Introduction

Lecture Graph Drawing Algorithms · 192.053

Martin Nöllenburg

13.03.2018

Lecturer and Dates

Lecturer

- Martin Nöllenburg
- noellenburg@ac.tuwien.ac.at
- Office: Favoritenstr. 9–11, room HC 04 11
- Office hours: by appointment

Lecturer and Dates

Lecturer

- Martin Nöllenburg
- noellenburg@ac.tuwien.ac.at
- Office: Favoritenstr. 9–11, room HC 04 11
- Office hours: by appointment

Lecture schedule

- Tue 9:00 – 11:00, seminar room 186
- 12 units (pay attention to teach-free days)

Organization

Website

$\rightarrow$ visit TISS page for course 192.053 $\leftarrow$

- TUWEL course will go online this week
- up-to-date course information
- lecture slides
- exercise information
- additional literature and material
- registration for exercise groups and topic selection (mandatory for certificate, deadline March 23!)

Goals and Requirements

Objectives: At the end of the course you will be able to...

- explain concepts, structures, and problem definitions
- understand the discussed algorithms, explain them intuitively, analyze them formally and prove their properties
- use graph drawing tools and libraries to create your own visualizations
- select and adapt appropriate graph drawing algorithms
- analyze new graph drawing problems and build abstract models
- develop and analyze efficient algorithms in these models

Goals and Requirements

Objectives: At the end of the course you will be able to...

- explain concepts, structures, and problem definitions
- understand the discussed algorithms, explain them intuitively, analyze them formally and prove their properties
- use graph drawing tools and libraries to create your own visualizations
- select and adapt appropriate graph drawing algorithms
- analyze new graph drawing problems and build abstract models
- develop and analyze efficient algorithms in these models

Prior Knowledge: graph algorithms and basic geometry, e.g.

- VU Algorithmen und Datenstrukturen (1+2) **required**
- VUs Algorithmics, Algorithmic Geometry, Visualization (1+2), InfoVis **optional**

Goals and Requirements

Objectives: At the end of the course you will be able to...

- explain concepts, structures, and problem definitions
- understand the discussed algorithms, explain them intuitively, analyze them formally and prove their properties
- use graph drawing tools and libraries to create your own visualizations
- select and adapt appropriate graph drawing algorithms
- analyze new graph drawing problems and build abstract models
- develop and analyze efficient algorithms in these models

Prior Knowledge: graph algorithms and basic geometry, e.g.

- VU Algorithmen und Datenstrukturen (1+2) **required**
- VUs Algorithmics, Algorithmic Geometry, Visualization (1+2), InfoVis **optional**

Course Time Breakdown: 3LP = 75h

- time in lectures and exercise sessions ca. 24h
- self-study of lecture material, exam preparation, and exam ca. 21h
- exercise (presentation or programming project) ca. 30h

Structure of the Lectures

Attendance:

- recommended (interaction, exercises), but not mandatory

Structure of the Lectures

Attendance:

- recommended (interaction, exercises), but not mandatory

Media:

- slides & blackboard (take notes!)
- text books, original literature (mostly background reading)

Structure of the Lectures

Attendance:

- recommended (interaction, exercises), but not mandatory

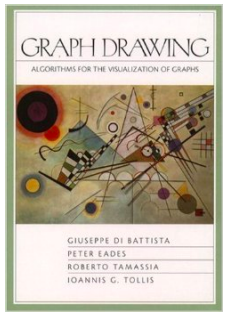
Media:

- slides & blackboard (take notes!)
- text books, original literature (mostly background reading)

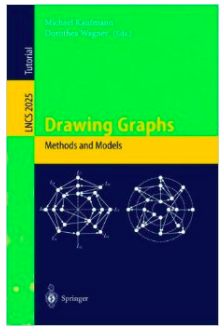
Content:

- aesthetic quality and optimization criteria, layout styles
- formal algorithmic perspective on graph visualization
- **not:** rendering or graphics
- algorithm types
  - divide & conquer / recursion
  - combinatorial optimization (network flow, ILP, ...)
  - incremental algorithms
  - iterative methods
  - algorithms for special graph classes

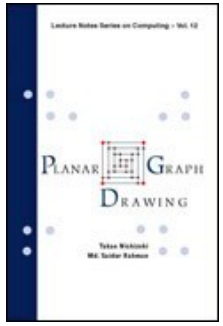
Literature (available in TU library)



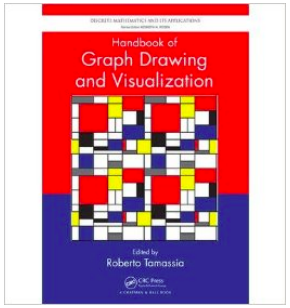
G. di Battista, P. Eades, R. Tamassia, I. Tollis:
Graph Drawing
Prentice Hall, 1998



M. Kaufmann, D. Wagner:
Drawing Graphs: Methods and Models
Springer, 2001
→ e-book via TU Wien



T. Nishizeki, Md. S. Rahman:
Planar Graph Drawing
World Scientific, 2004



R. Tamassia:
Handbook of Graph Drawing and Visualization
CRC Press, 2013 `http://cs.brown.edu/~rt/gdhandbook/`

What you should know

Basics graph theory:

- graph, vertices, edges
- vertex degree, neighborhood, adjacent, incident
- connectivity, tree, cycle, path
- planarity
- ...

What you should know

Basics graph theory:

- graph, vertices, edges
- vertex degree, neighborhood, adjacent, incident
- connectivity, tree, cycle, path
- planarity
- ...

Basics algorithmics:

- asymptotic running times, big-O notation
- complexity, NP-completeness
- network flow algorithms
- linear programming
- recursion
- divide & conquer
- ...

What you should know

Basics graph theory:

- graph, vertices, edges
- vertex degree, neighborhood, adjacent, incident
- connectivity, tree, cycle, path
- planarity
- ...

Basics algorithmics:

- asymptotic running times, big-O notation
- complexity, NP-completeness
- network flow algorithms
- linear programming
- recursion
- divide & conquer
- ...

general rule:
ask if s.th. is unclear

Exercises

Lecture is complemented by an exercise. You can choose between more theoretical or more practical exercises

- exercise will run in parallel to lecture throughout course
- presentation of your work at the end of semester (dates tba)
- regular meetings with Fabian to discuss progress
- exercises count for **30%** of the final grade

Lecture Assistant

- Fabian Klute
- `fklute@ac.tuwien.ac.at`
- Office: Favoritenstr. 9–11, room HA 04 06
- Office hours: by appointment

Graph Drawing – The GD Conference

Annual Symposium on Graph Drawing and Network Visualization

This year in Barcelona, usually iterates between EU and US

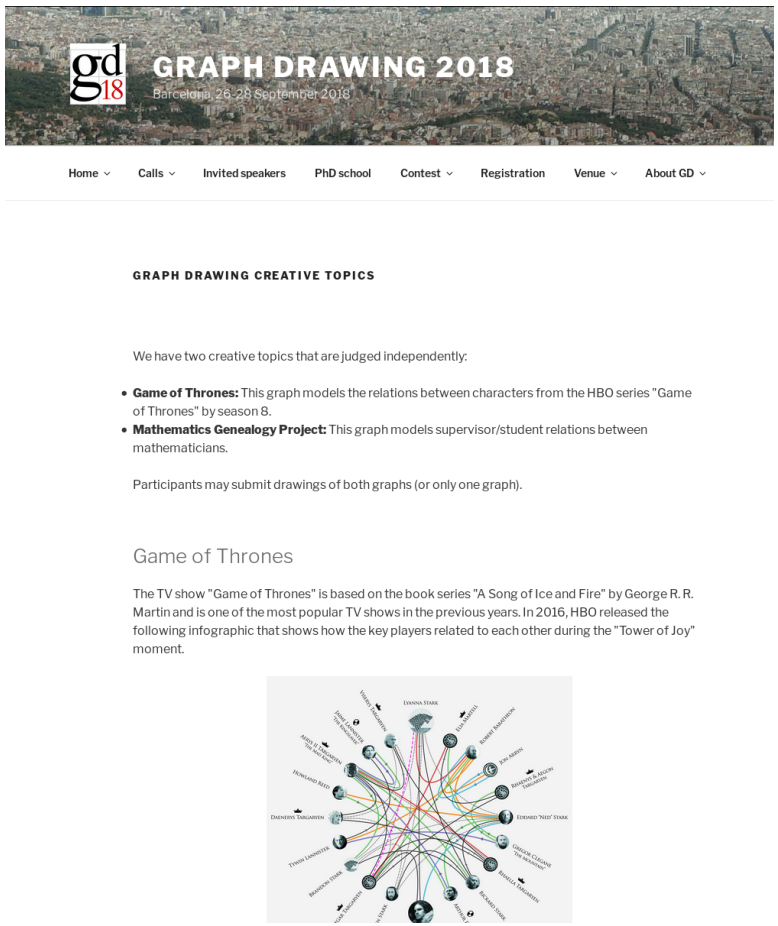
Also: Graph Drawing Contest at every GD!

Graph Drawing – The GD Conference

Annual Symposium on Graph Drawing and Network Visualization

This year in Barcelona, usually iterates between EU and US

Also: Graph Drawing Contest at every GD!



The practical exercises will be the contest topics of 2018!

Challenging topics!

Visual design and algorithmic tasks!

More details on the website!

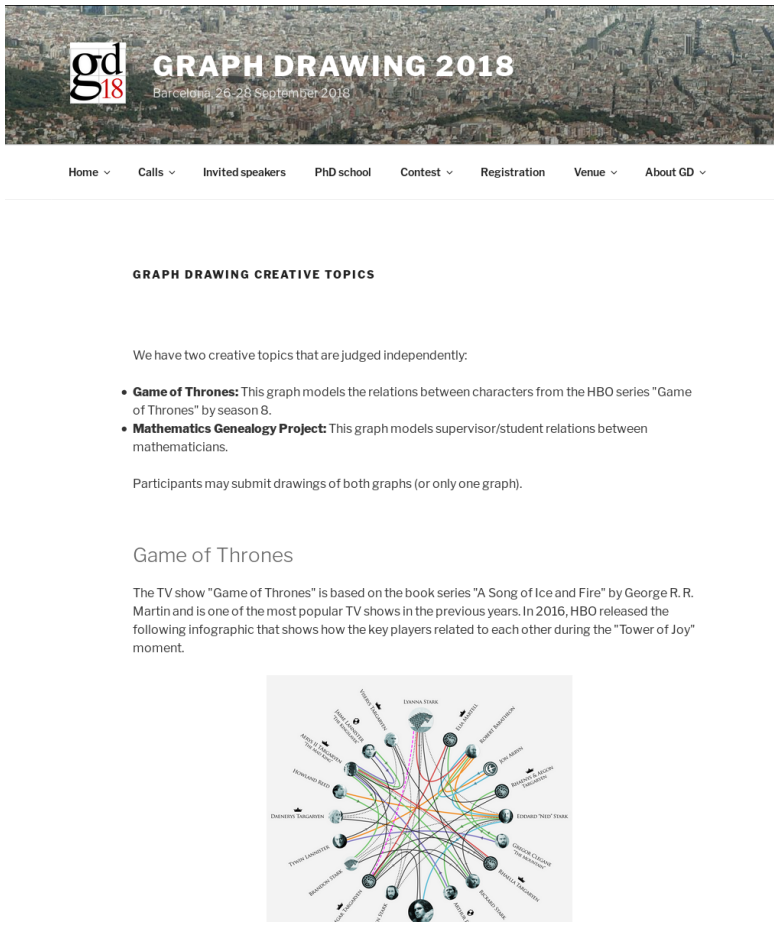
`http://graphdrawing.de/contest2018/topics.html`

Graph Drawing – The GD Conference

Annual Symposium on Graph Drawing and Network Visualization

This year in Barcelona, usually iterates between EU and US

Also: Graph Drawing Contest at every GD!



The practical exercises will be the contest topics of 2018!

Challenging topics!

Visual design and algorithmic tasks!

More details on the website!

You can win eternal fame, months of fun, new knowledge, paper ideas...

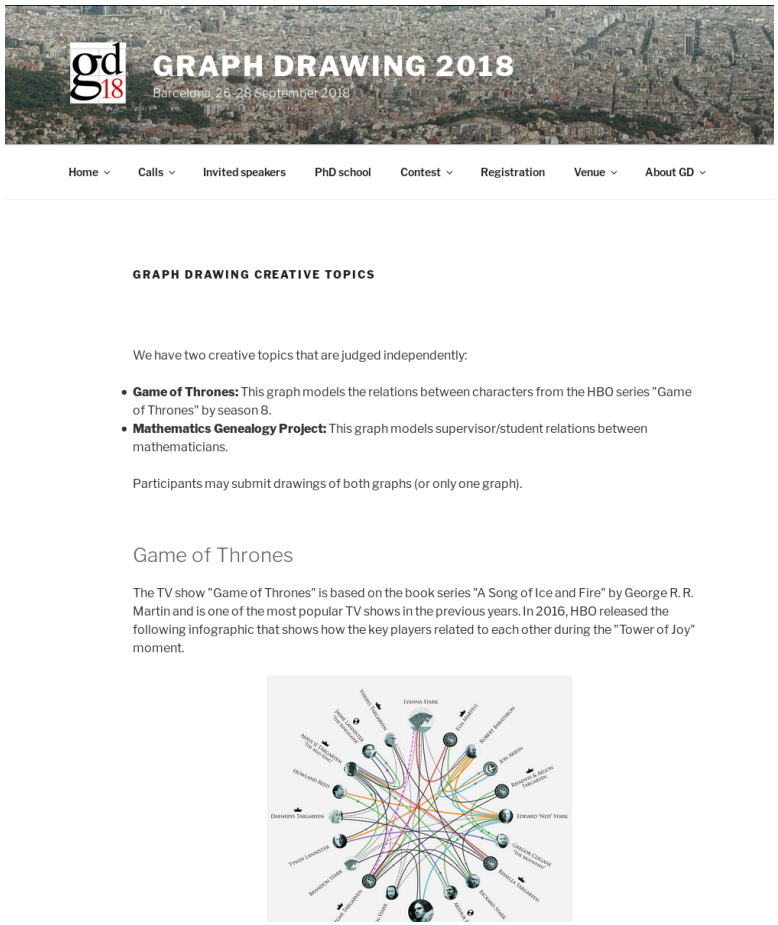
`http://graphdrawing.de/contest2018/topics.html`

Graph Drawing – The GD Conference

Annual Symposium on Graph Drawing and Network Visualization

This year in Barcelona, usually iterates between EU and US

Also: Graph Drawing Contest at every GD!



The practical exercises will be the contest topics of 2018!

Challenging topics!

Visual design and algorithmic tasks!

More details on the website!

You can win eternal fame, months of fun, new knowledge, paper ideas...

(And money!)

`http://graphdrawing.de/contest2018/topics.html`

Exercises

Programming?

Exercises

Programming?

No!

Paper presentation

- Theory paper!
- Read and understand
- Presentation at the end of semester

Exercises

Programming?

No!

Paper presentation

- Theory paper!
- Read and understand
- Presentation at the end of semester

Yes!

Coding Project in Teams (3 ppl/team)

Exercises

Programming?

No!

Paper presentation

- Theory paper!
- Read and understand
- Presentation at the end of semester

Yes!

Coding Project in Teams (3 ppl/team)

Design?

Exercises

Programming?

No!

Paper presentation

- Theory paper!
- Read and understand
- Presentation at the end of semester

Yes!

Coding Project in Teams (3 ppl/team)

Design?

No!

Angle Optimization

- Create a drawing
- Straight lines
- Max min angles
- Your ideas needed!
- Short presentation

Exercises

Programming?

No!

Paper presentation

- Theory paper!
- Read and understand
- Presentation at the end of semester

Yes!

Coding Project in Teams (3 ppl/team)

Design?

No!

Yes!

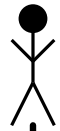
Angle Optimization

- Create a drawing
- Straight lines
- Max min angles
- Your ideas needed!
- Short presentation

Visualization

- Given some (graph) data
- Visualize the data using methods from graph drawing
- Be creative!
- Short presentation

Exercises



Programming?

No!

Paper presentation

- Theory paper!
- Read and understand
- Presentation at the end of semester

Yes!

Coding Project in Teams (3 ppl/team)

Design?

No!

Yes!

Visualization

- Given some (graph) data
- Visualize the data using methods from graph drawing
- Be creative!
- Short presentation

Angle Optimization

- Create a drawing
- Straight lines
- Max min angles
- Your ideas needed!
- Short presentation

GD contest!

Final Grading

Exam

- individual oral exams in July and September 2018
- each exam 15-20 minutes
- 70% of the final grade

Final grade

- 30% exercises + 70% oral exam

Requirements for positive grade

Successful participation in exercise part (incl. attendance of student presentations)

Introduction to Graph Drawing

Graphs and their Representations

What is a Graph?

Graphs and their Representations

What is a Graph?

Tuple $G = (V, E)$
Set of vertices $V = \{v_1, \ldots, v_n\}$
Set of edges $E = \{e_1, \ldots, e_m\}$,
$e_i = \{v_j, v_k\},\ 1 \leq i \leq m,\ 1 \leq j, k \leq n$

Graphs and their Representations

What is a Graph?

Tuple $G = (V, E)$
Set of vertices $V = \{v_1, \ldots, v_n\}$
Set of edges $E = \{e_1, \ldots, e_m\}$,
$e_i = \{v_j, v_k\},\ 1 \leq i \leq m,\ 1 \leq j, k \leq n$

Representations?

Graphs and their Representations

What is a Graph?

Tuple $G = (V, E)$
Set of vertices $V = \{v_1, \ldots, v_n\}$
Set of edges $E = \{e_1, \ldots, e_m\}$,
$e_i = \{v_j, v_k\},\ 1 \leq i \leq m,\ 1 \leq j, k \leq n$

Representations?

Set representation:

$$V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}$$
$$E = \{\{v_1, v_2\}, \{v_1, v_8\}, \{v_2, v_3\}, \{v_3, v_5\}, \{v_3, v_9\}, \{v_3, v_{10}\}, \{v_4, v_5\}, \{v_4, v_6\}, \{v_4, v_9\}, \{v_5, v_8\}, \{v_6, v_8\}, \{v_6, v_9\}, \{v_7, v_8\}, \{v_7, v_9\}, \{v_8, v_{10}\}, \{v_9, v_{10}\}\}$$

Graphs and their Representations

What is a Graph?

Tuple $G = (V, E)$
Set of vertices $V = \{v_1, \ldots, v_n\}$
Set of edges $E = \{e_1, \ldots, e_m\}$,
$e_i = \{v_j, v_k\},\ 1 \leq i \leq m,\ 1 \leq j, k \leq n$

Representations?

Set representation
Adjacency lists

$$
\begin{array}{ll}
v_1: & v_2, v_8 \\
v_2: & v_1, v_3 \\
v_3: & v_2, v_5, v_9, v_{10} \\
v_4: & v_5, v_6, v_9 \\
v_5: & v_3, v_4, v_8 \\
v_6: & v_4, v_8, v_9 \\
v_7: & v_8, v_9 \\
v_8: & v_1, v_5, v_6, v_7, v_9, v_{10} \\
v_9: & v_3, v_4, v_6, v_7, v_8, v_{10} \\
v_{10}: & v_3, v_8, v_9
\end{array}
$$

Graphs and their Representations

What is a Graph?

Tuple $G = (V, E)$
Set of vertices $V = \{v_1, \ldots, v_n\}$
Set of edges $E = \{e_1, \ldots, e_m\}$,
$e_i = \{v_j, v_k\},\ 1 \leq i \leq m,\ 1 \leq j, k \leq n$

Representations?

Set representation
Adjacency lists
Adjacency matrix

$$\begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0
\end{pmatrix}$$

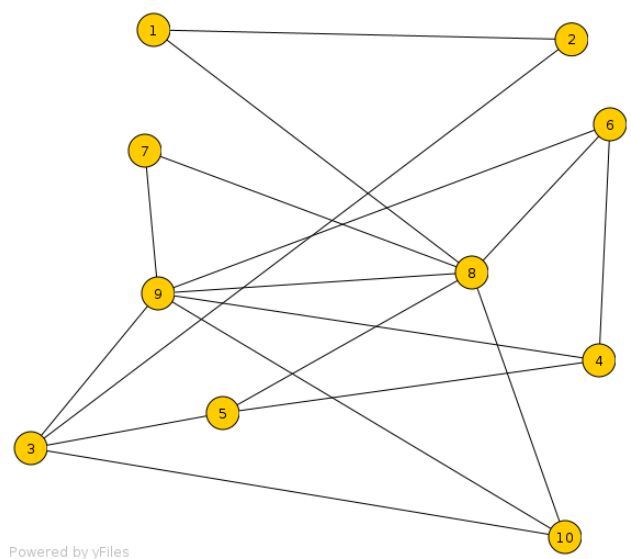
Graphs and their Representations

What is a Graph?

Tuple $G = (V, E)$
Set of vertices $V = \{v_1, \ldots, v_n\}$
Set of edges $E = \{e_1, \ldots, e_m\}$,
$e_i = \{v_j, v_k\},\ 1 \leq i \leq m,\ 1 \leq j, k \leq n$

Representations?

Set representation
Adjacency lists
Adjacency matrix
Drawing



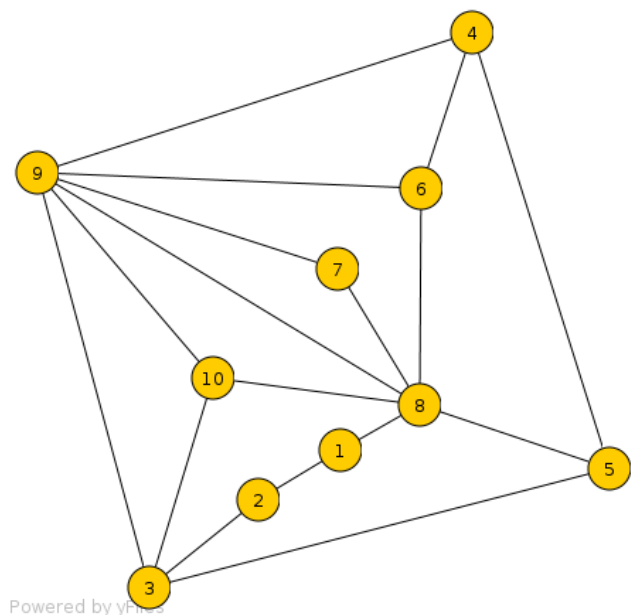
Graphs and their Representations

What is a Graph?

Tuple $G = (V, E)$
Set of vertices $V = \{v_1, \ldots, v_n\}$
Set of edges $E = \{e_1, \ldots, e_m\}$,
$e_i = \{v_j, v_k\},\ 1 \leq i \leq m,\ 1 \leq j, k \leq n$

Representations?

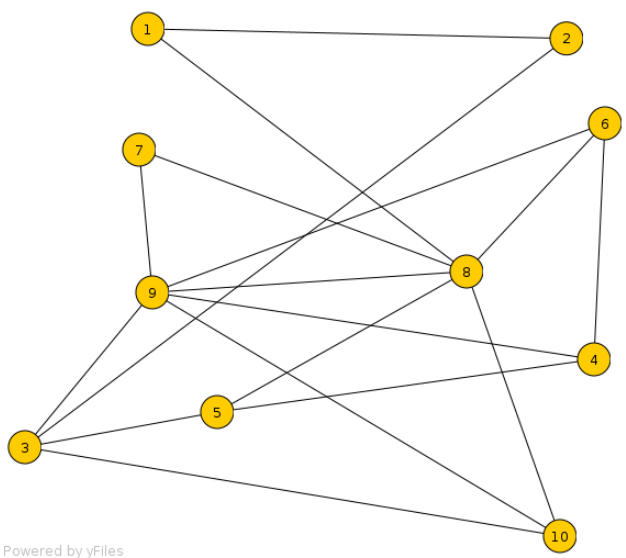
Set representation
Adjacency lists
Adjacency matrix
Drawing



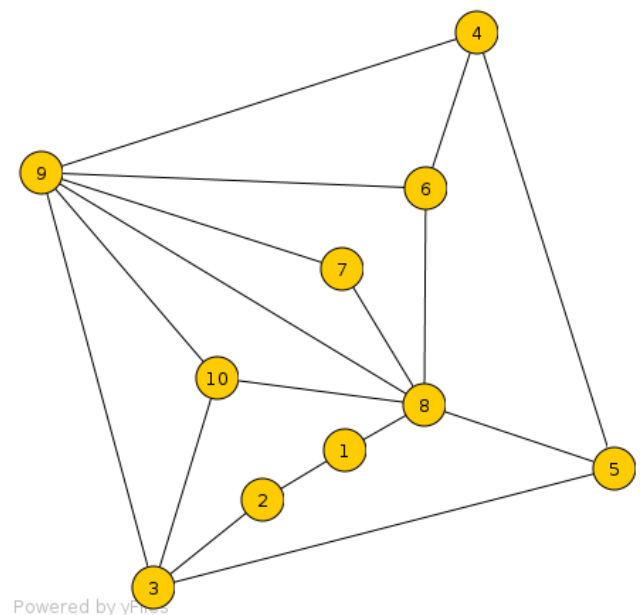
Graphs and their Representations

$V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}$
$E = \{\{v_1, v_2\}, \{v_1, v_8\}, \{v_2, v_3\}, \{v_3, v_5\}, \{v_3, v_9\}, \{v_3, v_{10}\}, \{v_4, v_5\}, \{v_4, v_6\}, \{v_4, v_9\}, \{v_5, v_8\}, \{v_6, v_8\}, \{v_6, v_9\}, \{v_7, v_8\}, \{v_7, v_9\}, \{v_8, v_{10}\}, \{v_9, v_{10}\}\}$

v_1 : v_2, v_8
v_2 : v_1, v_3
v_3 : v_2, v_5, v_9, v_{10}
v_4 : v_5, v_6, v_9
v_5 : v_3, v_4, v_8
v_6 : v_4, v_8, v_9
v_7 : v_8, v_9
v_8 : $v_1, v_5, v_6, v_7, v_9, v_{10}$
v_9 : $v_3, v_4, v_6, v_7, v_8, v_{10}$
v_{10} : v_3, v_8, v_9



$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$



Why Draw Graphs?

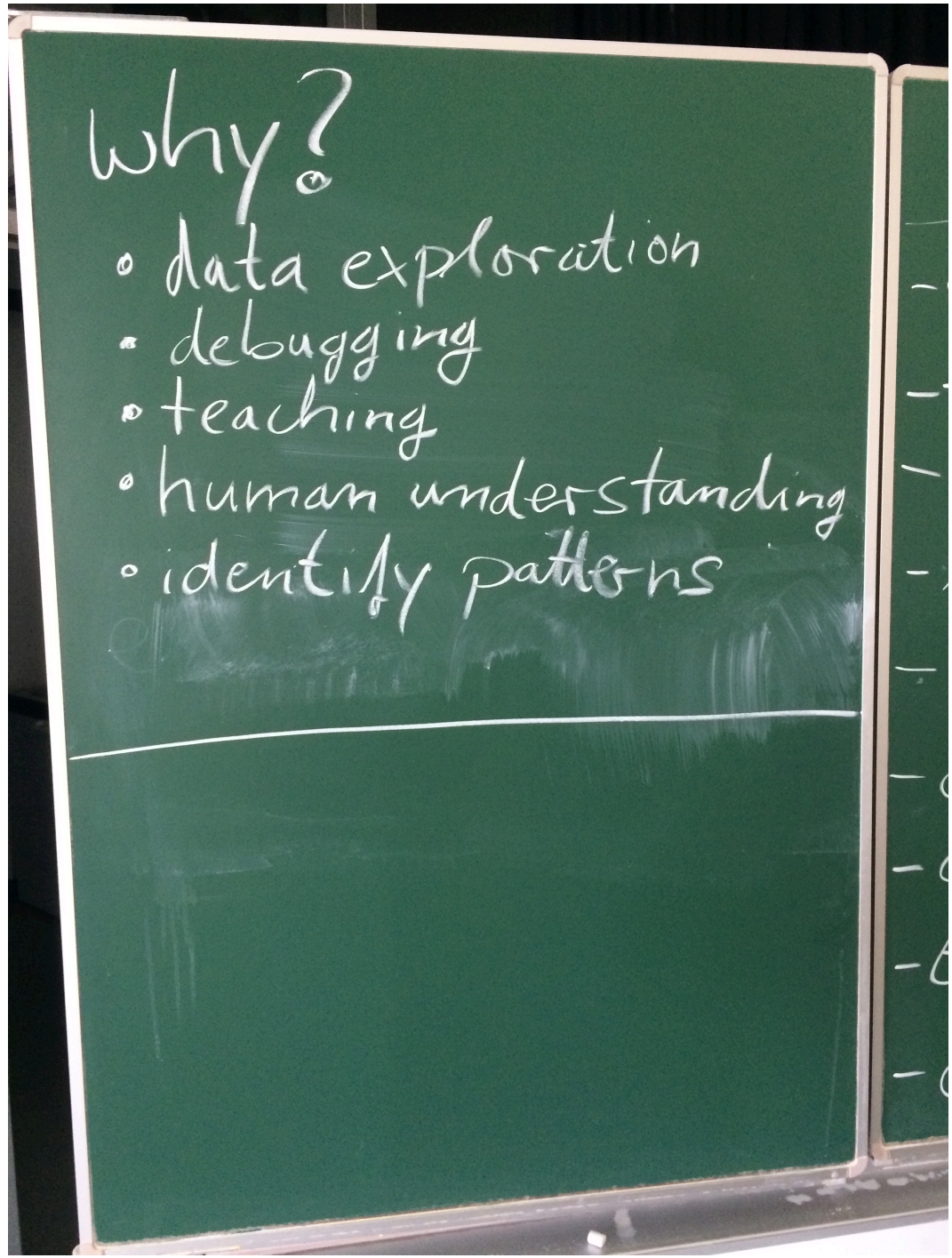
Why do we need graph/network visualizations?

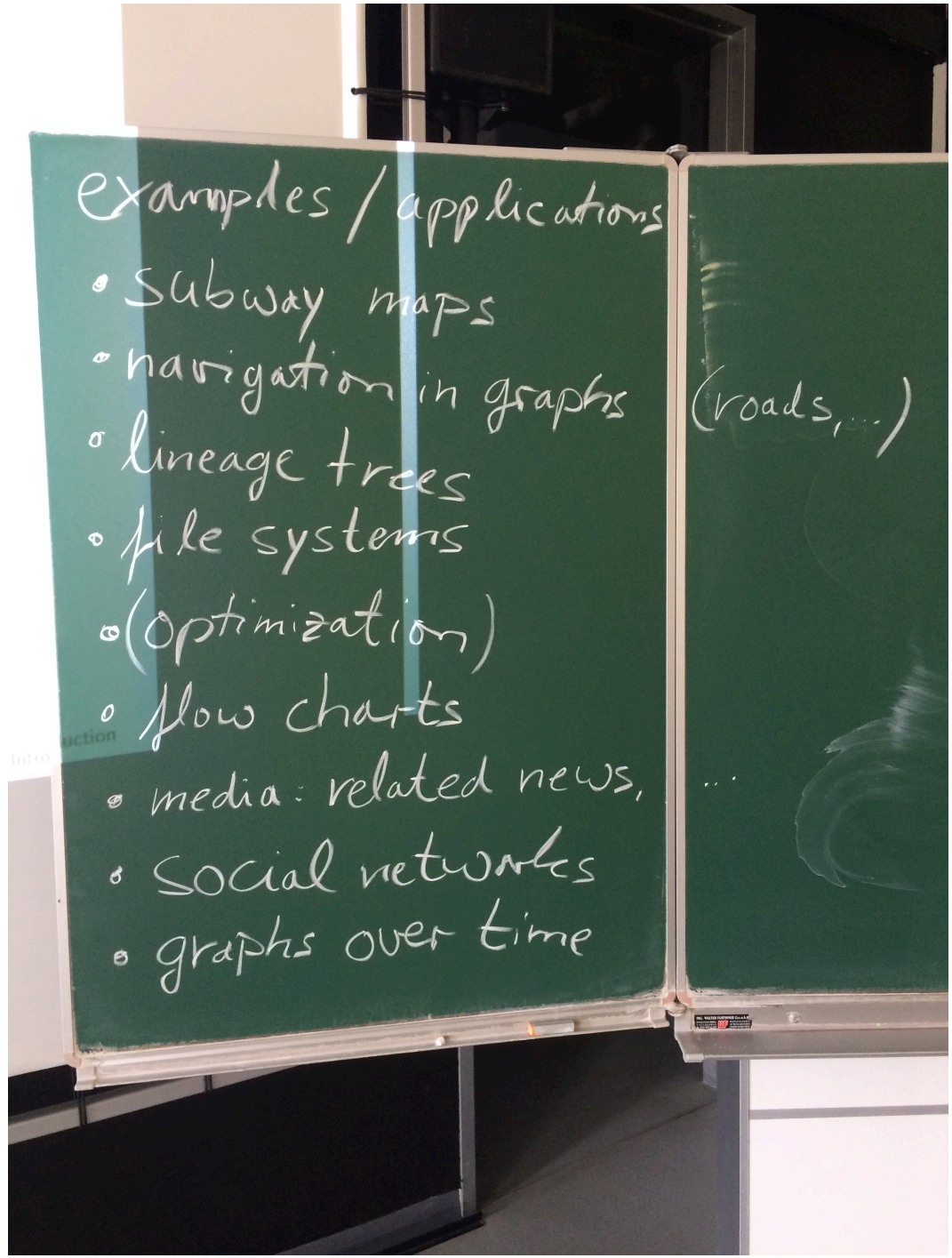
Which applications or examples do you know?

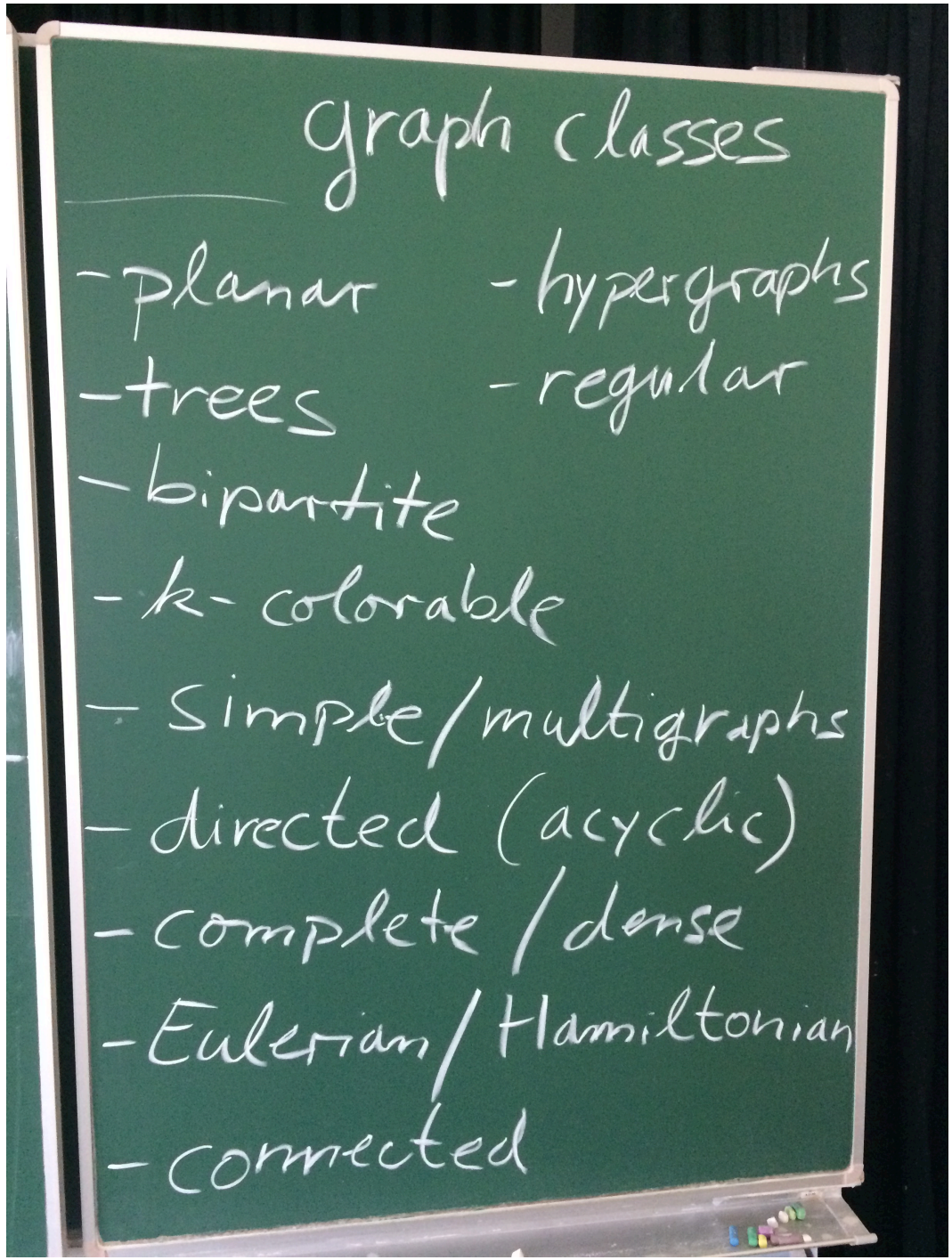
Which relevant graph classes do you know?

In groups of two or three: discuss and write down your answers

5 min


Why?
• data exploration
• debugging
• teaching
• human understanding
• identify patterns


examples / applications
• subway maps
• navigation in graphs (roads, ...)
• lineage trees
• file systems
• (optimization)
• flow charts
• media: related news, ...
• social networks
• graphs over time

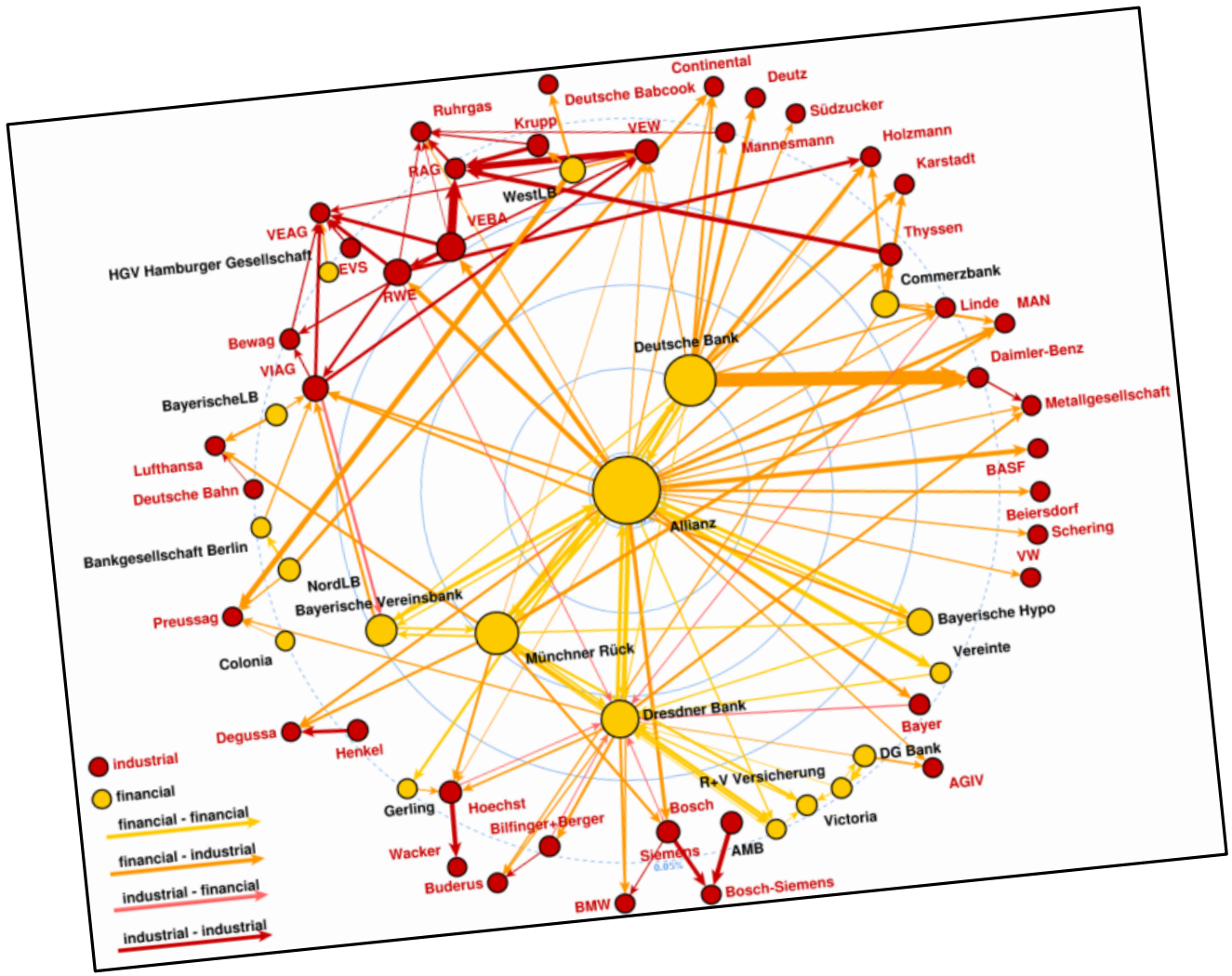

graph classes
- planar
- trees
- bipartite
- k-colorable
- simple / multigraphs
- directed (acyclic)
- complete / dense
- Eulerian / Hamiltonian
- connected
- hypergraphs
- regular

Why Draw Graphs?

Graphs are mathematical models of real physical and abstract networks (social networks, metabolical networks, VLSI-network, UML-diagrams, citation networks, ...)

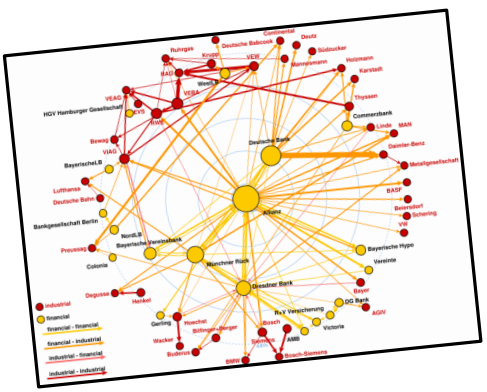
Why Draw Graphs?

Graphs are mathematical models of real physical and abstract networks (social networks, metabolical networks, VLSI-network, UML-diagrams, citation networks, ...)



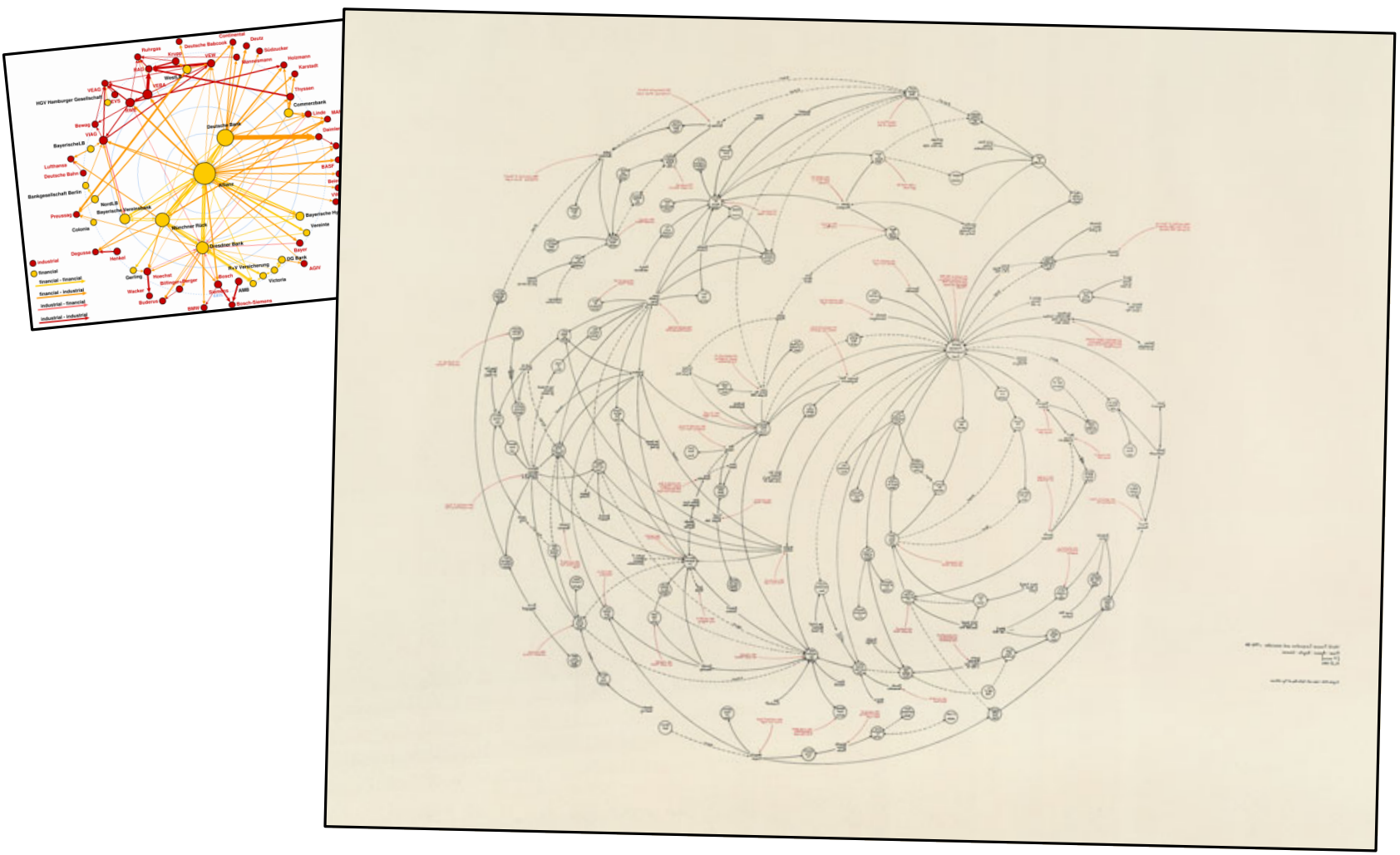
Why Draw Graphs?

Graphs are mathematical models of real physical and abstract networks (social networks, metabolical networks, VLSI-network, UML-diagrams, citation networks, ...)



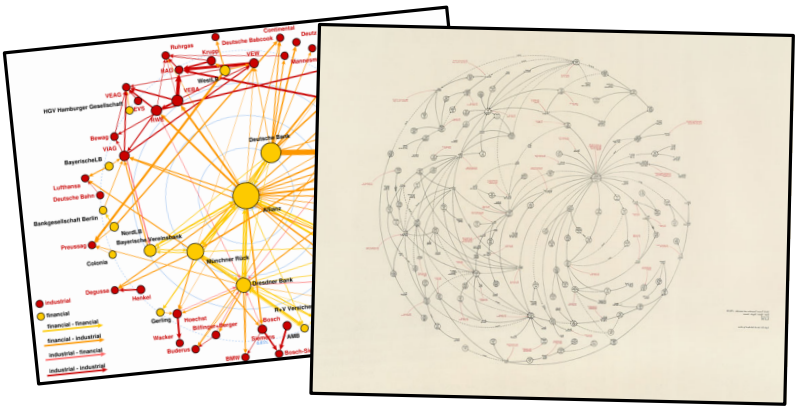
Why Draw Graphs?

Graphs are mathematical models of real physical and abstract networks (social networks, metabolical networks, VLSI-network, UML-diagrams, citation networks, ...)



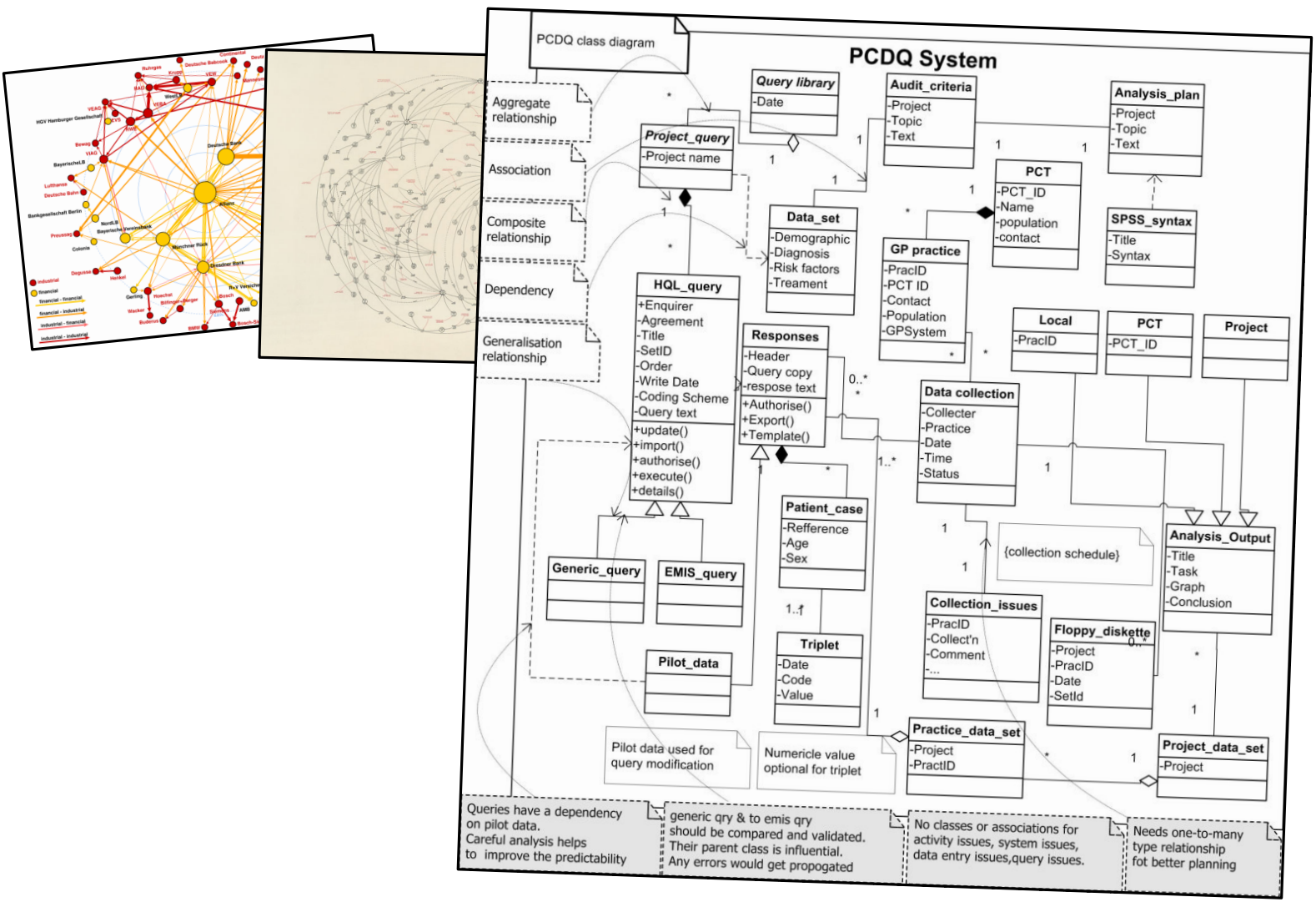
Why Draw Graphs?

Graphs are mathematical models of real physical and abstract networks (social networks, metabolical networks, VLSI-network, UML-diagrams, citation networks, ...)



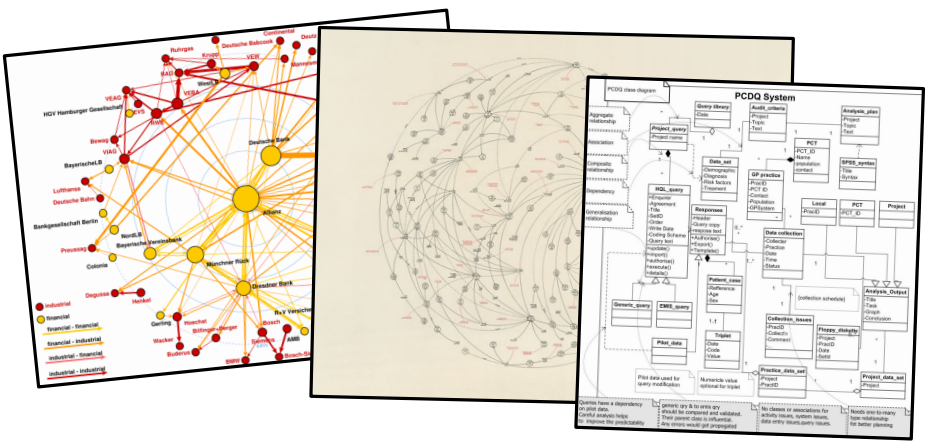
Why Draw Graphs?

Graphs are mathematical models of real physical and abstract networks (social networks, metabolical networks, VLSI-network, UML-diagrams, citation networks, ...)



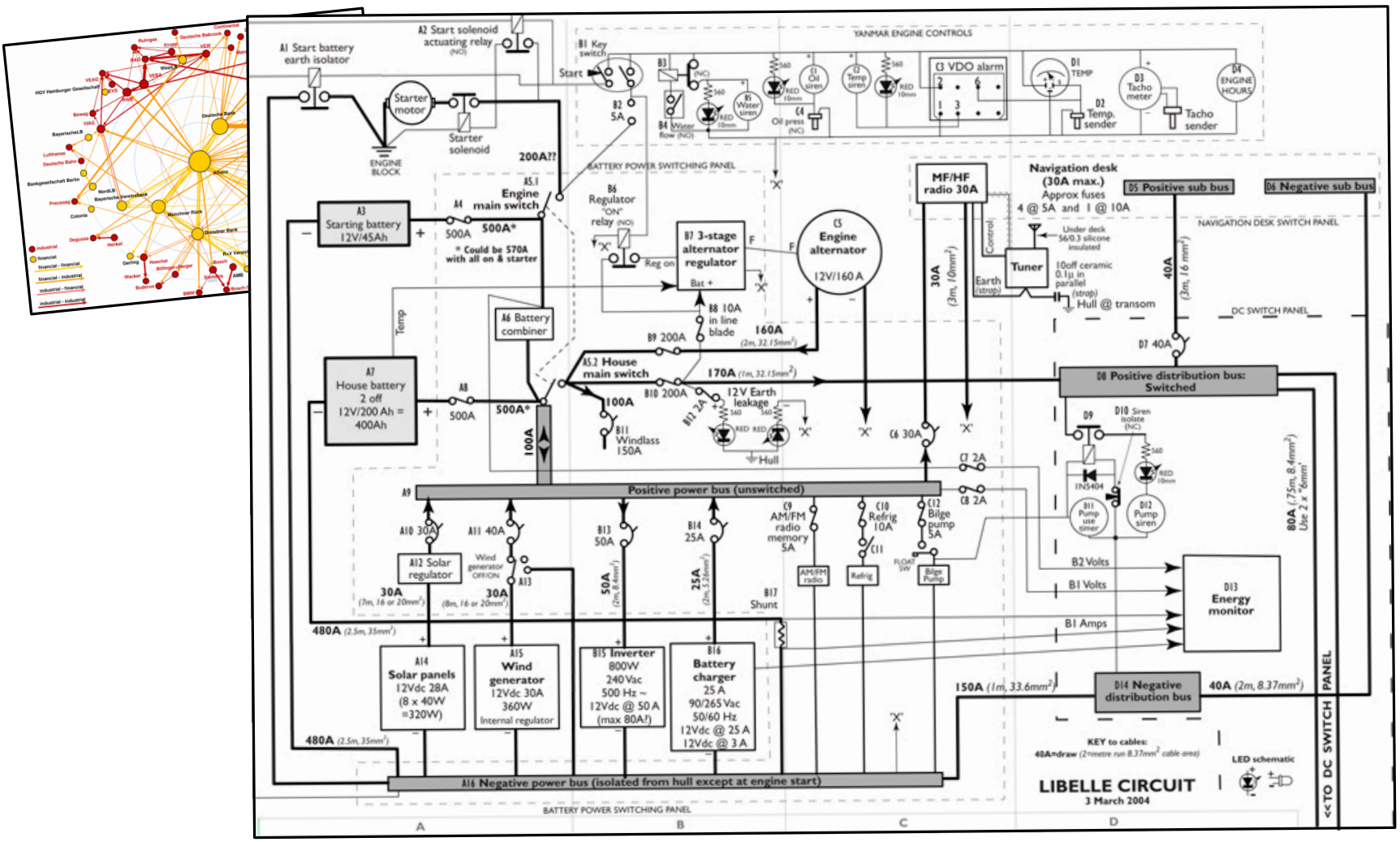
Why Draw Graphs?

Graphs are mathematical models of real physical and abstract networks (social networks, metabolical networks, VLSI-network, UML-diagrams, citation networks, ...)



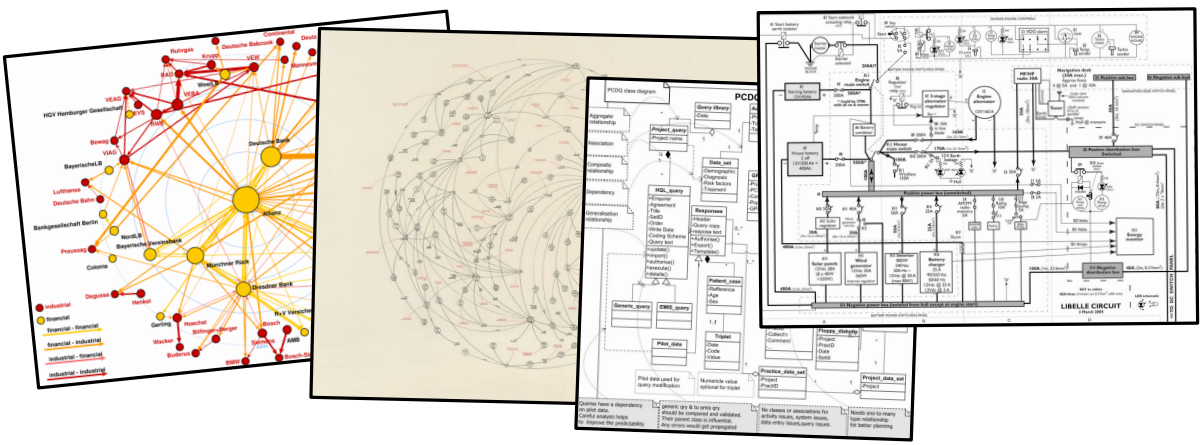
Why Draw Graphs?

Graphs are mathematical models of real physical and abstract networks (social networks, metabolical networks, VLSI-network, UML-diagrams, citation networks, ...)



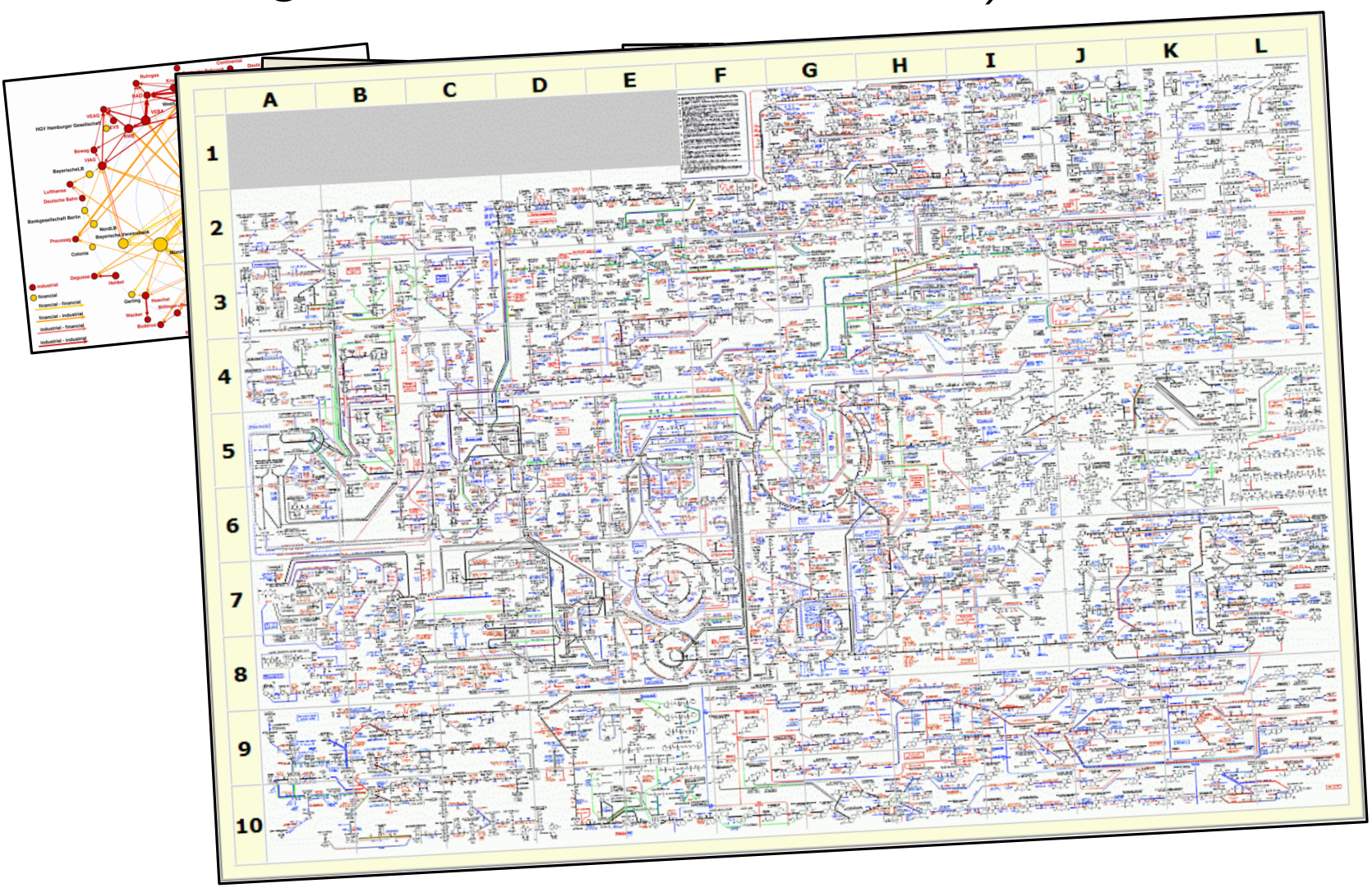
Why Draw Graphs?

Graphs are mathematical models of real physical and abstract networks (social networks, metabolical networks, VLSI-network, UML-diagrams, citation networks, ...)



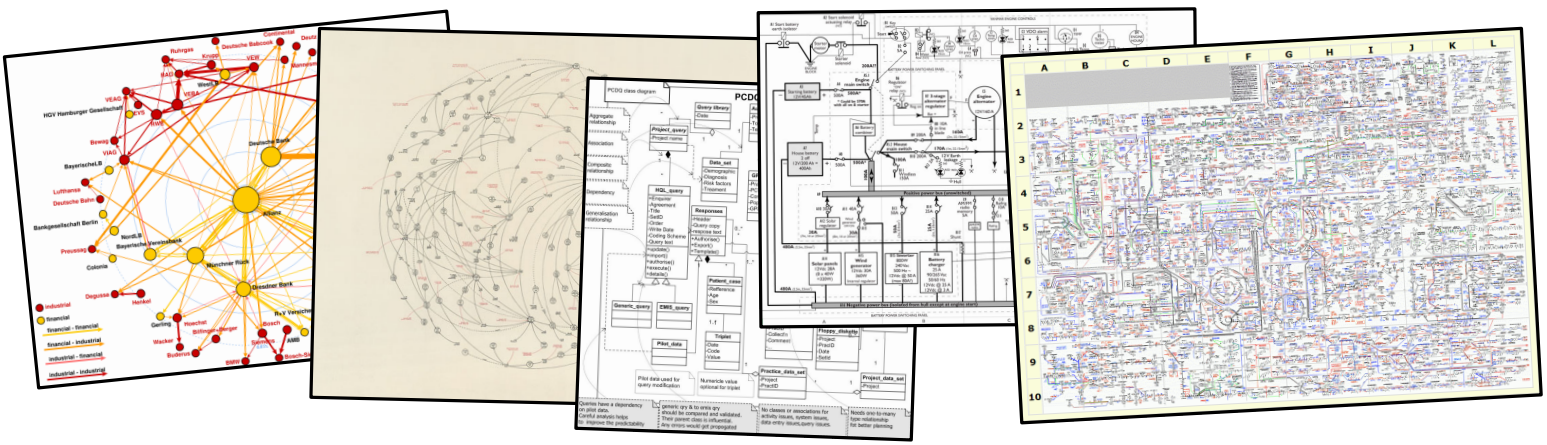
Why Draw Graphs?

Graphs are mathematical models of real physical and abstract networks (social networks, metabolical networks, VLSI-network, UML-diagrams, citation networks, ...)



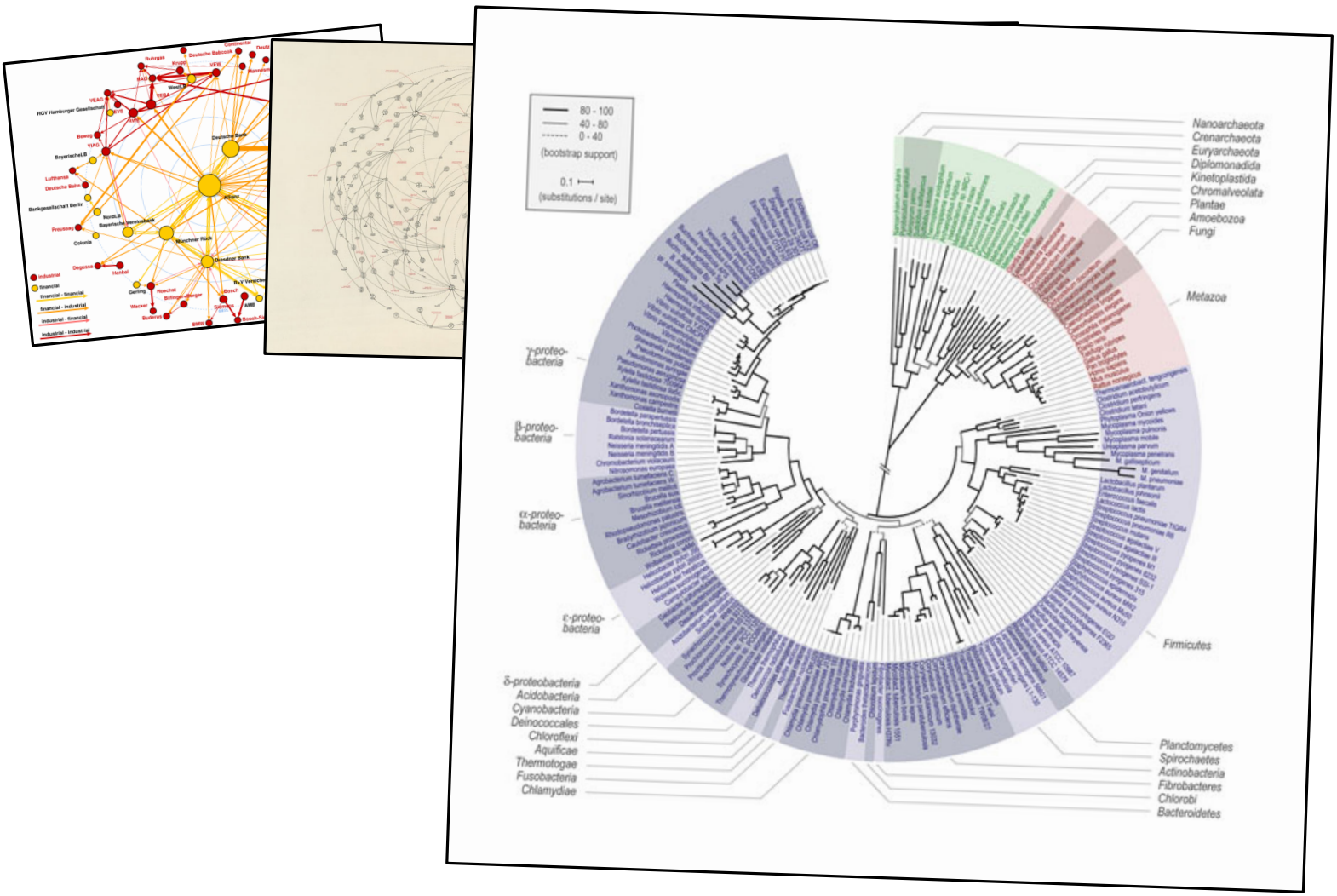
Why Draw Graphs?

Graphs are mathematical models of real physical and abstract networks (social networks, metabolical networks, VLSI-network, UML-diagrams, citation networks, ...)



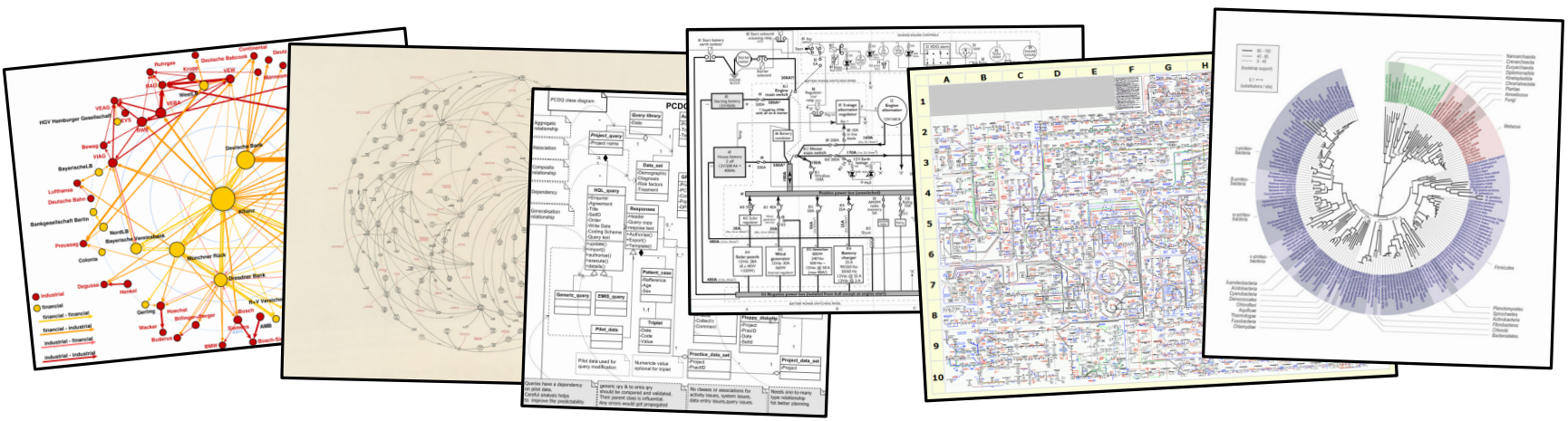
Why Draw Graphs?

Graphs are mathematical models of real physical and abstract networks (social networks, metabolical networks, VLSI-network, UML-diagrams, citation networks, ...)



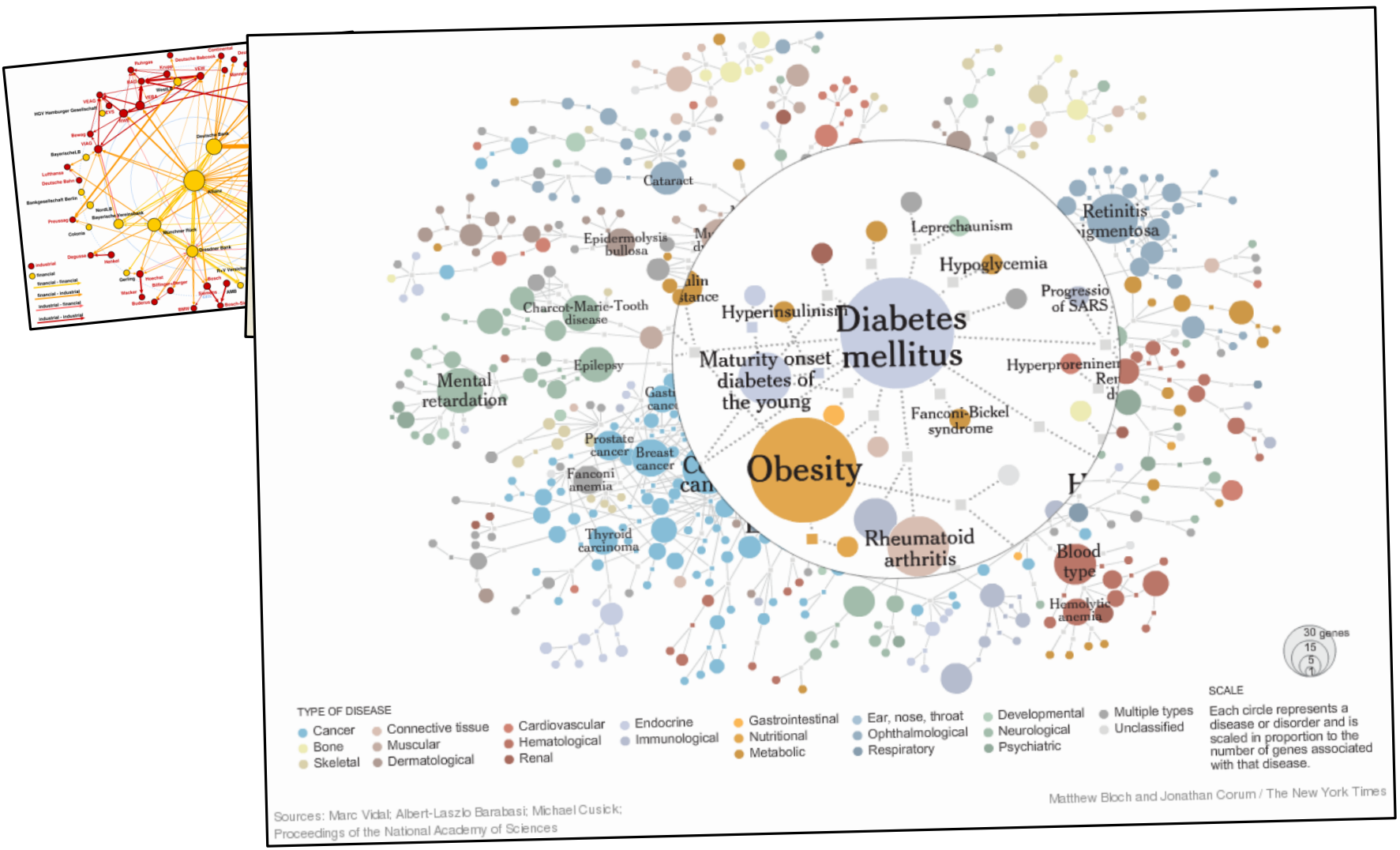
Why Draw Graphs?

Graphs are mathematical models of real physical and abstract networks (social networks, metabolical networks, VLSI-network, UML-diagrams, citation networks, ...)



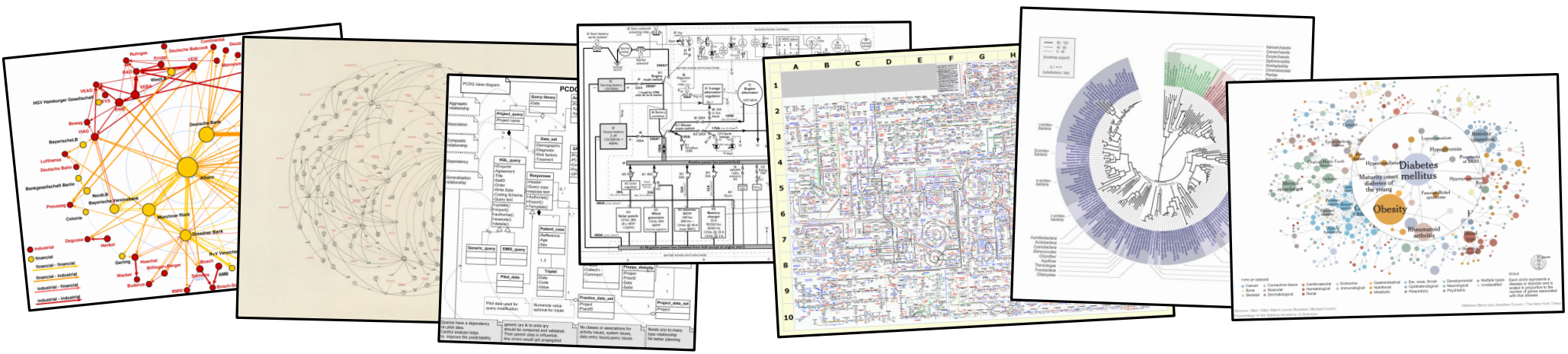
Why Draw Graphs?

Graphs are mathematical models of real physical and abstract networks (social networks, metabolical networks, VLSI-network, UML-diagrams, citation networks, ...)



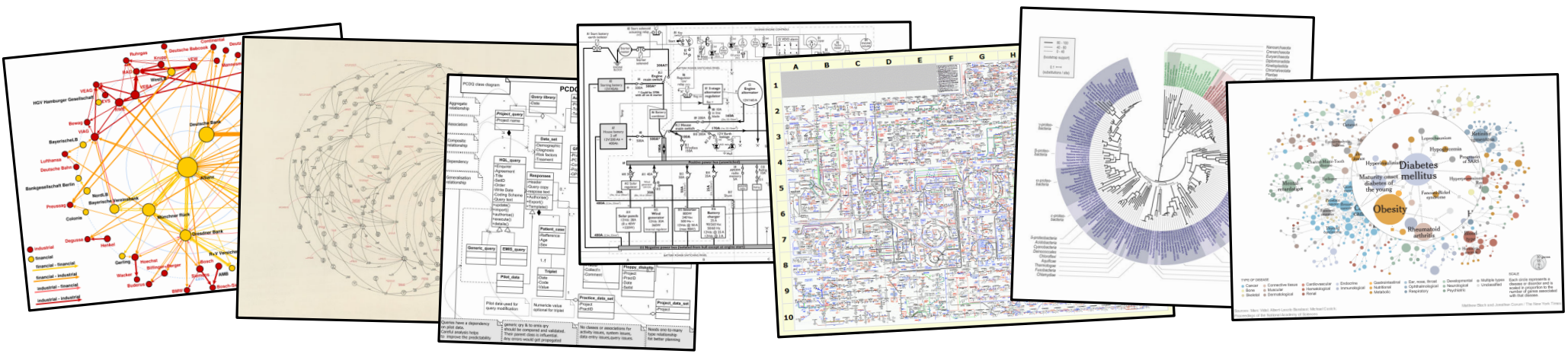
Why Draw Graphs?

Graphs are mathematical models of real physical and abstract networks (social networks, metabolical networks, VLSI-network, UML-diagrams, citation networks, ...)



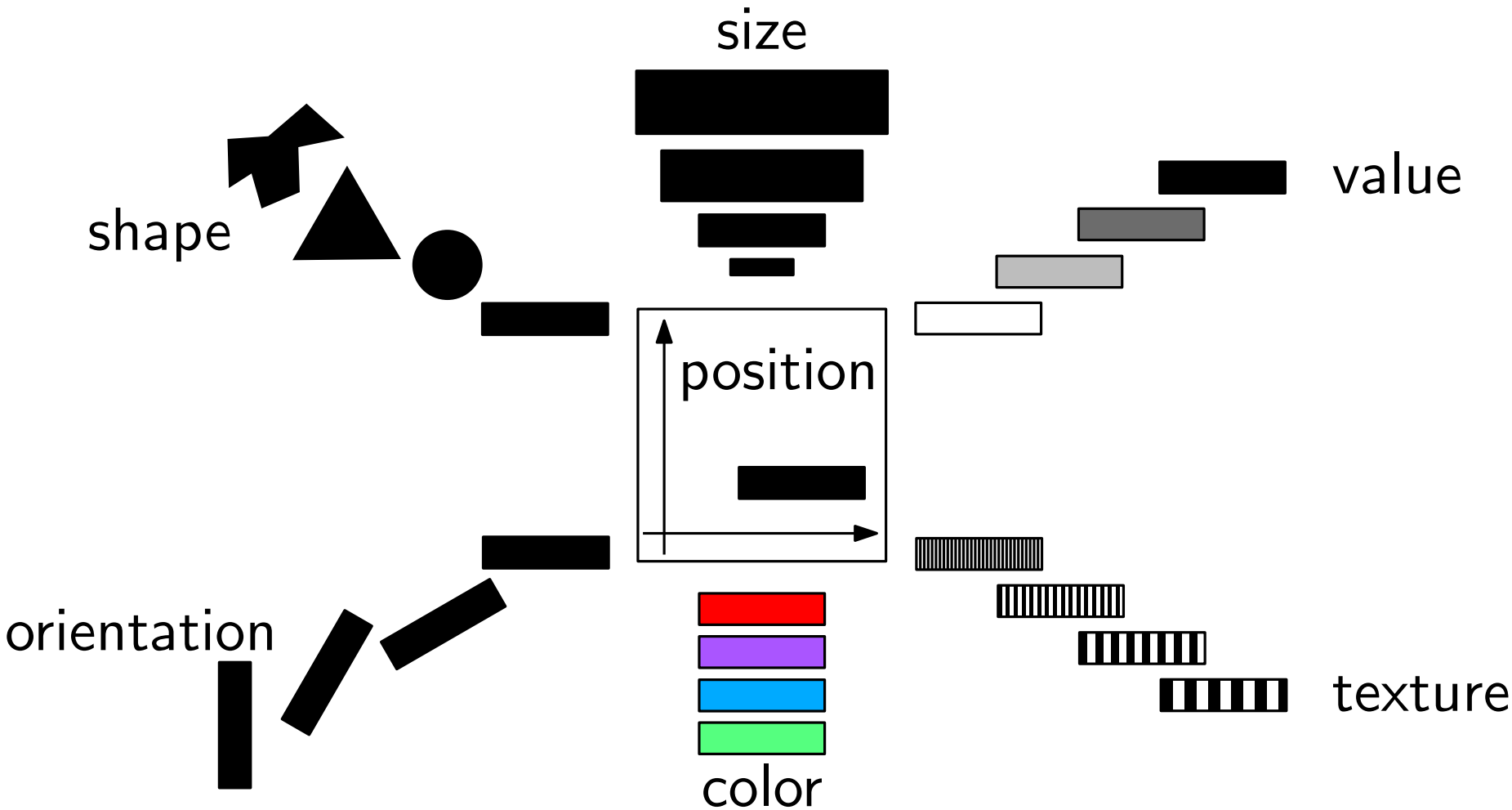
Why Draw Graphs?

Graphs are mathematical models of real physical and abstract networks (social networks, metabolical networks, VLSI-network, UML-diagrams, citation networks, ...)

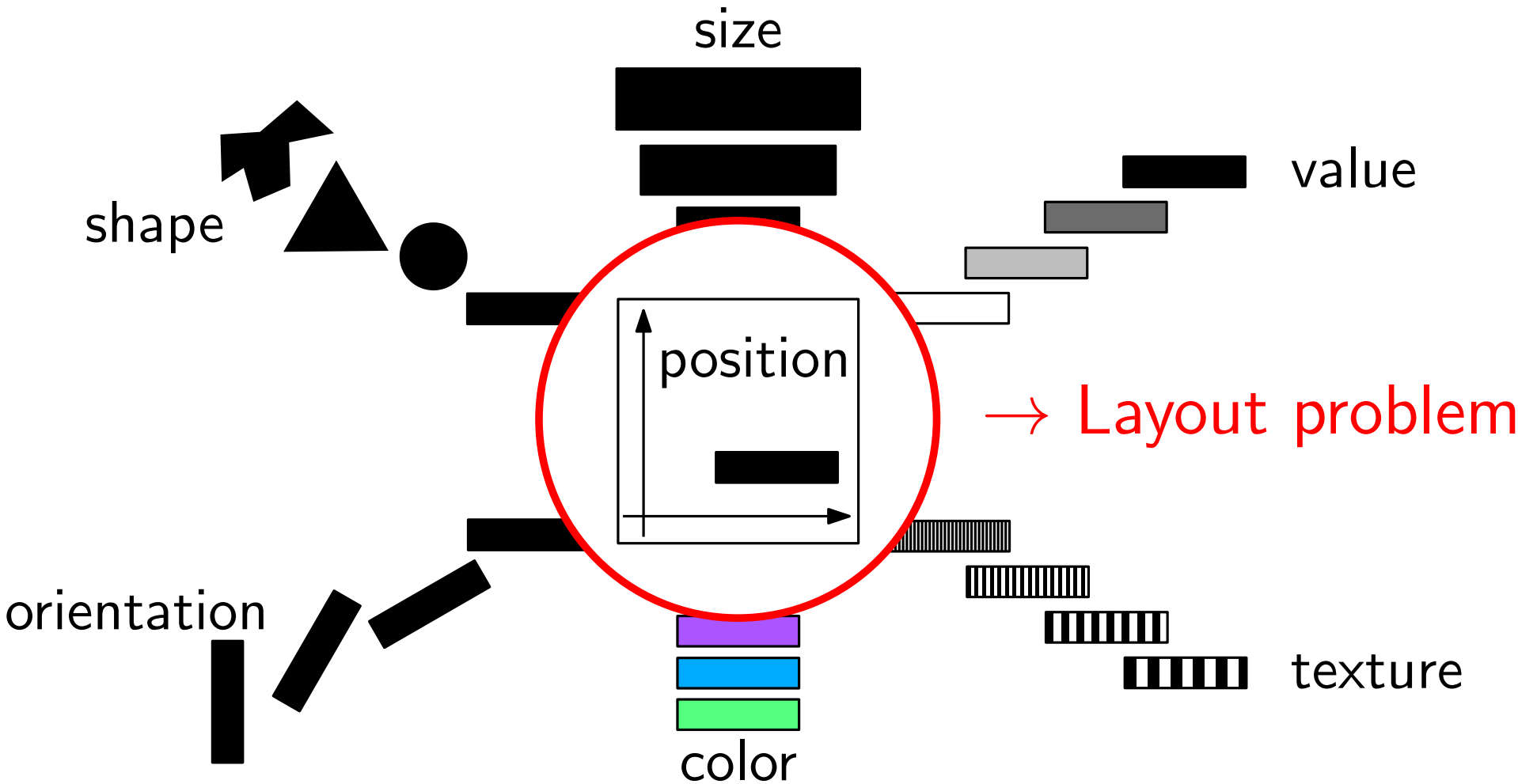


- **People think visually** – without a good visualization, complex graphs are not understandable to us
- A visualization helps to **communicate** and **explore** the graphs/networks
- We need **algorithms** to draw graphs, and make graphs and networks accessible to people

Visual Variables according to Bertin (1967)

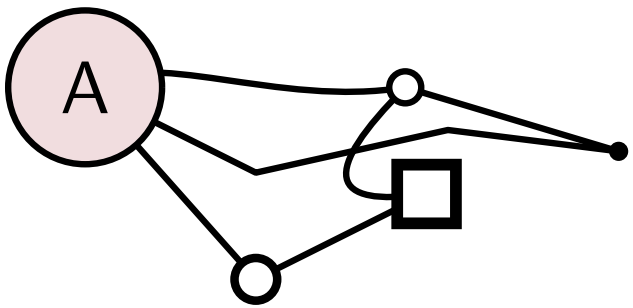


Visual Variables according to Bertin (1967)



Layout Problem

Here: drawing is always meant to be in **standard representation** (aka node-link diagram)



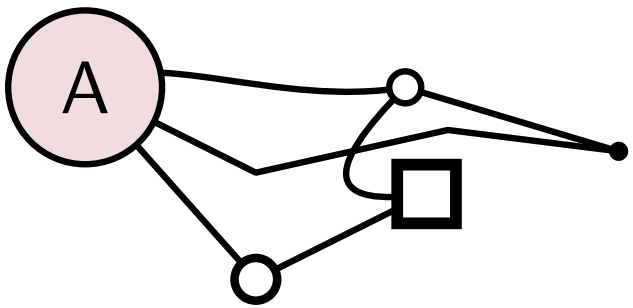
Graph visualization problem

given: graph $G = (V, E)$
find: **good** drawing Γ of G

- $\Gamma : V \to \mathbb{R}^2$, vertex $v \mapsto$ point $\Gamma(v)$
- $\Gamma : E \to$ curves in $\mathbb{R}^2$, edge $\{u, v\} \mapsto$ simple open curve $c_{uv} : [0, 1] \to \mathbb{R}^2$ where $c_{uv}(0) = \Gamma(u)$ and $c_{uv}(1) = \Gamma(v)$

Layout Problem

Here: drawing is always meant to be in **standard representation** (aka node-link diagram)



Graph visualization problem

given: graph $G = (V, E)$
find: **good** drawing Γ of G

- $\Gamma : V \to \mathbb{R}^2$, vertex $v \mapsto$ point $\Gamma(v)$
- $\Gamma : E \to$ curves in $\mathbb{R}^2$, edge $\{u, v\} \mapsto$ simple open curve $c_{uv} : [0, 1] \to \mathbb{R}^2$ where $c_{uv}(0) = \Gamma(u)$ and $c_{uv}(1) = \Gamma(v)$

But what is a **good** drawing?

Let's draw some graphs to find out

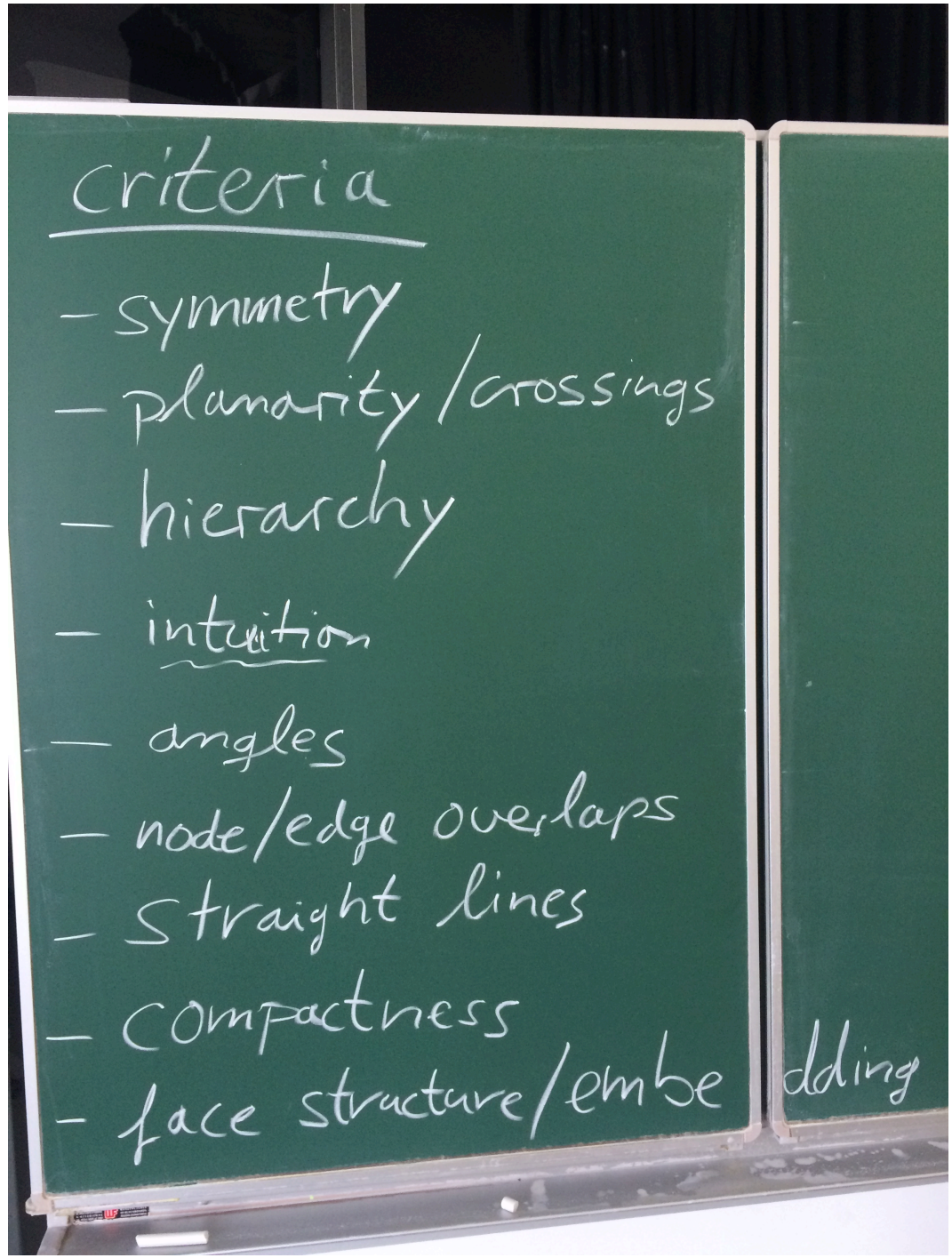
Central questions

- what makes a good drawing?
- what to avoid in a good drawing?
- are the criteria quantifiable?

Task:
Draw the graphs by hand as nicely as possible.

- work with your neighbor
- input is adjacency matrix or list

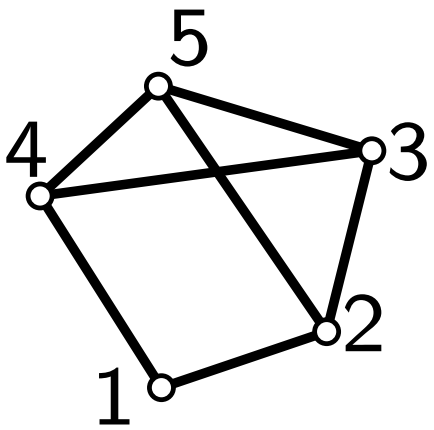
10 min

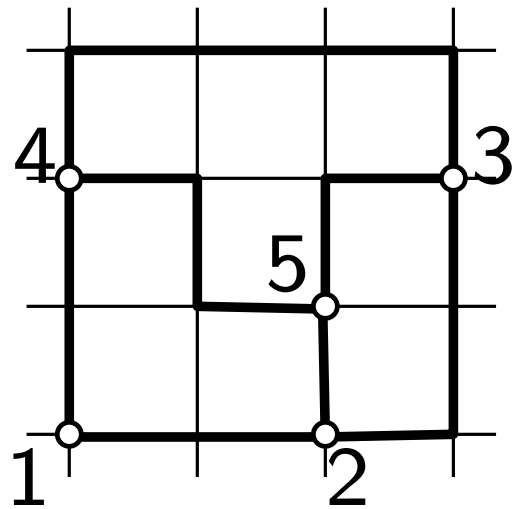

criteria
- symmetry
- planarity / crossings
- hierarchy
- intuition
- angles
- node/edge overlaps
- straight lines
- compactness
- face structure/embe
dding

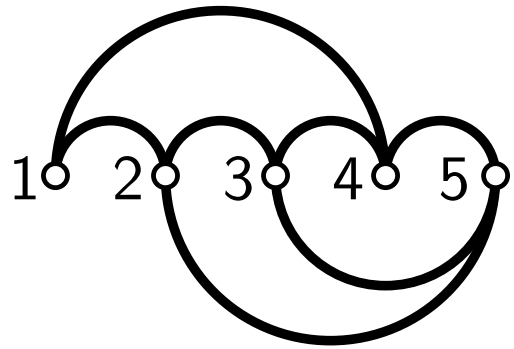
Quality Criteria for Graph Drawings

1) **Drawing conventions**, required properties, for example

- straight-line edges
- orthogonal edges (with 90° bends)
- grid drawings
- crossing-free
- ...





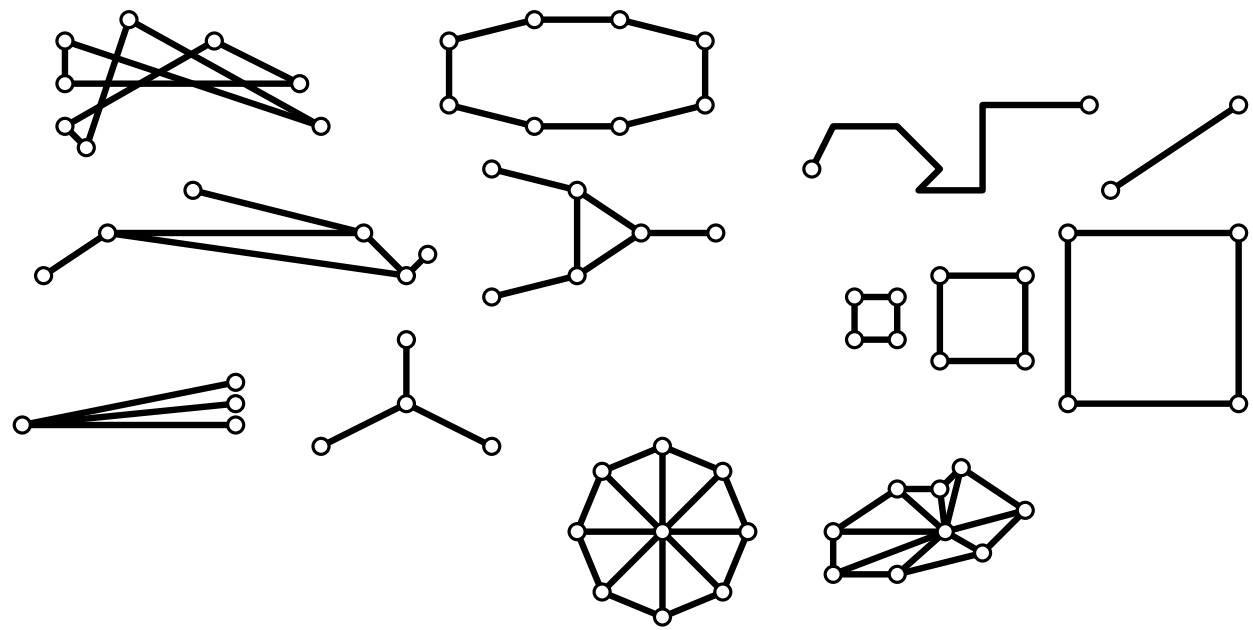


Quality Criteria for Graph Drawings

1) **Drawing conventions**, required properties

2) **Aesthetics** (to be optimized), for example:

- number of crossing
- number of bends
- uniform edge length
- area/length
- angular resolution
- symmetries
- ...



Quality Criteria for Graph Drawings

1) **Drawing conventions**, required properties

2) **Aesthetics** (to be optimized)

3) **Partial/local constraints**, for example:

- constraints on positions of some vertices
- constrained relative positions of vertices
- groups of vertices drawn close to each other

Layout Problem (2nd attempt)

Graph visualization problem

given: graph $G = (V, E)$
find: drawing Γ of G that

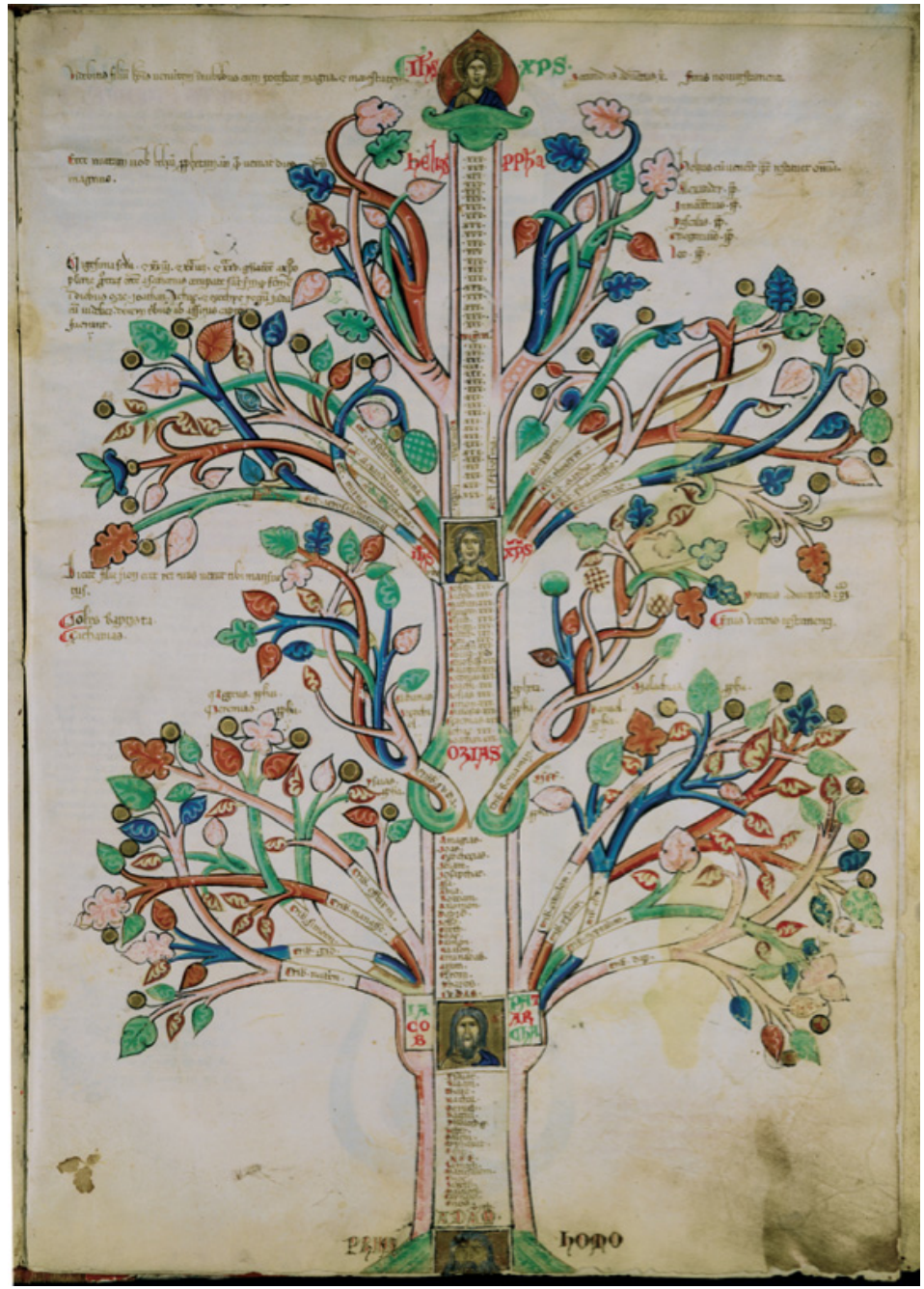
- complies with the given drawing conventions
- optimizes the given aesthetics
- satisfies the partial/local constraints

$\rightarrow$ often lead to NP-hard optimization problems!
$\rightarrow$ often several competing criteria

Example Gallery*

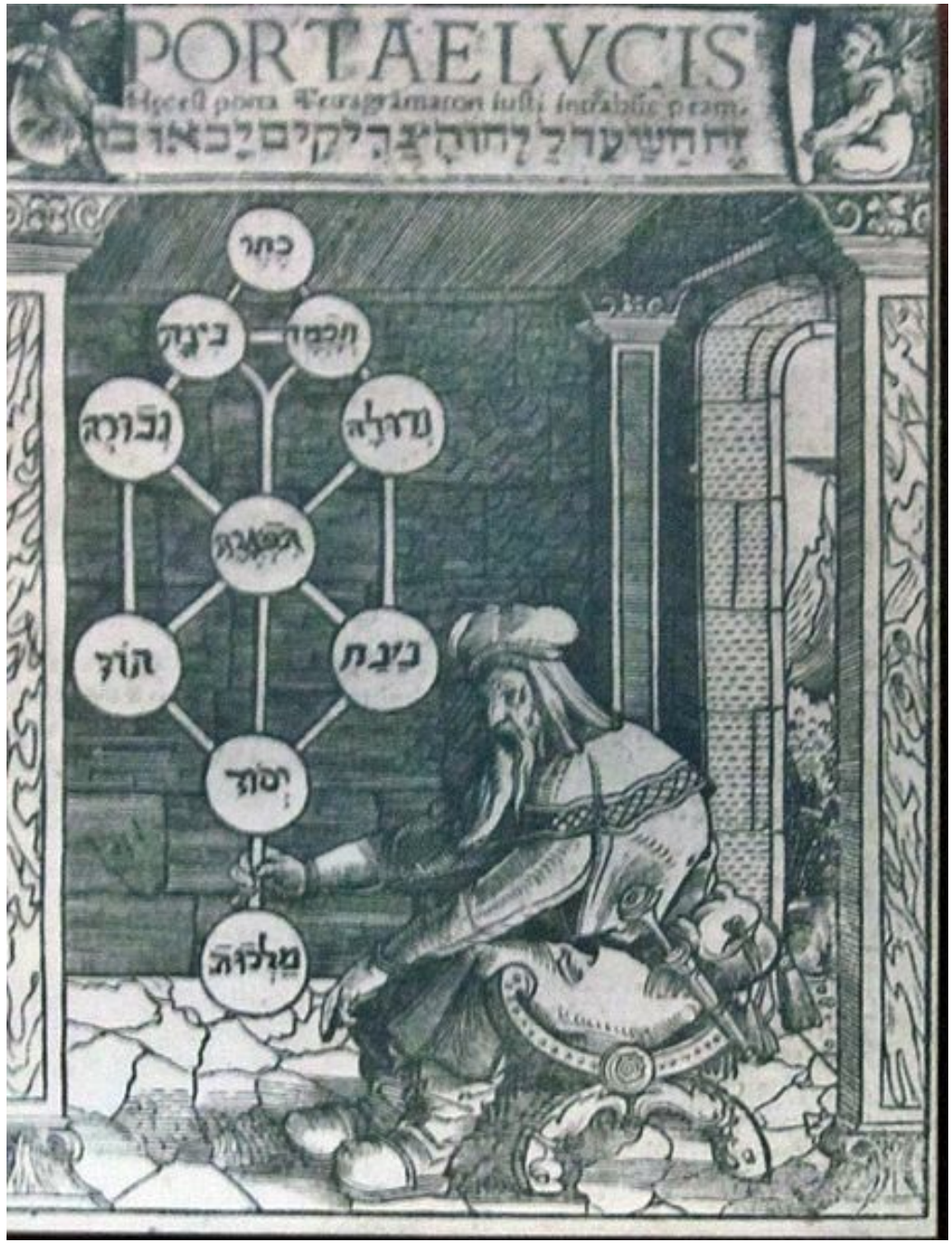
$\star$ for more examples see, e.g., collections like `www.visualcomplexity.com`

Biblical Characters and Events (1202)



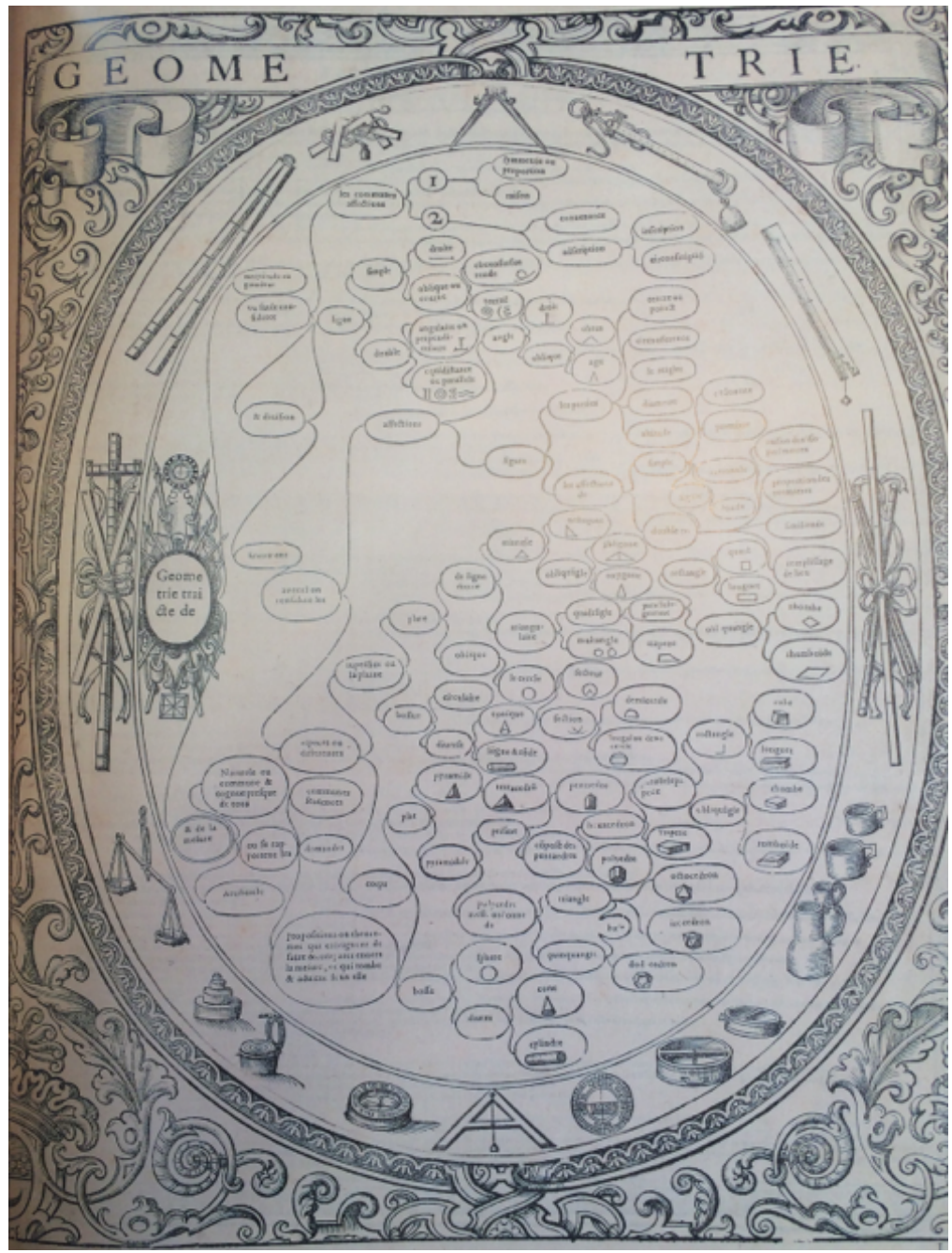
source: Joachim de Fiore

Tree of Life (1516)



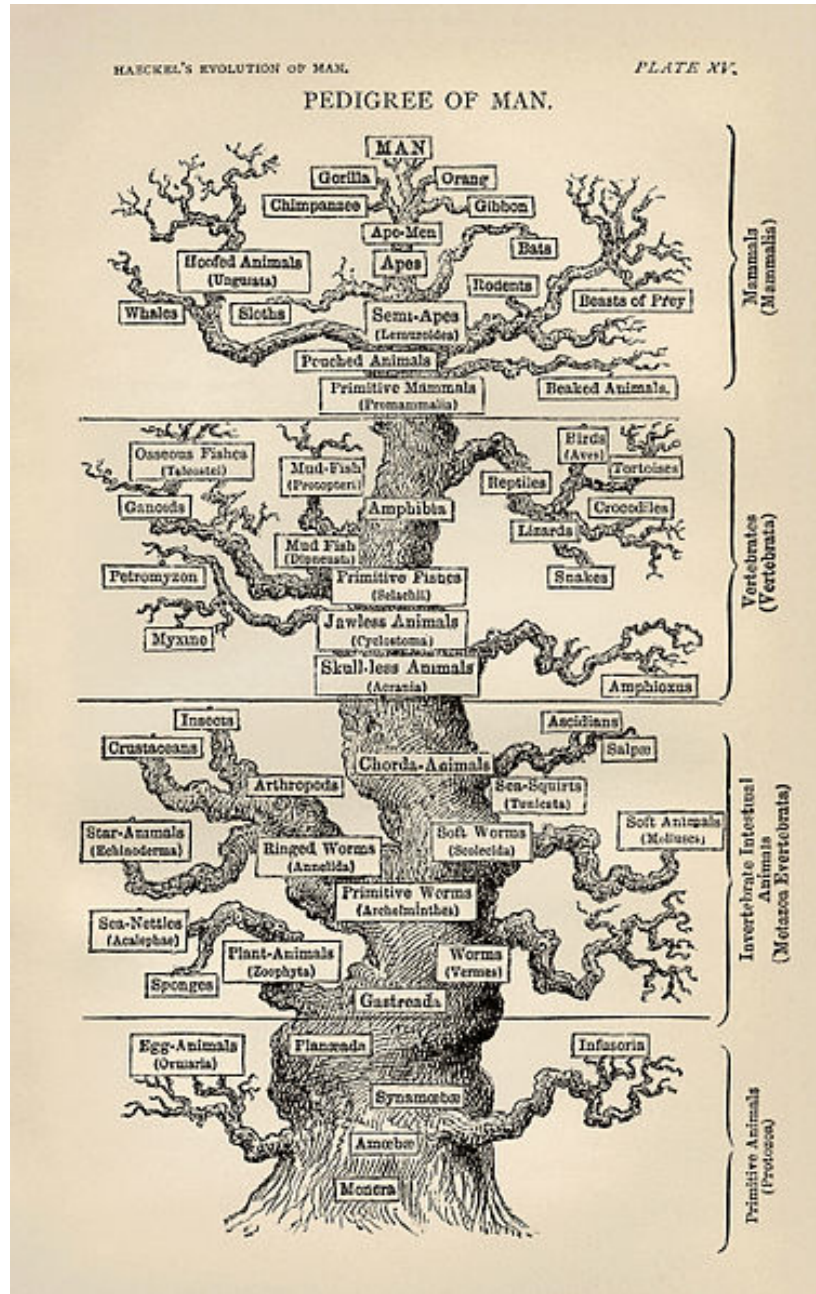
source: Paul Riccius, Portae Lucis

Geometrical Concepts (1587)



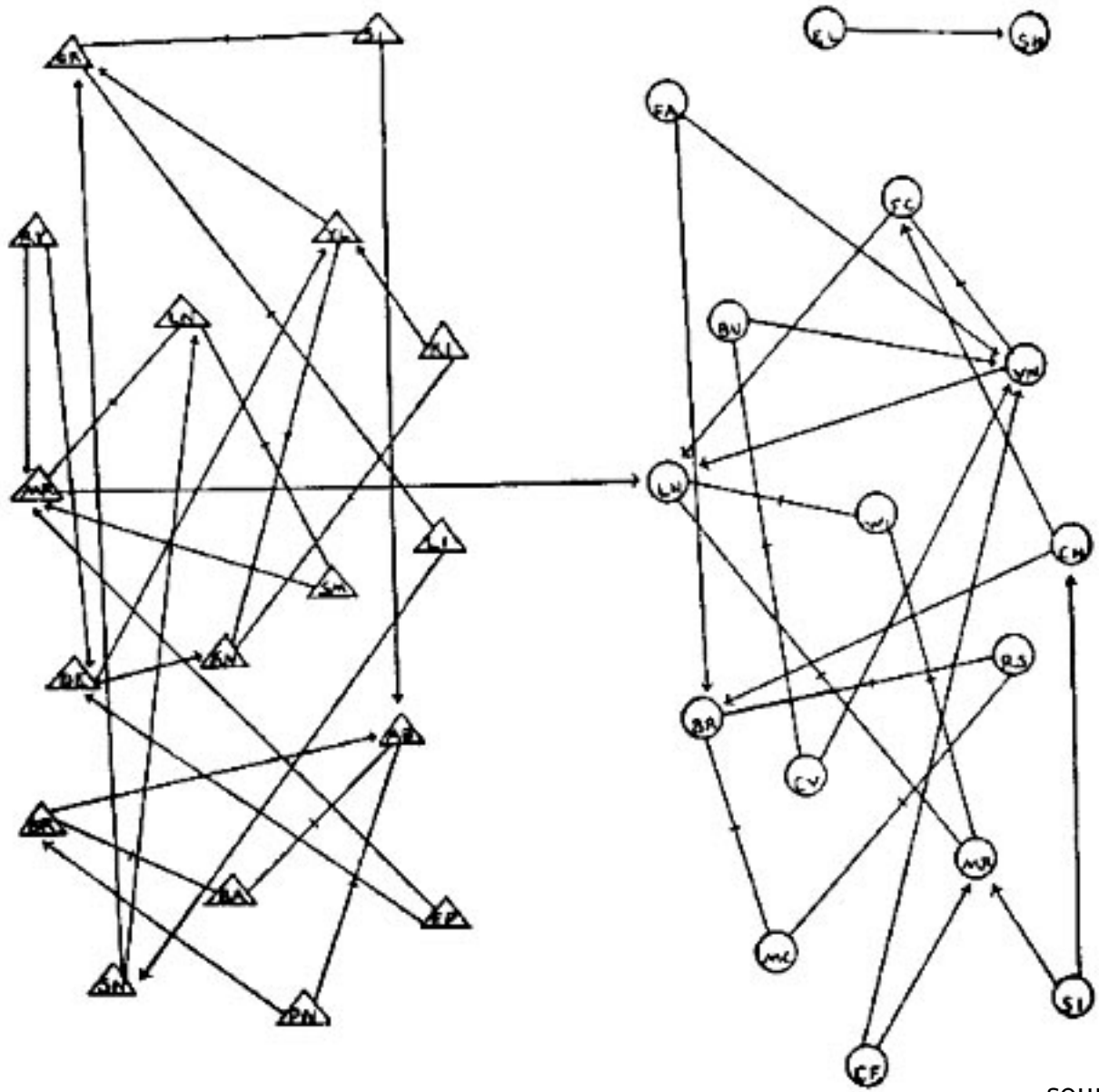
source: Christophe de Savigny

Evolutionary Tree (1879)



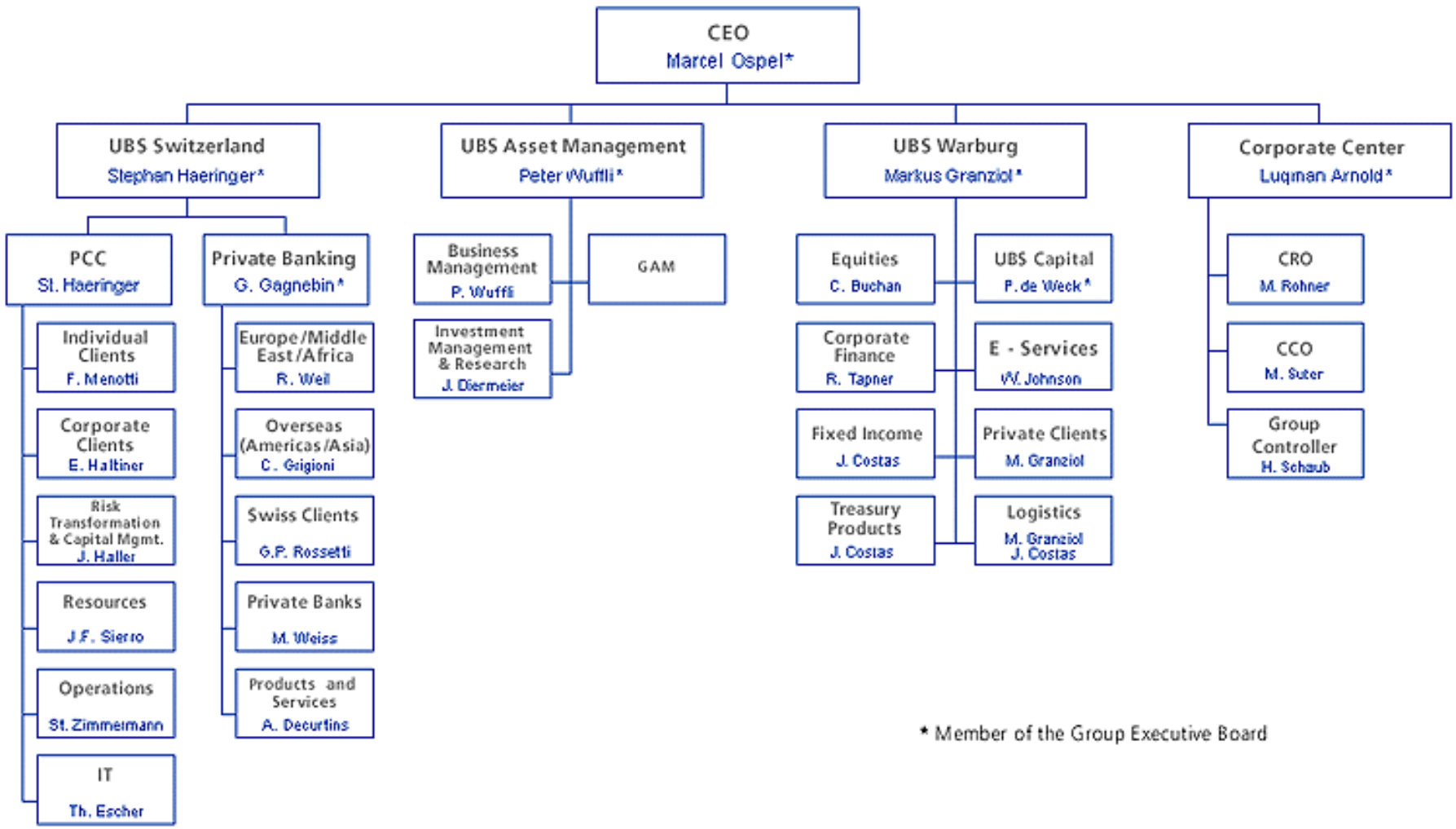
source: Ernst Haeckel

Sociogram (1933)



source: Moreno

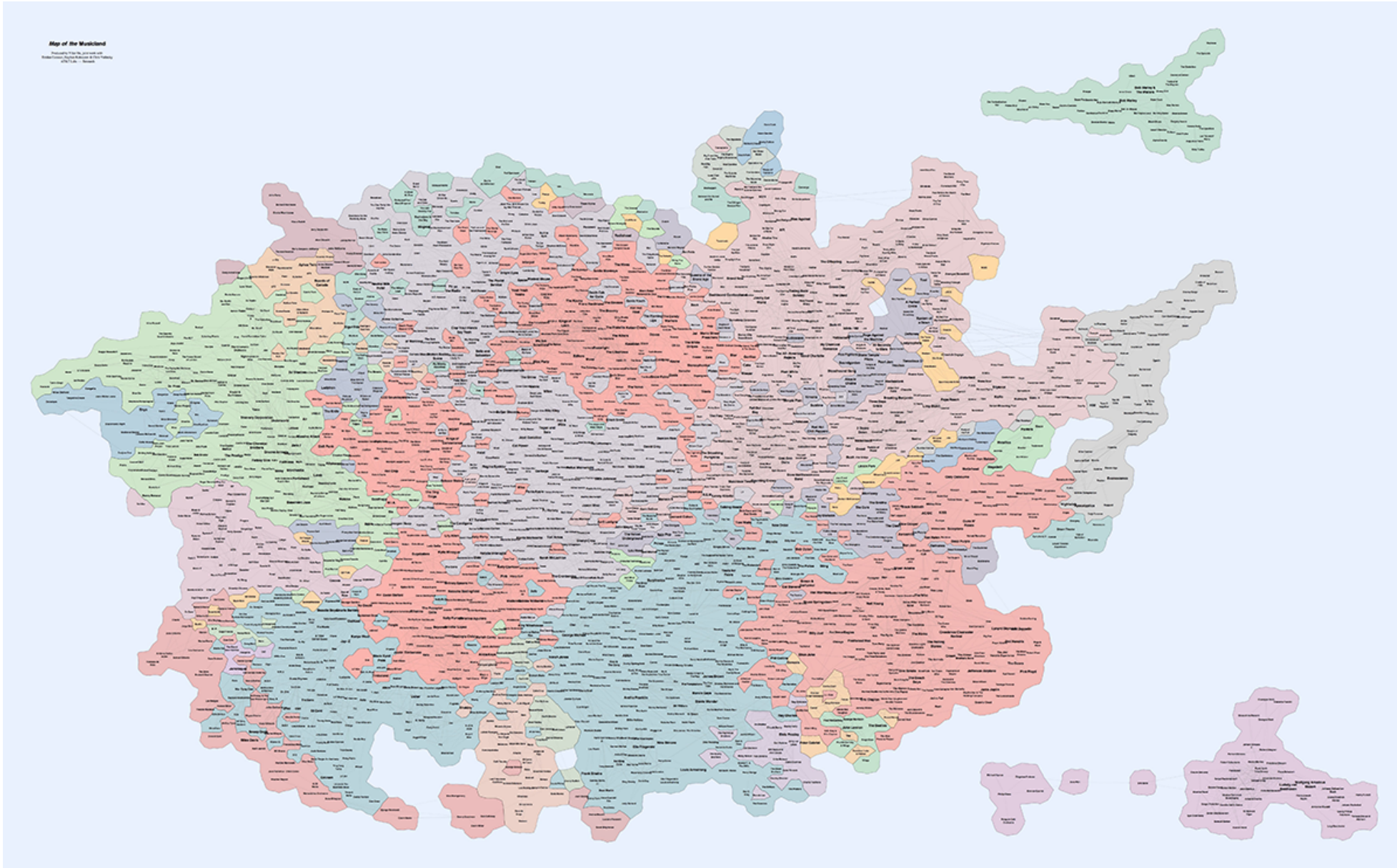
Social Network – Organigram UBS



CPAN Developer Graph

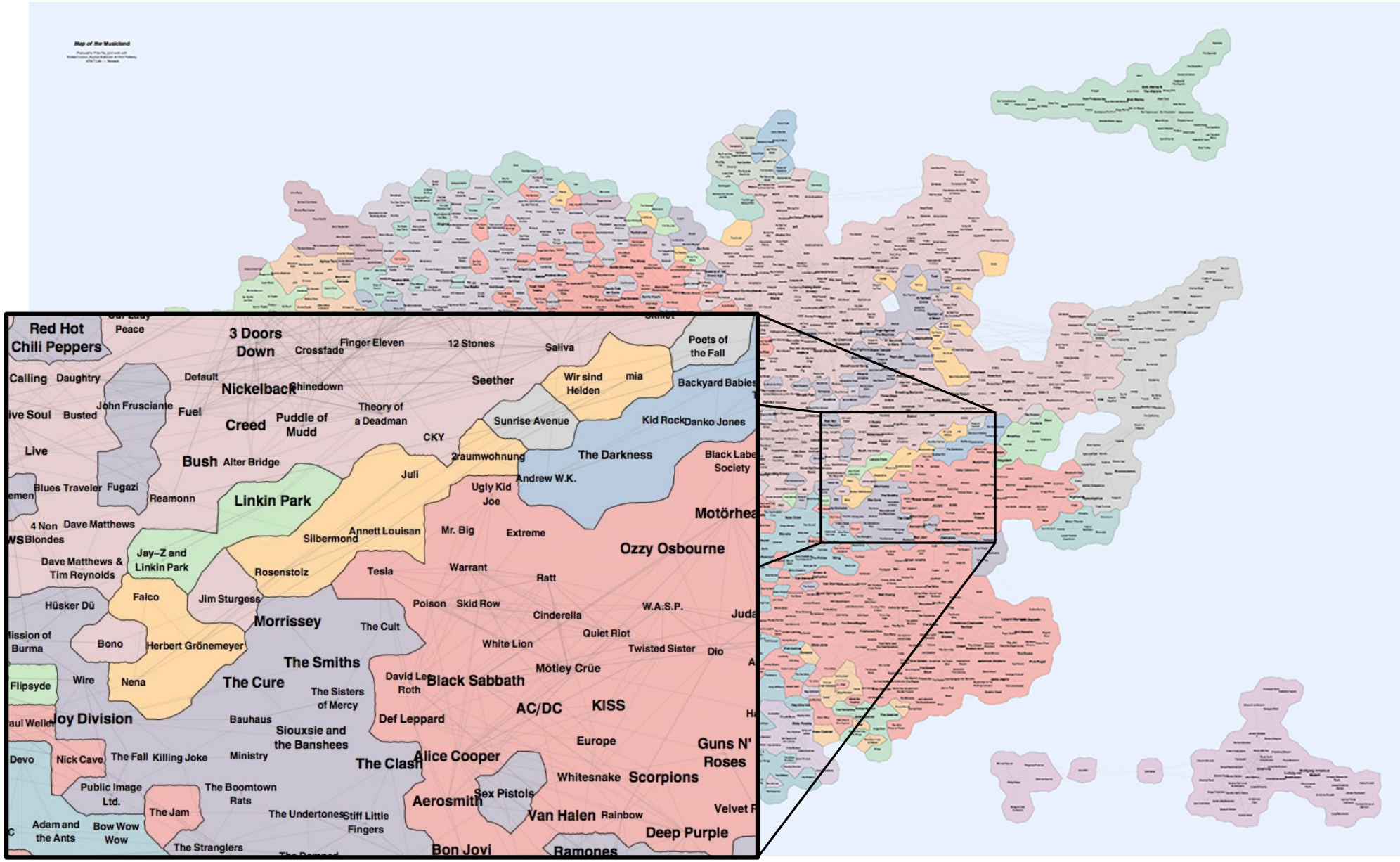
source: cpan-explorer.org

last.fm music graph as political map



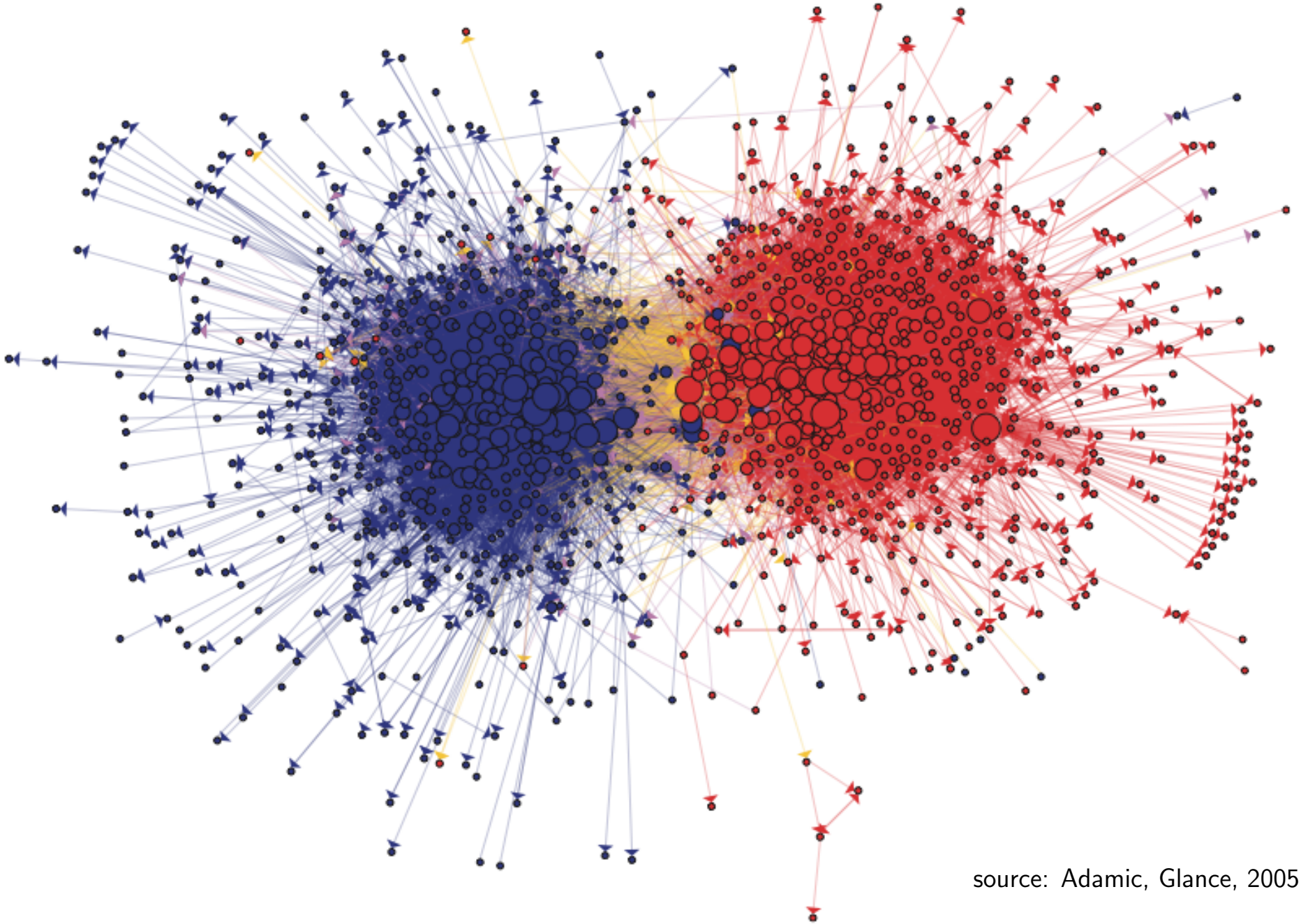
(Gansner, Hu, Kobourov: GMap, 2009)

last.fm music graph as political map



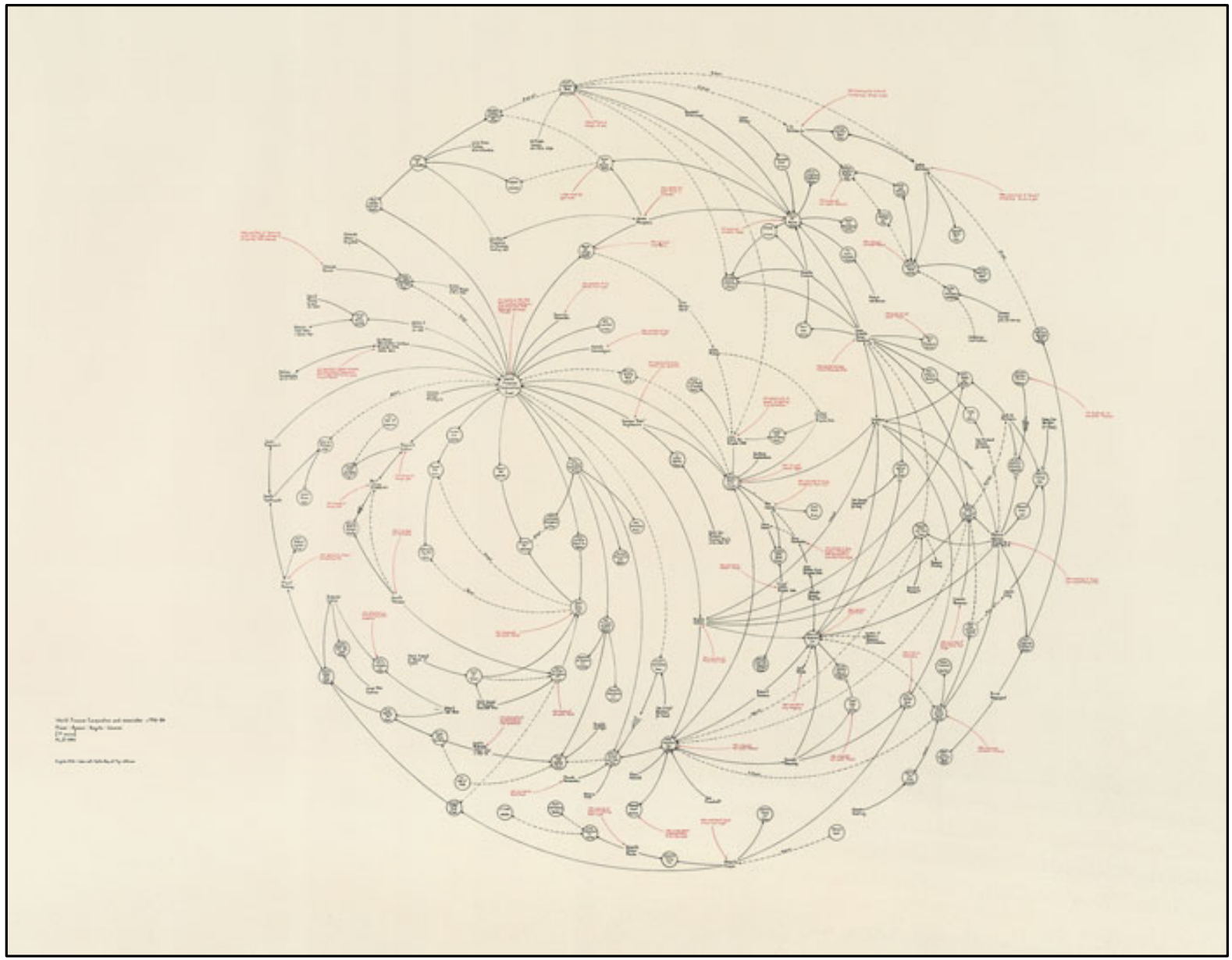
(Gansner, Hu, Kobourov: GMap, 2009)

Blogosphere 2004 Elections USA



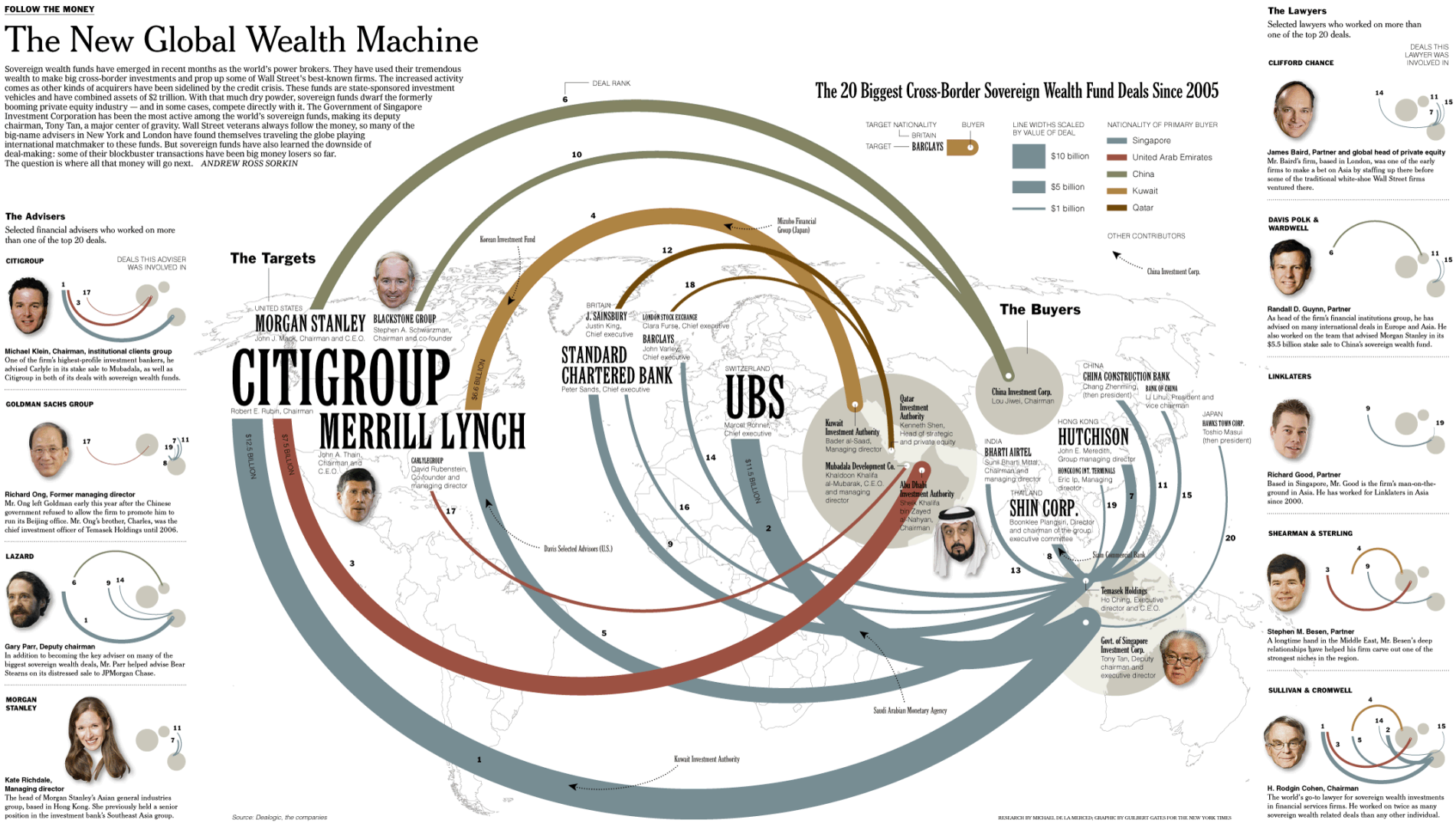
source: Adamic, Glance, 2005

Social Network – World Financial System

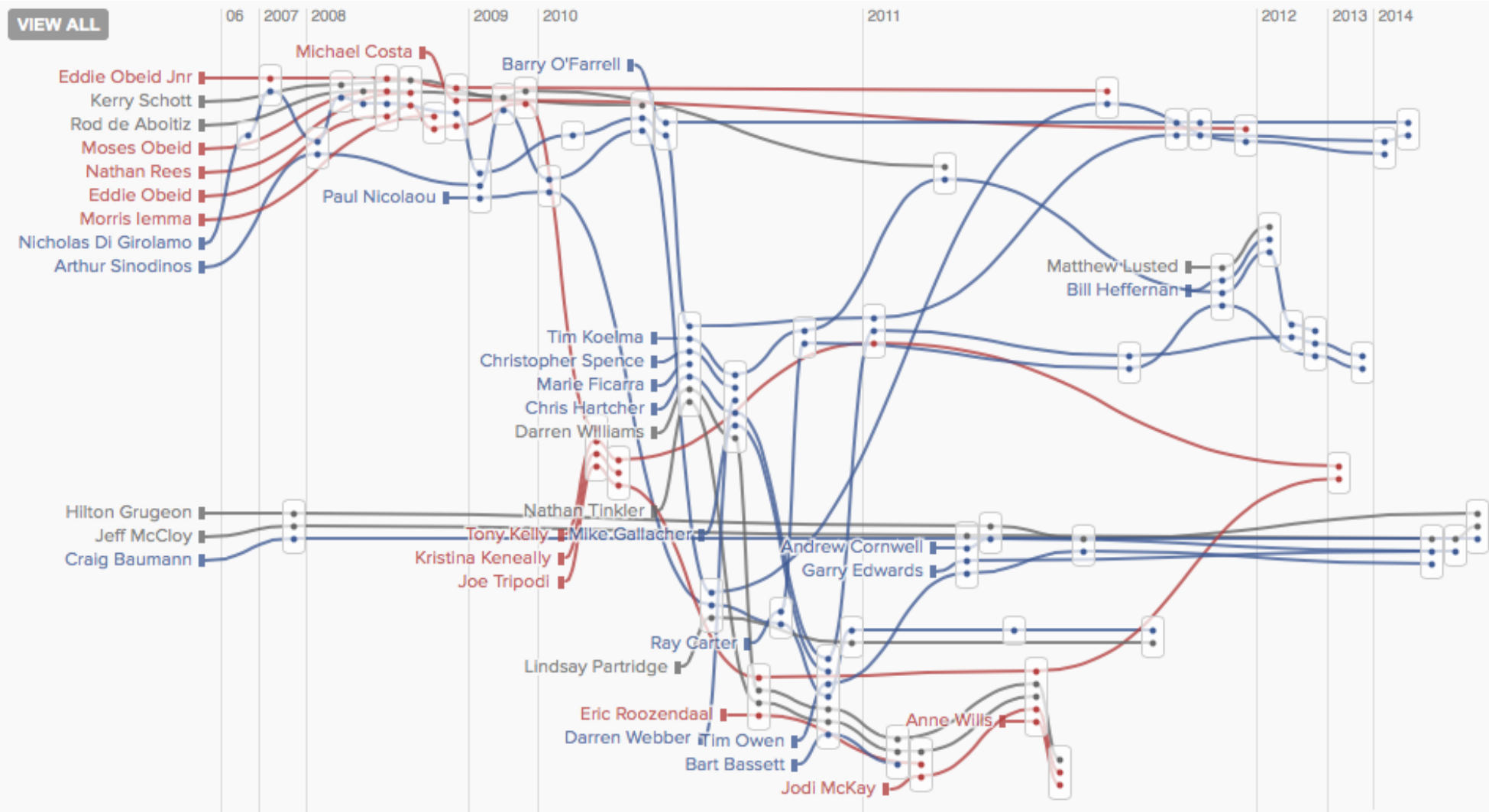


World Finance Corporation
© Mark Lombardi

Social Networks – State Funds

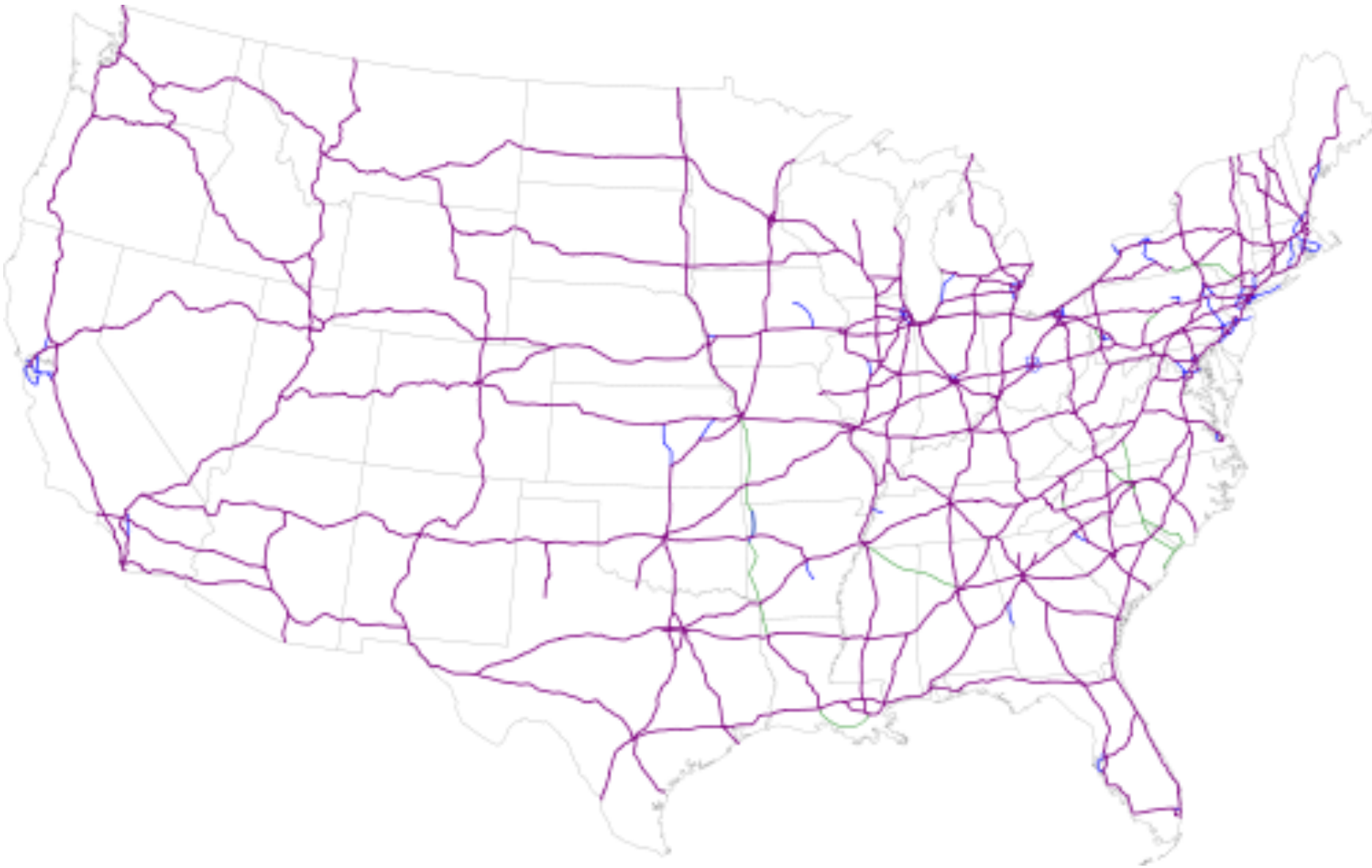


Temporal Graph Layout: Storylines

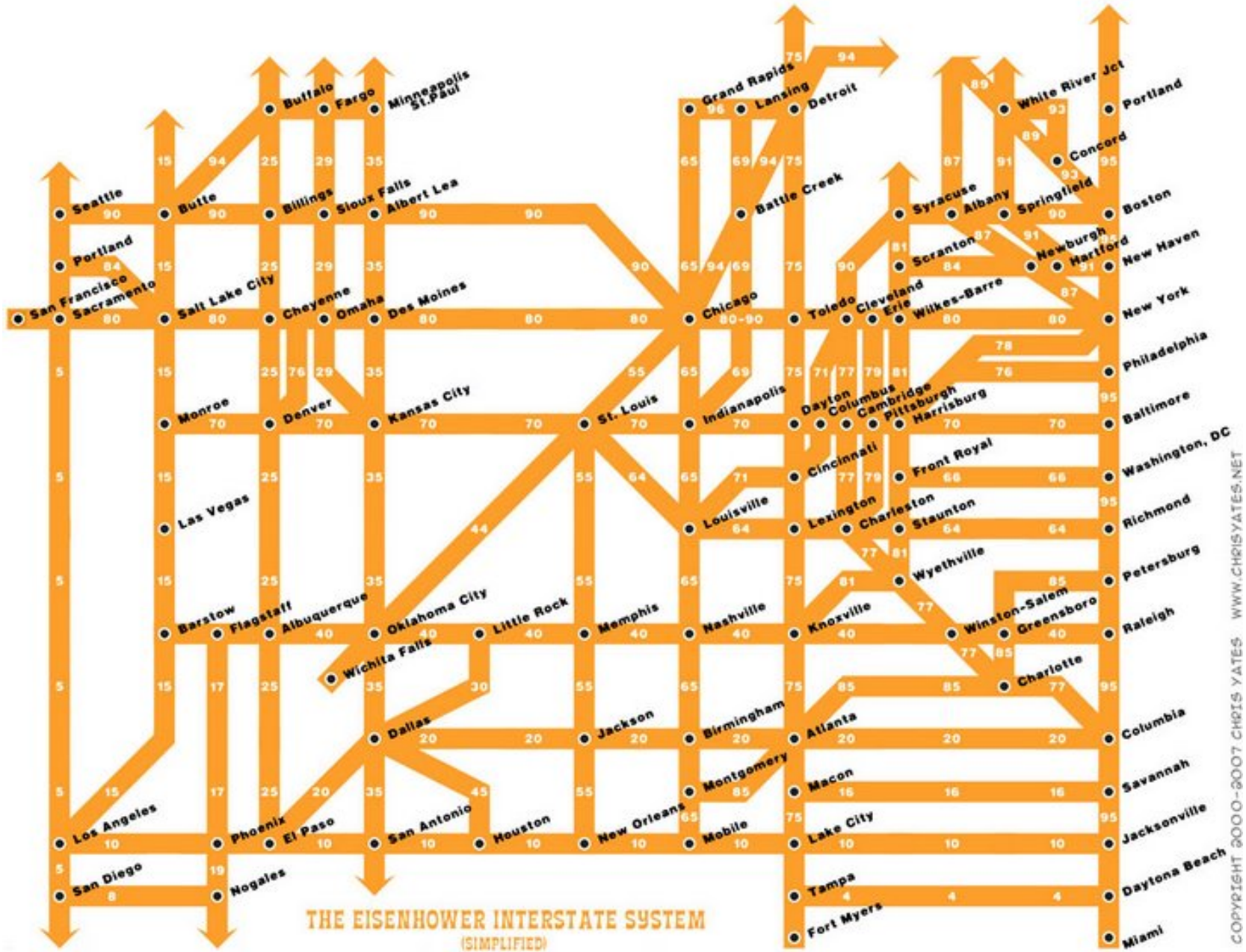


source: ABC news, Australia

Traffic Network – Highways USA



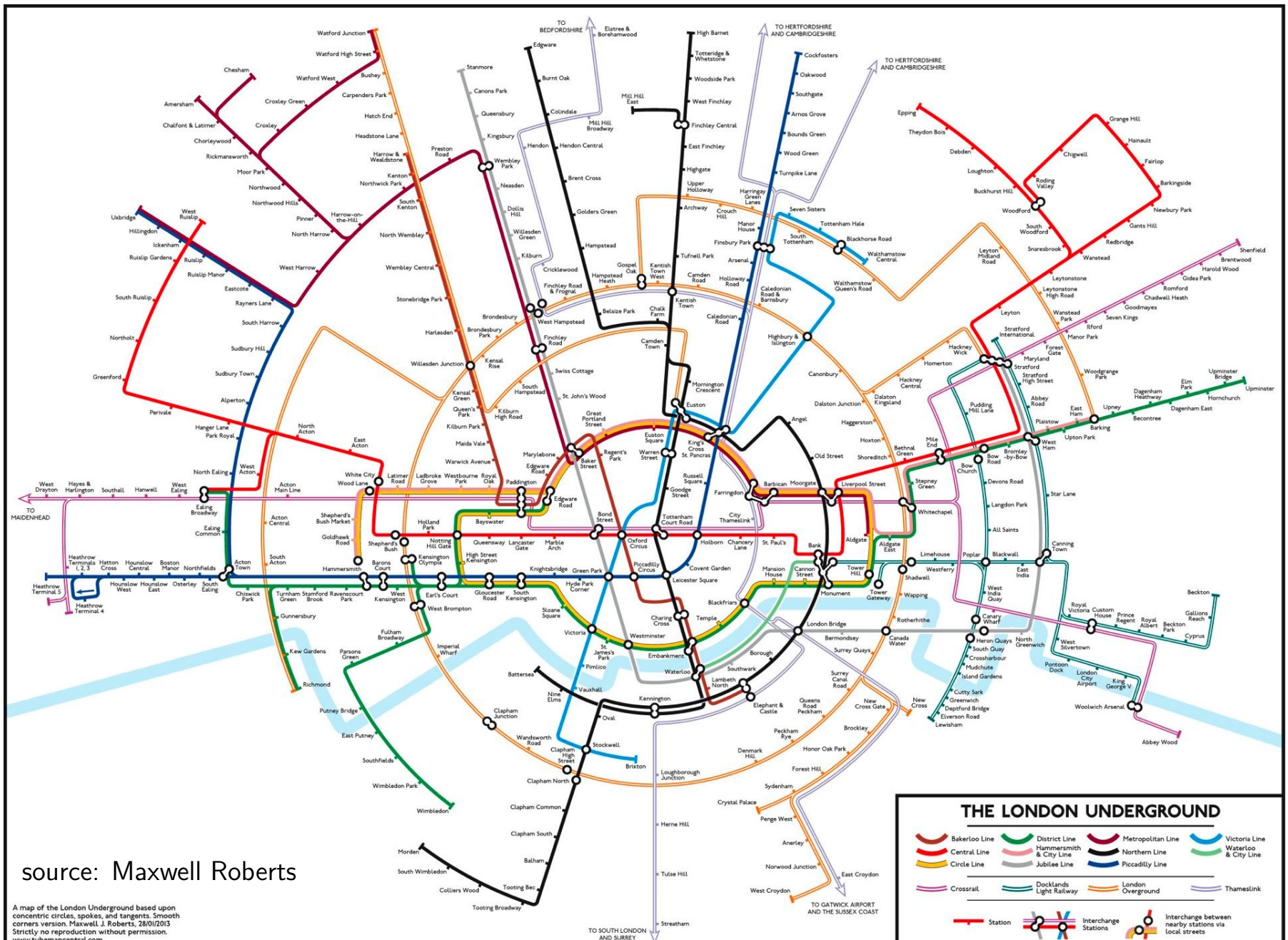
Traffic Network – Highways USA



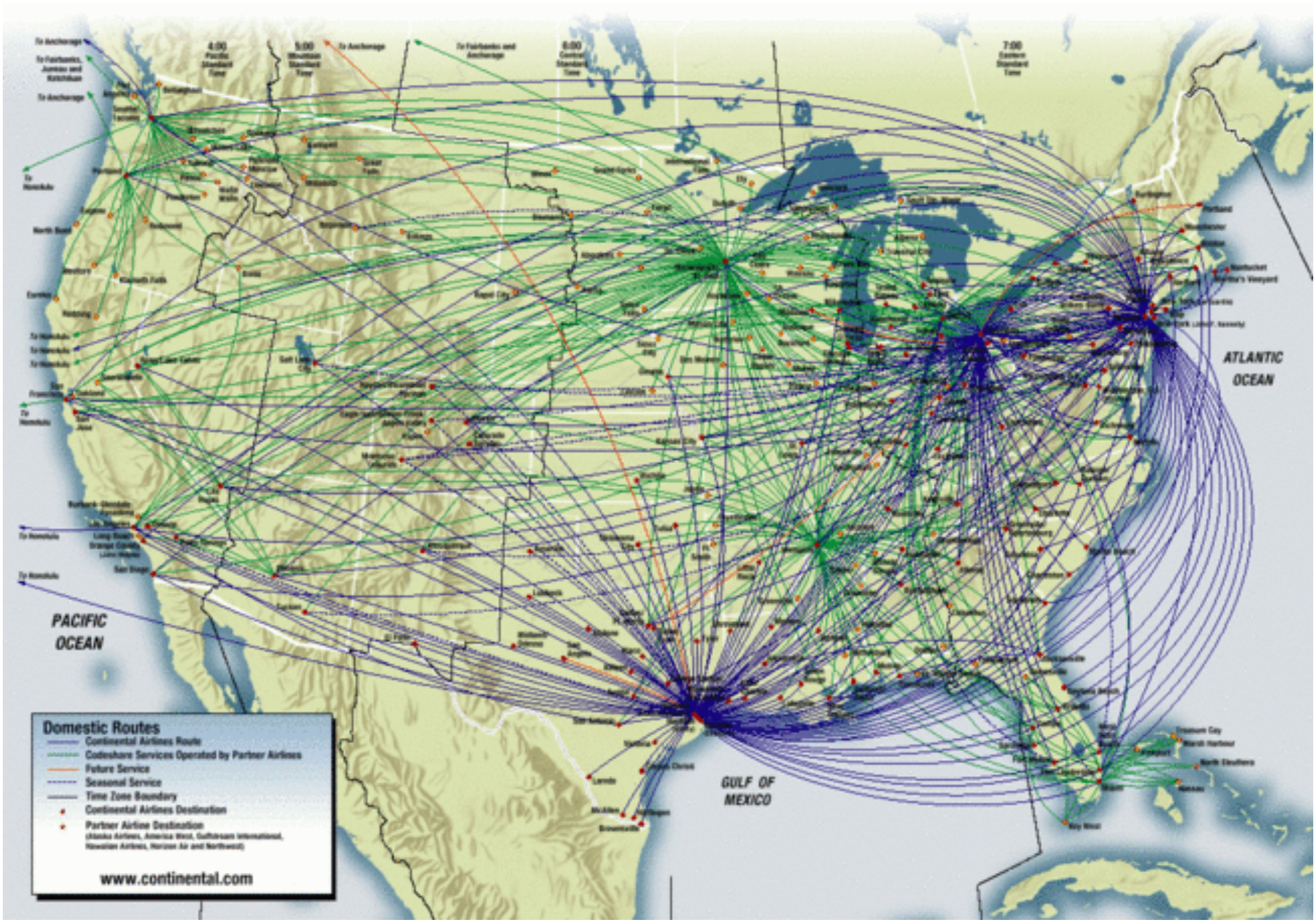
London Tube Map (1933)

source: Henry Beck

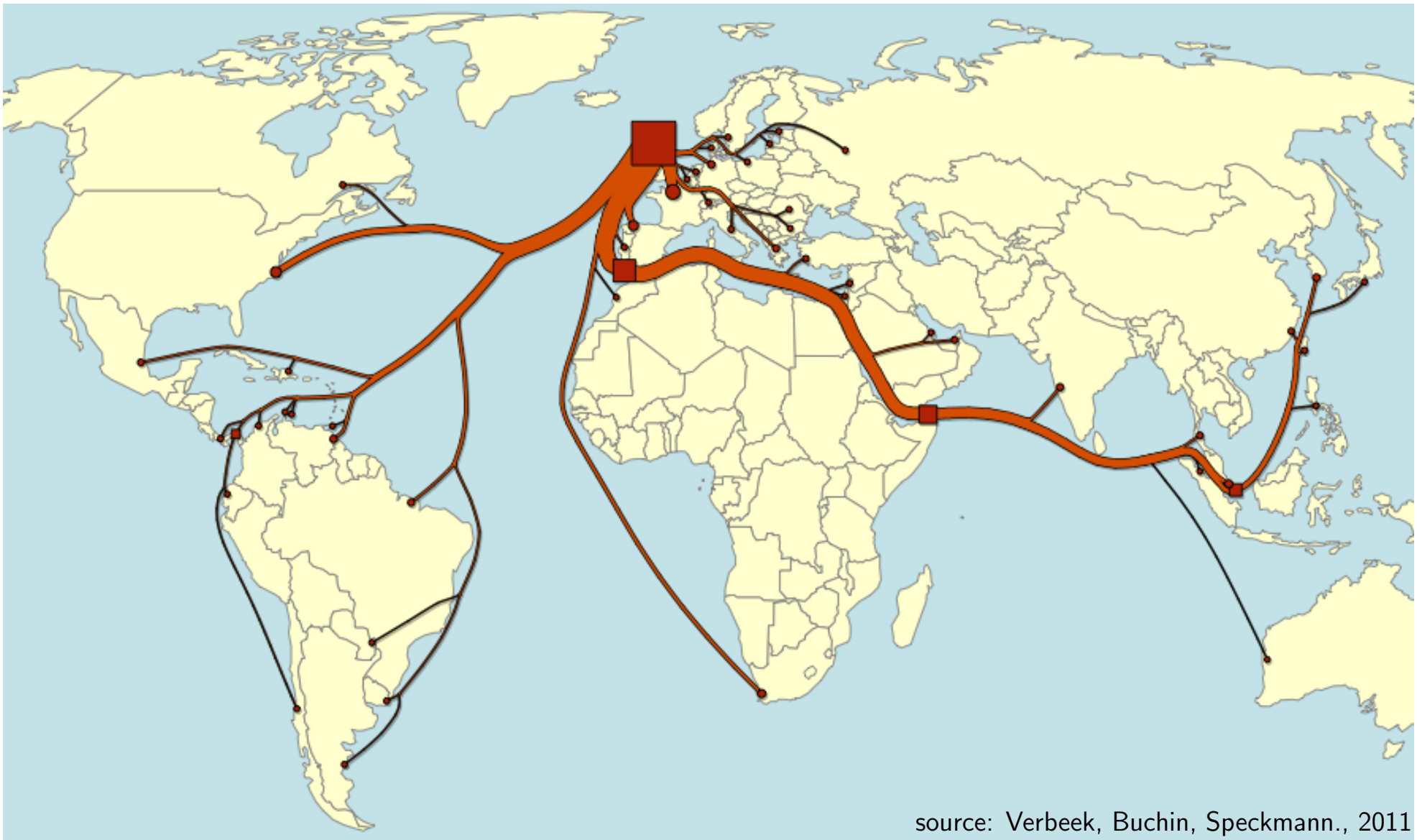
Concentric Tube Map



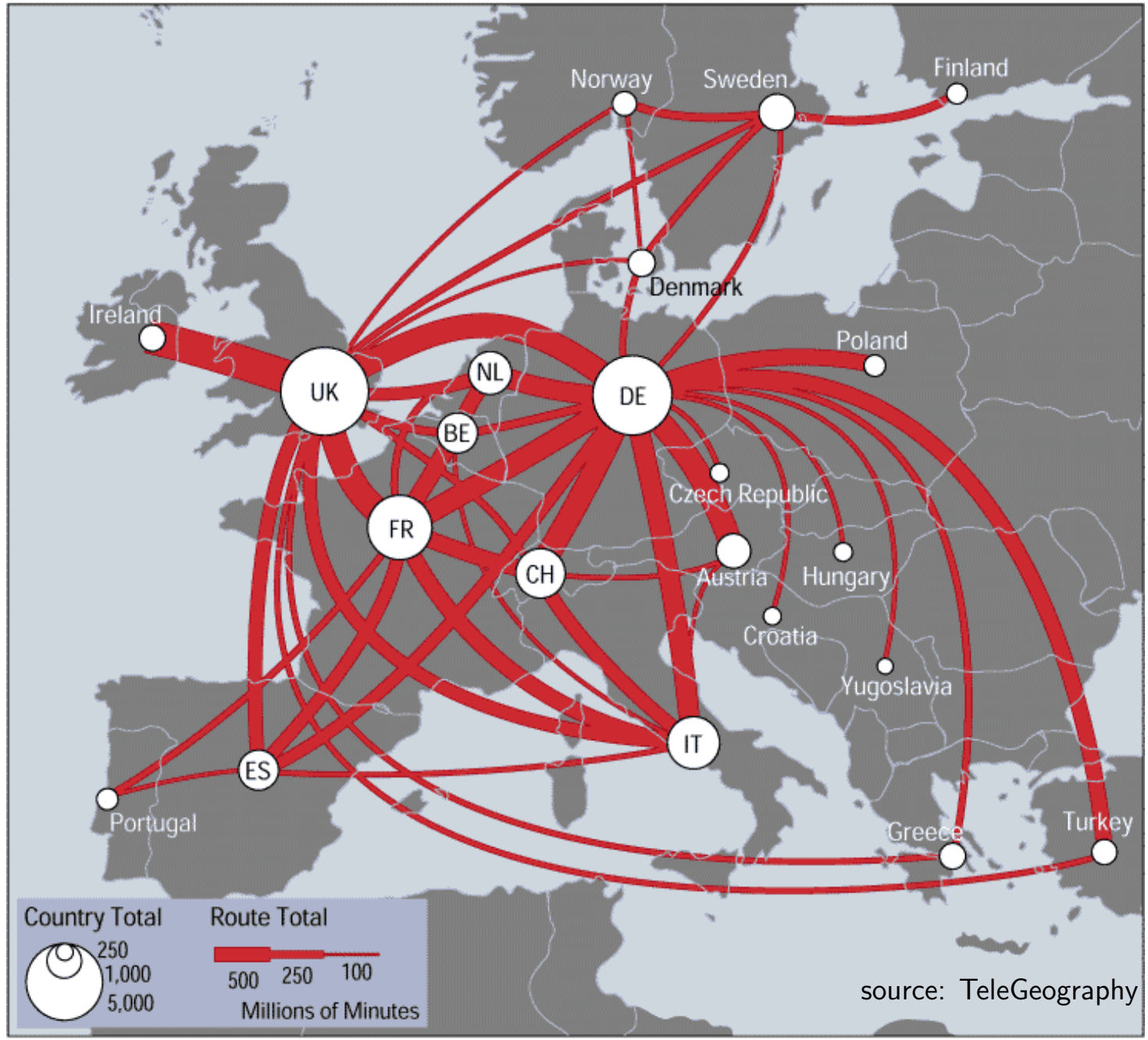
Traffic Network – Flight Destinations Continental



Flow-Map: Whiskey Exports



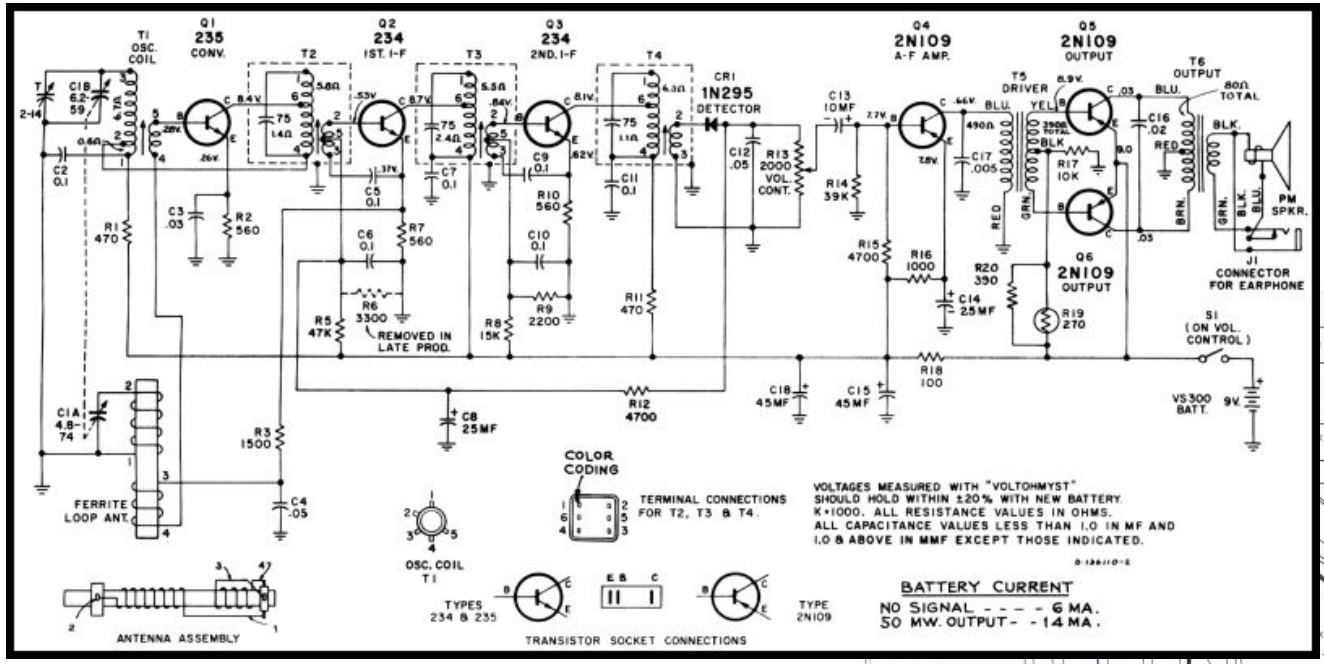
Phone Volume Map

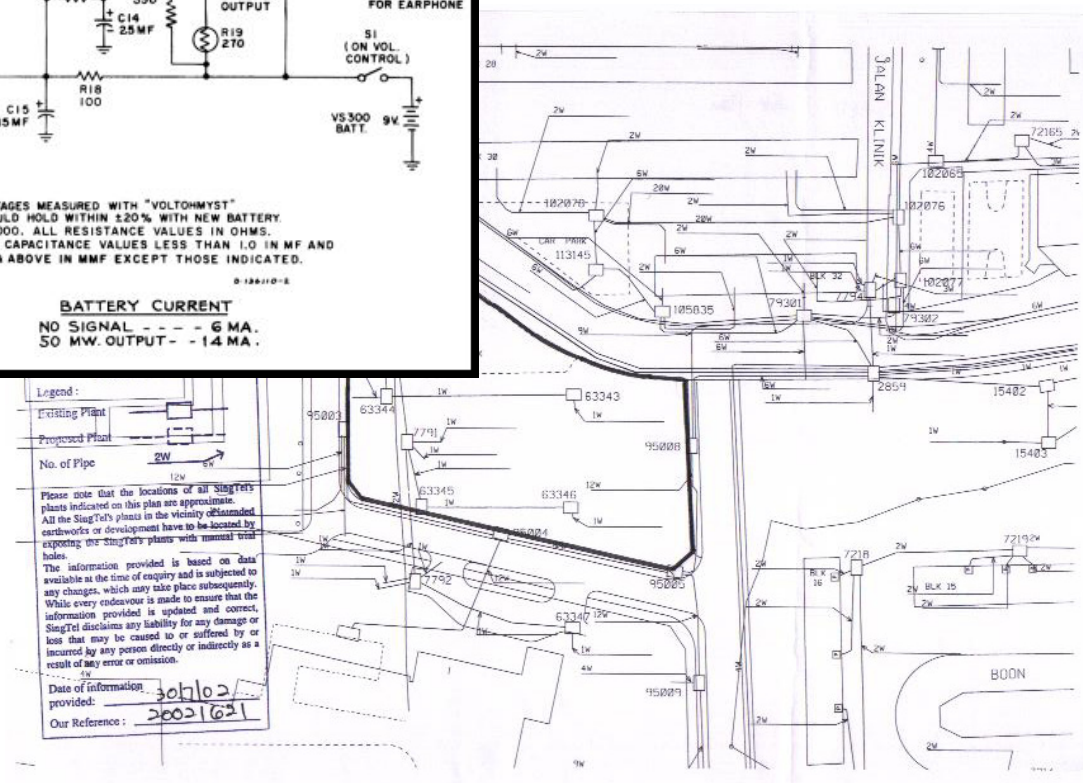


Monitoring Energy Networks

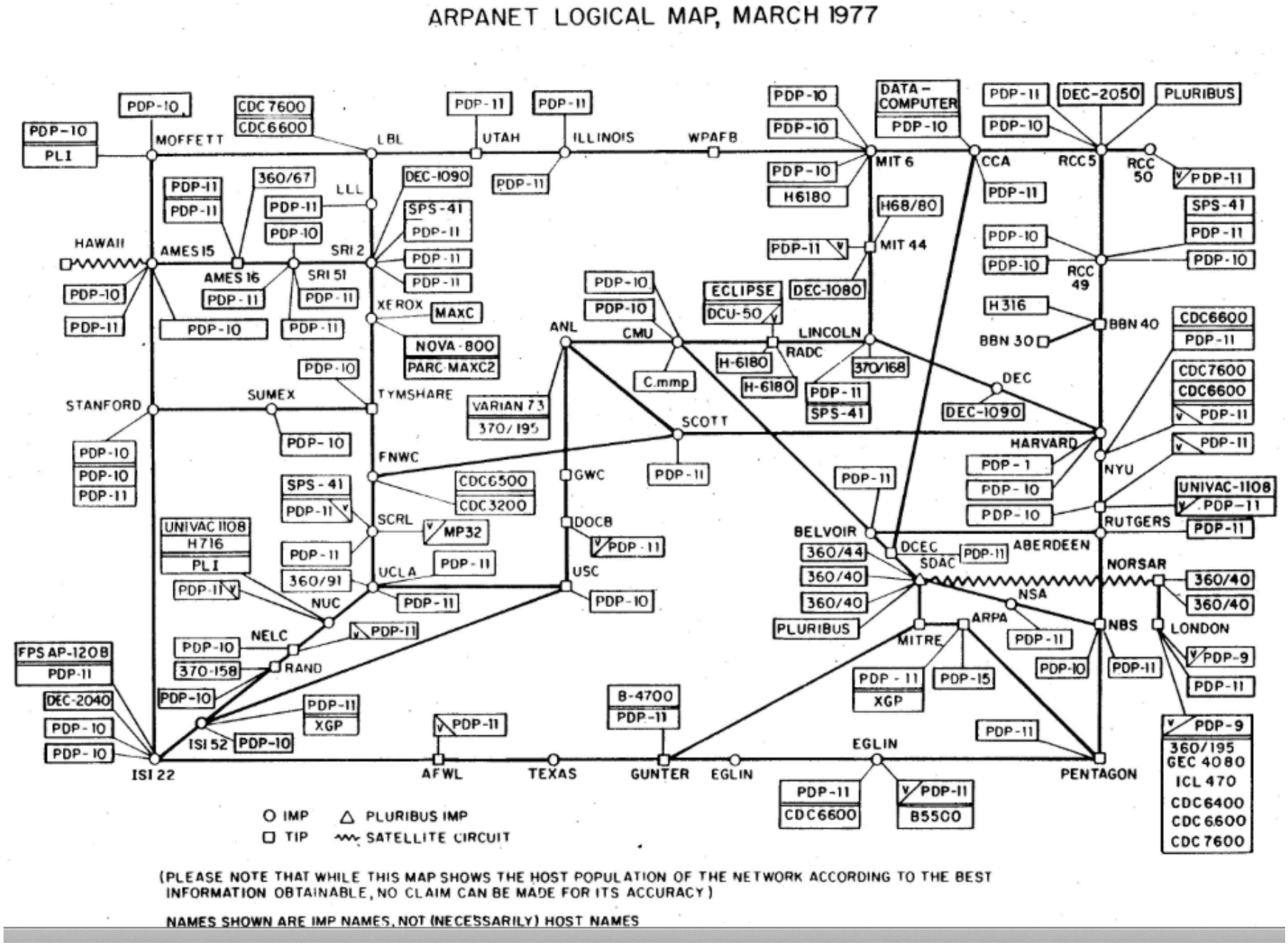
source: Eir Grid, Irland

Technical Networks – Circuit Diagram/Cable Plan

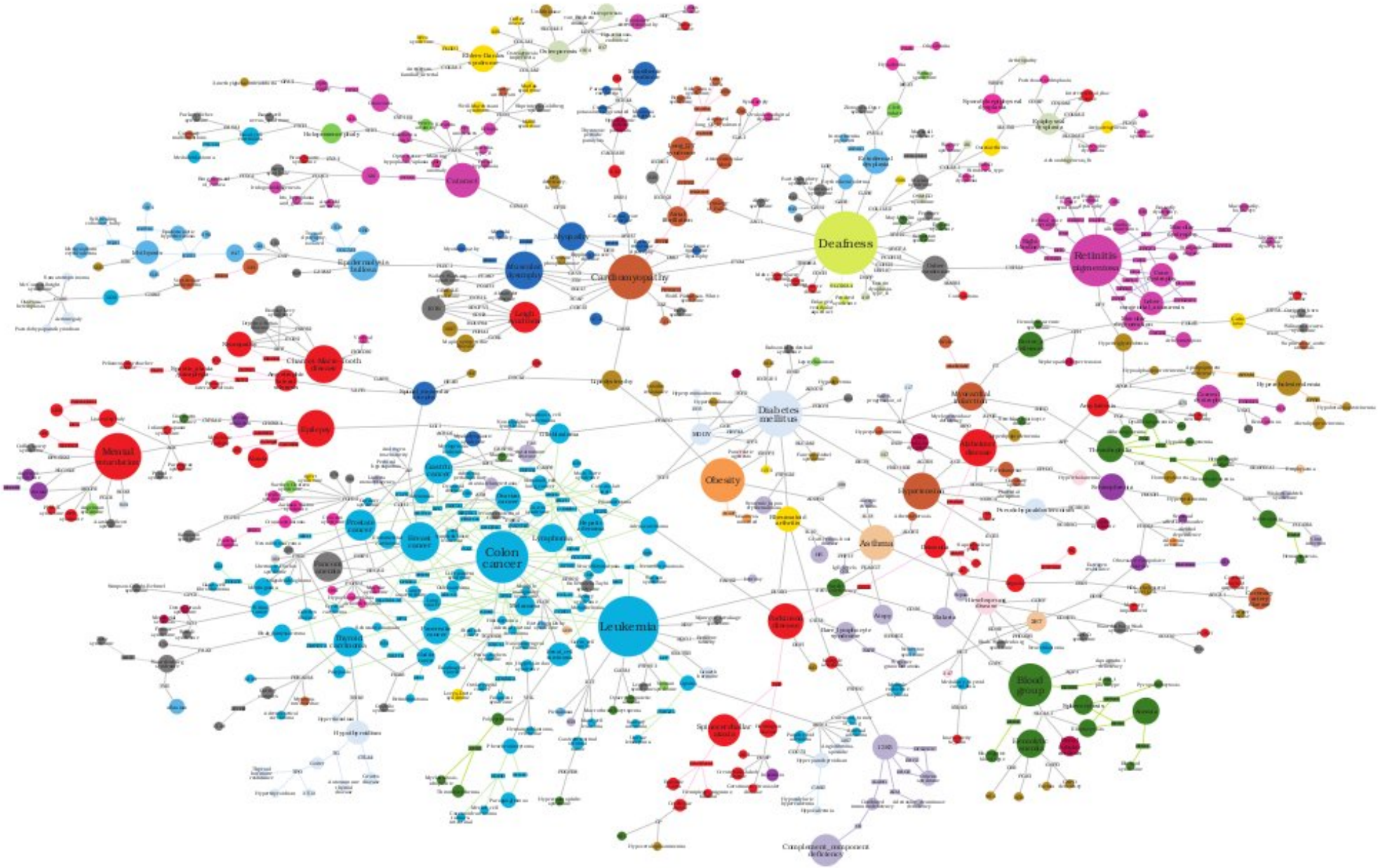




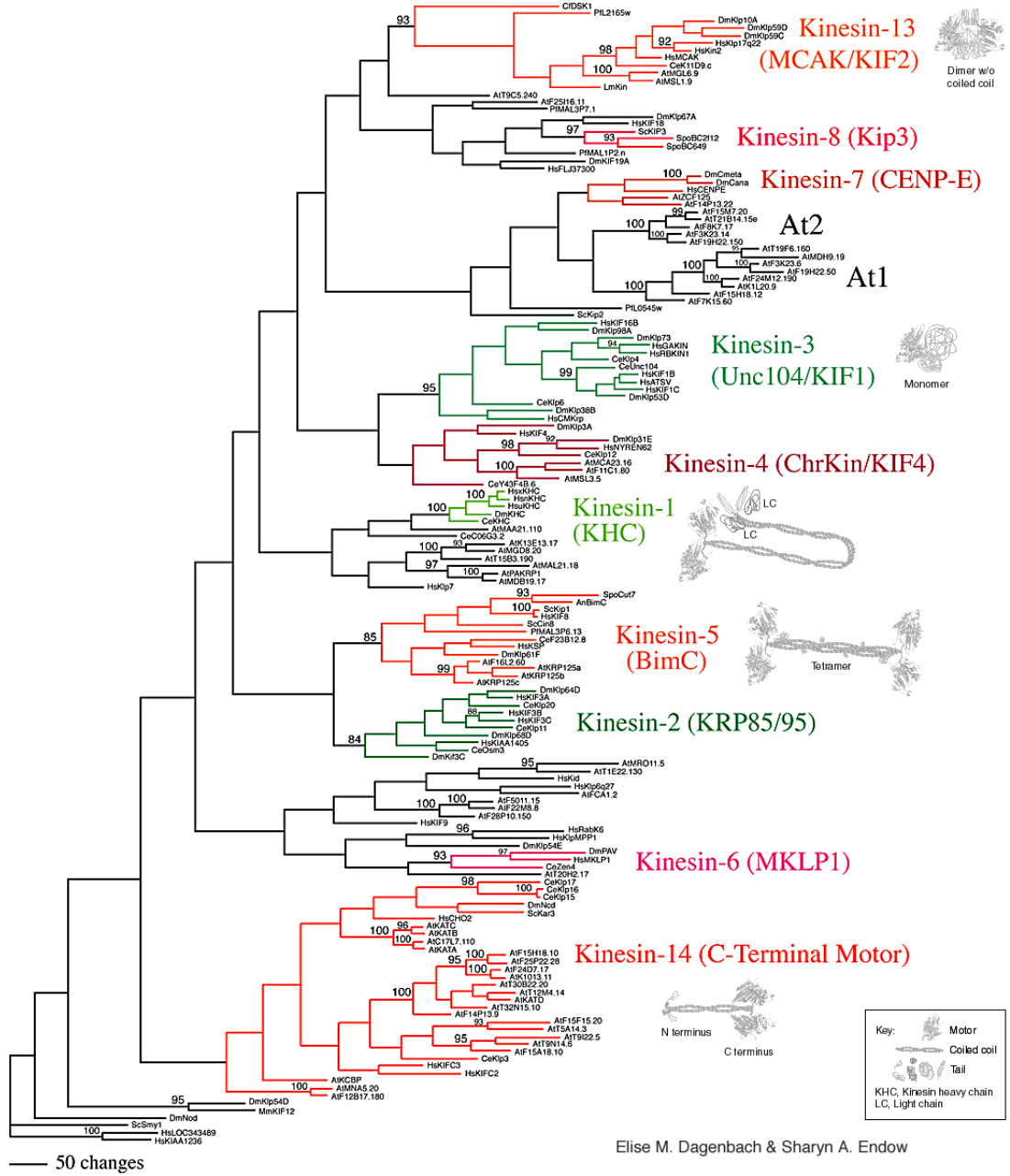
ARPANET Sketch (1977)



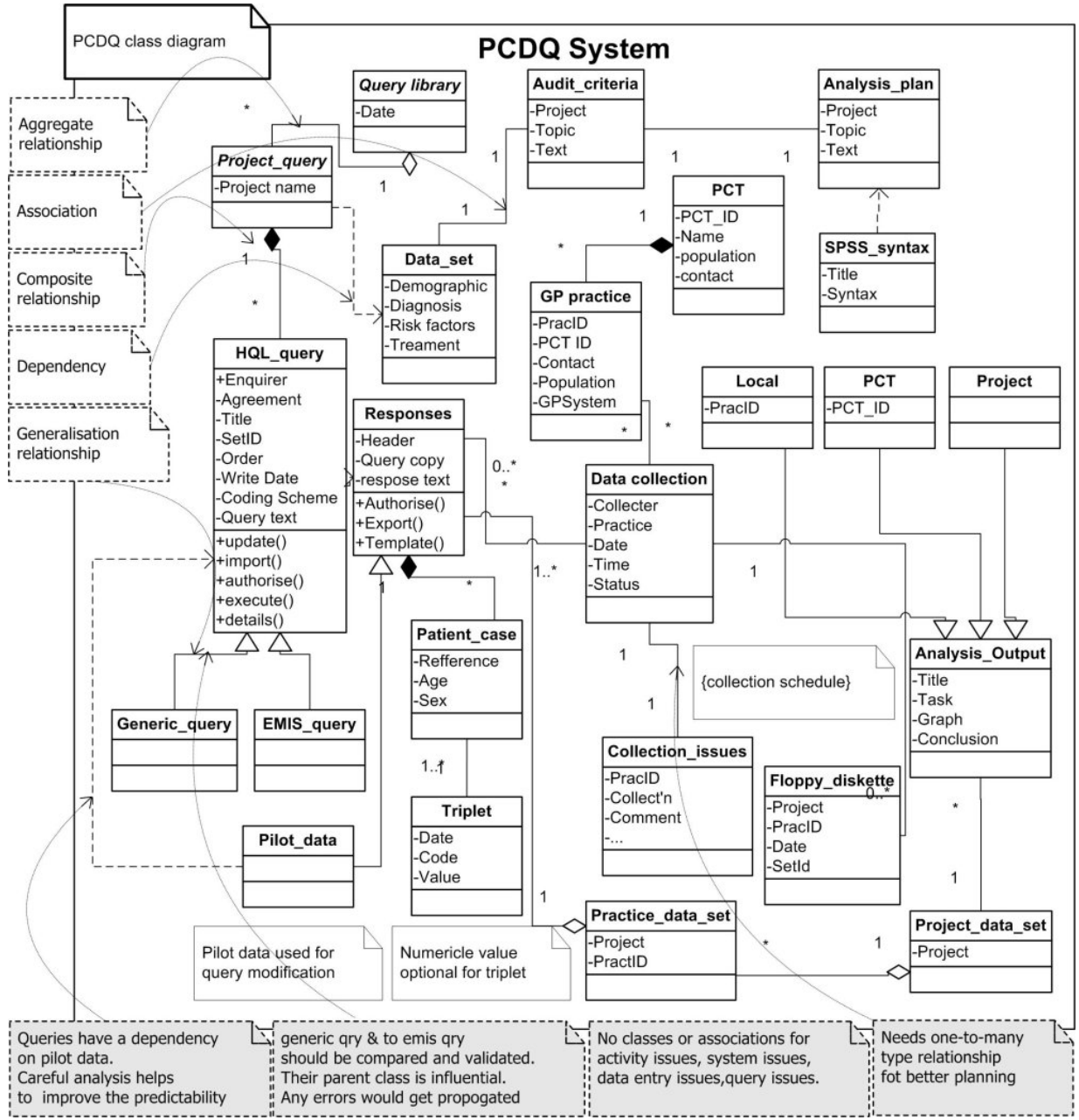
Biomedical Network – Diseasome



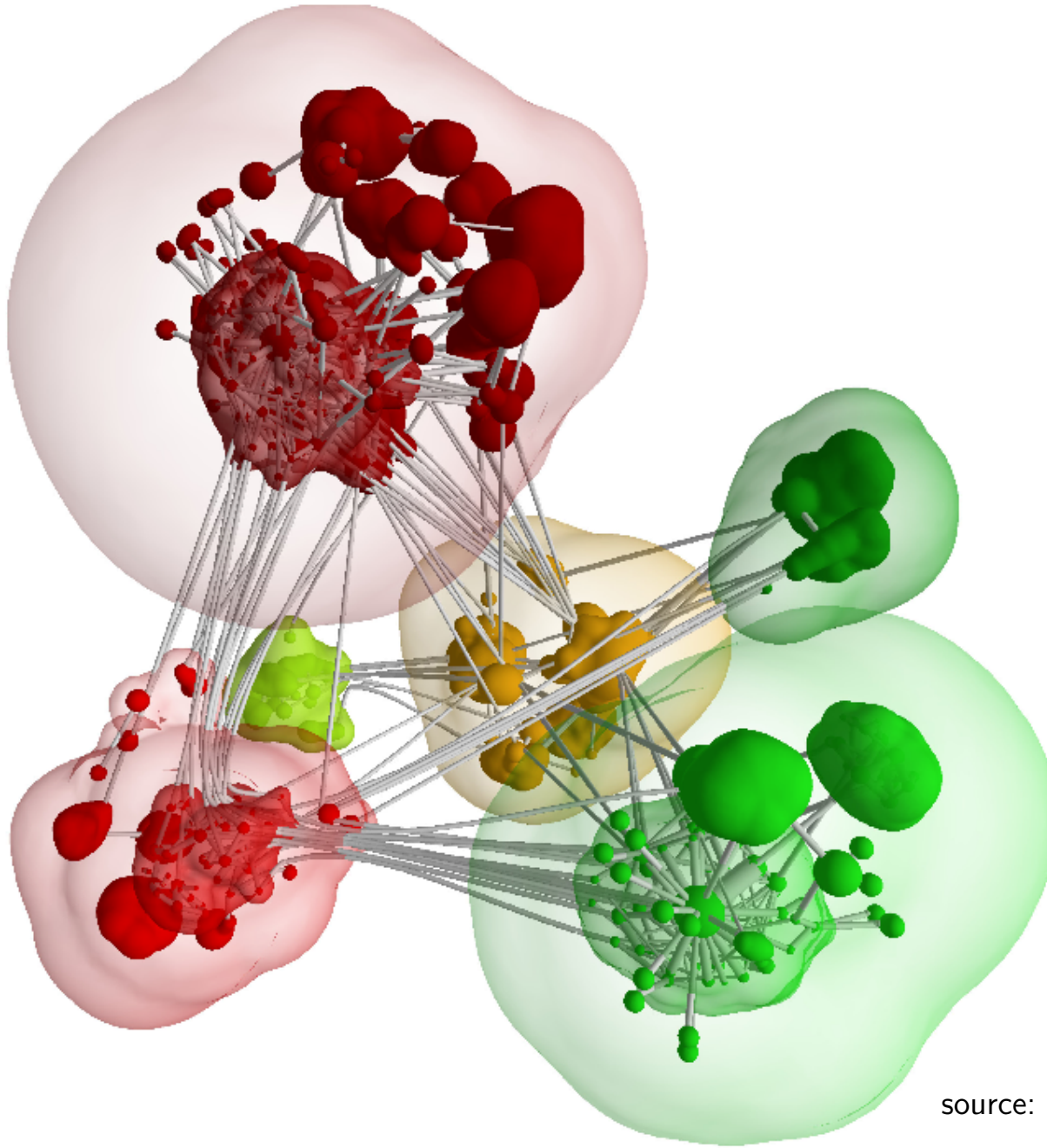
Biology – Phylogenetic Trees



Software Engineering – UML Diagrams

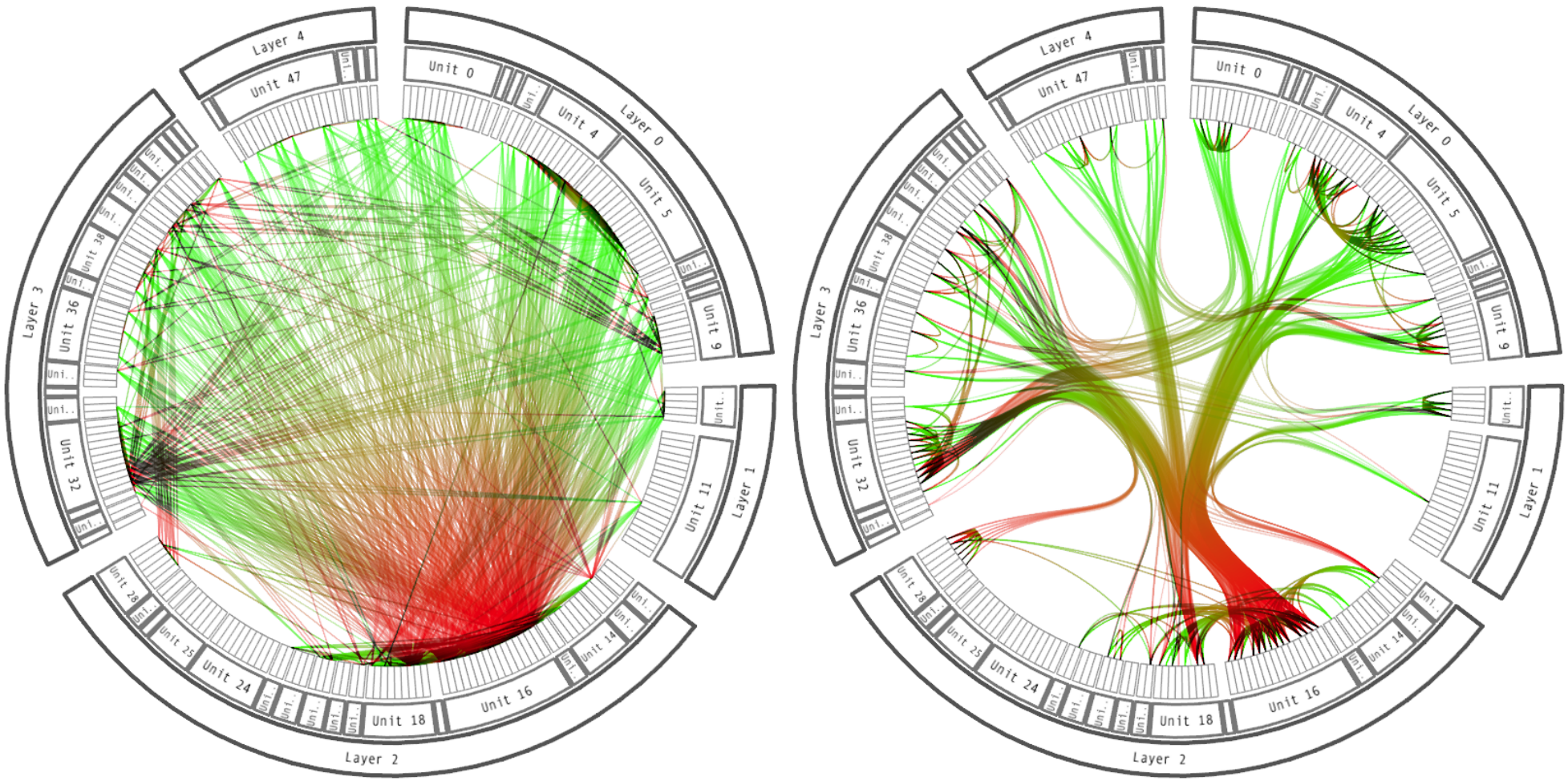


Clustered Software Graph in 3D



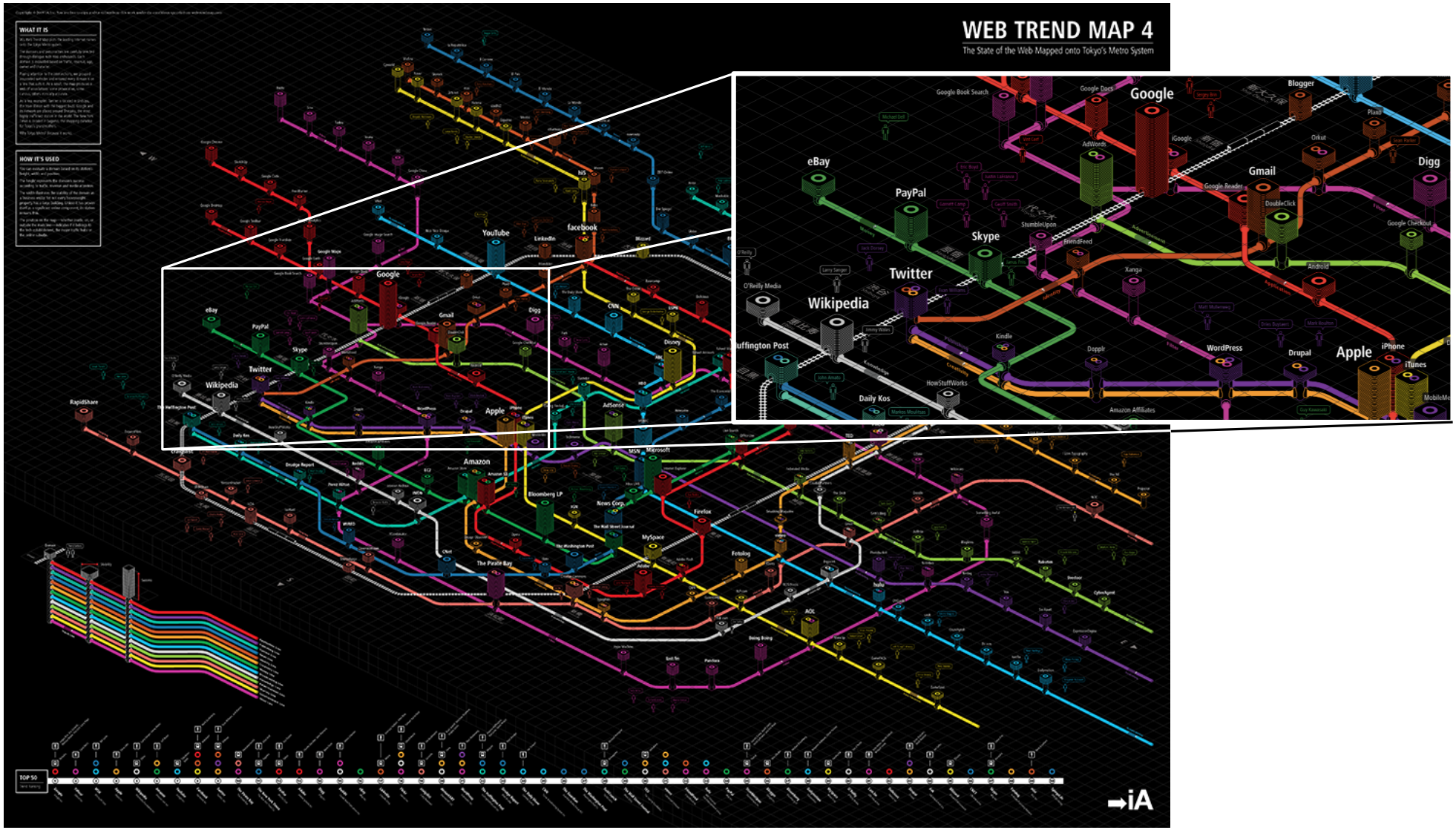
source: Balzer, Deussen, 2007

Software Call Graph with Edge Bundling



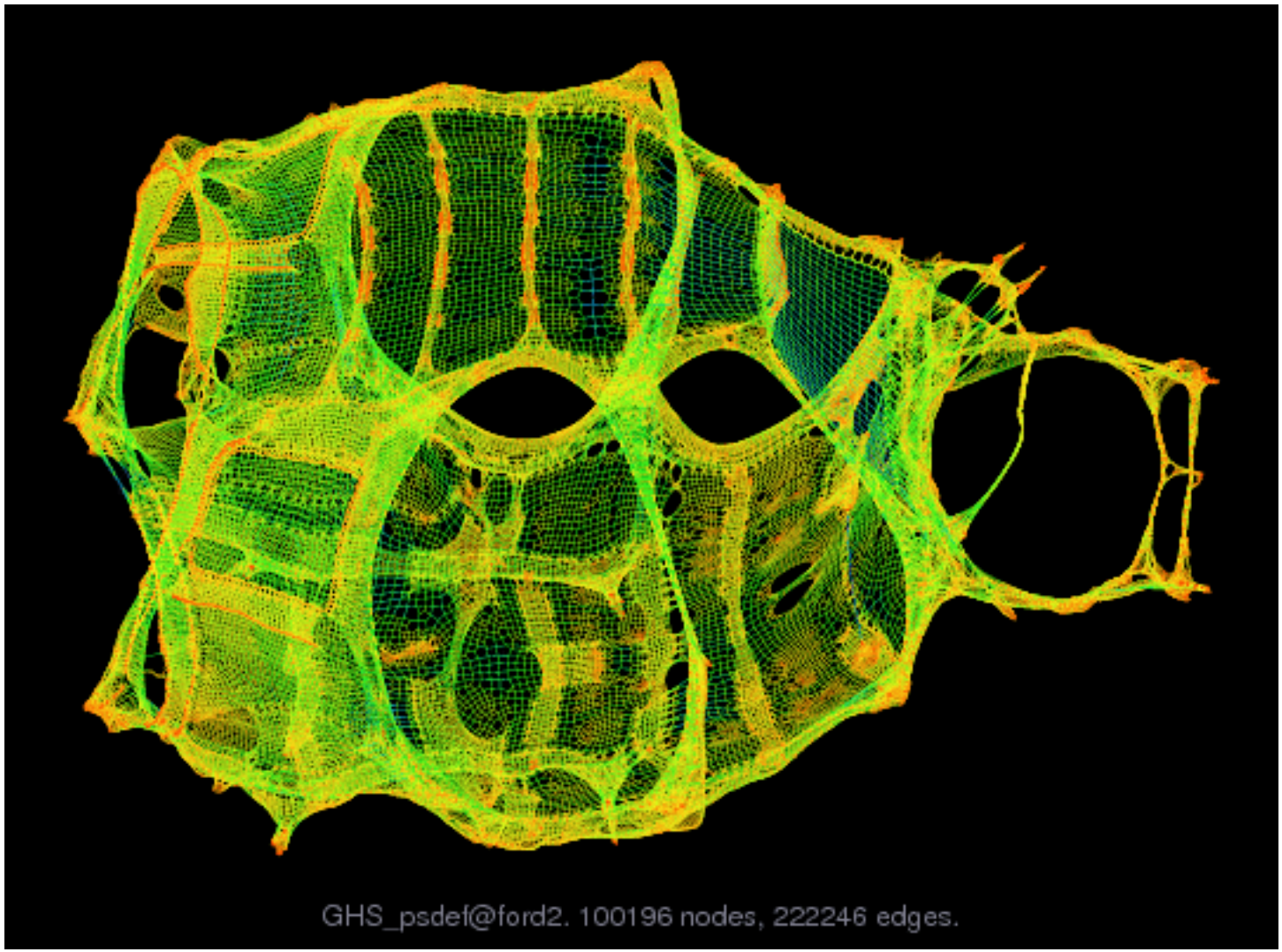
source: Danny Holten, 2011

Web Trend Map



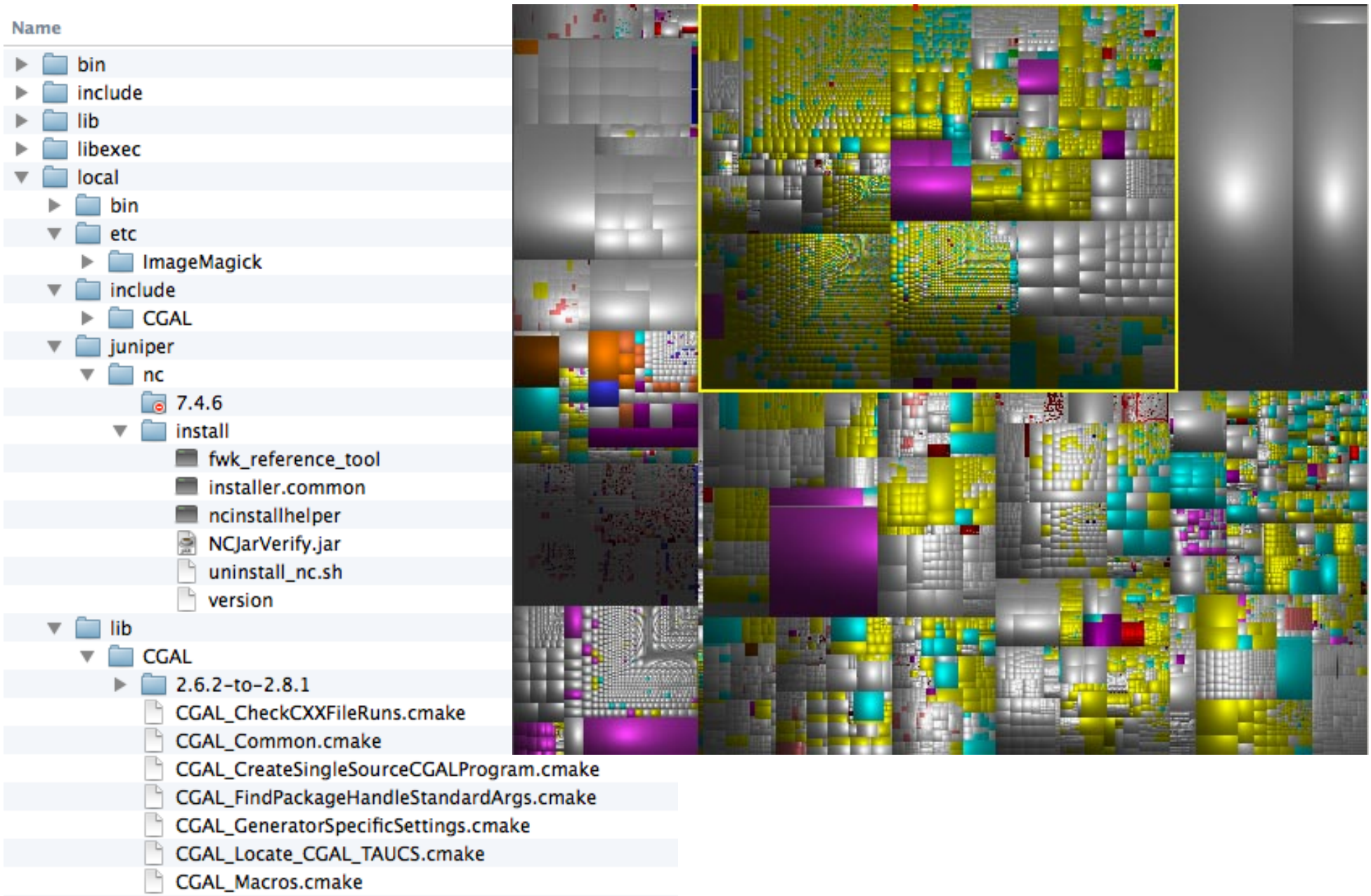
source: information Architects, 2009

Large Graphs – Component Mesh

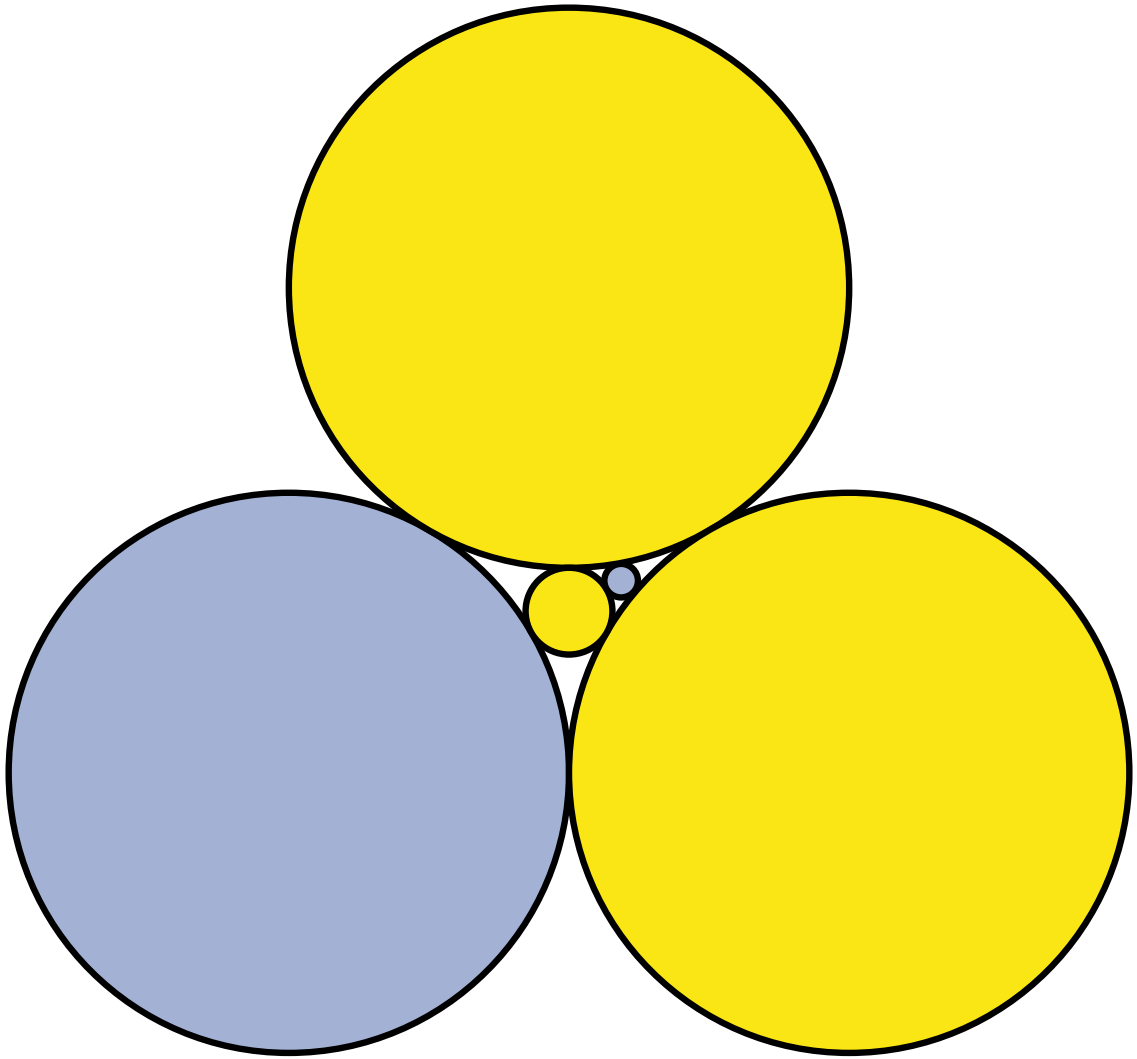


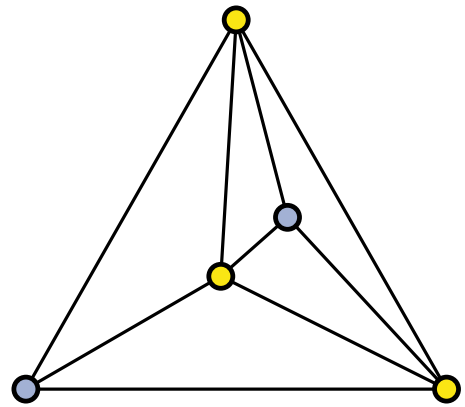
source: Yifan Hu

File System: Explorer vs Treemap



Alternative Representations – Contact Graph





Graph Drawing Software

Graph visualization libraries

- JUNG `jung.sourceforge.net` (Java)
- yFiles `www.yworks.com` (Java(FX), .NET, HTML)
- OGDF `www.ogdf.net` (C++)

Visualization tools

- visone `visone.info`
- graphviz `www.graphviz.org`
- yEd `www.yworks.com`
- Gephi `www.gephi.org`
- Tulip `tulip.labri.fr`
- Tom Sawyer Software `www.tomsawyer.com`

Topics

We will cover the following topics:

- basic definitions: layout problem, aesthetics (today)
- recursive algorithms: trees and related graph classes
- planar straight-line graph drawings
- force- and stress-based layouts
- hierarchical drawings of directed graphs
- orthogonal graph drawing
- ...
- (special drawing styles)

Topics

We will cover the following topics:

- basic definitions: layout problem, aesthetics (today)
- recursive algorithms: trees and related graph classes
- planar straight-line graph drawings
- force- and stress-based layouts
- hierarchical drawings of directed graphs
- orthogonal graph drawing
- ...
- (special drawing styles)

Next week: tree layout algorithms

Homework:
Keep your eyes open for tree visualizations around you!
What do you see?