polygons is maximised.	NSLH
Children of a vertex are summitrically placed.	NMGH
Crossings among outlines are eliminated.	NMGB
The density of the placement and the routing is uniform.	NMGB
The drawing area is minimised.	NMGB
The total edge length is minimised.	NMGB
The difference in sizes of vertices is minimized.	NMGF
The average length of edges is minimized.	NMGF
The difference between the length of contours of vertices and the length of edges is maximised.	NMGF
The differences in edge lengths is minimized.	NMGF
The length of the longest edge is minimized.	NMLF
Vertices on the boundary are placed with uniform density.	NMLF


STRUCTURAL RULES
Central placement of high degree vertices
Hierarchical layout
Minimize edge crossings
Balance of length and breadth
Symmetrical layout
Minimize edge bends
Draw faces as convex polygons

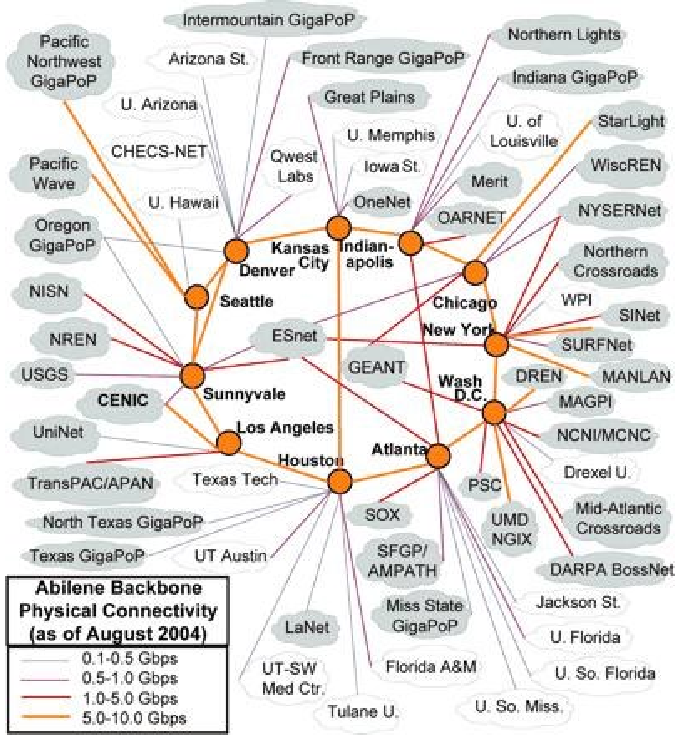

Identical layout of isomorphic subgraphs
Place children symmetrically
Avoid crossings among outlines
Uniform placement
Minimize drawing area
Minimize total edge length
Minimize difference in sizes of vertices
Minimize average length of edges

Graph Layout Algorithms

Different layout algorithms



Ellson, J. and Gansner, E., “Graphviz - graph visualization software,” 2004, URL: http://www.graphviz.org.



Doyle, J. C., Alderson, D. L., Li, L., Low, S., Roughan, M., Shalunov, S., Tanaka, R., and Willinger, W., “The “robust yet fragile” nature of the internet,” Proceedings of the National Academy of Science of the United States of America, Vol. 102, 2005, pp. 14497 – 14502.

Popular Graph Layout Algorithms

- Tutte’s barycentre algorithm
- Forced Directed Layout
  - Eades’ Force Directed Layout

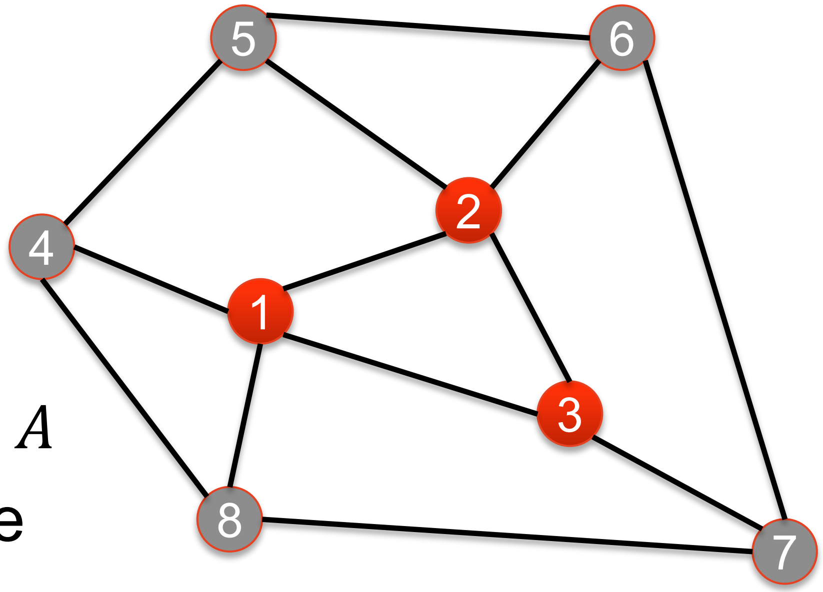
Tutte's barycentre algorithm

Input: A graph $G = (V, E)$

Output: A straight-line drawing p

- Step 1: Choose a subset A of V
- Step 2: Choose a location $p(a)$ for each vertex $a \in A$
- Step 3: For all $u \in V - A$, place u at the barycentre of its graph-theoretic neighbours

$$p(u) = \frac{1}{\deg(u)} \sum_{v \in N(u)} p(v)$$



Tutte's barycentre algorithm (2n equations)

$$p(u) = \frac{1}{\deg(u)} \sum_{v \in N(u)} p(v)$$

$$x(u) = \frac{1}{\deg(u)} \sum_{v \in N(u)} x(v)$$

$$y(u) = \frac{1}{\deg(u)} \sum_{v \in N(u)} y(v)$$

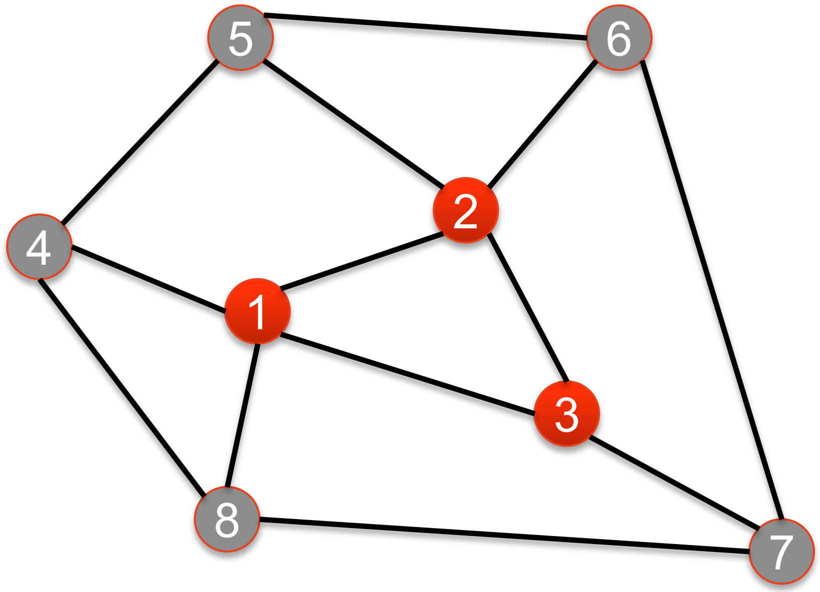
- The Euclidean distance between u and v in the drawing p is:

$$d(u,v) = \sqrt{(x_u - x_v)^2 + (y_u - y_v)^2}$$

- The energy in the edge (u,v) is $d(u,v)^2 = (x_u - x_v)^2 + (y_u - y_v)^2$
- The energy in the drawing p is the sum of the energy in its edges, ie,

$$energy(p) = \sum_{(u,v) \in E} d(u,v)^2 = \sum_{(u,v) \in E} (x_u - x_v)^2 + (y_u - y_v)^2$$

Planar Example



- Step 1: $A = \{4, 5, 6, 7, 8\}$
- Step 2: For all $i = 4, 5, 6, 7, 8$, choose x_i and y_i in some way.
- Step 3: Find x_1, y_1, x_2, y_2, x_3, and y_3 such that:

$$x_1 = \frac{1}{4}(x_2 + x_3 + x_4' + x_8')$$
$$x_2 = \frac{1}{4}(x_1 + x_3 + x_5' + x_6')$$
$$x_3 = \frac{1}{3}(x_1 + x_2 + x_7')$$

and

$$y_1 = \frac{1}{4}(y_2 + y_3 + y_4' + y_8')$$
$$y_2 = \frac{1}{4}(y_1 + y_3 + y_5' + y_6')$$
$$y_3 = \frac{1}{3}(y_1 + y_2 + y_7')$$

Where x_i' and y_i' are the values chosen at Step 2.

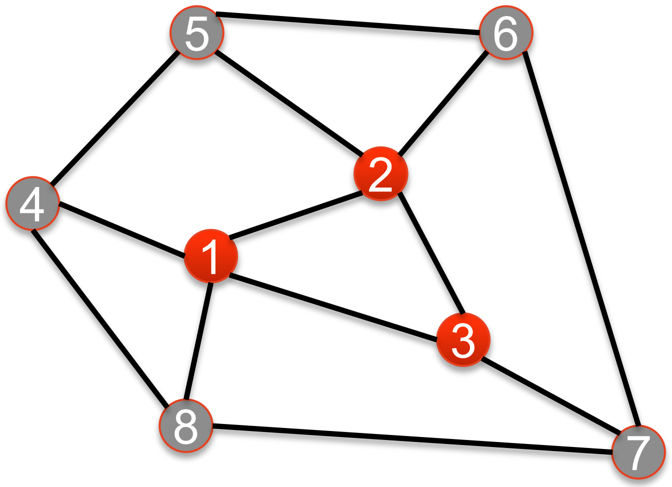
Planar Example (cont.)

- Step 4: Assign constants c_i and d_i to equations we got from Step 3:

$$4x_1 - x_2 - x_3 = x_4' + x_8' = c_1$$
$$-x_1 + 4x_2 - x_3 = x_5' + x_6' = c_2$$
$$-x_1 - x_2 + 3x_3 = x_7' = c_3$$

and

$$4y_1 - y_2 - y_3 = y_4' + y_8' = d_1$$
$$-y_1 + 4y_2 - y_3 = y_5' + y_6' = d_2$$
$$-y_1 - y_2 + 3y_3 = y_7' = d_3$$



- Step 5: Find vectors $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$ and $\mathrm{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$ such that

$$Mx = c$$

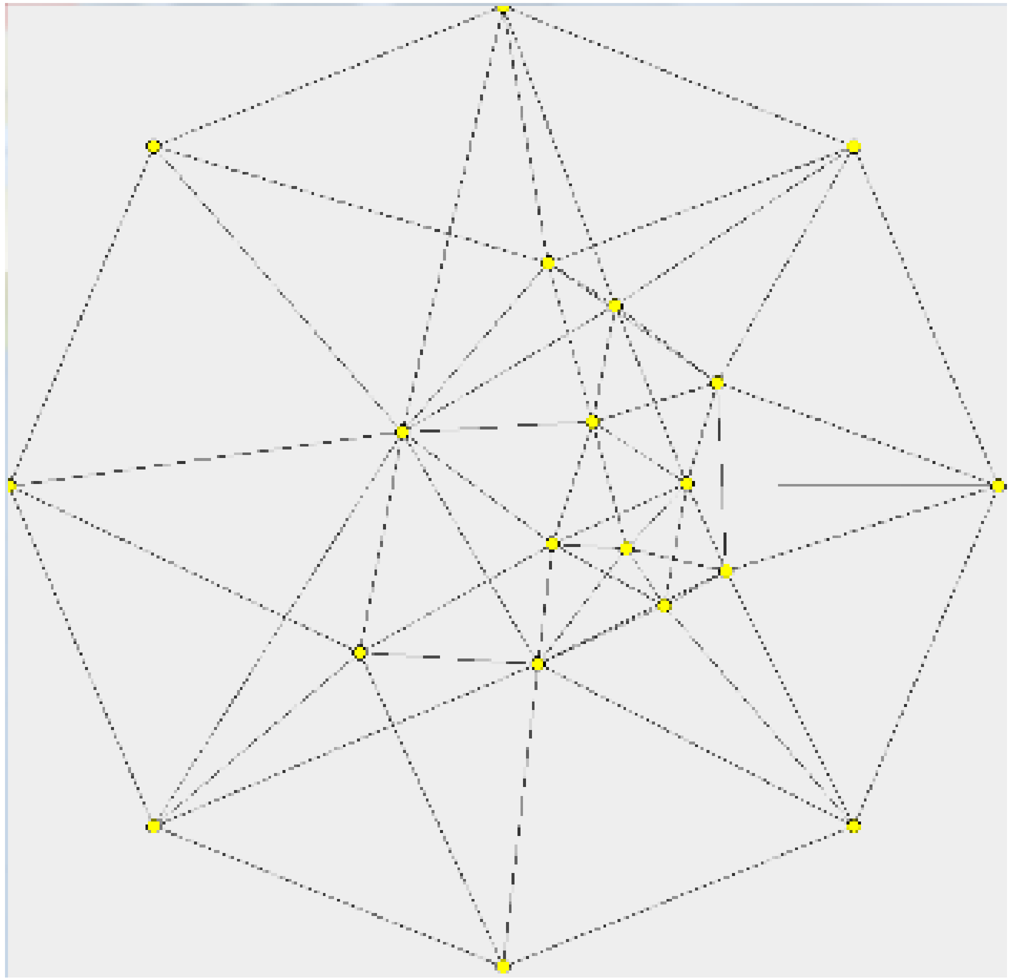
And

$$My = d$$

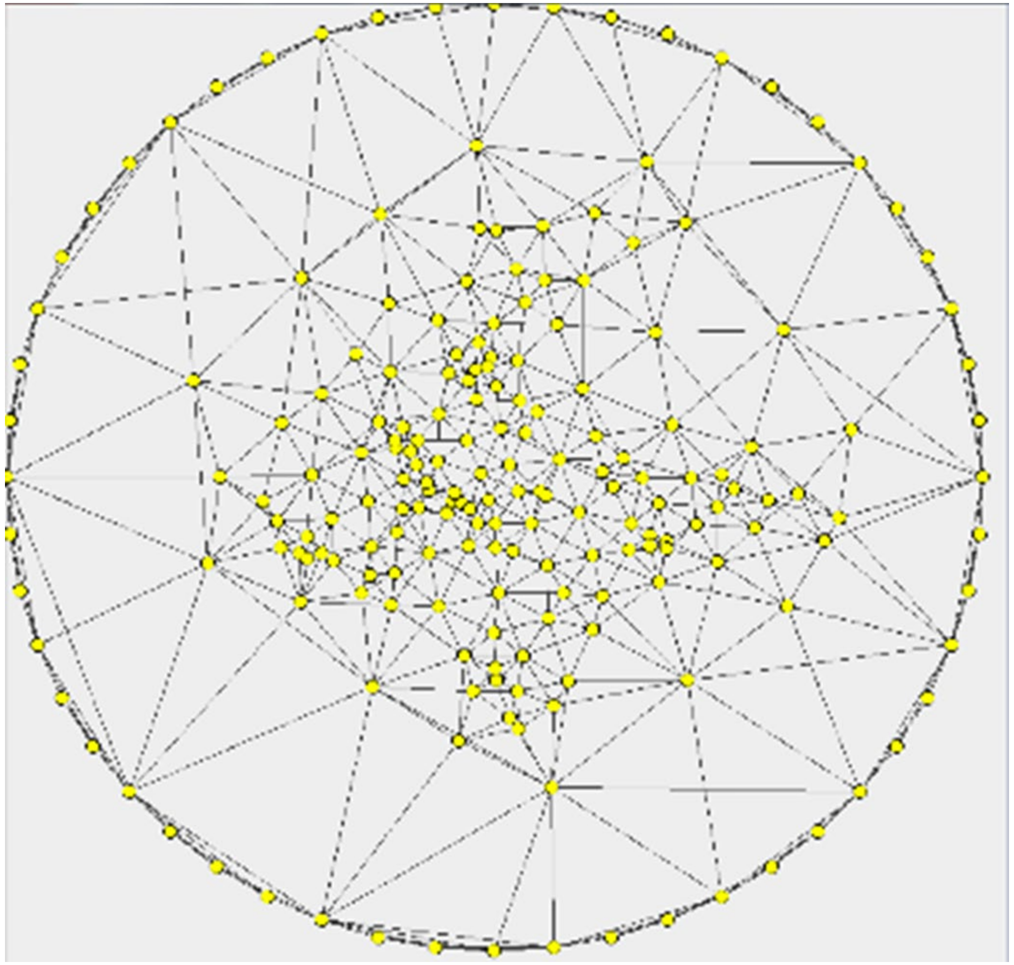
Where

$$M = \begin{bmatrix} 4 & -1 & -1 \\ -1 & 4 & -1 \\ -1 & -1 & 3 \end{bmatrix}$$

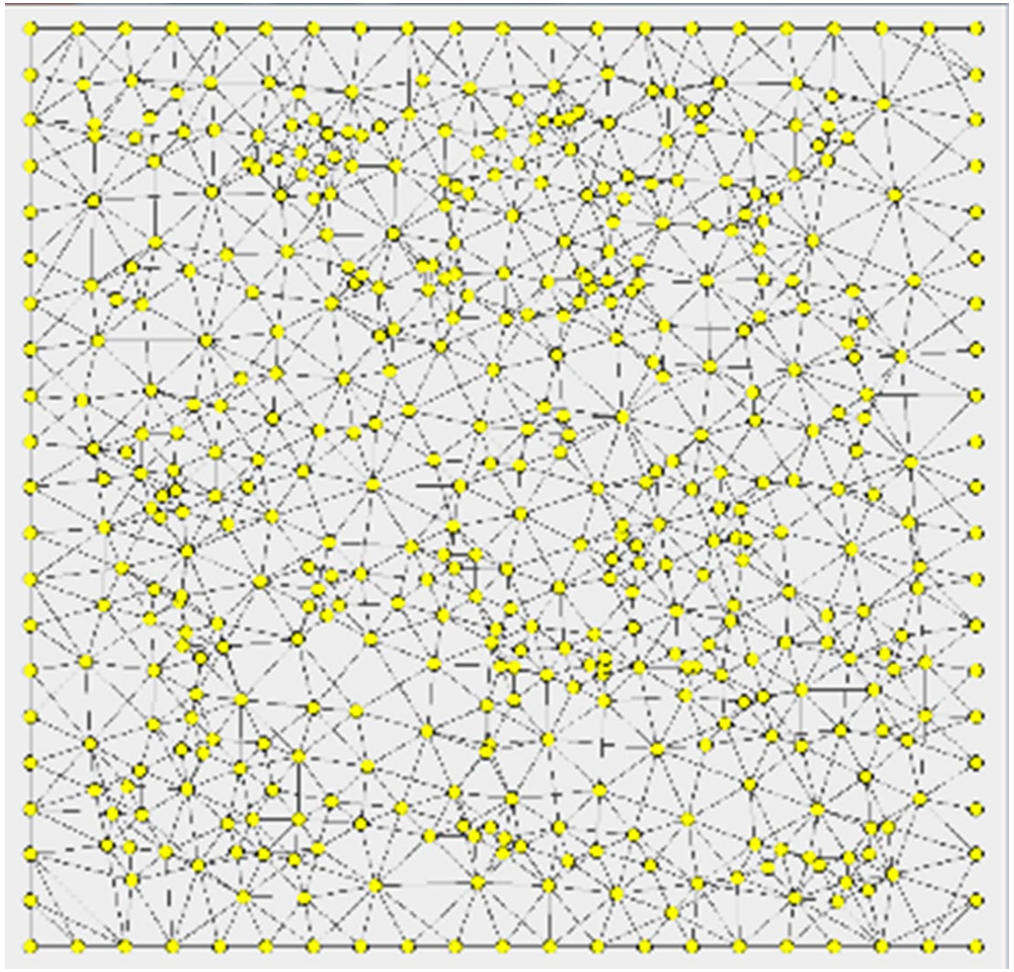
Non-planar Example (20 vertices)



Non-planar Example (200 vertices)



Non-planar Example (500 vertices)

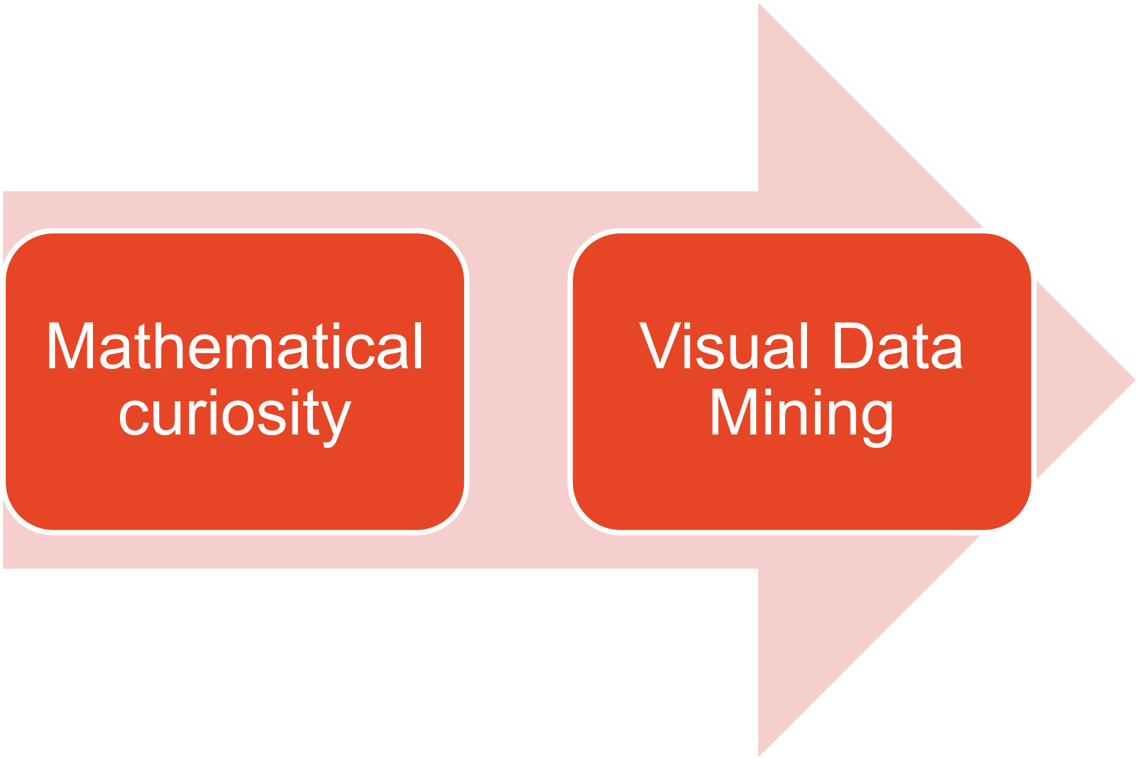


Forced Directed Layout

Background – Part I

After Tutte…

- Sometime in the 1980s, the motivation for graph drawing changed



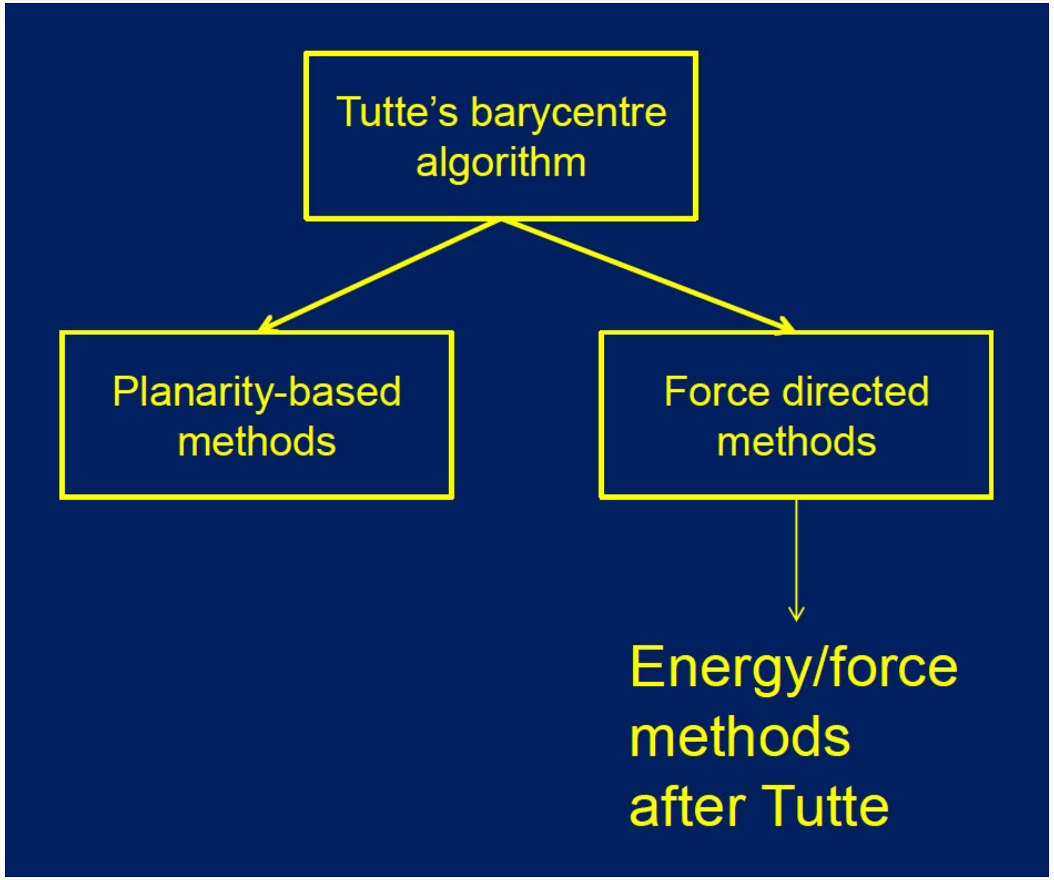
Background – Part II

From the 1980s, industrial demand for graph drawing algorithms has grown

- Software engineering: CASE systems, reverse engineering
- Biology: PPI networks, gene regulatory networks
- Physical networks: network management tools
- Security: risk management, money movements
- Social network analysis
- Customer relationship management: value identification

Many companies buy graph drawing algorithms, many code them

Background – Part III



Forced Directed Layout – Part I

- Based on VLSI placement / routing method

    Breuer, Min-cut placement, J. of Design Automation and Fault Tolerant Computing, 1-4, 343/382 (1977)

- Peter Eades' method is one of most used algorithm

1. Use springs of nonzero natural length
2. Use an inverse square law repulsive force between nonadjacent vertices

Forced Directed Layout – Part II

Force exerted by a vertex v on a vertex u :

- If u and v are adjacent :

$$f_{spring}(u,v) = k_{uv}(\, d(u,v) - q_{uv})i_{uv}$$

where k_{uv} is the strength of the spring between u and v

$d(u,v)$ is the Euclidean distance between u and v

q_{uv} is the natural length of the $u - v$ spring

i_{uv} is a unit vector in the direction from u to v

- If u and v are not adjacent :

$$f_{non-adjac}(u,v) = \frac{r_{uv}}{d(u,v)^2} i_{uv}$$

where r_{uv} is the strength of the repulsive force

Forced Directed Layout – Part III

- Total force on a vertex u:

$$F(u) = \sum_{(u,v)\in E} f_{spring}(u,v) + \sum_{(u,v)\notin E} f_{non-adjac}(u,v)$$

- A (locally) minimum energy configuration satisfies

$$F(u) = 0$$

  for each vertex u.

Note:

- The solution to this system of equations is not unique, that is, there are local minima that may not be global.

Characteristics of Forced Directed Layout

- Adjacency vertices are placed near each other (closeness)
- Vertices are placed so as not to be too close to each other (smallest separation)
- Edge lengths are fixed (fixed edge length)
- Symmetry is clearly shown (symmetry)
- Within the given drawing frame, vertices are uniformly distributed (uniform distribution)
- The shape of the drawing is adapted to the drawing frame (adaptation to the frame)
- Edge crossings are minimised (edge crossing minimisation)

Eades' Force Directed Layout

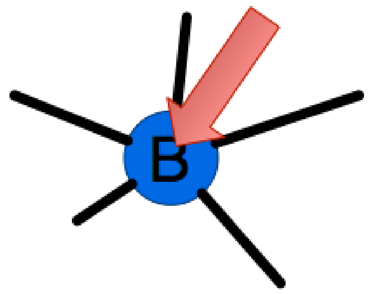
Input: Undirected graph G, the number of iteration is M
Output: Drawing of the undirected graph.
Method:

1. The vertices of G are placed in random positions;
2. The following is repeated M times;
    1) The forces working at each vertex are computed;
    2) Each vertex is moved by c4 x (the force working at that vertex);
3. G is drawn.

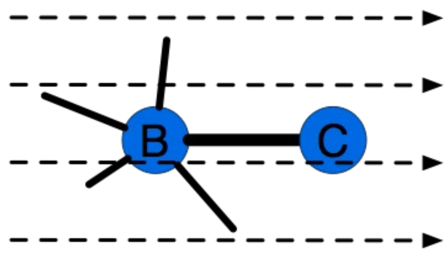
- Computation of repulsive force : $O(|V|2)$
- Computation of attractive force between adjacents : $O(|A|)$

Extra conditions of Eades' FDL

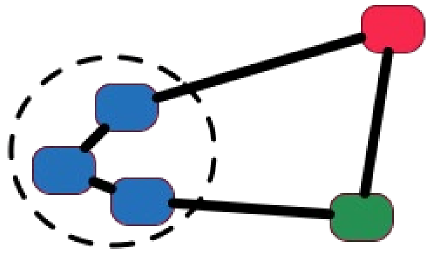
- Nails can be used to hold a node in place



- Magnetic fields and magnetised springs can be used to align nodes in various ways

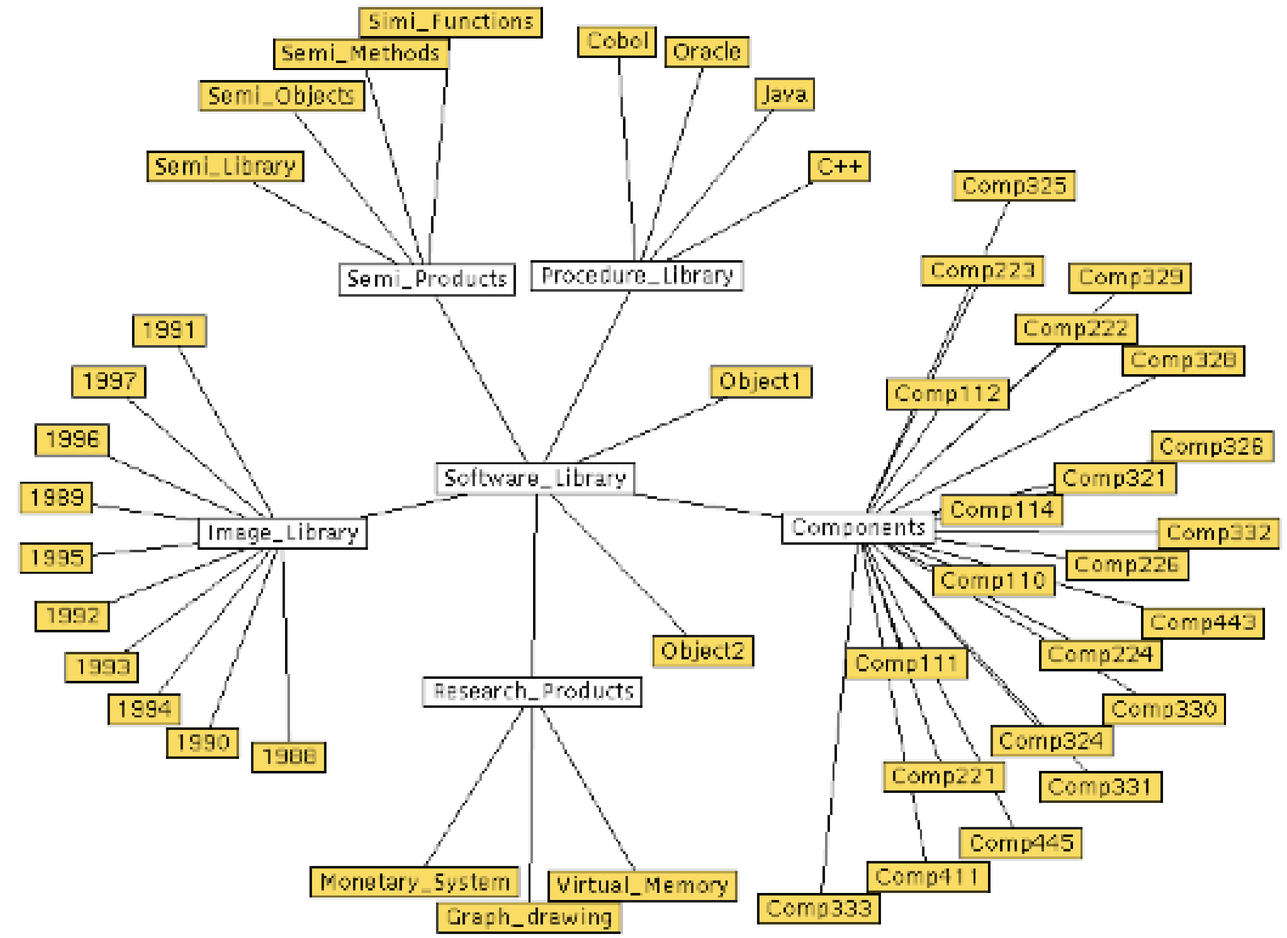


- Attractive forces can be used to keep clusters together



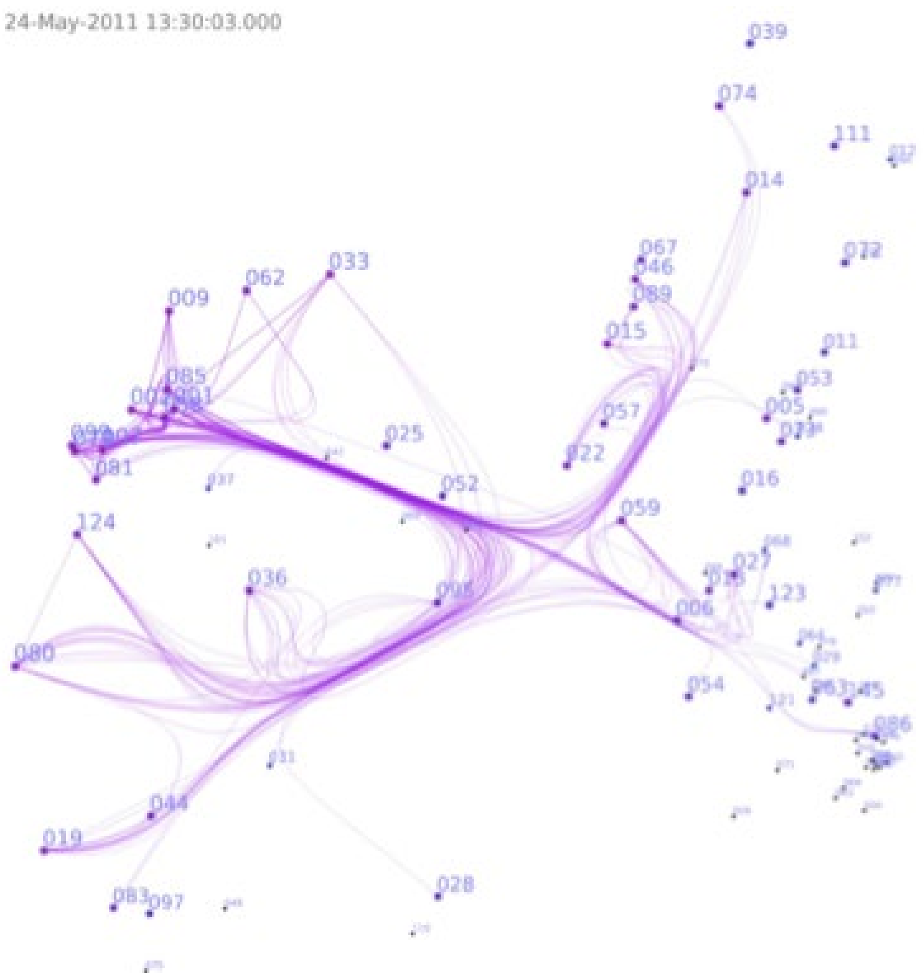
Examples of Visualisation using Graph Drawings

Example 1



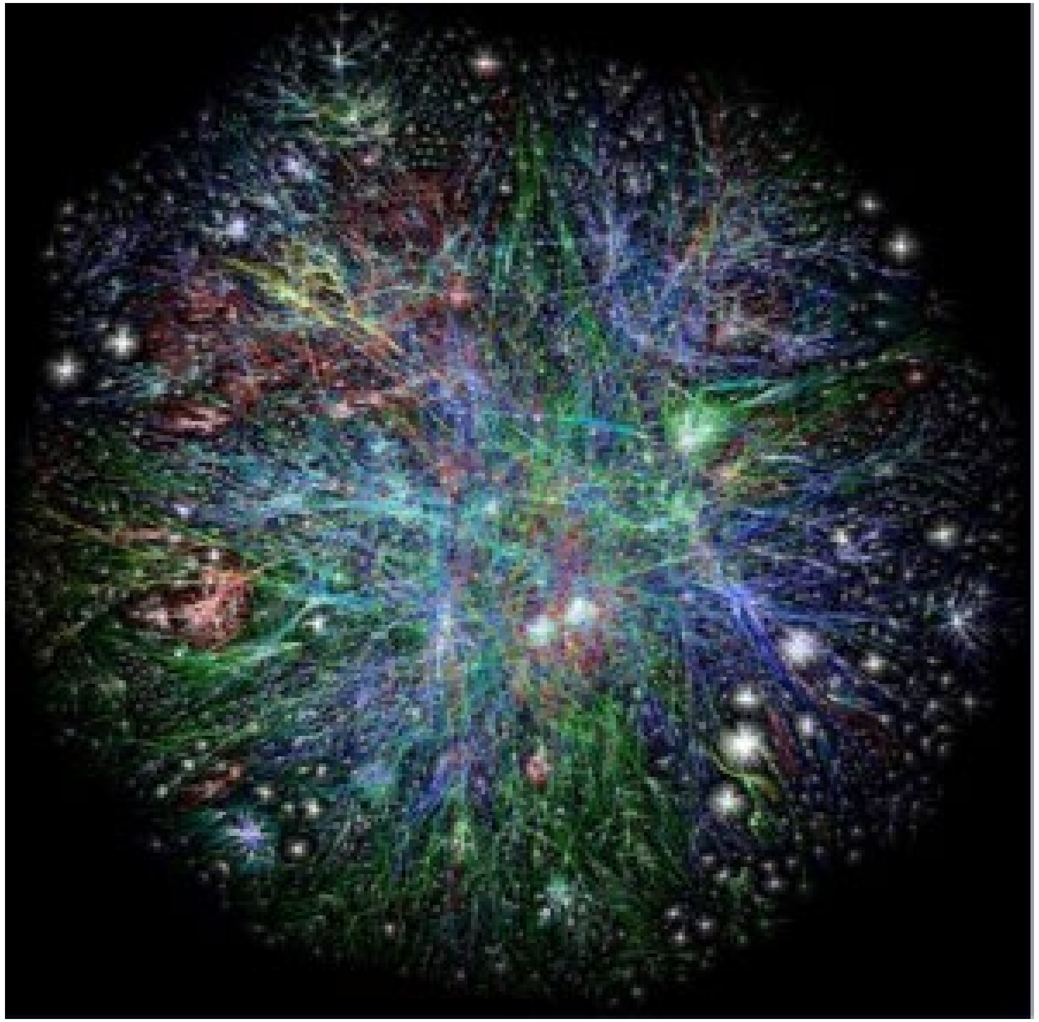
© Huang

Example 2



© Nguyen

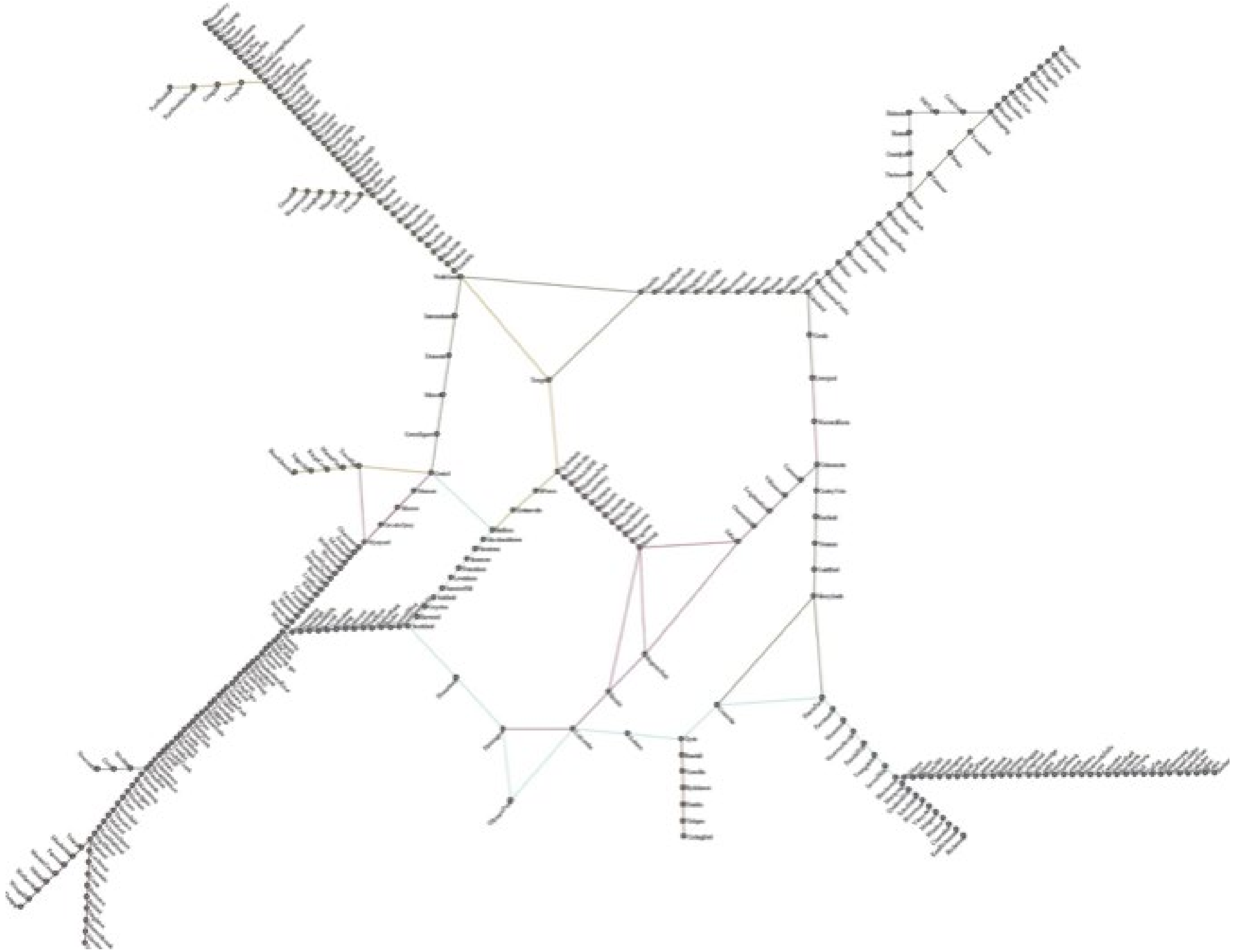
Example 3



Example 4

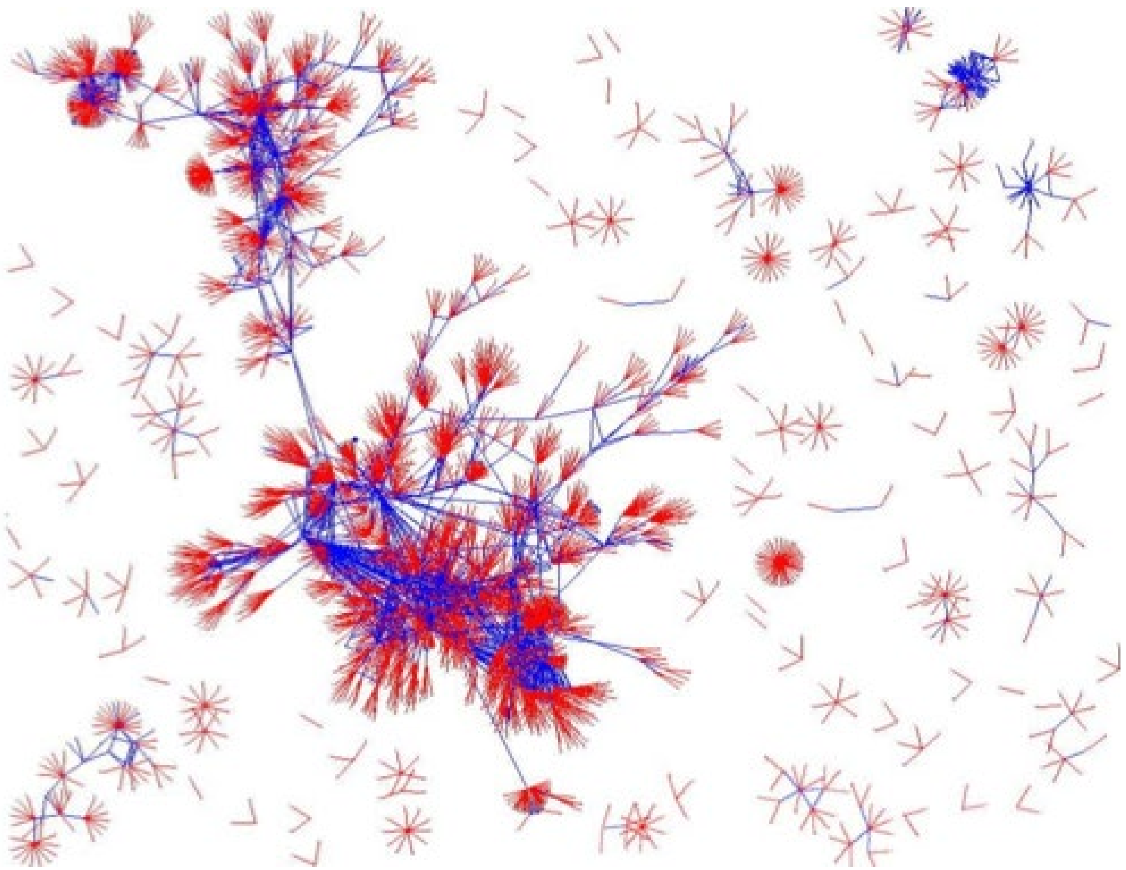

CityRail Network
Sydney Suburban Lines
Intercity Lines
North
CityRail

Example 5



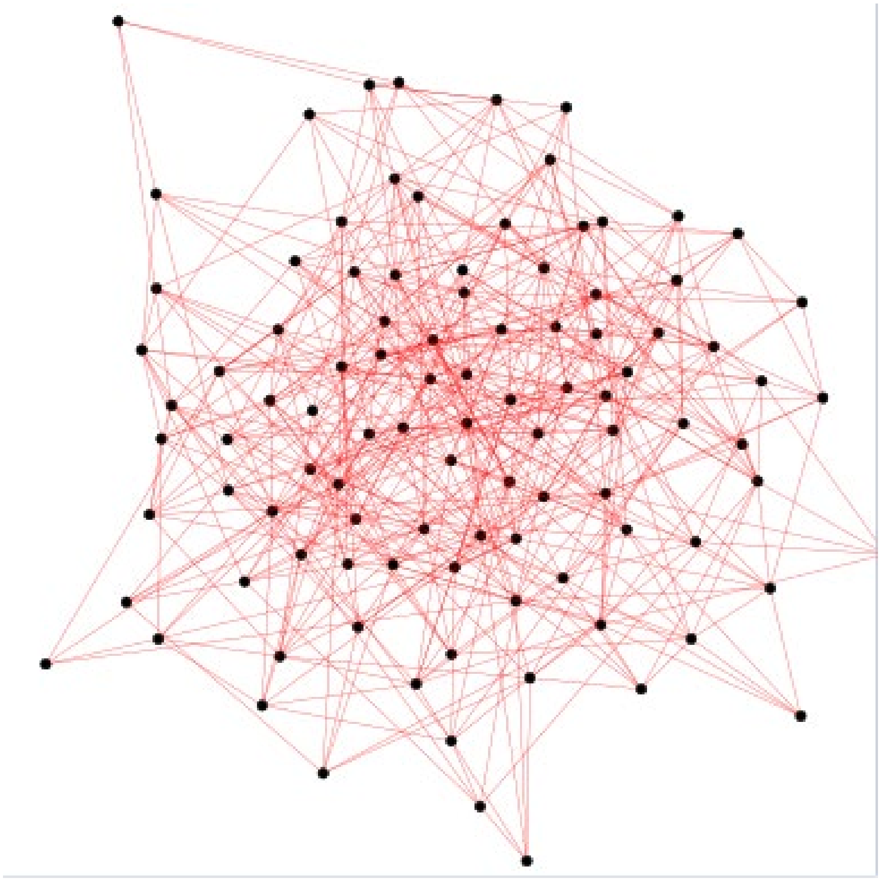
Example 6

- For some data sets, Force-Directed methods give reasonably good outputs

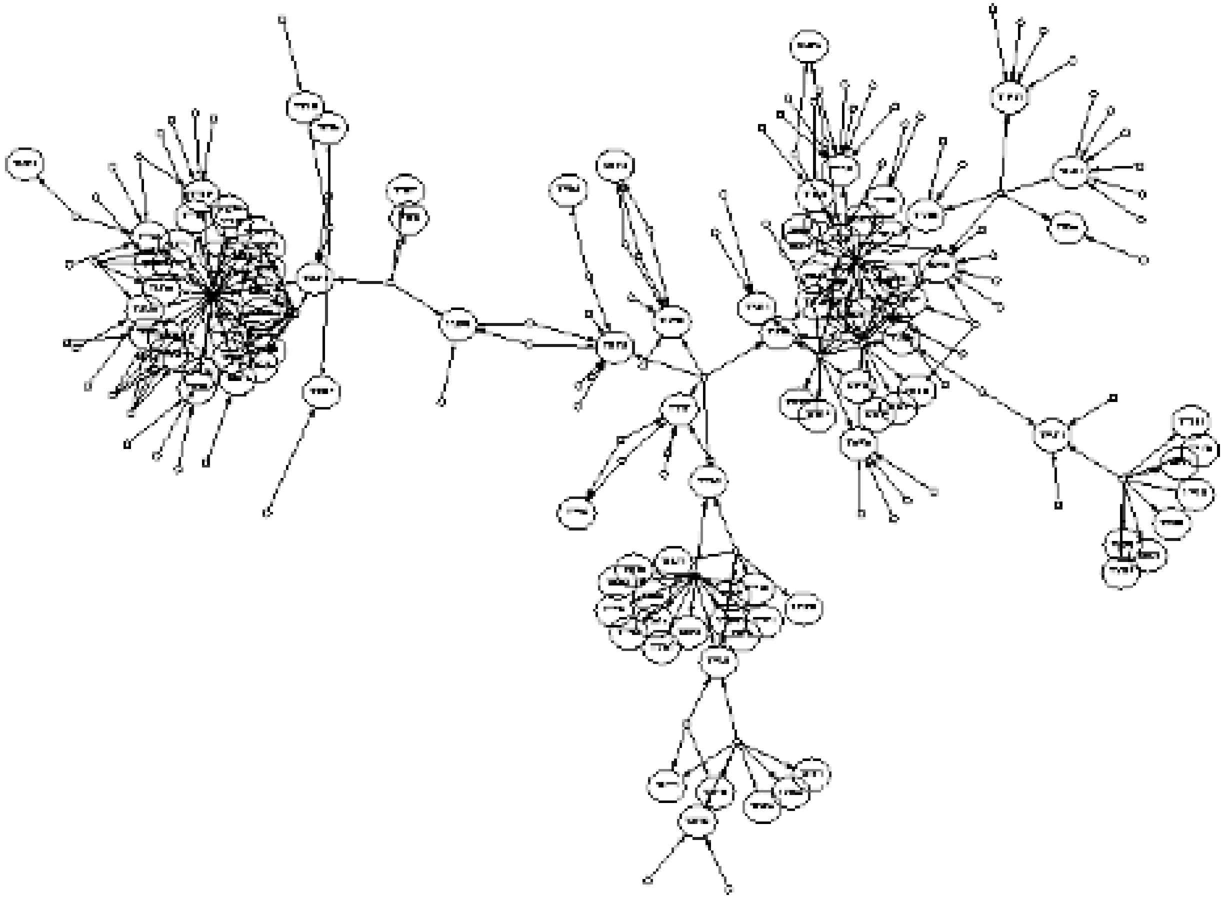


Example 7

- For some data sets, Force-Directed methods give bad outputs

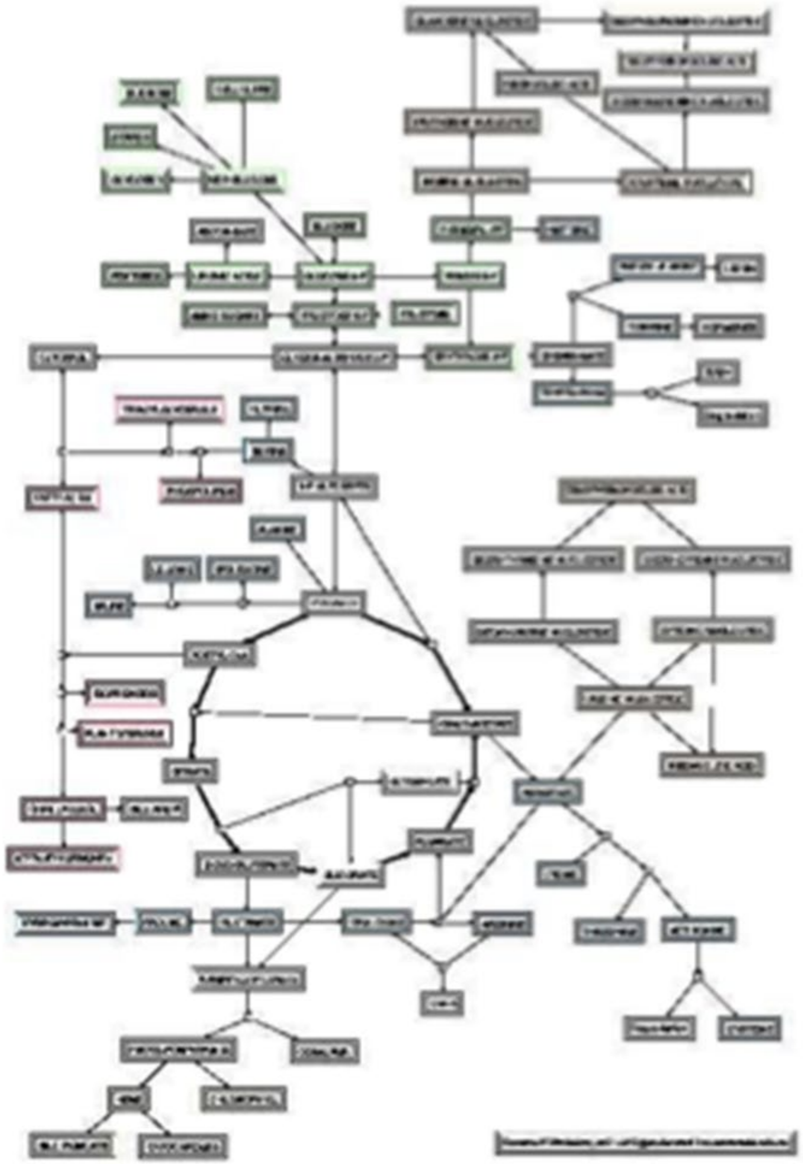


Example 8 - Software

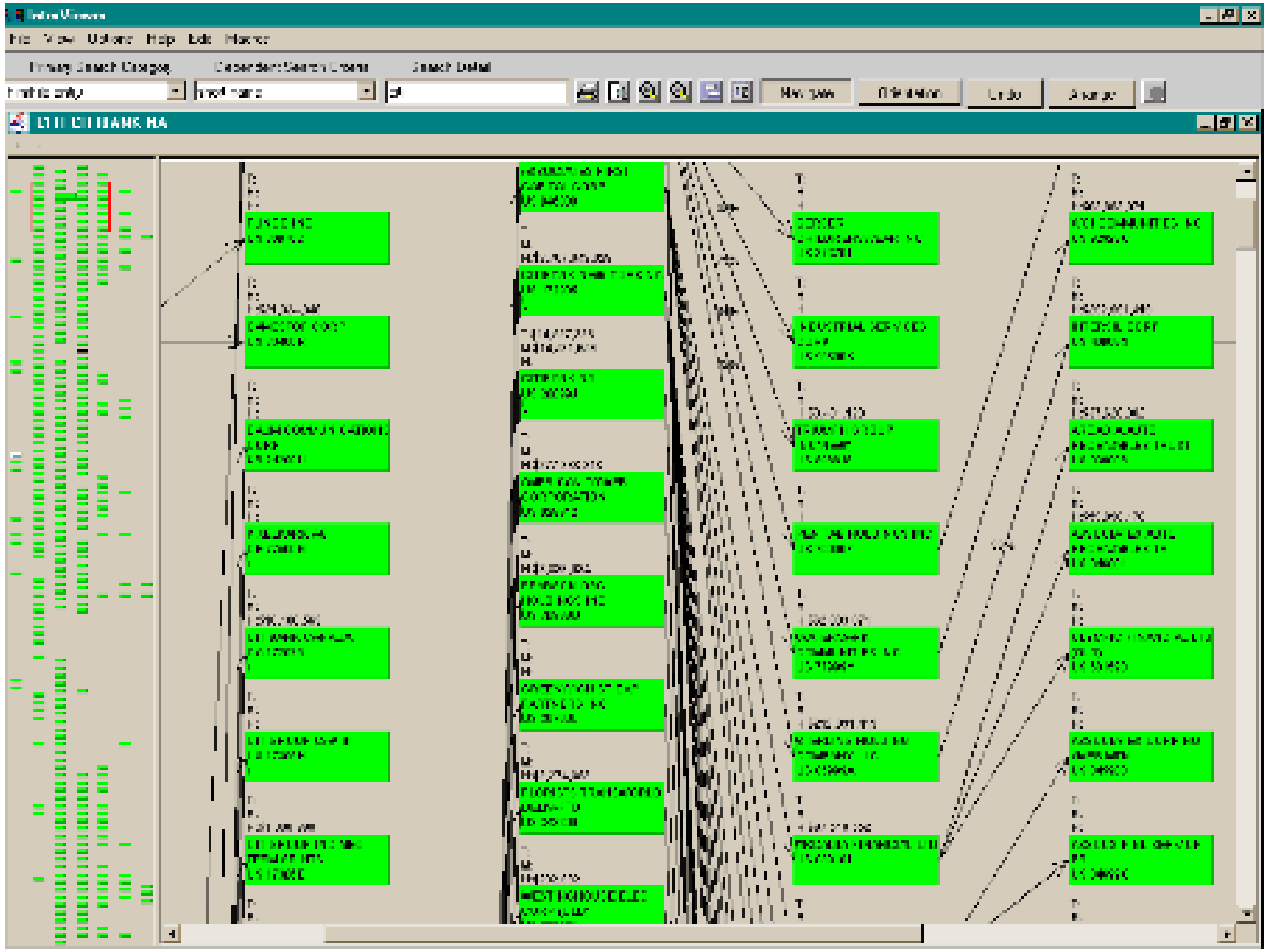


© AT&T

Example 9 - Biology



Example 10 – Business: Risk Exposure



Summary

- Graph based visualization can be used for describing relational information
- Force-directed methods account for 90% of commercial and free graph drawing software for undirected graphs.
- Very few scientific human experiments have been donw on the results of force directed methods
- Many informal (unscientific?) investigations have been donw
  - Appeal to developer intuition
  - Case studies in context

THE UNIVERSITY OF SYDNEY