

# Introduction

Lecture Graph Drawing Algorithms · 192.053

Martin Nöllenburg

13.03.2018



ALGORITHMS AND  
COMPLEXITY GROUP

## Lecturer



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

## 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

→ visit TISS page for course 192.053 ←

- 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

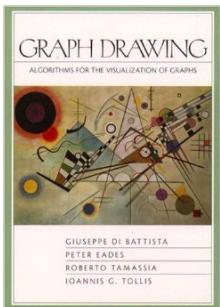
## Media:

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

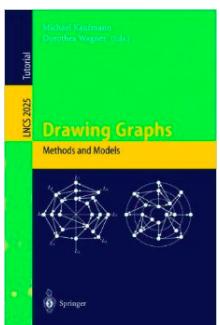
## Content:

- 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

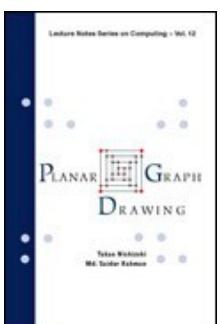
# Literature (available in TU library)



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/>

## 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](mailto: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 screenshot shows the homepage of the Graph Drawing 2018 conference website. At the top, it says "gd18 GRAPH DRAWING 2018" and "Barcelona, 26-28 September 2018". Below this is a banner image of the city of Barcelona. The navigation menu includes "Home", "Calls", "Invited speakers", "PhD school", "Contest", "Registration", "Venue", and "About GD".

The practical exercises will be the contest topics of 2018!

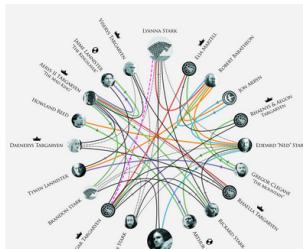
Challenging topics!

Visual design and algorithmic tasks!

More details on the website!

## Game of Thrones

The TV show "Game of Thrones" is based on the book series "A Song of Ice and Fire" by George R.R. Martin and is one of the most popular TV shows in the previous years. In 2016, HBO released the following infographic that shows how the key players related to each other during the "Tower of Joy" moment.



<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 screenshot shows the homepage of the Graph Drawing 2018 conference website. At the top, there is a banner with the text "gd18 GRAPH DRAWING 2018" and "Barcelona, 26-28 September 2018". Below the banner is a photograph of the city of Barcelona. The navigation menu includes links for Home, Calls, Invited speakers, PhD school, Contest, Registration, Venue, and About GD.

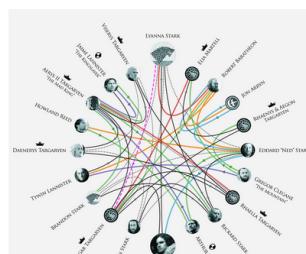
The practical exercises will be the contest topics of 2018!

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 screenshot shows the homepage of the Graph Drawing 2018 conference website. At the top, there is a banner with the text "gd18 GRAPH DRAWING 2018" and "Barcelona, 26-28 September 2018". Below the banner is a photograph of the city of Barcelona. The navigation menu includes links for Home, Calls, Invited speakers, PhD school, Contest, Registration, Venue, and About GD.

The practical exercises will be the contest topics of 2018!

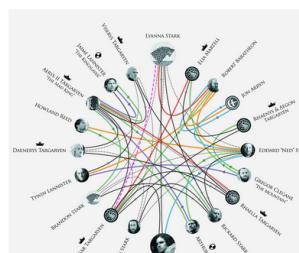
Challenging topics!

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?

Programming?

No!



## Paper presentation

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

Programming?

Yes!

No!

Coding Project in Teams (3 ppl/team)

## Paper presentation

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

Programming?

Yes!

Coding Project in Teams (3 ppl/team)

Design?



No!

## Paper presentation

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

Programming?

Yes!

No!

## Paper presentation

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

Coding Project in Teams (3 ppl/team)

Design?

No!

## Angle Optimization

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

Programming?

Yes!

No!

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

## Paper presentation

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

## Angle Optimization

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

Programming?

No!

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

## Paper presentation

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

## Angle Optimization

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

GD contest!

## 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  $V = \{v_1, \dots, v_n\}$

Set of edges  $E = \{e_1, \dots, e_m\}$ ,

$e_i = \{v_j, v_k\}$ ,  $1 \leq i \leq m$ ,  $1 \leq j, k \leq n$

# Graphs and their Representations



## What is a Graph?

Tuple  $G = (V, E)$

Set of vertices  $V = \{v_1, \dots, v_n\}$

Set of edges  $E = \{e_1, \dots, e_m\}$ ,

$e_i = \{v_j, v_k\}$ ,  $1 \leq i \leq m$ ,  $1 \leq j, k \leq n$

## Representations?

# Graphs and their Representations



## What is a Graph?

Tuple  $G = (V, E)$

Set of vertices  $V = \{v_1, \dots, v_n\}$

Set of edges  $E = \{e_1, \dots, e_m\}$ ,

$e_i = \{v_j, v_k\}$ ,  $1 \leq i \leq m$ ,  $1 \leq j, k \leq n$

## Representations?

**Set representation:**

$$\begin{aligned} V &= \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\} \\ E &= \{\{v_1, v_2\}, \{v_1, v_8\}, \{v_2, v_3\}, \{v_3, v_5\}, \{v_3, v_9\}, \\ &\quad \{v_3, v_{10}\}, \{v_4, v_5\}, \{v_4, v_6\}, \{v_4, v_9\}, \{v_5, v_8\}, \\ &\quad \{v_6, v_8\}, \{v_6, v_9\}, \{v_7, v_8\}, \{v_7, v_9\}, \{v_8, v_{10}\}, \\ &\quad \{v_9, v_{10}\}\} \end{aligned}$$

# Graphs and their Representations



## What is a Graph?

Tuple  $G = (V, E)$

Set of vertices  $V = \{v_1, \dots, v_n\}$

Set of edges  $E = \{e_1, \dots, e_m\}$ ,

$e_i = \{v_j, v_k\}$ ,  $1 \leq i \leq m$ ,  $1 \leq j, k \leq n$

## Representations?

Set representation

Adjacency lists

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

# Graphs and their Representations

## What is a Graph?

Tuple  $G = (V, E)$

Set of vertices  $V = \{v_1, \dots, v_n\}$

Set of edges  $E = \{e_1, \dots, e_m\}$ ,

$e_i = \{v_j, v_k\}$ ,  $1 \leq i \leq m$ ,  $1 \leq j, k \leq n$

## 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  $V = \{v_1, \dots, v_n\}$

Set of edges  $E = \{e_1, \dots, e_m\}$ ,

$e_i = \{v_j, v_k\}$ ,  $1 \leq i \leq m$ ,  $1 \leq j, k \leq n$

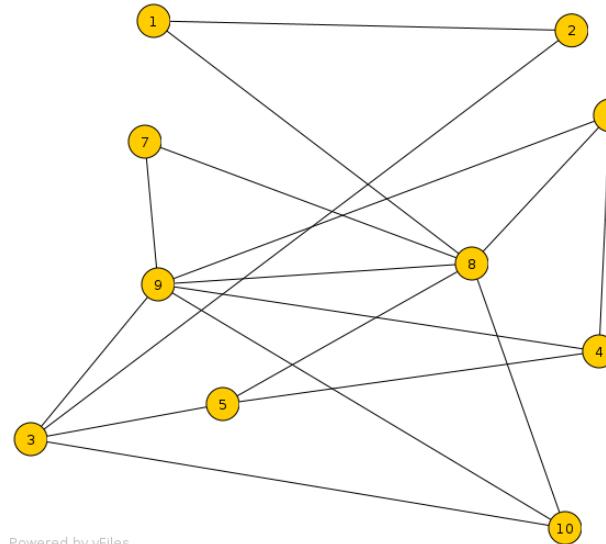
## Representations?

Set representation

Adjacency lists

Adjacency matrix

Drawing



Powered by yFiles

# Graphs and their Representations

## What is a Graph?

Tuple  $G = (V, E)$

Set of vertices  $V = \{v_1, \dots, v_n\}$

Set of edges  $E = \{e_1, \dots, e_m\}$ ,

$e_i = \{v_j, v_k\}$ ,  $1 \leq i \leq m$ ,  $1 \leq j, k \leq n$

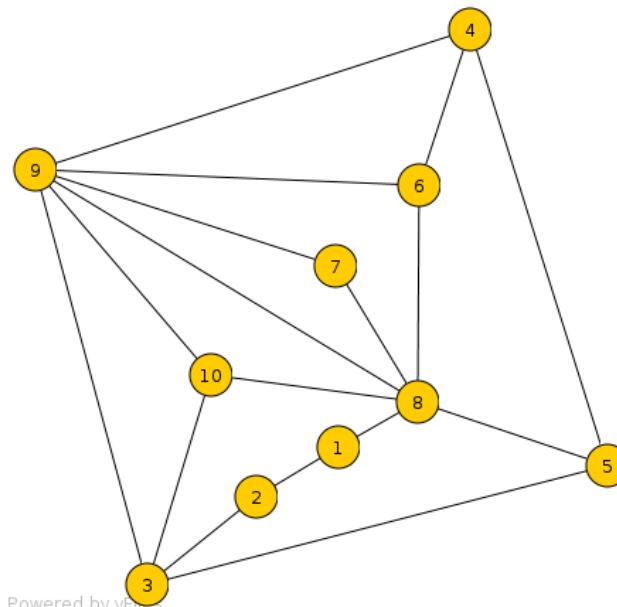
## Representations?

Set representation

Adjacency lists

Adjacency matrix

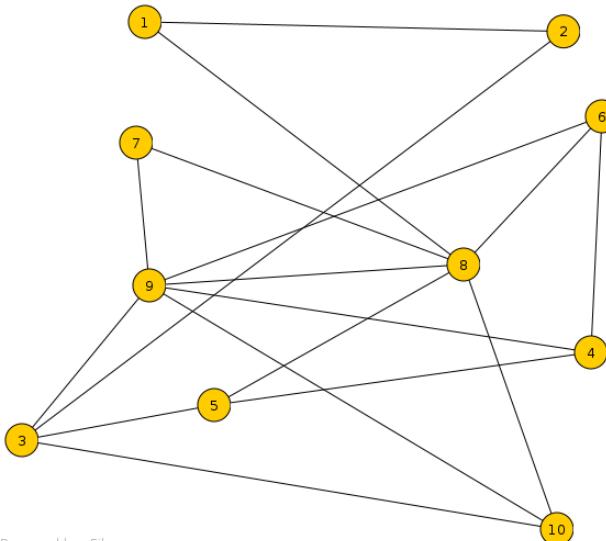
Drawing



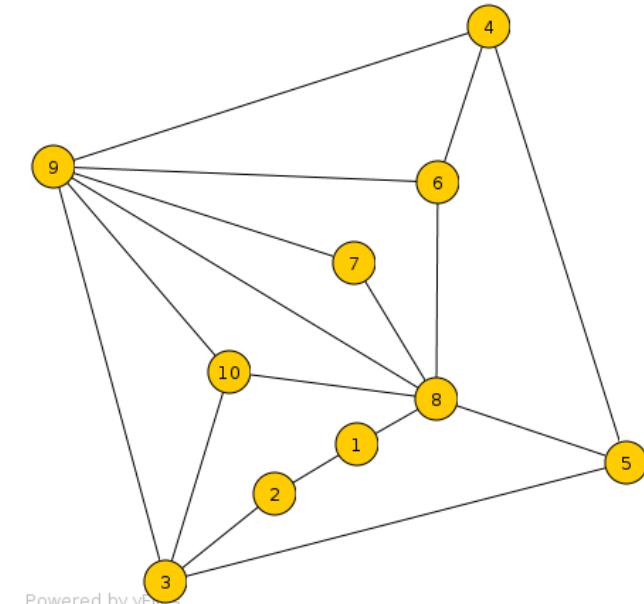
# Graphs and their Representations

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

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



$$\left( \begin{array}{cccccccccc} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{array} \right)$$



# 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      - hypergraphs
- trees        - regular
- bipartite
- $k$ -colorable
- simple/multigraphs
- directed (acyclic)
- complete / dense
- Eulerian / Hamiltonian
- connected

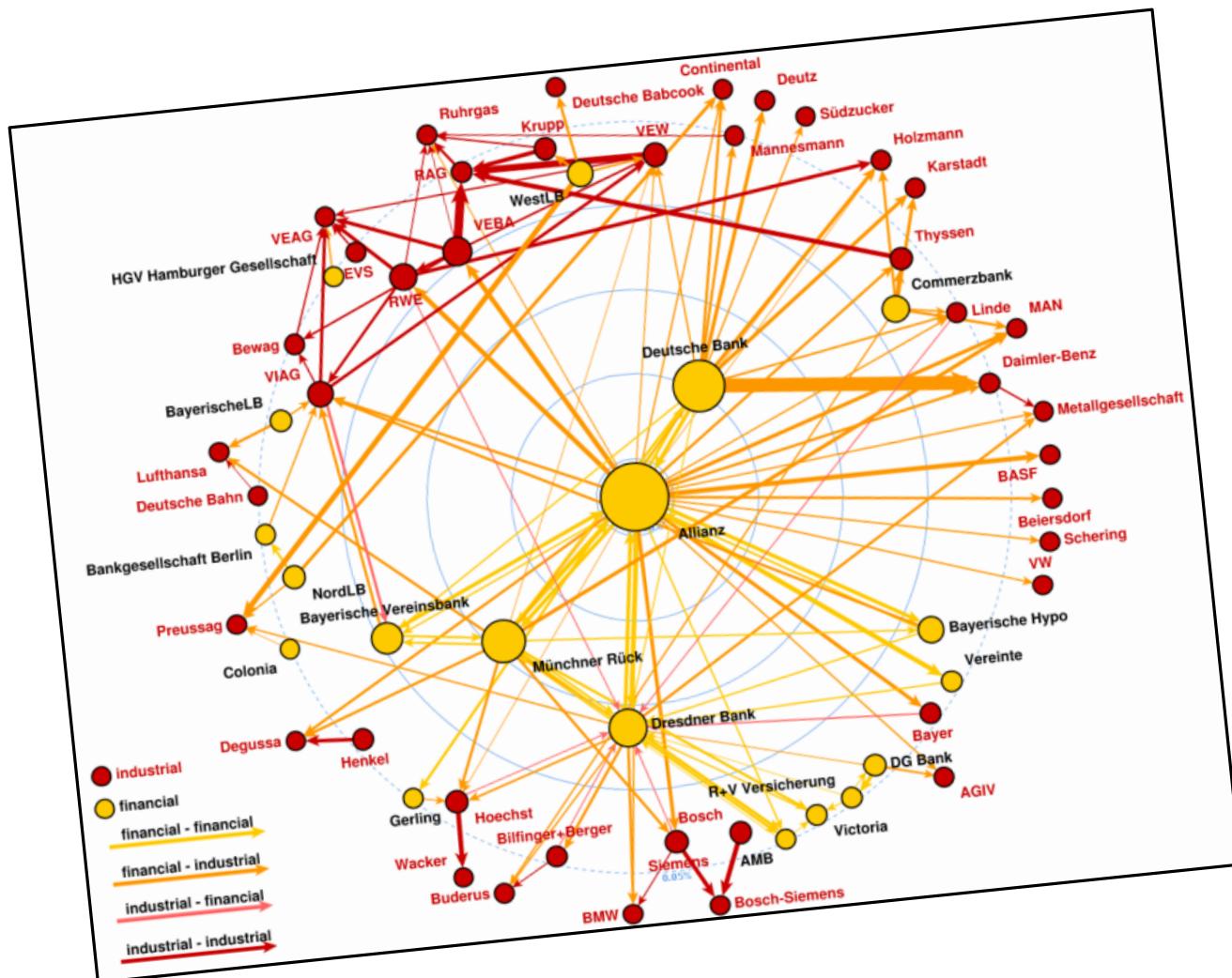
# Why Draw Graphs?



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

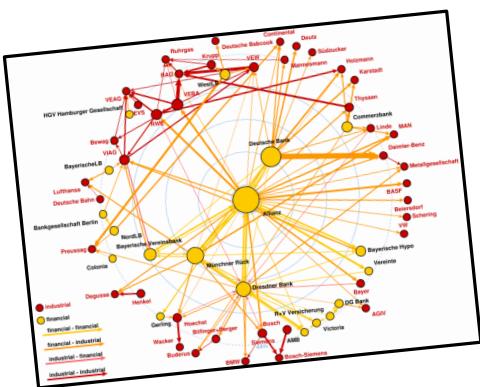
# Why Draw Graphs?

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



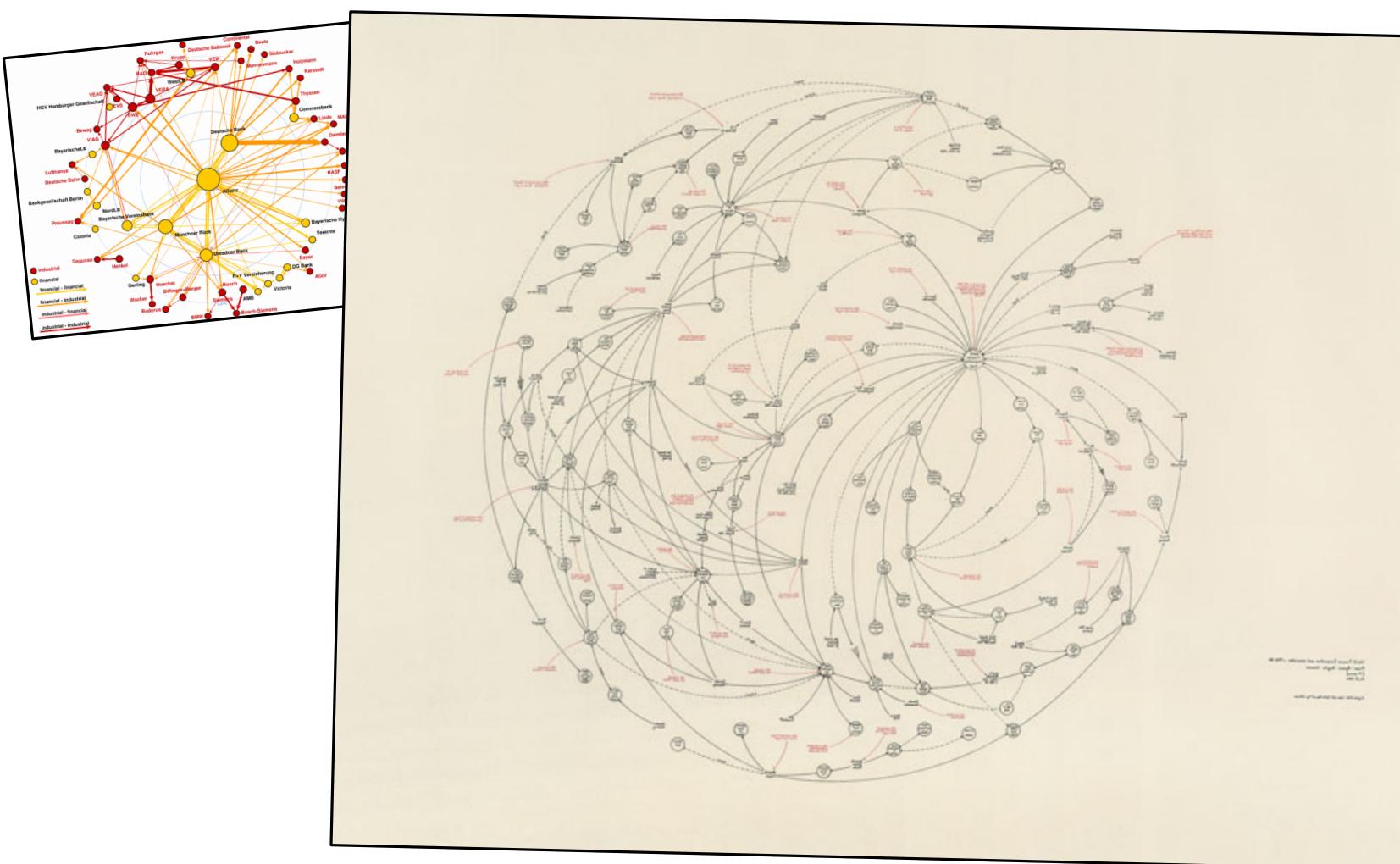
# Why Draw Graphs?

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



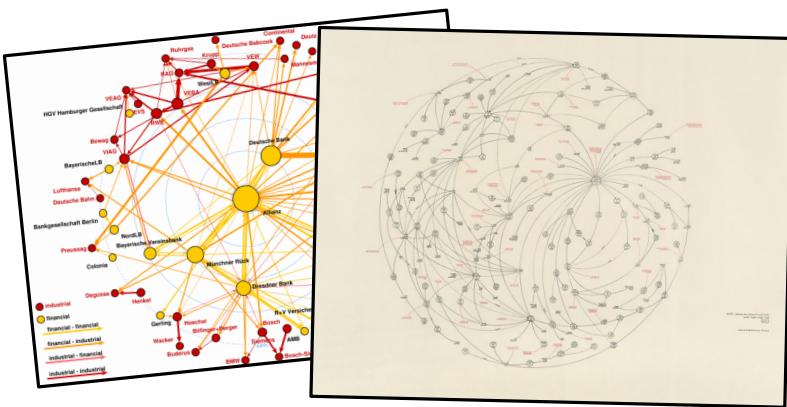
# Why Draw Graphs?

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



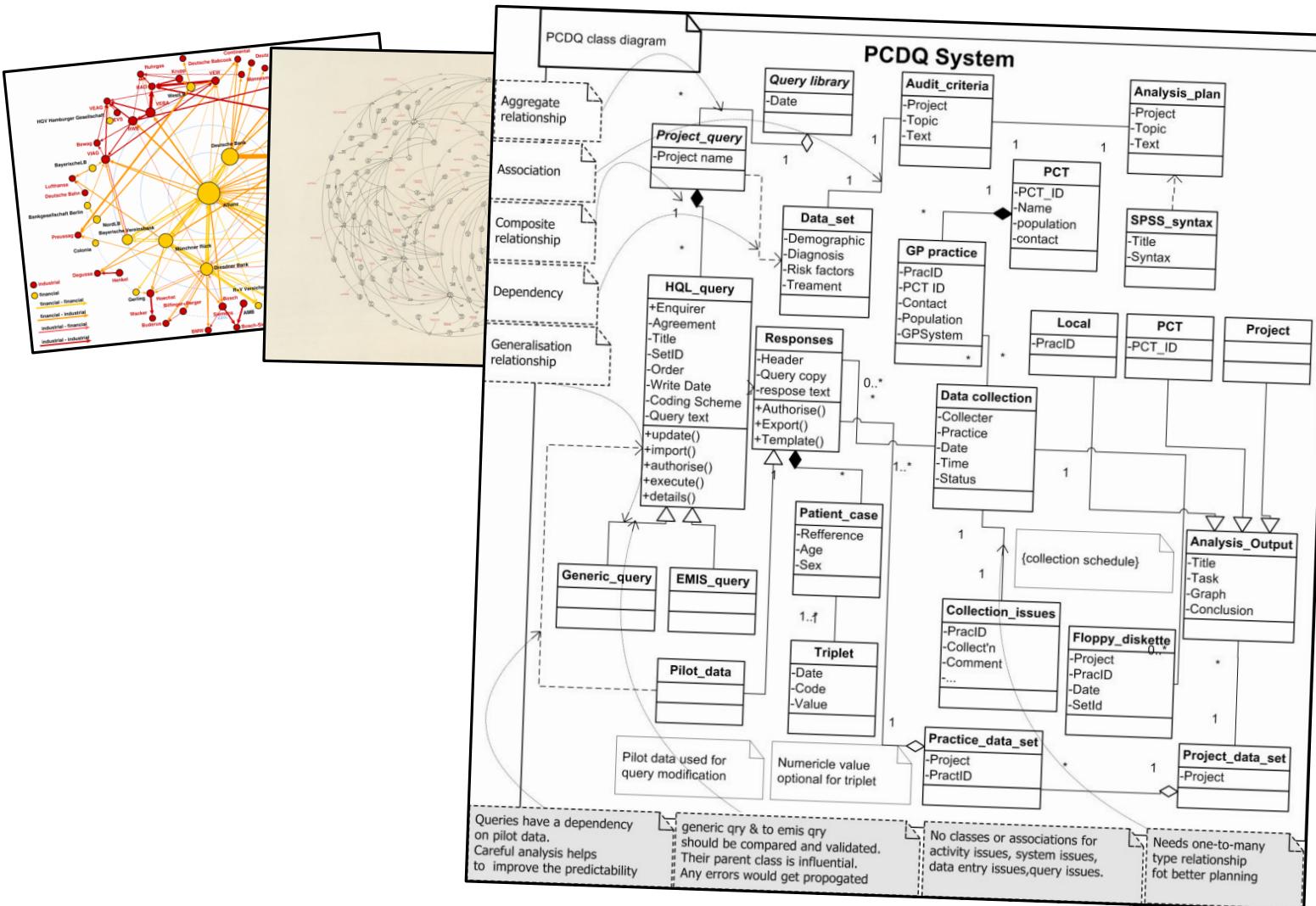
# Why Draw Graphs?

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



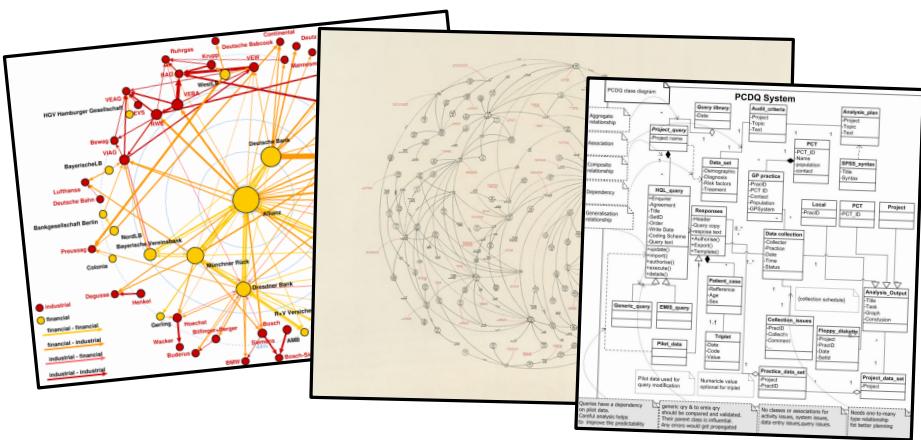
# Why Draw Graphs?

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



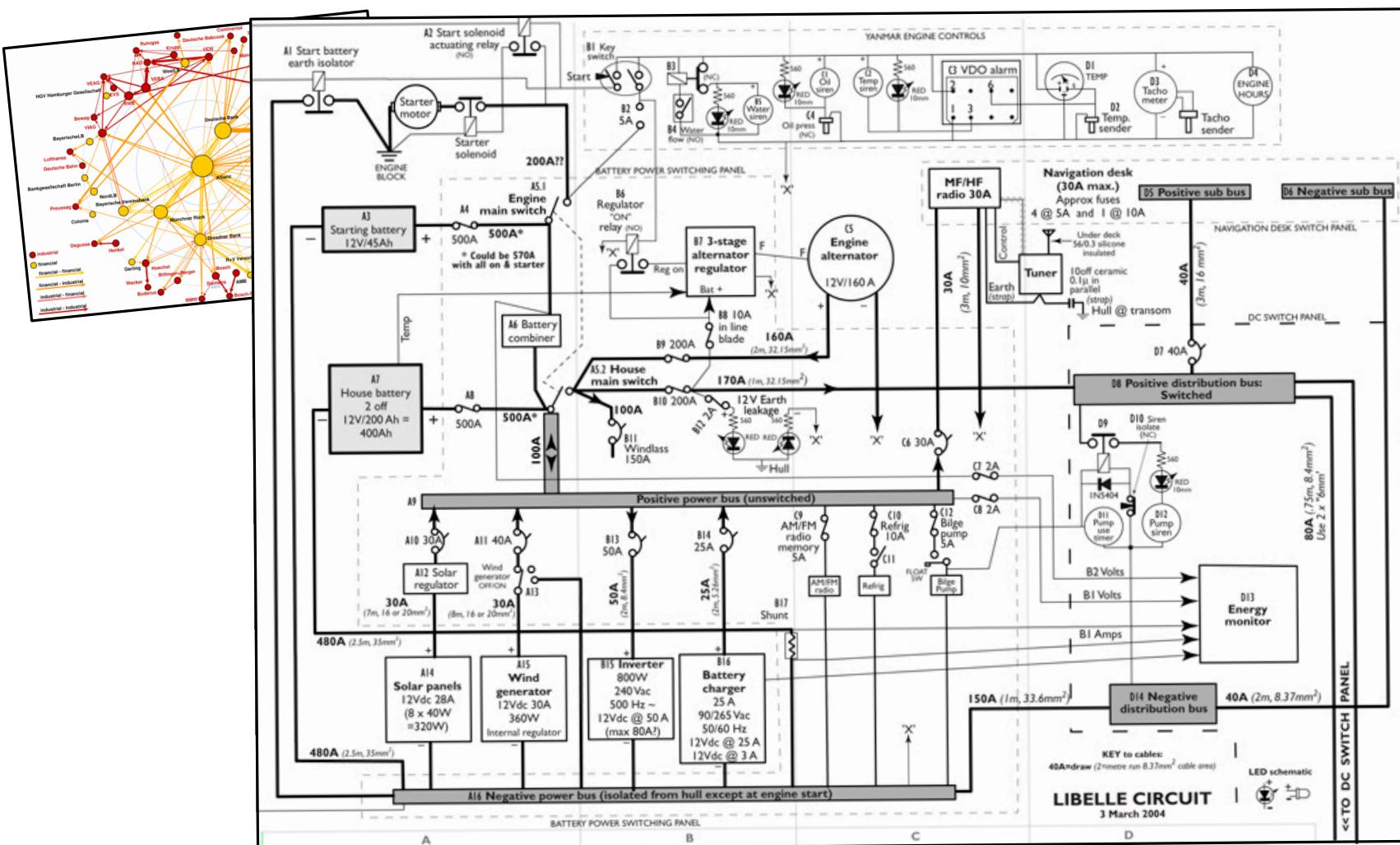
# Why Draw Graphs?

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



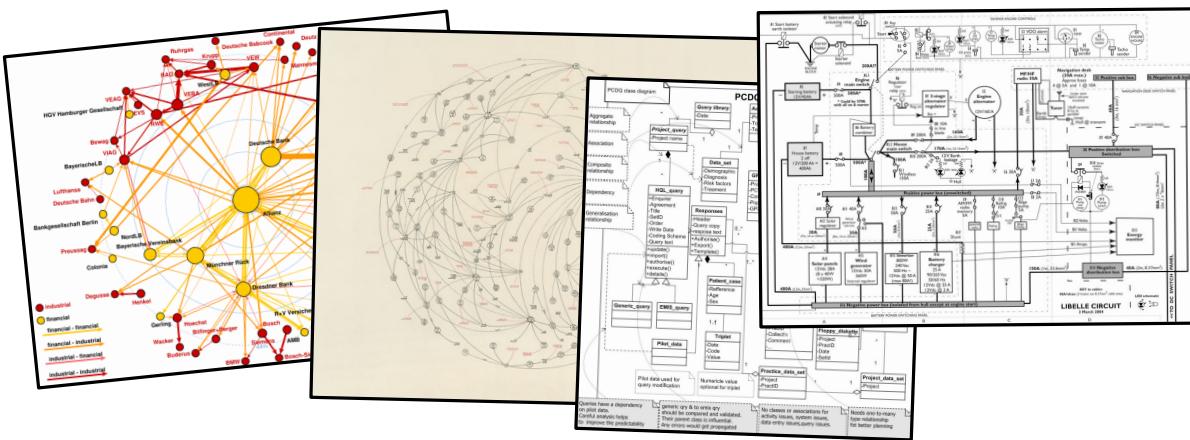
# Why Draw Graphs?

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



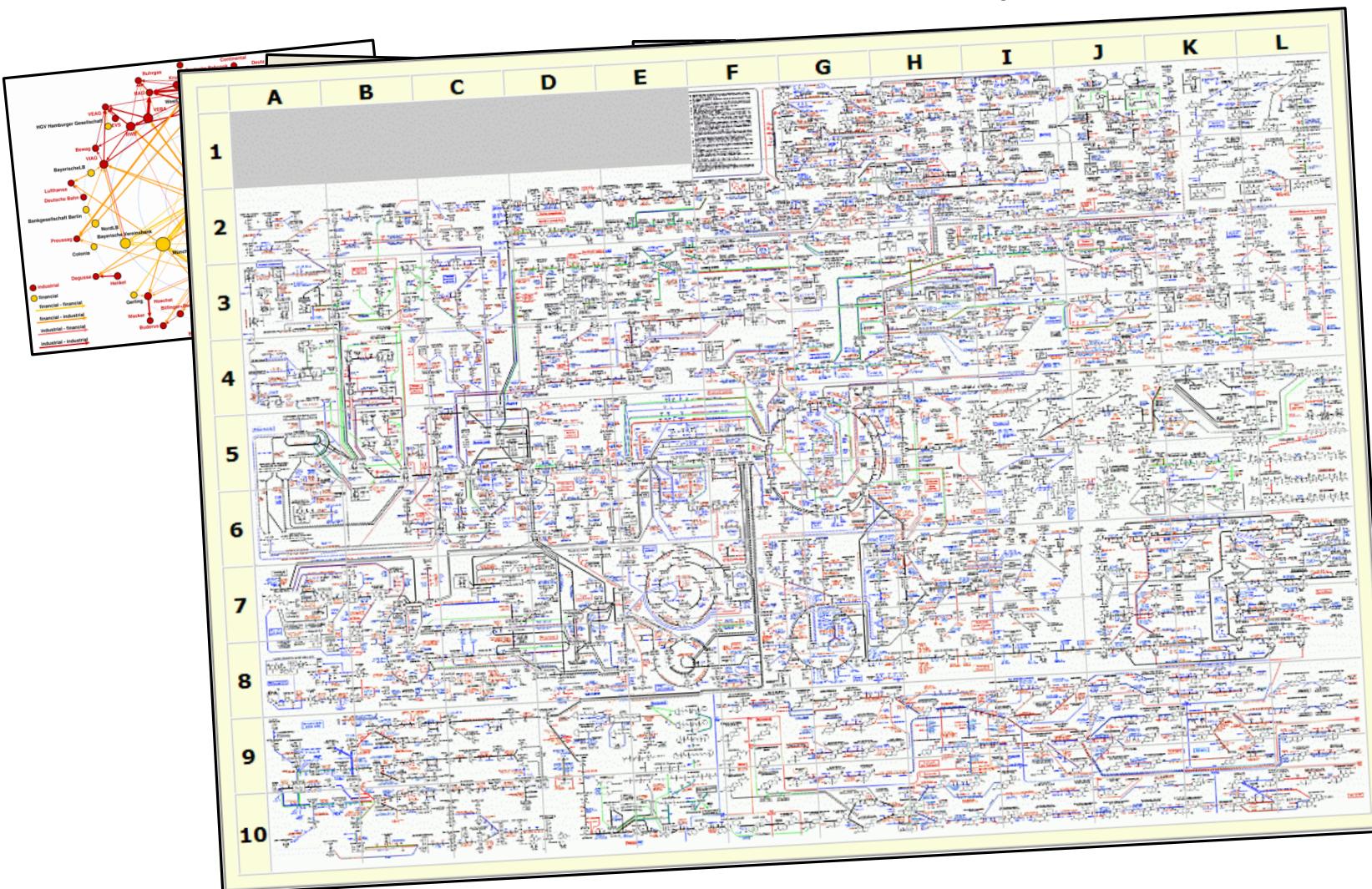
# Why Draw Graphs?

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



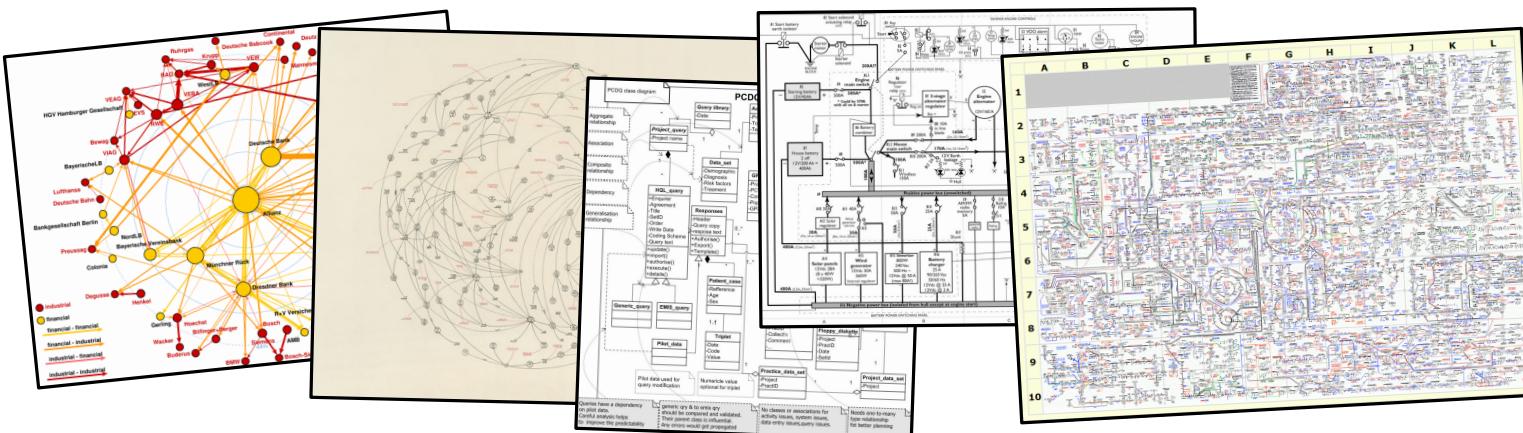
# Why Draw Graphs?

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



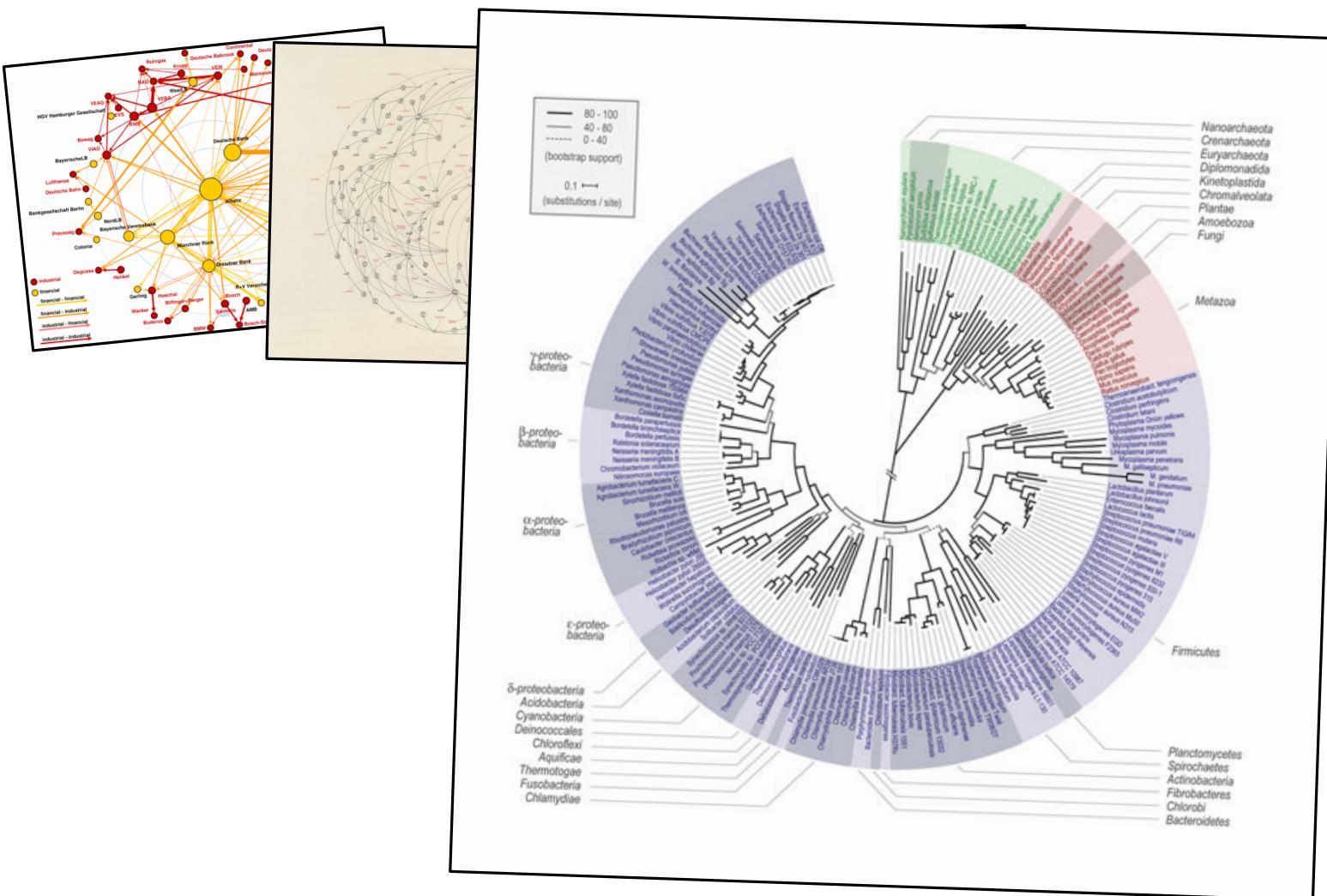
# Why Draw Graphs?

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



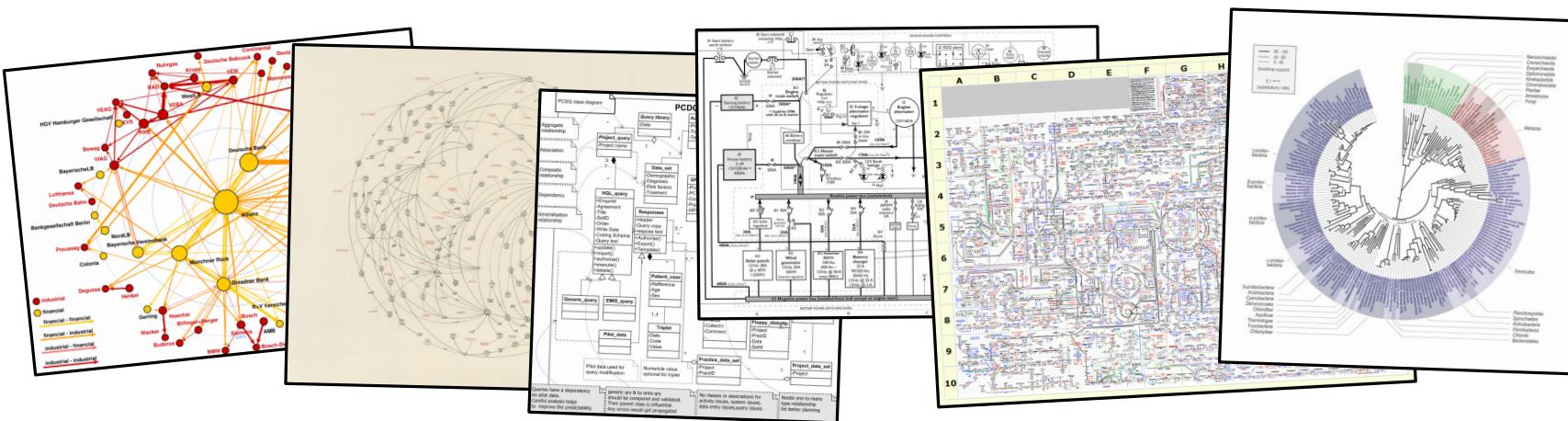
# Why Draw Graphs?

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



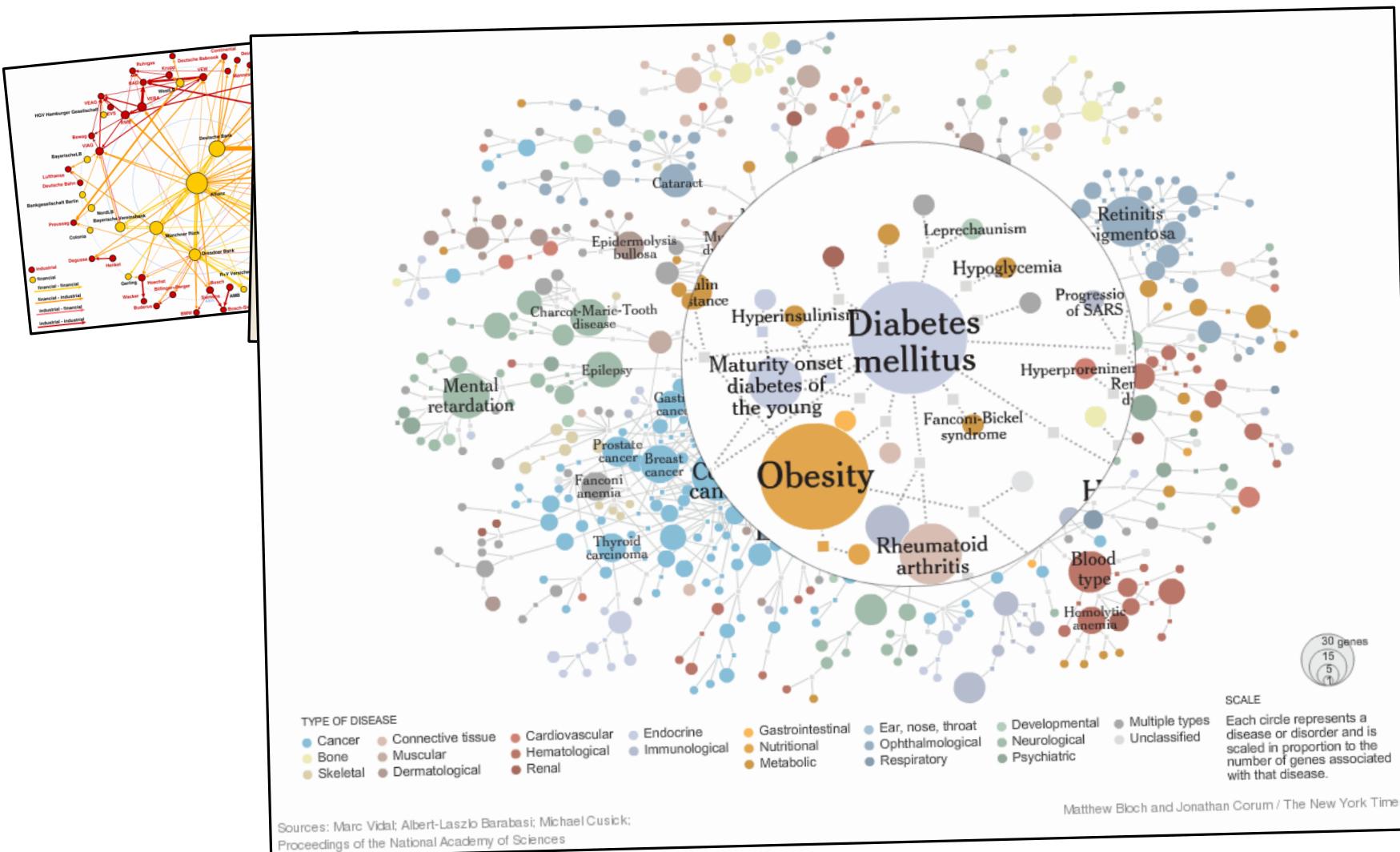
# Why Draw Graphs?

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



# Why Draw Graphs?

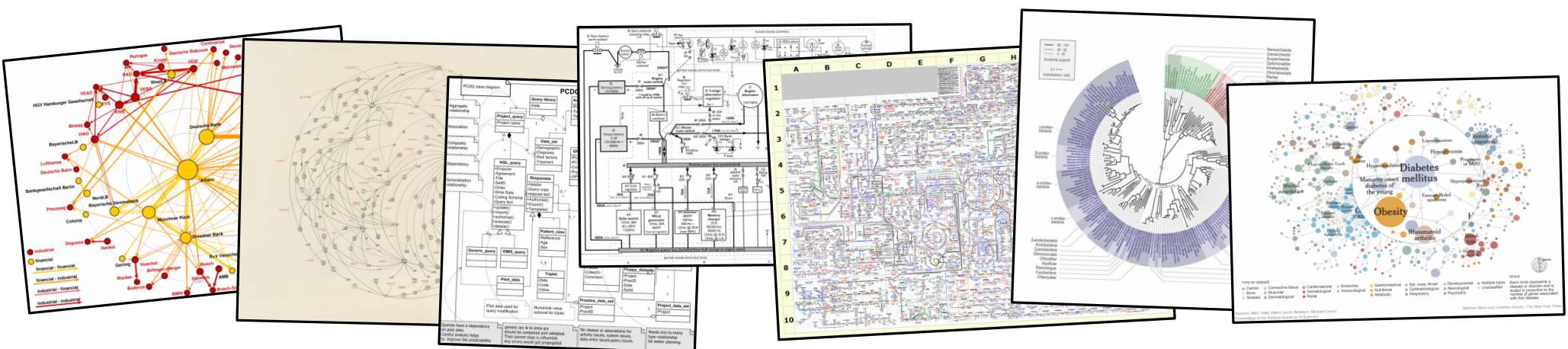
Graphs are mathematical models of real physical and abstract networks (social networks, metabolical networks, VLSI-network, UML-diagrams, citation networks, ...)



# Why Draw Graphs?

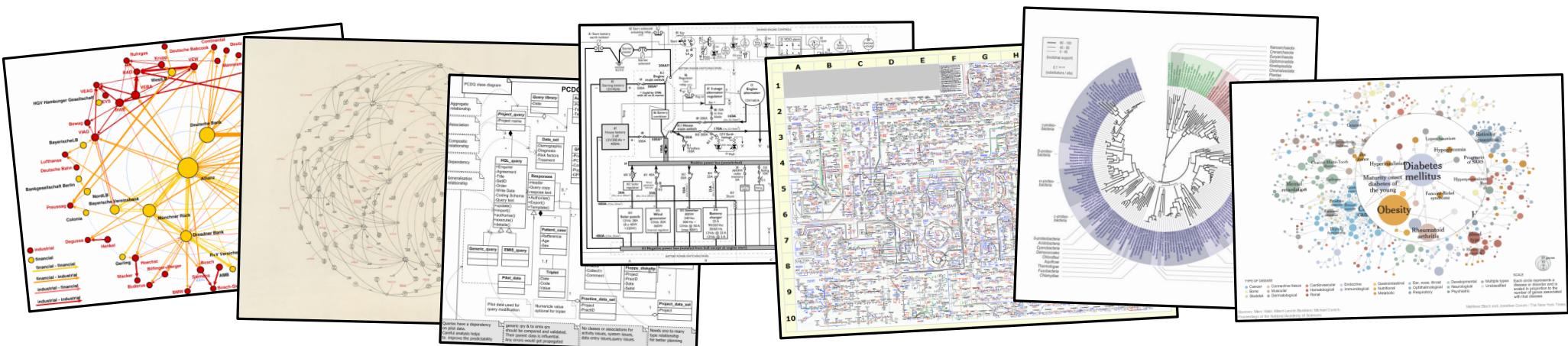


Graphs are mathematical models of real physical and abstract networks (social networks, metabolical networks, VLSI-network, UML-diagrams, citation networks, ...)



# Why Draw Graphs?

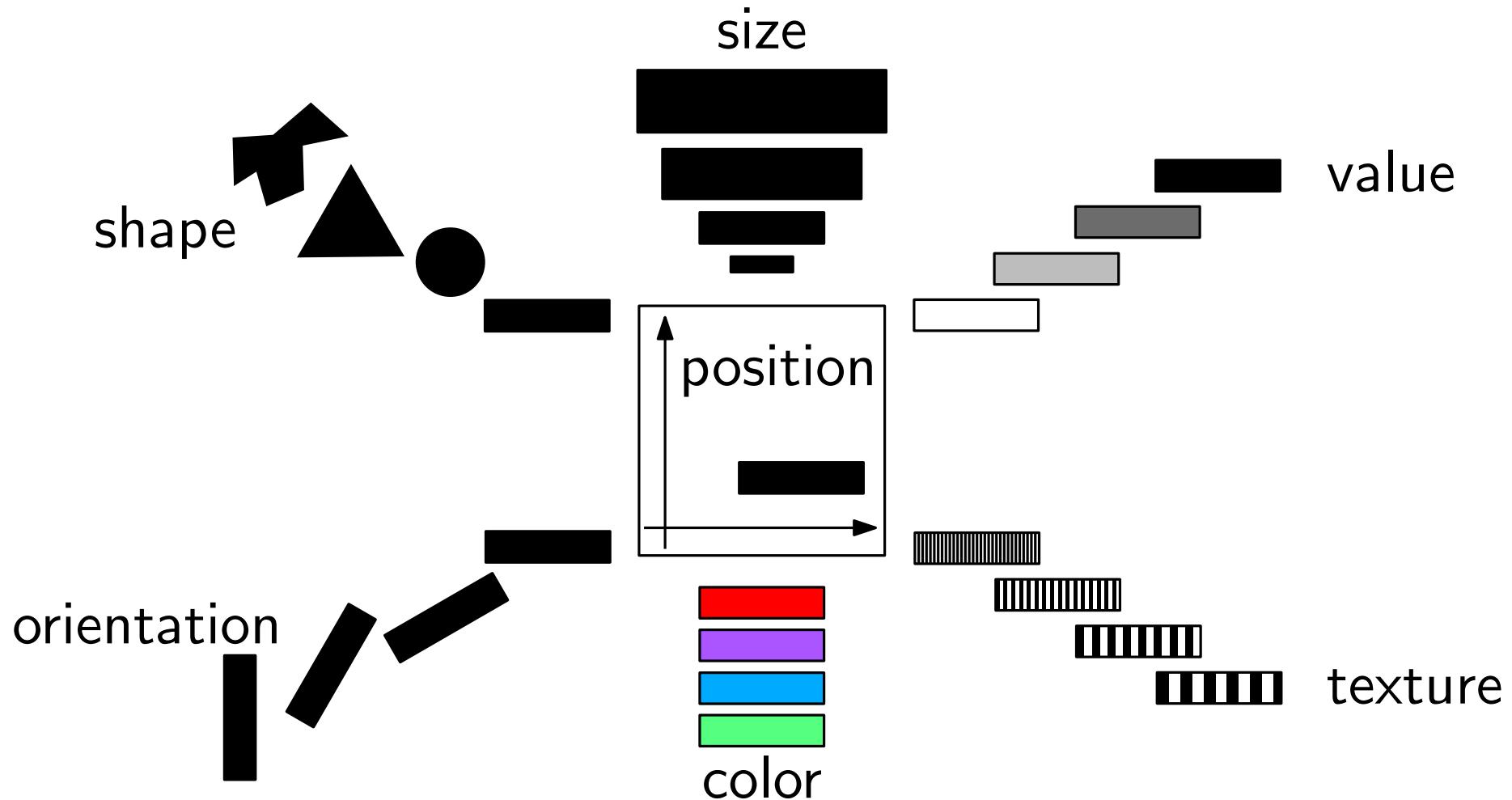
Graphs are mathematical models of real physical and abstract networks (social networks, metabolical networks, VLSI-network, UML-diagrams, citation networks, ...)



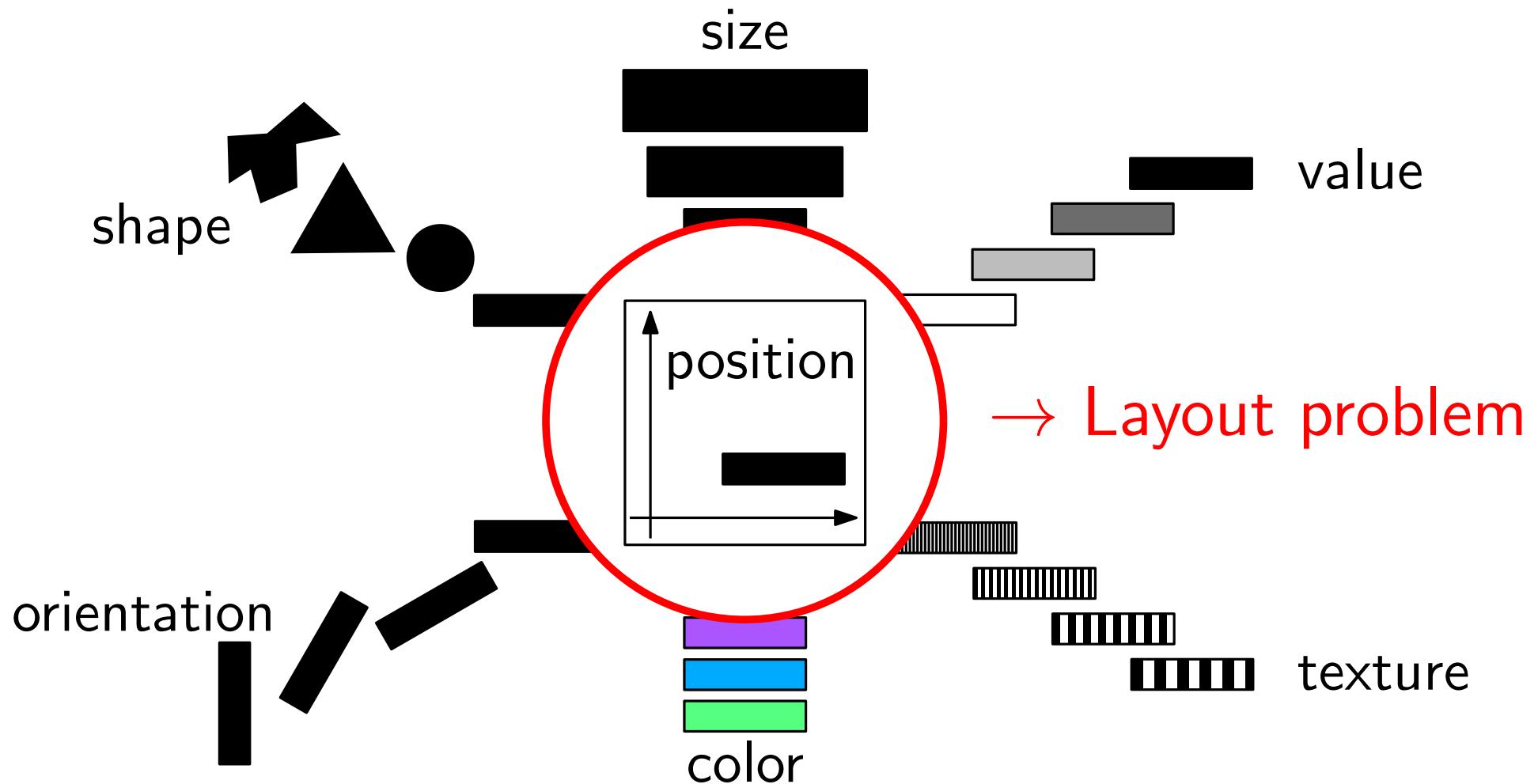
- **People think visually** – without a good visualization, complex graphs are not understandable to us
- A visualization helps to **communicate** and **explore** the graphs/networks
- We need **algorithms** to draw graphs, and make graphs and networks accessible to people

# Visual Variables according to Bertin (1967)

ac 

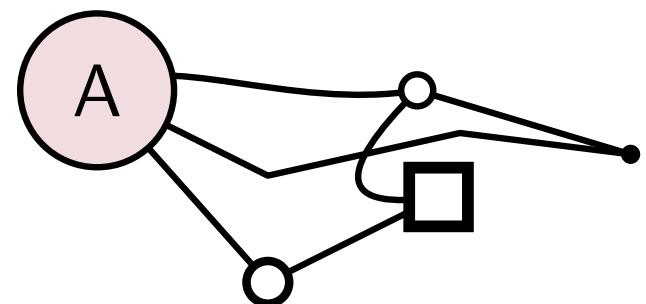


# Visual Variables according to Bertin (1967)



# Layout Problem

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



## Graph visualization problem

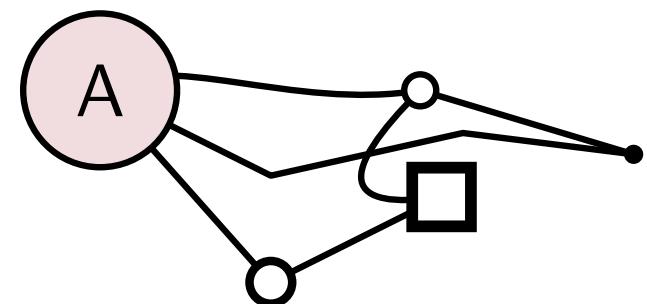
**given:** graph  $G = (V, E)$

**find: good** drawing  $\Gamma$  of  $G$

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

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

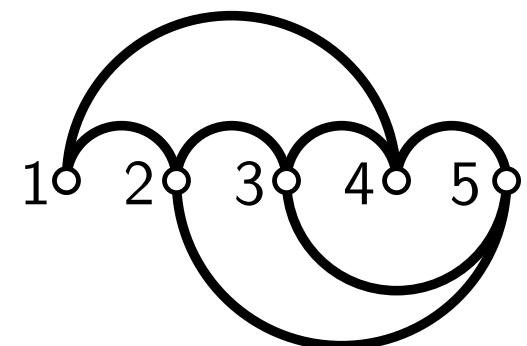
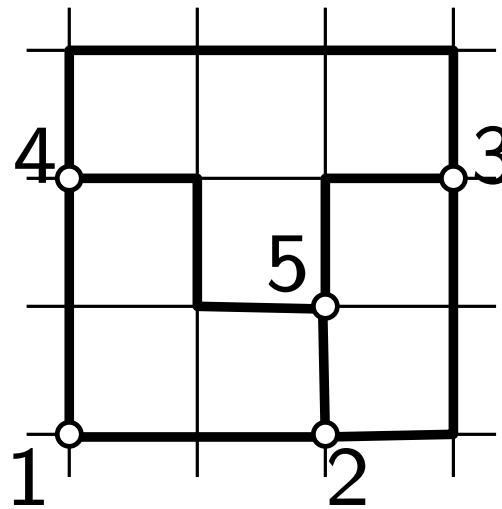
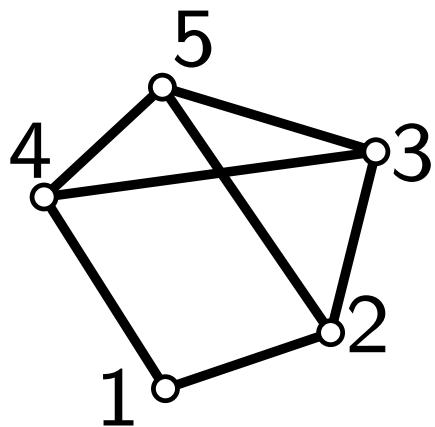
## criteria

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

# Quality Criteria for Graph Drawings

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

- straight-line edges
- orthogonal edges (with  $90^\circ$  bends)
- grid drawings
- crossing-free
- ...

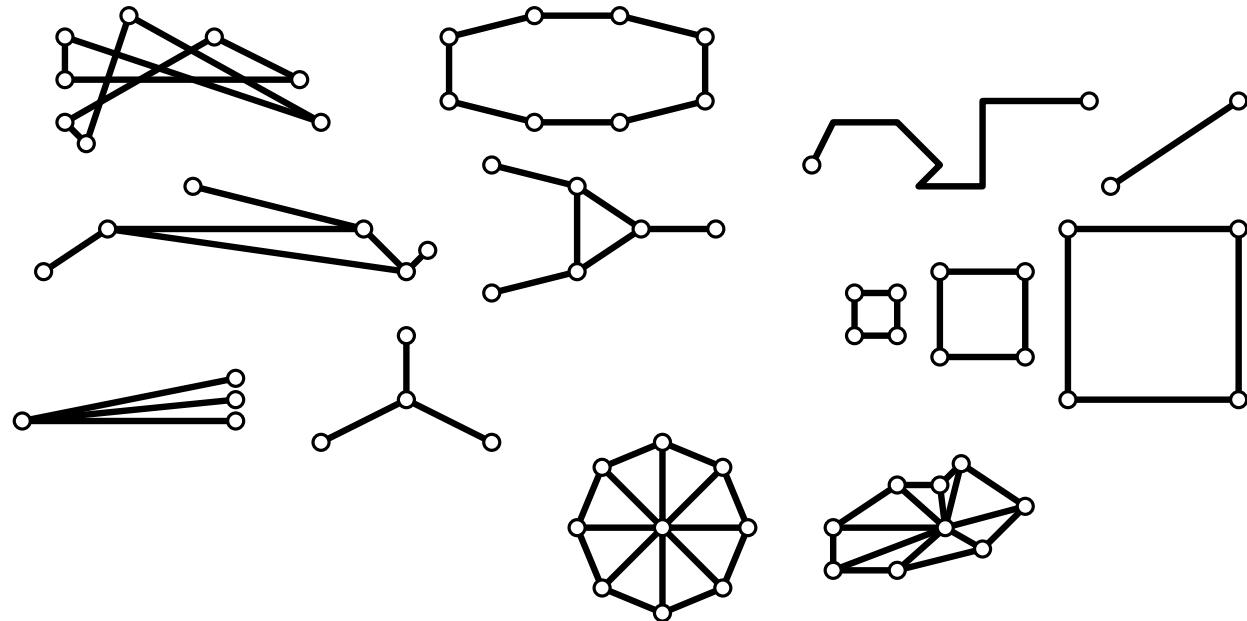


# Quality Criteria for Graph Drawings

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  $\Gamma$  of  $G$  that

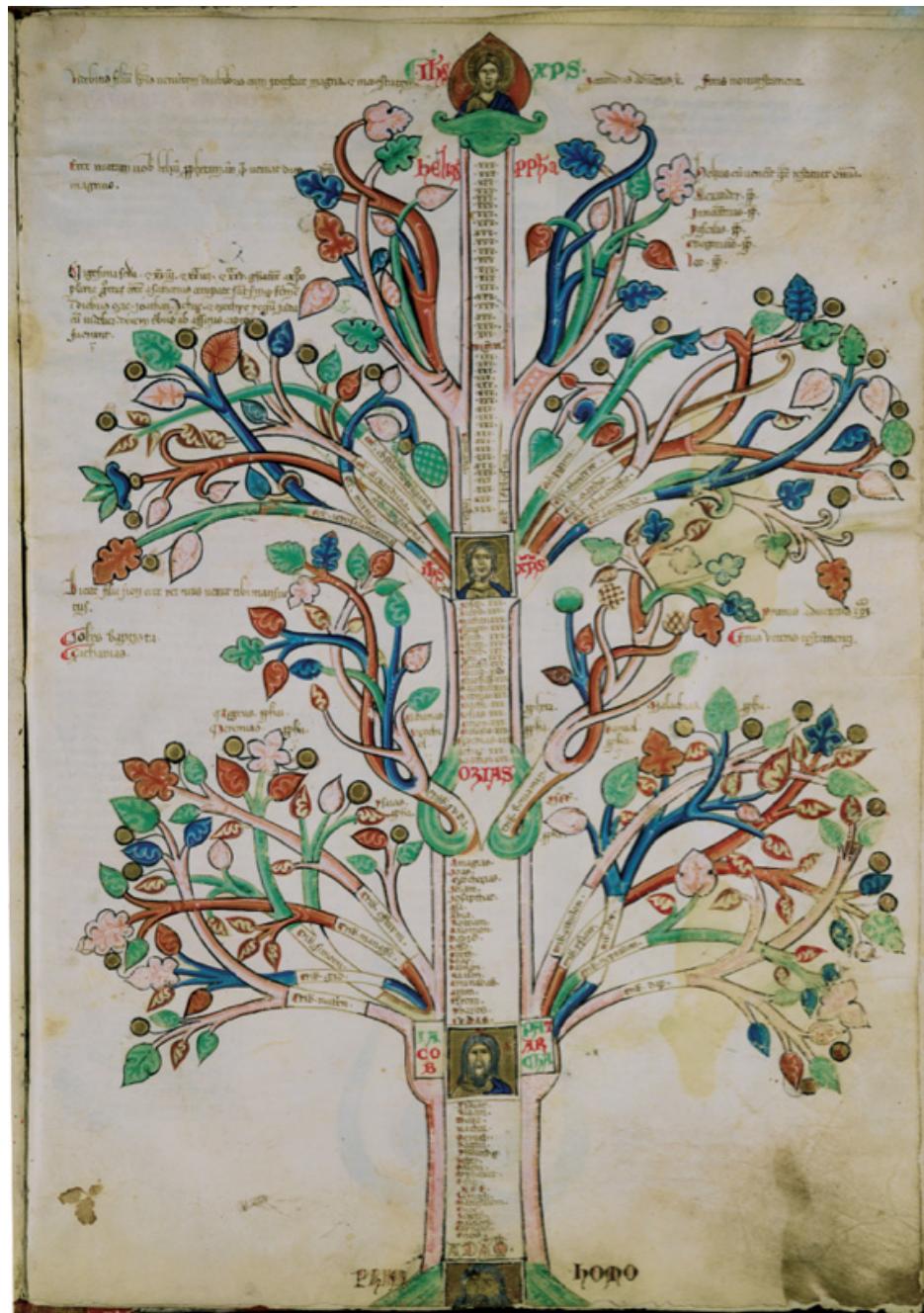
- complies with the given drawing conventions
- optimizes the given aesthetics
- satisfies the partial/local constraints

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

# Example Gallery<sup>\*</sup>

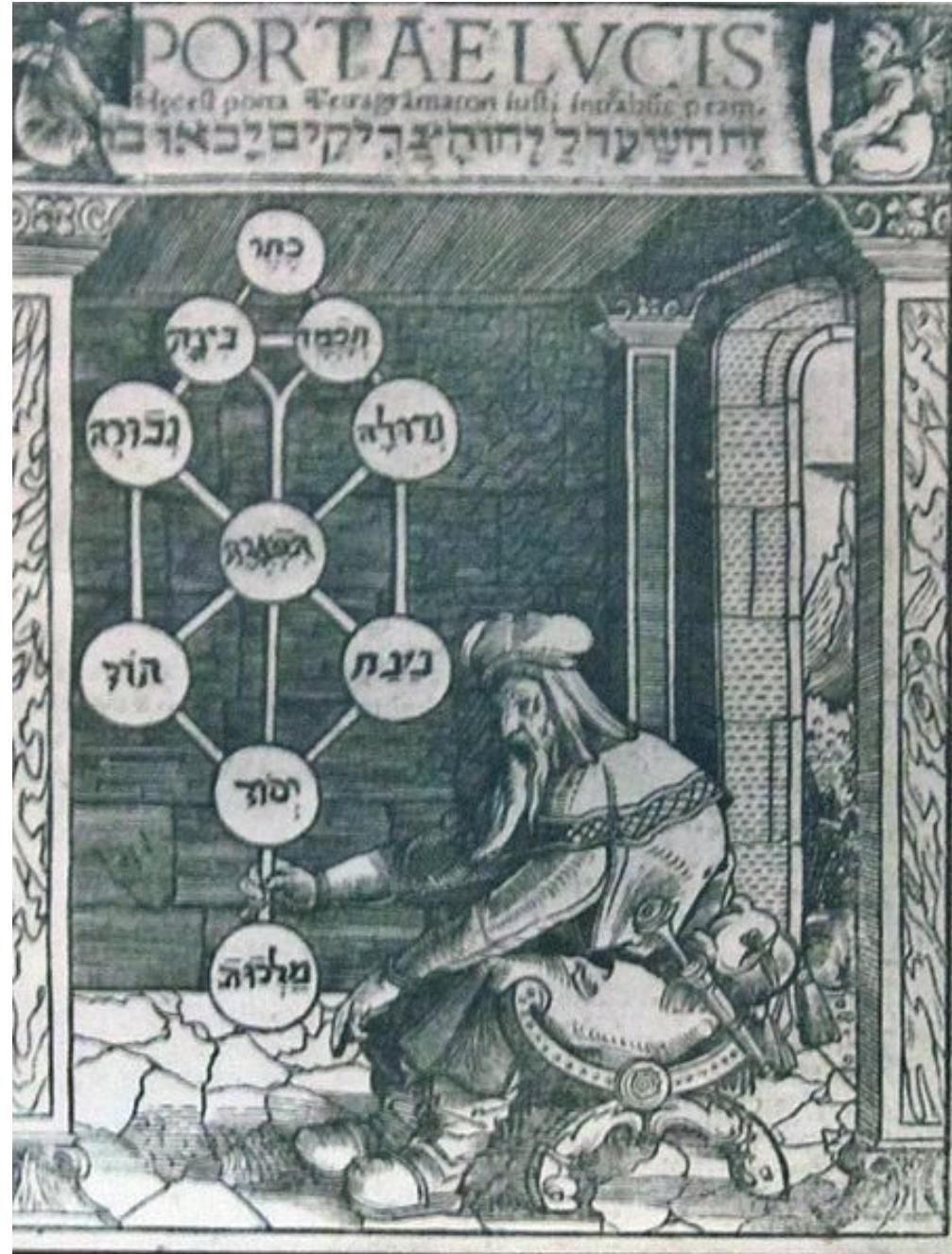
\* for more examples see, e.g., collections like [www.visualcomplexity.com](http://www.visualcomplexity.com)

# Biblical Characters and Events (1202)



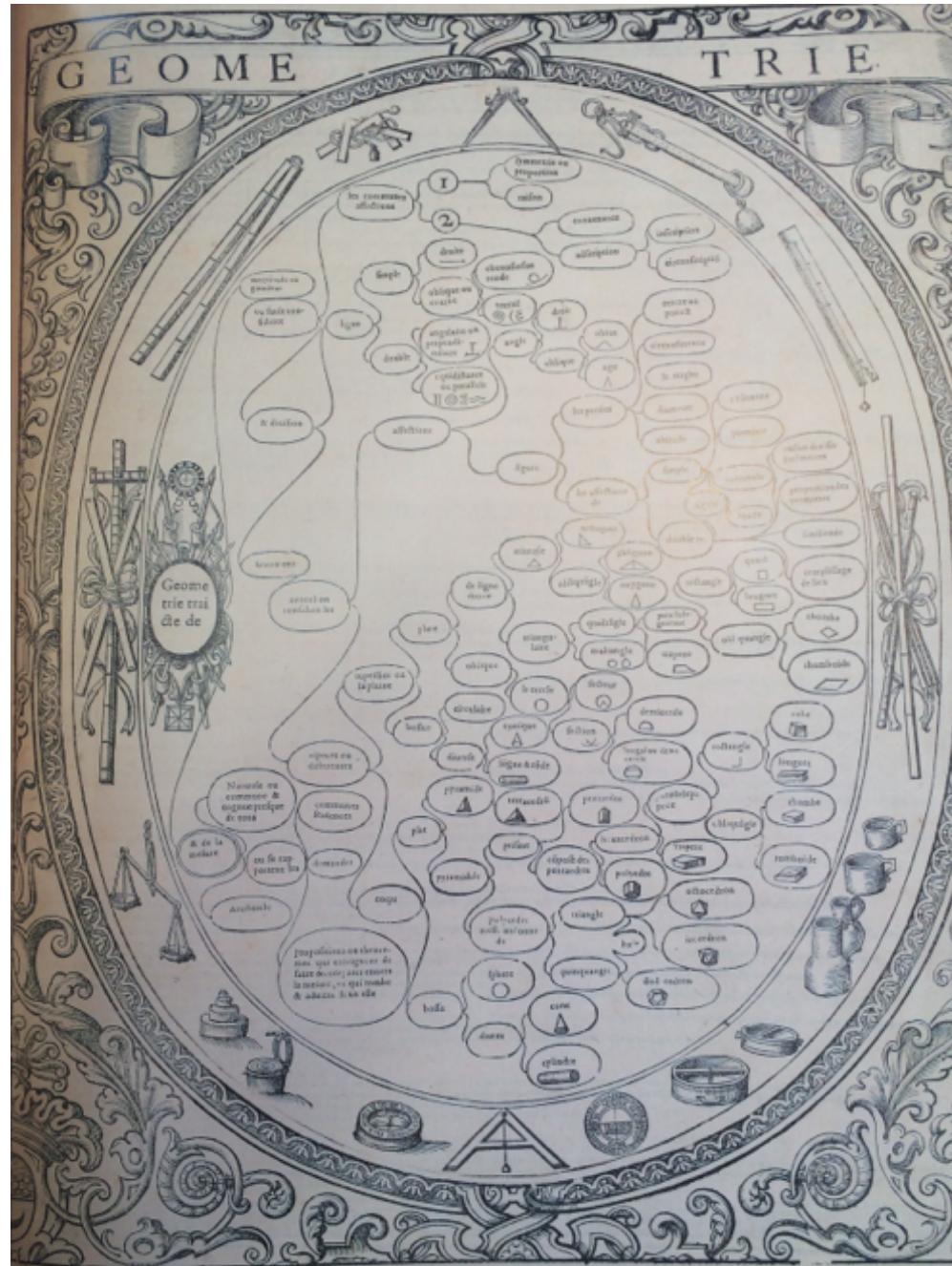
source: Joachim de Fiore

# Tree of Life (1516)



source: Paul Riccius, Portae Lucis

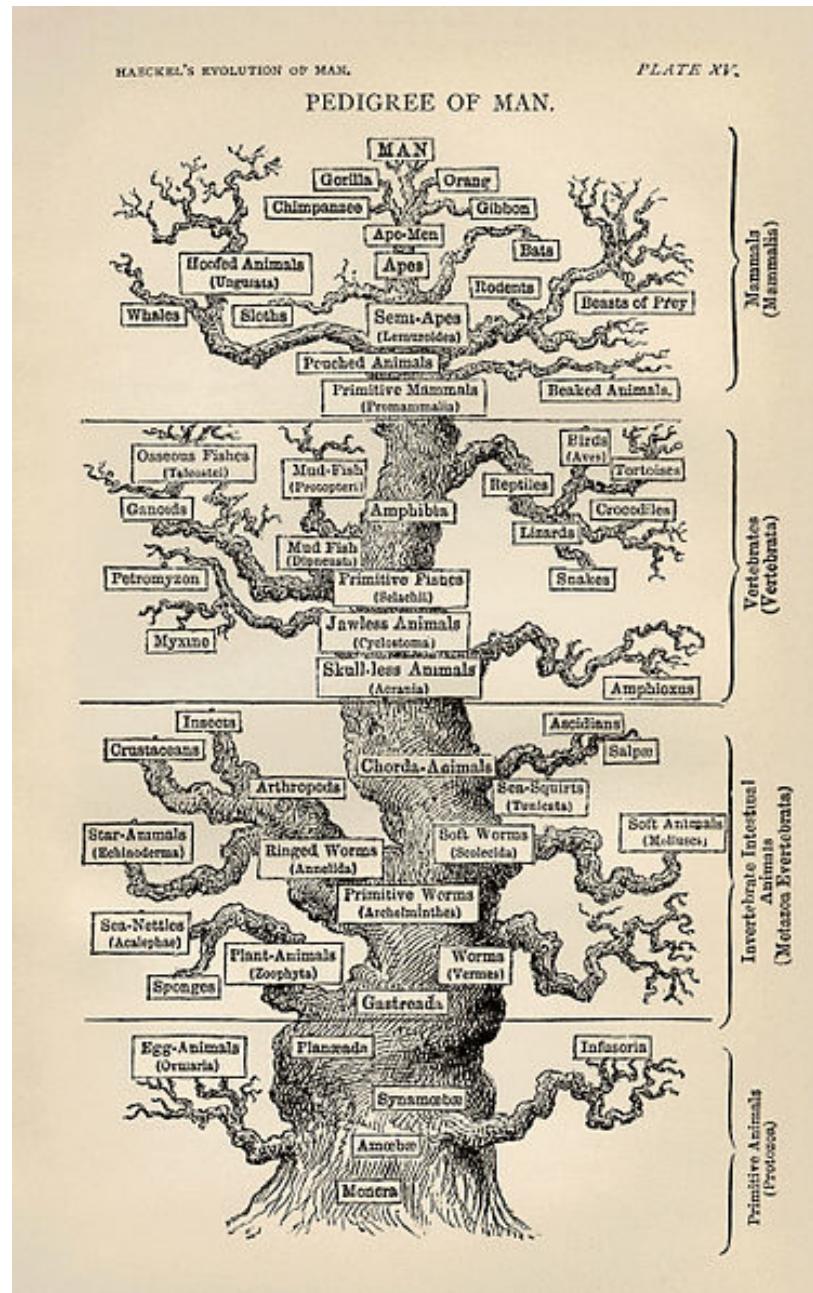
# Geometrical Concepts (1587)



source: Christophe de Savigny

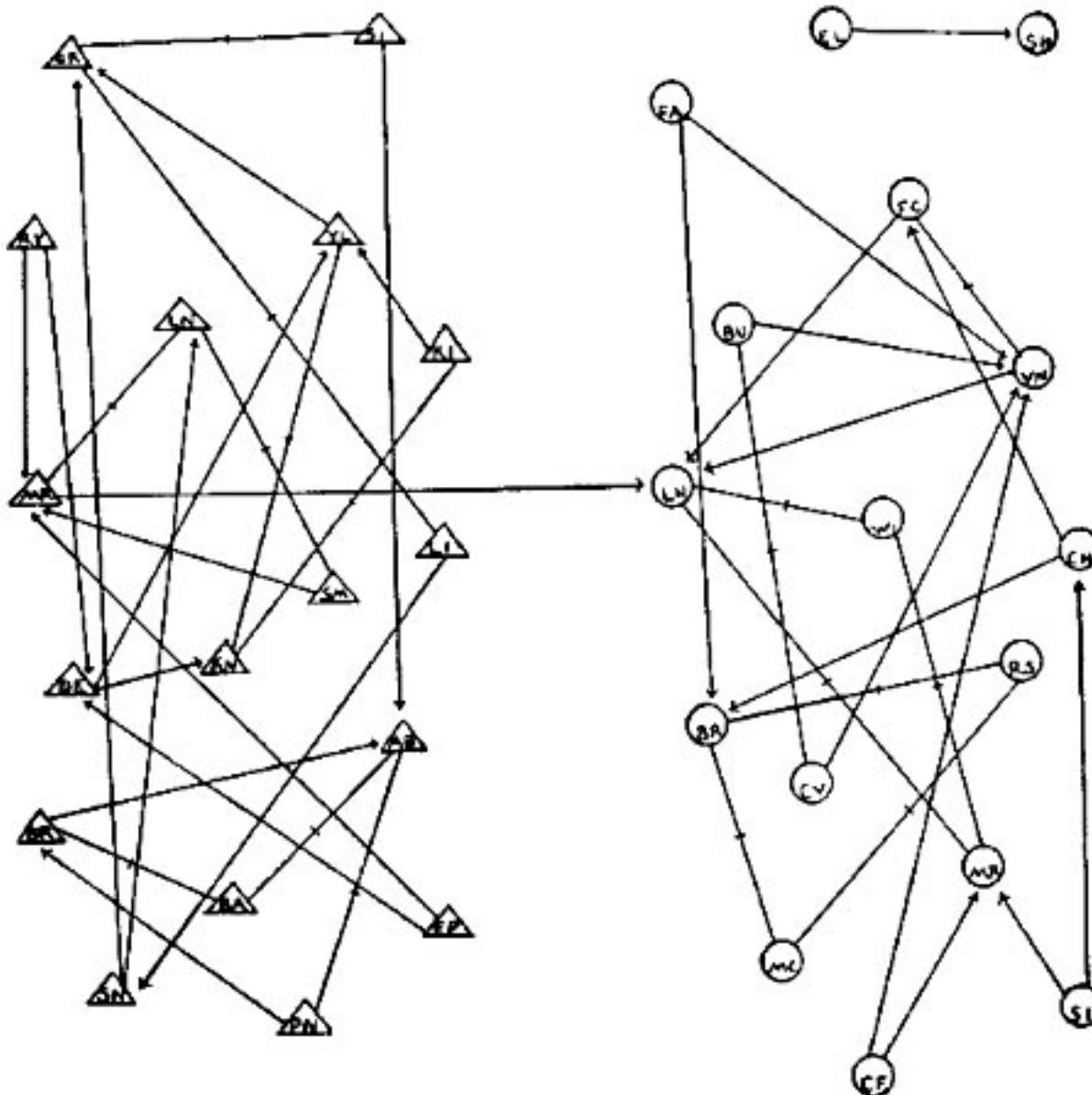
# Evolutionary Tree (1879)

ac



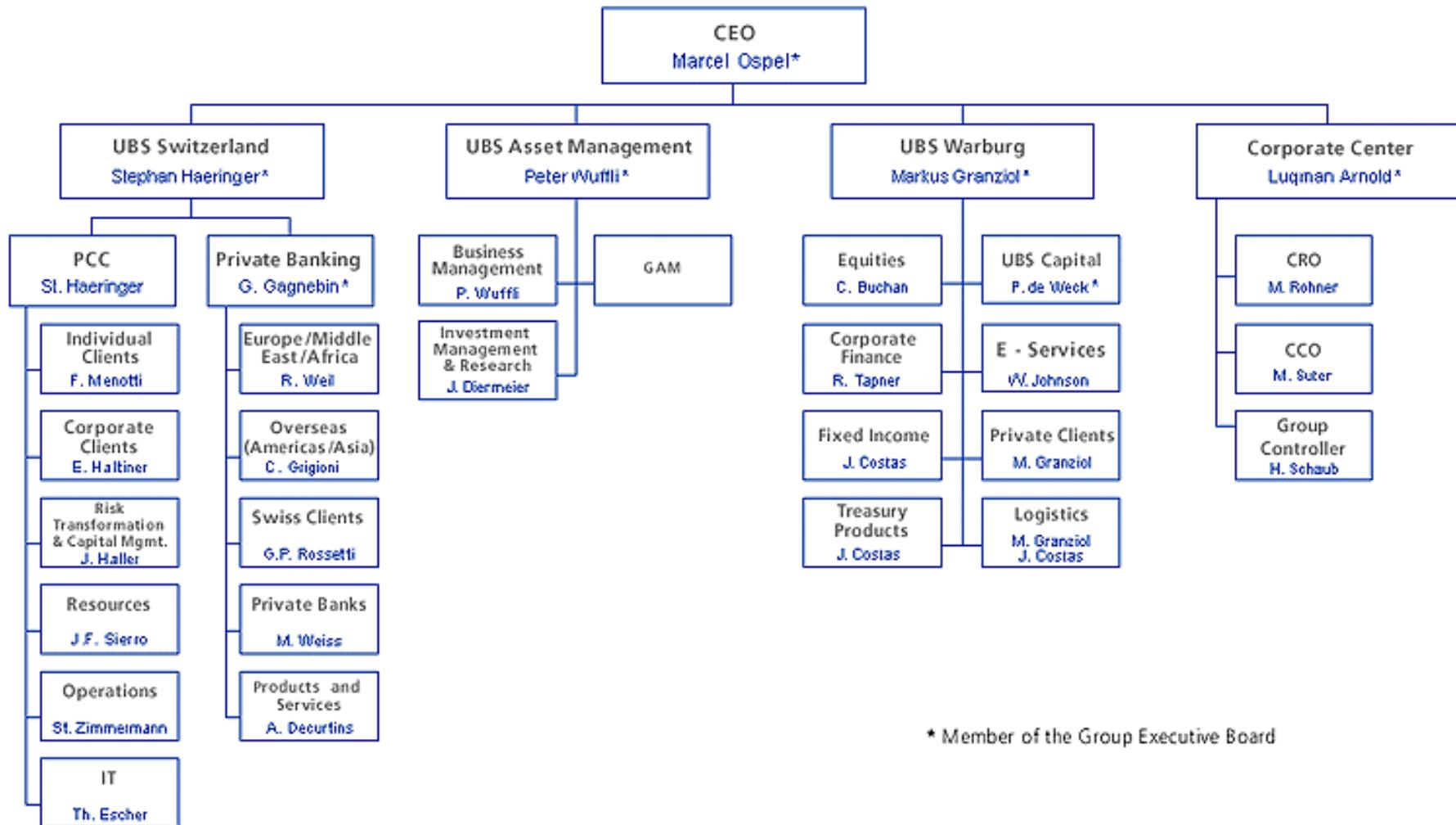
source: Ernst Haeckel

# Sociogram (1933)



source: Moreno

# Social Network – Organigram UBS



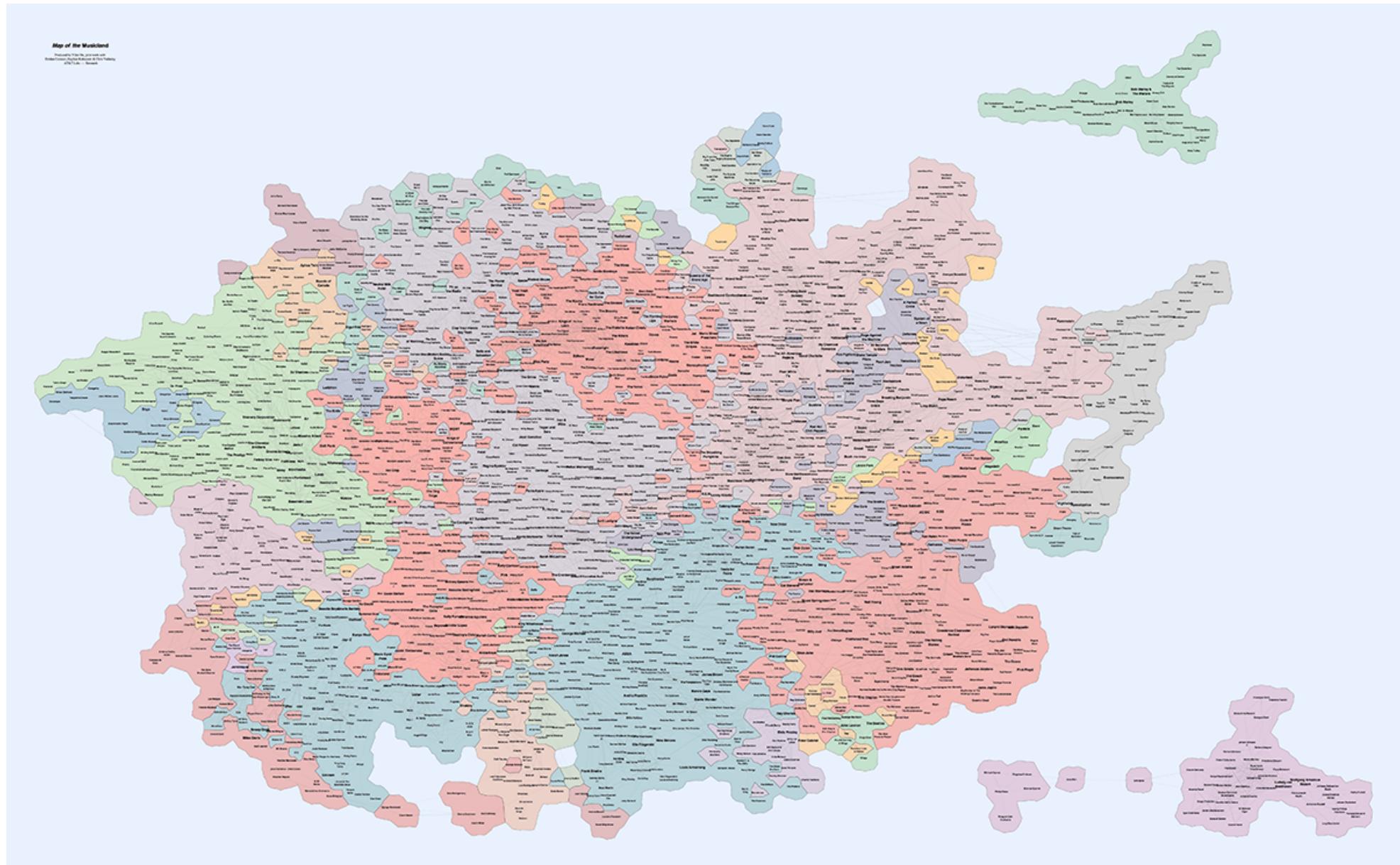
# CPAN Developer Graph



source: [cpan-explorer.org](http://cpan-explorer.org)

# last.fm music graph as political map

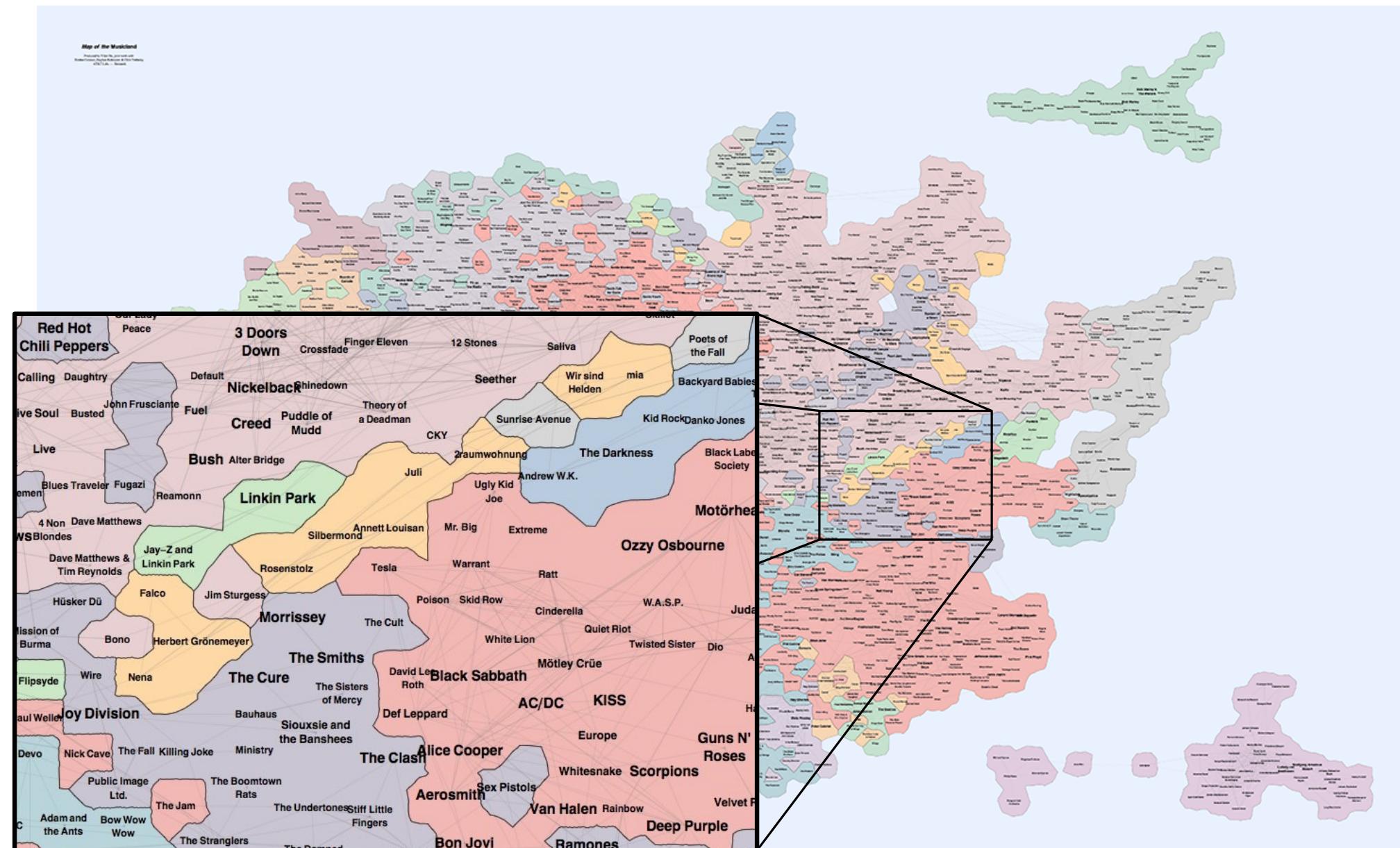
ac III



(Gansner, Hu, Kobourov: GMap, 2009)

# last.fm music graph as political map

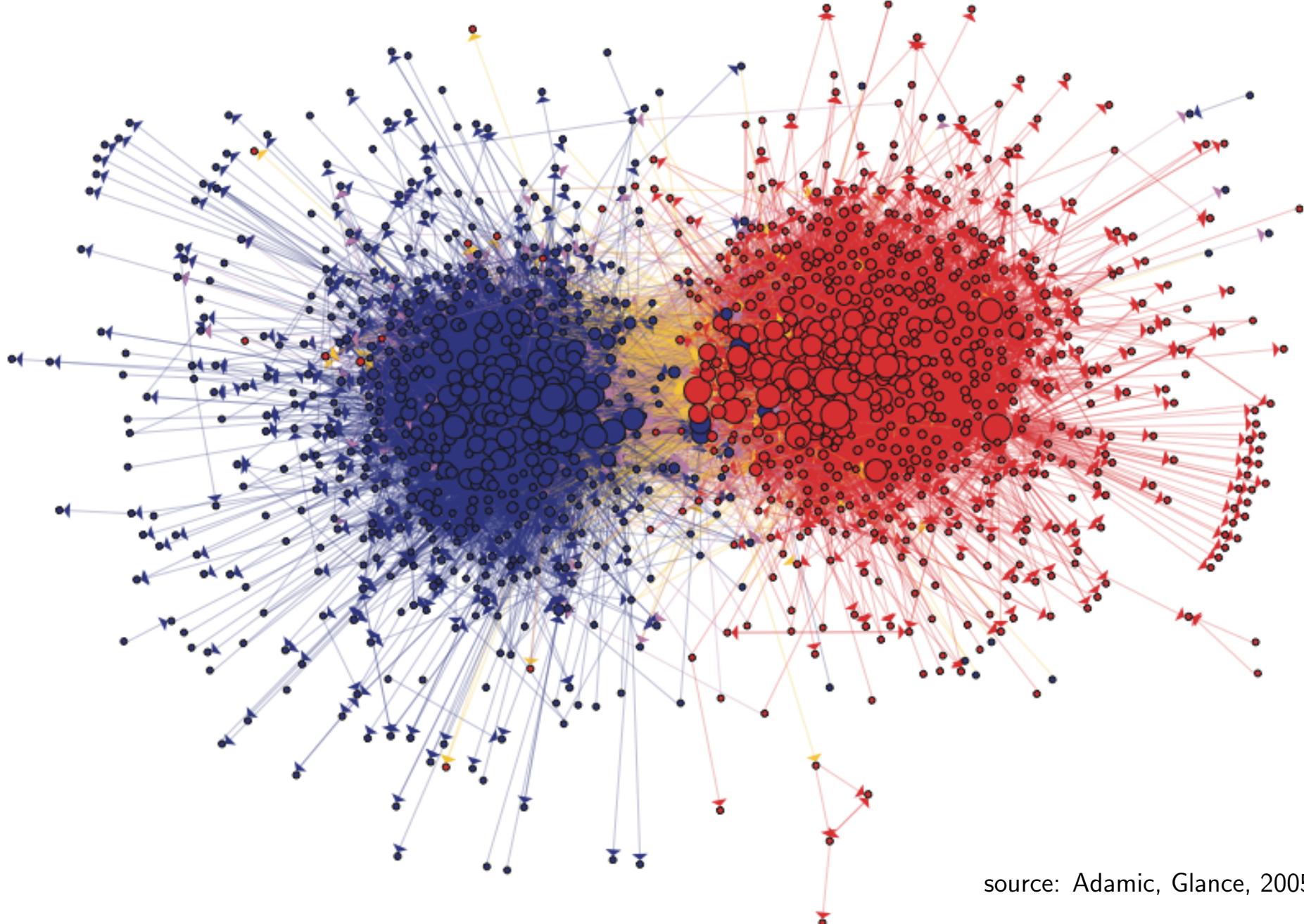
ac III



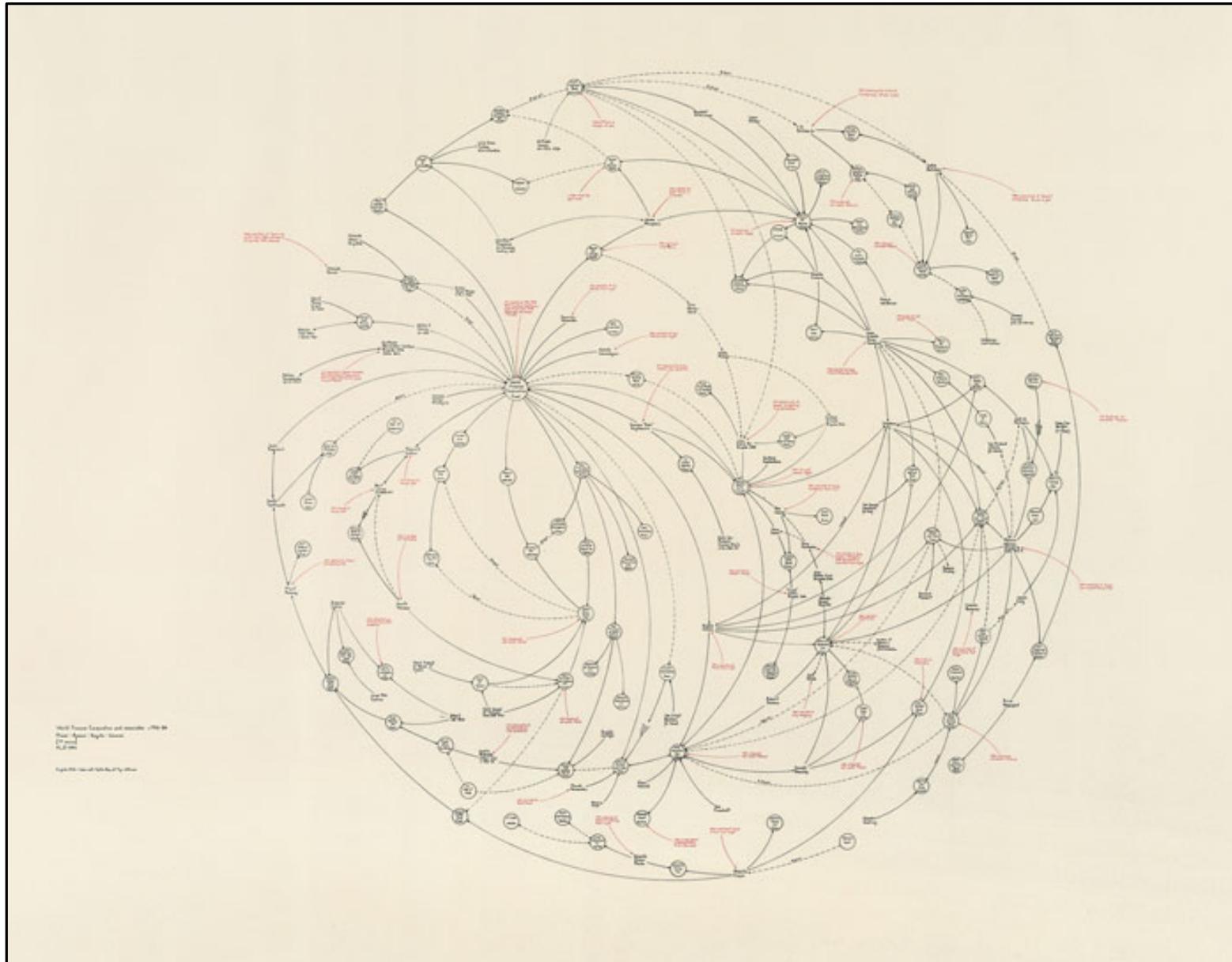
(Gansner, Hu, Kobourov: GMap, 2009)

# Blogosphere 2004 Elections USA

ac III



# Social Network – World Financial System



World Finance Corporation  
© Mark Lombardi

# Social Networks – State Funds

ac

## FOLLOW THE MONEY

### The New Global Wealth Machine

Sovereign wealth funds have emerged in recent months as the world's power brokers. They have used their tremendous wealth to make big cross-border investments and prop up some of Wall Street's best-known firms. The increased activity comes as other kinds of acquirers have been sidelined by the credit crisis. These funds are state-sponsored investment vehicles and have combined assets of \$2 trillion. With that much dry powder, sovereign funds dwarf the formerly booming private equity industry — and in some cases, compete directly with it. The Government of Singapore Investment Corporation has been the most active among the world's sovereign funds, making its deputy chairman, Tony Tan, a major center of gravity. Wall Street veterans always follow the money, so many of the big-name advisers to New York and London have found themselves traveling the globe playing international matchmaker to these funds. But sovereign funds have also learned the downside of deal-making: some of their blockbuster transactions have been big money losers so far.

The question is where all that money will go next. ANDREW ROSS SORKIN

#### The Advisers

Selected financial advisers who worked on more than one of the top 20 deals.

##### CITIGROUP



##### GOLDMAN SACHS GROUP



### CITIGROUP MERRILL LYNCH

**CITIGROUP**  
Robert E. Rubin, Chairman  
**MERRILL LYNCH**  
John A. Thain, Chairman and C.E.O.

**CARLYLE GROUP**  
David Rubenstein, Co-founder and managing director

**DAVIS SELECTED ADVISED (U.S.)**

**GEORGE SOHN FUND**

**JP MORGAN CHASE**

**KLINGER GROUP**

**LEHMAN BROTHERS**

**LJ LEWIS GROUP**

**MANAGEMENT CONSULTANTS**

**MERRILL LYNCH**

**MERRILL LYNCH & CO.**

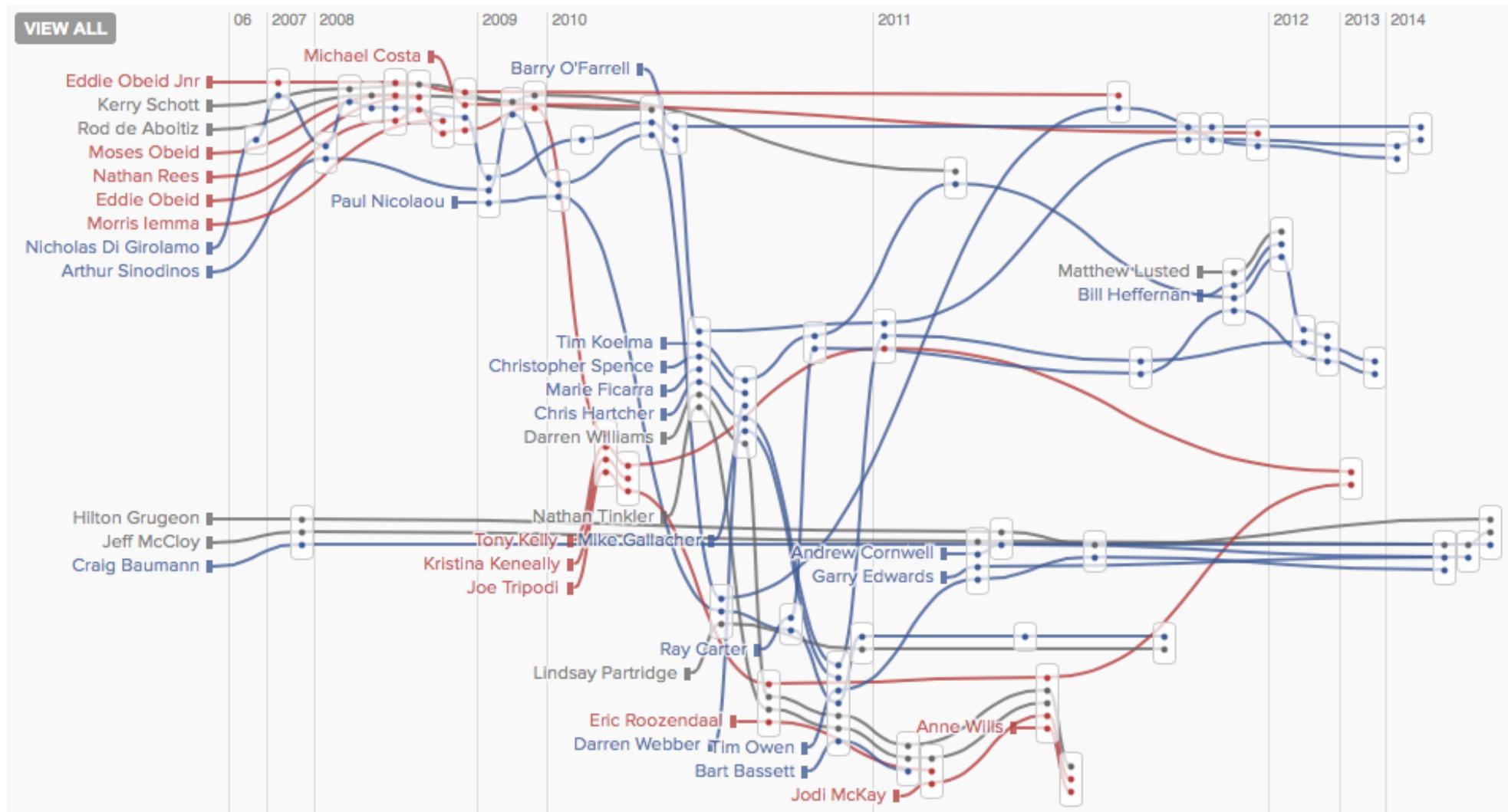
**MERRILL LYNCH & CO. INC.**

**MERRILL LYNCH & CO. INC. (U.S.)**

**MERRILL LYNCH & CO. INC. (U.S.A.)**

# Temporal Graph Layout: Storylines

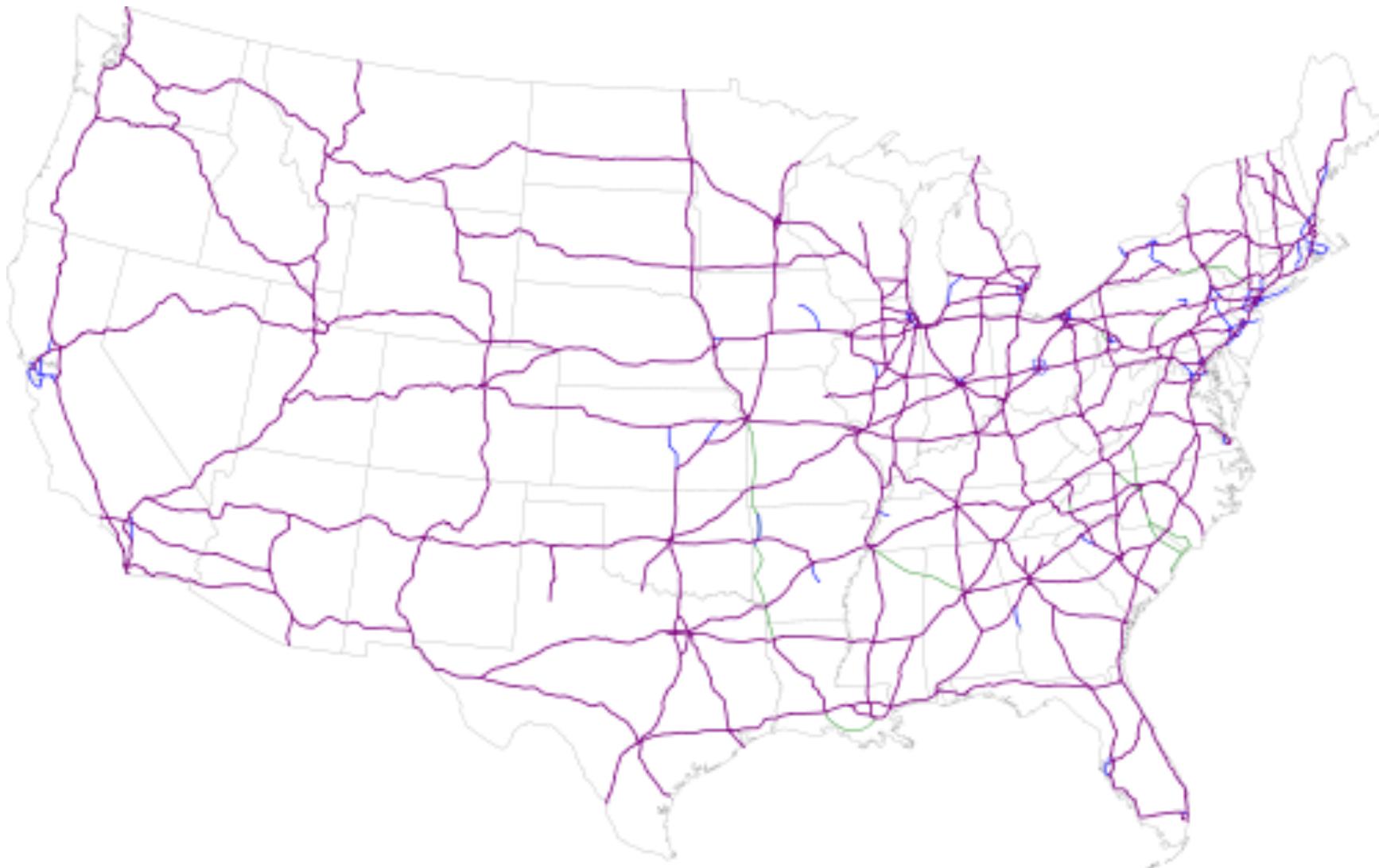
ac III



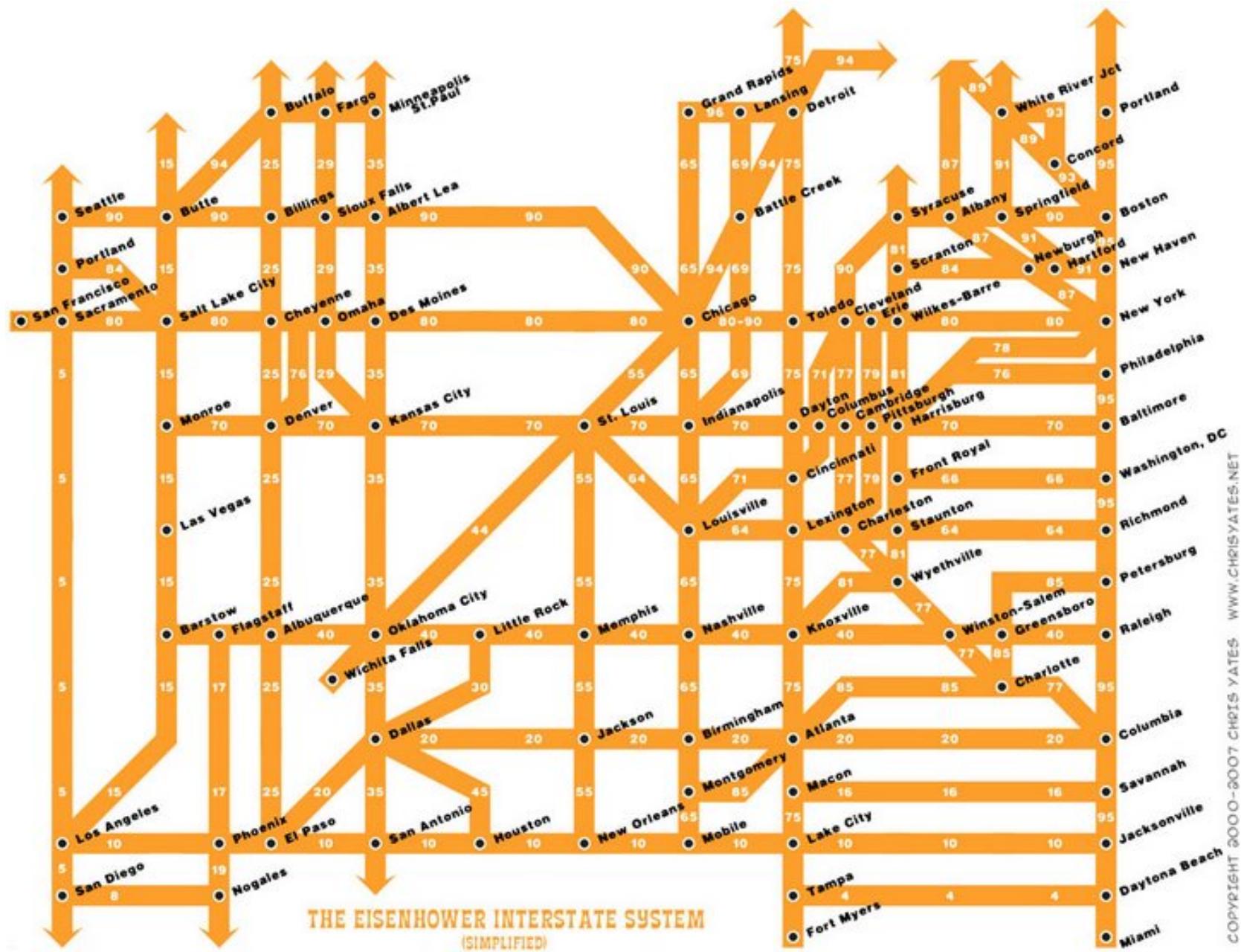
source: ABC news, Australia

# Traffic Network – Highways USA

ac III



# Traffic Network – Highways USA



# London Tube Map (1933)

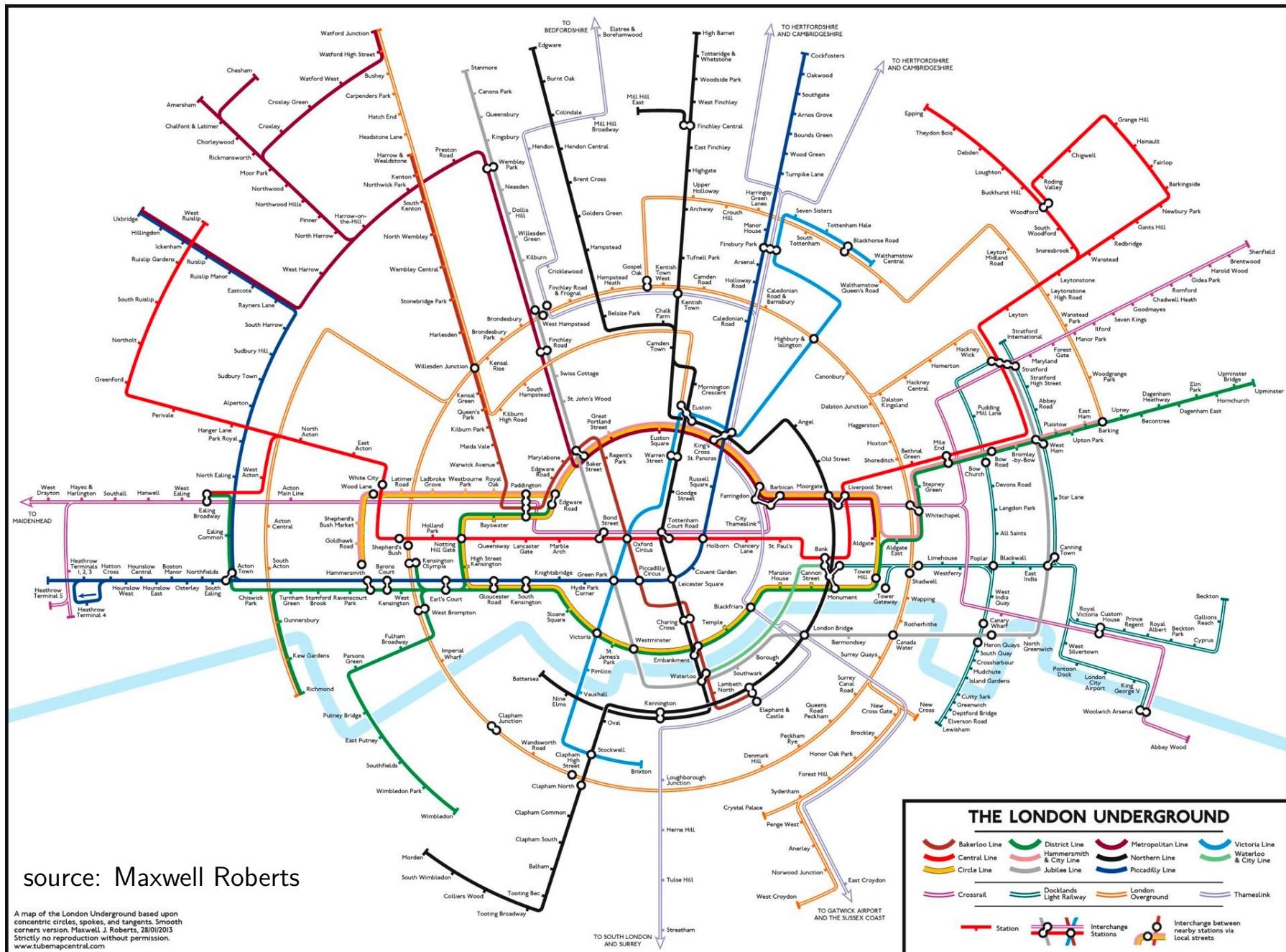
aci



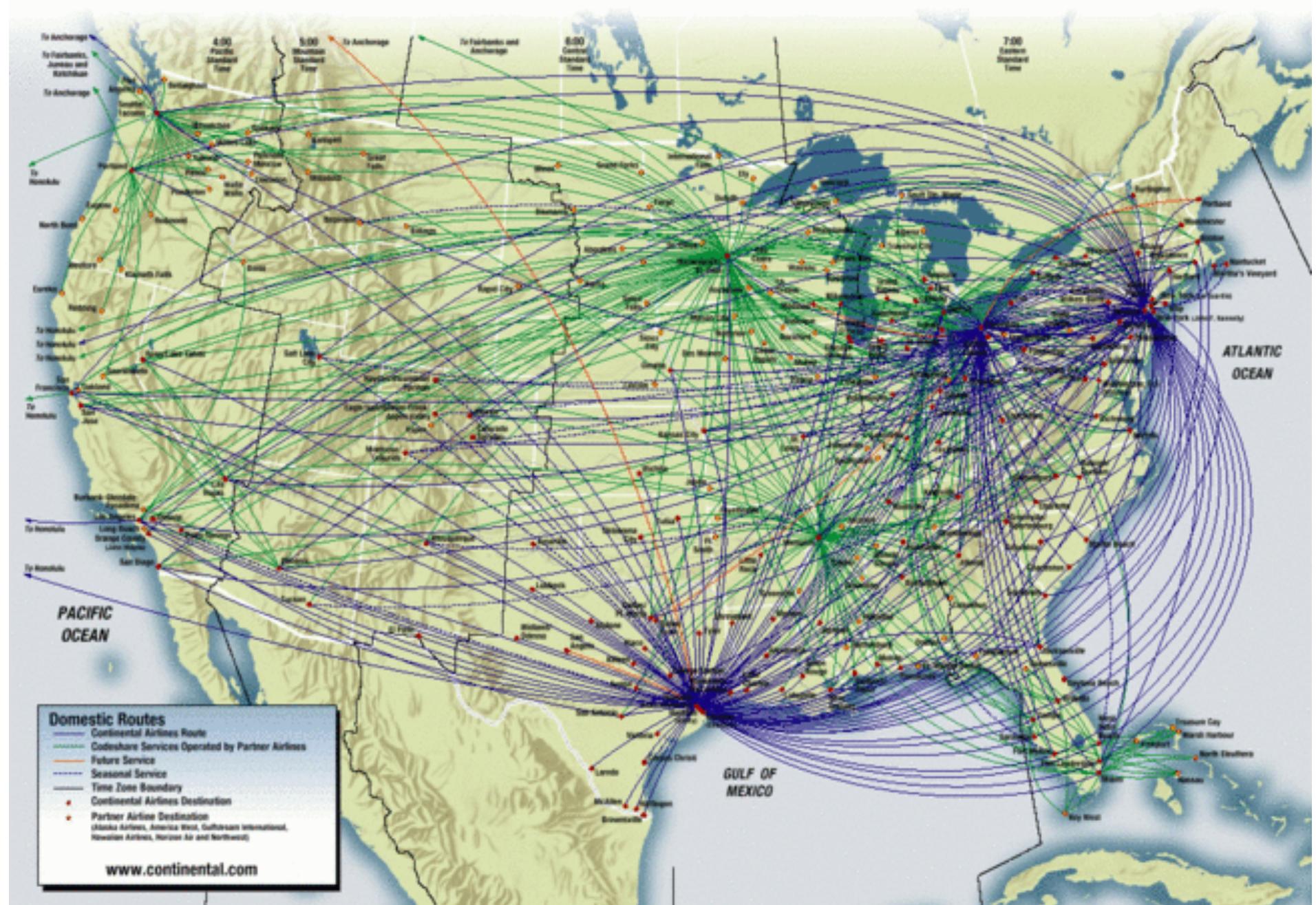
source: Henry Beck

# Concentric Tube Map

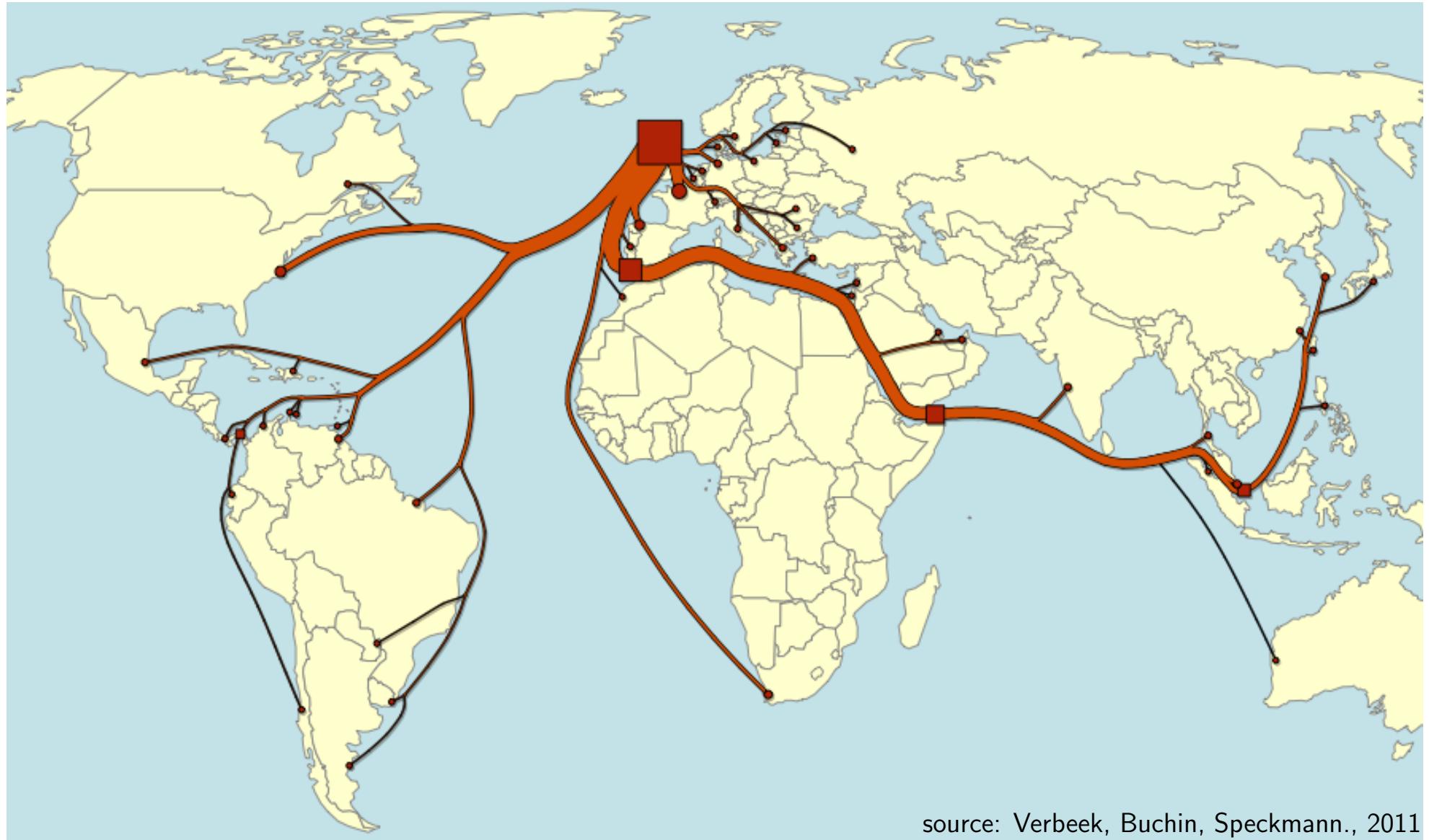
ac III



# Traffic Network – Flight Destinations Continental ac|||

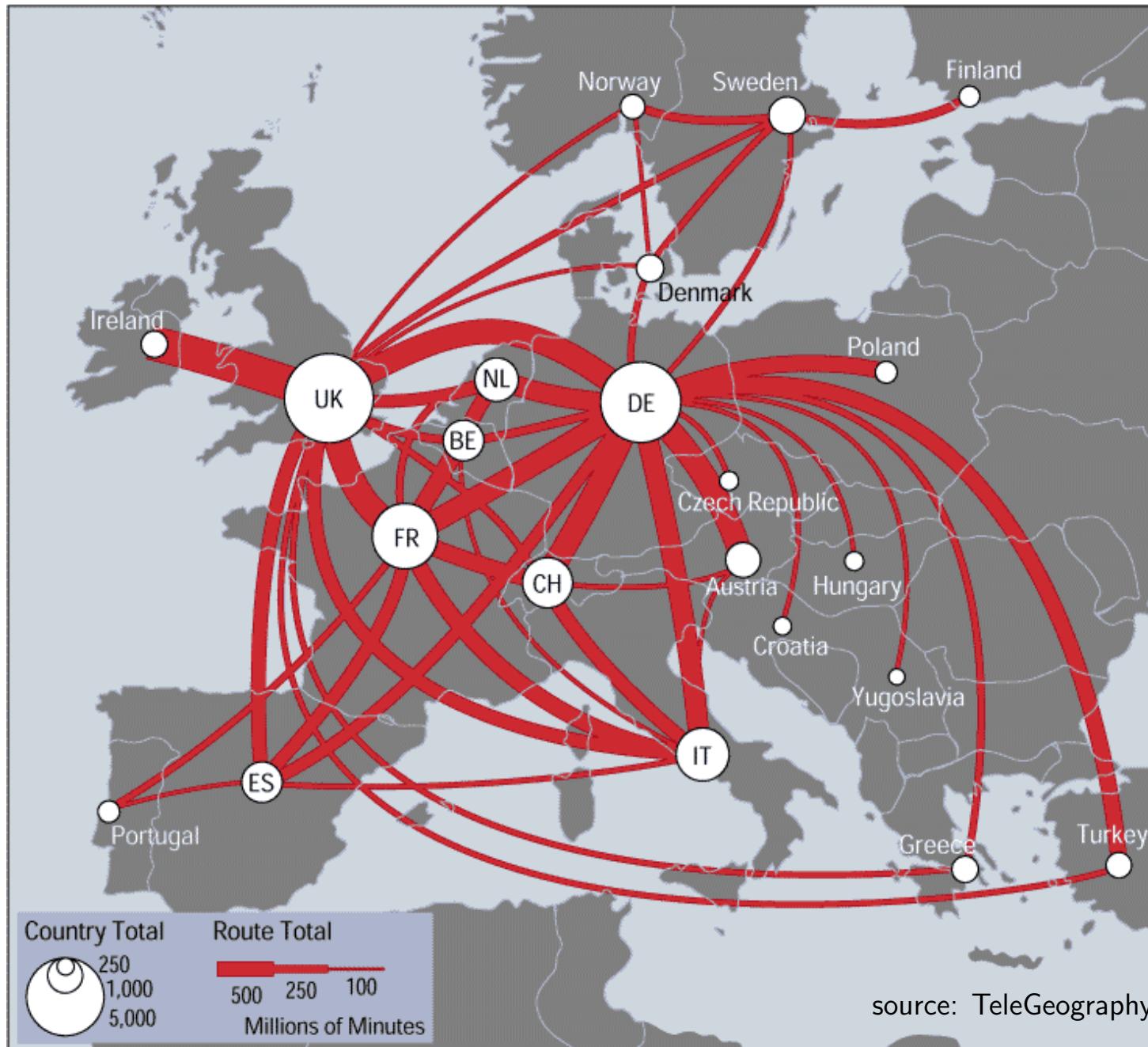


# Flow-Map: Whiskey Exports



# Phone Volume Map

ac III



# Monitoring Energy Networks

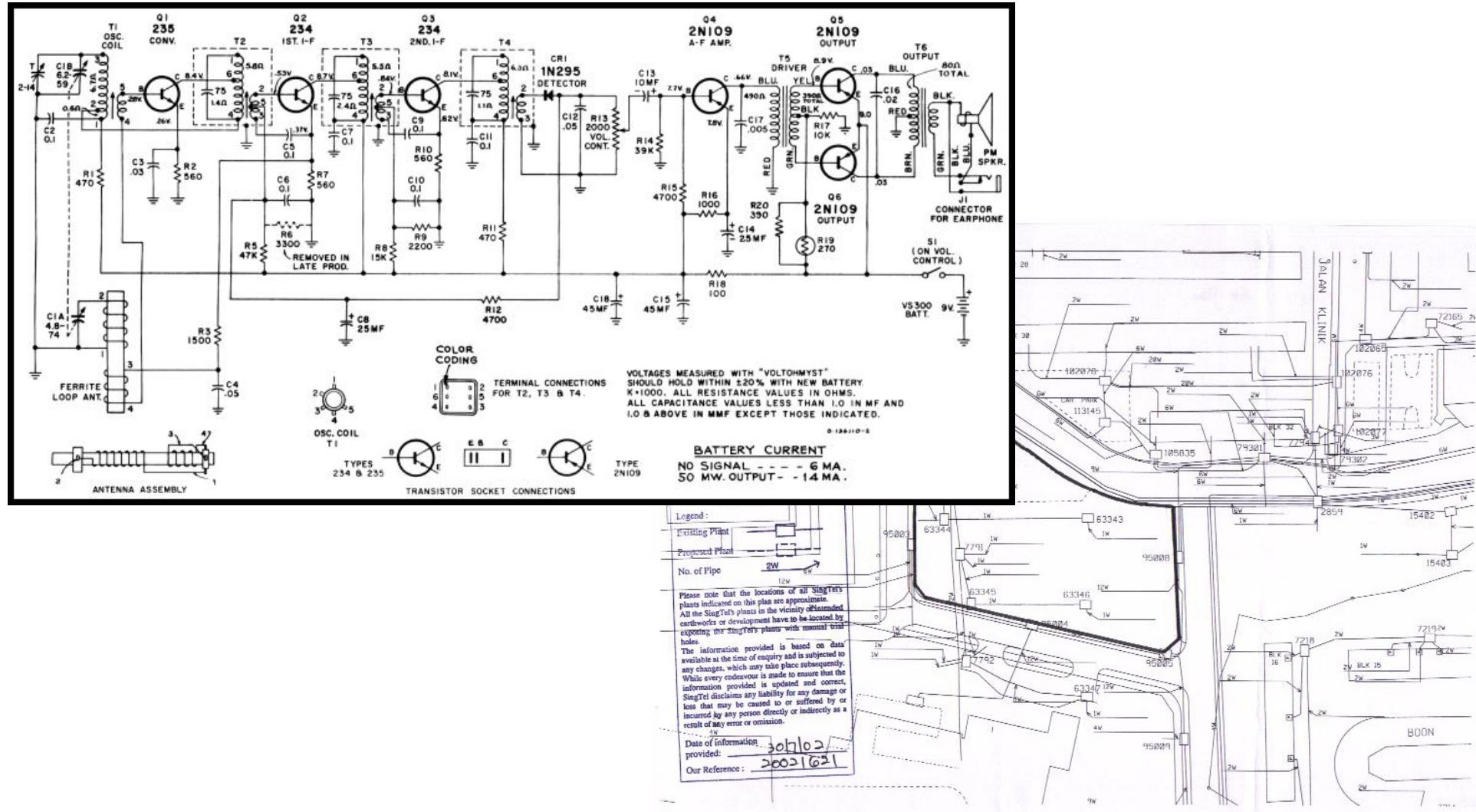
ac



source: Eir Grid, Irland

# Technical Networks – Circuit Diagram/Cable Plan

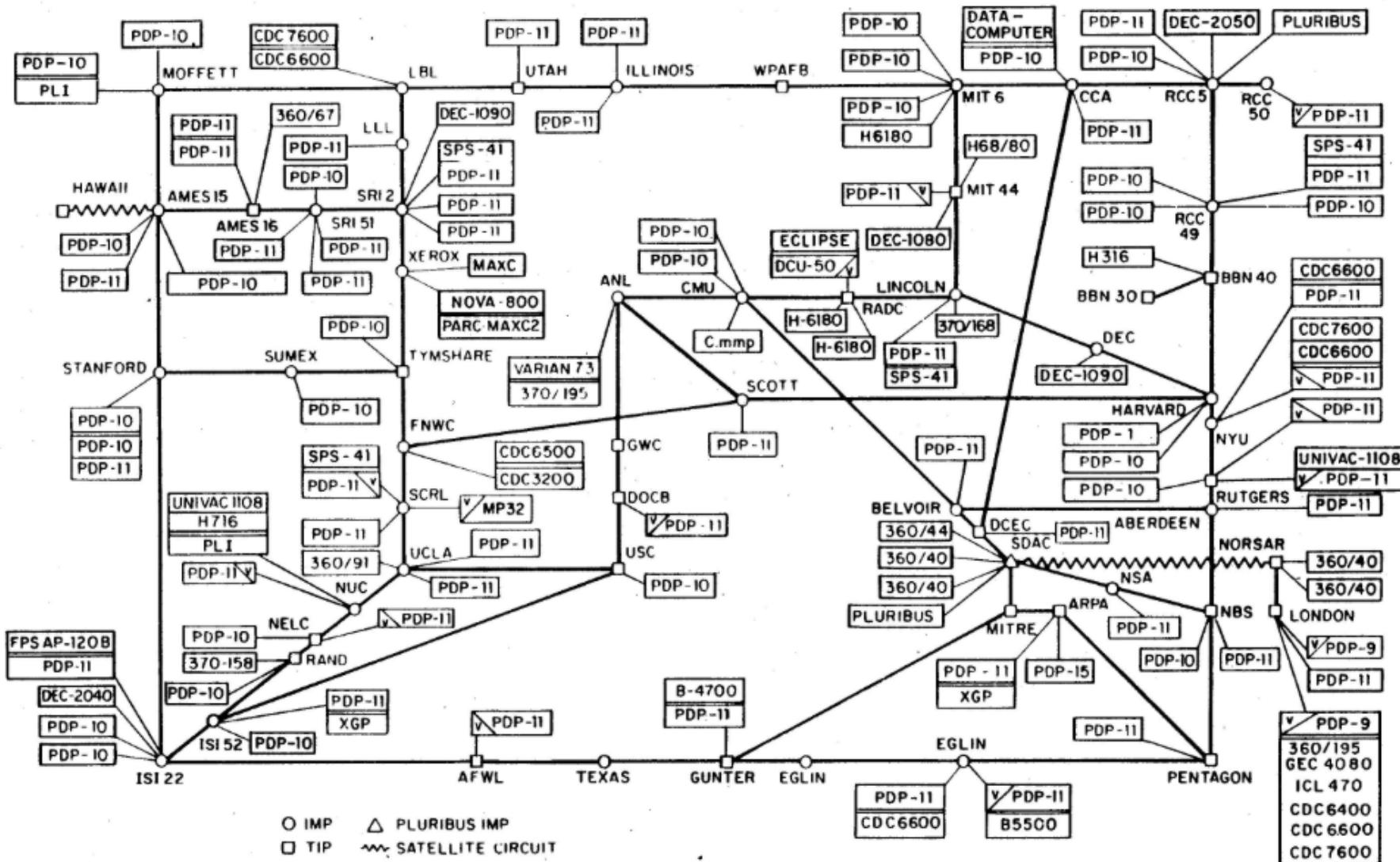
ac



# ARPANET Sketch (1977)

aci

## ARPANET LOGICAL MAP, MARCH 1977

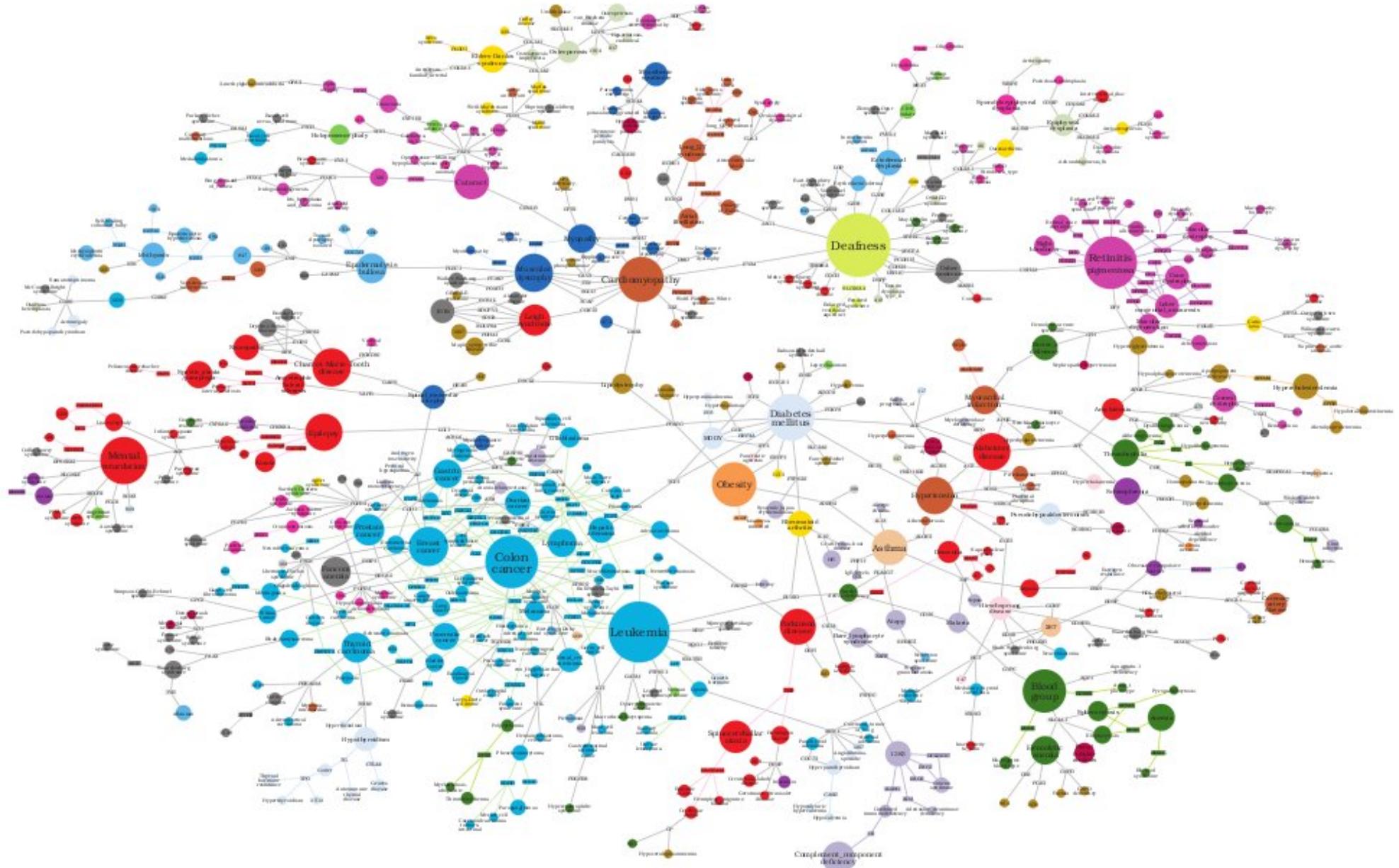


(PLEASE NOTE THAT WHILE THIS MAP SHOWS THE HOST POPULATION OF THE NETWORK ACCORDING TO THE BEST INFORMATION OBTAINABLE, NO CLAIM CAN BE MADE FOR ITS ACCURACY )

NAMES SHOWN ARE IMP NAMES, NOT (NECESSARILY) HOST NAMES

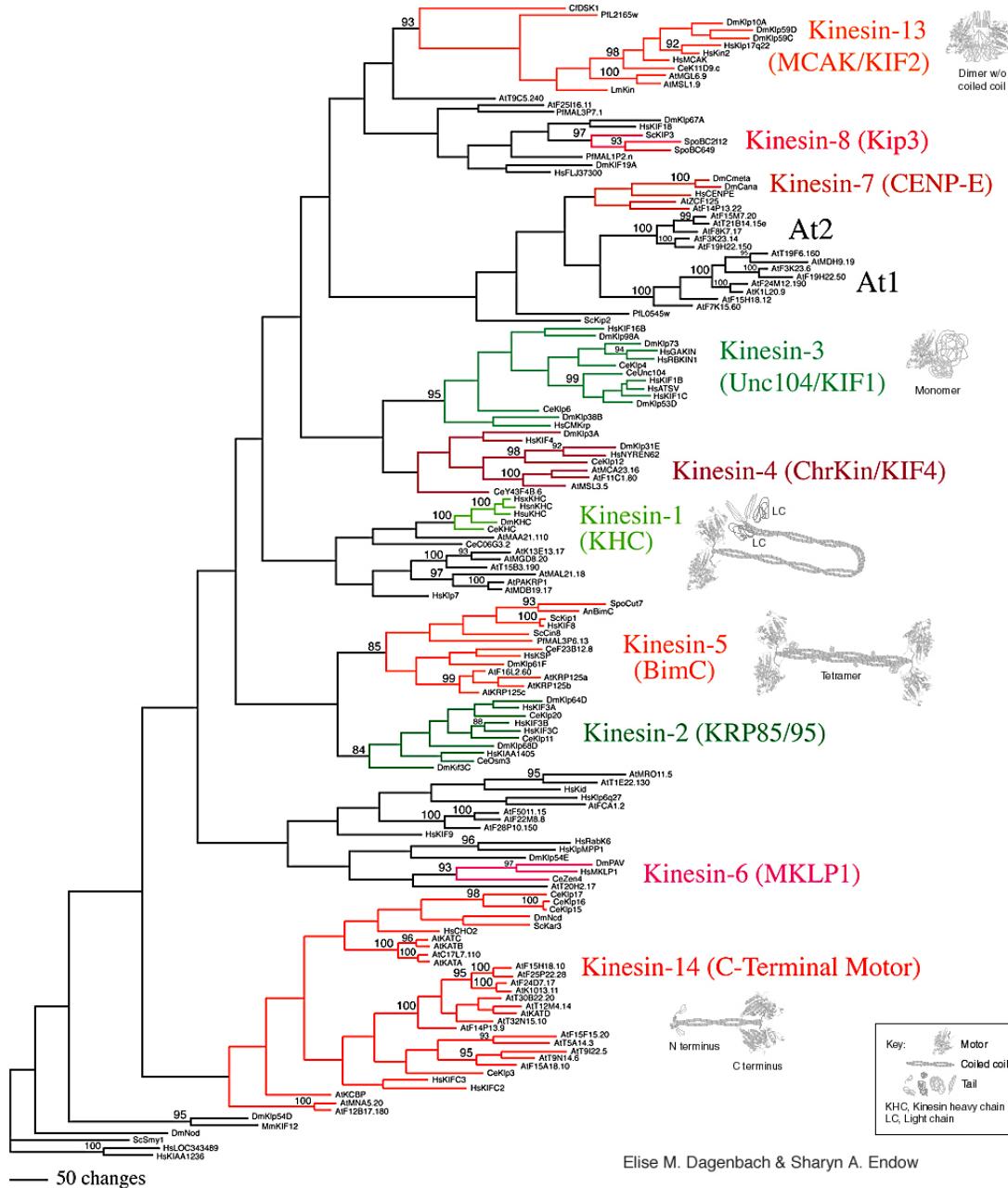
# Biomedical Network – Diseasesome

ac

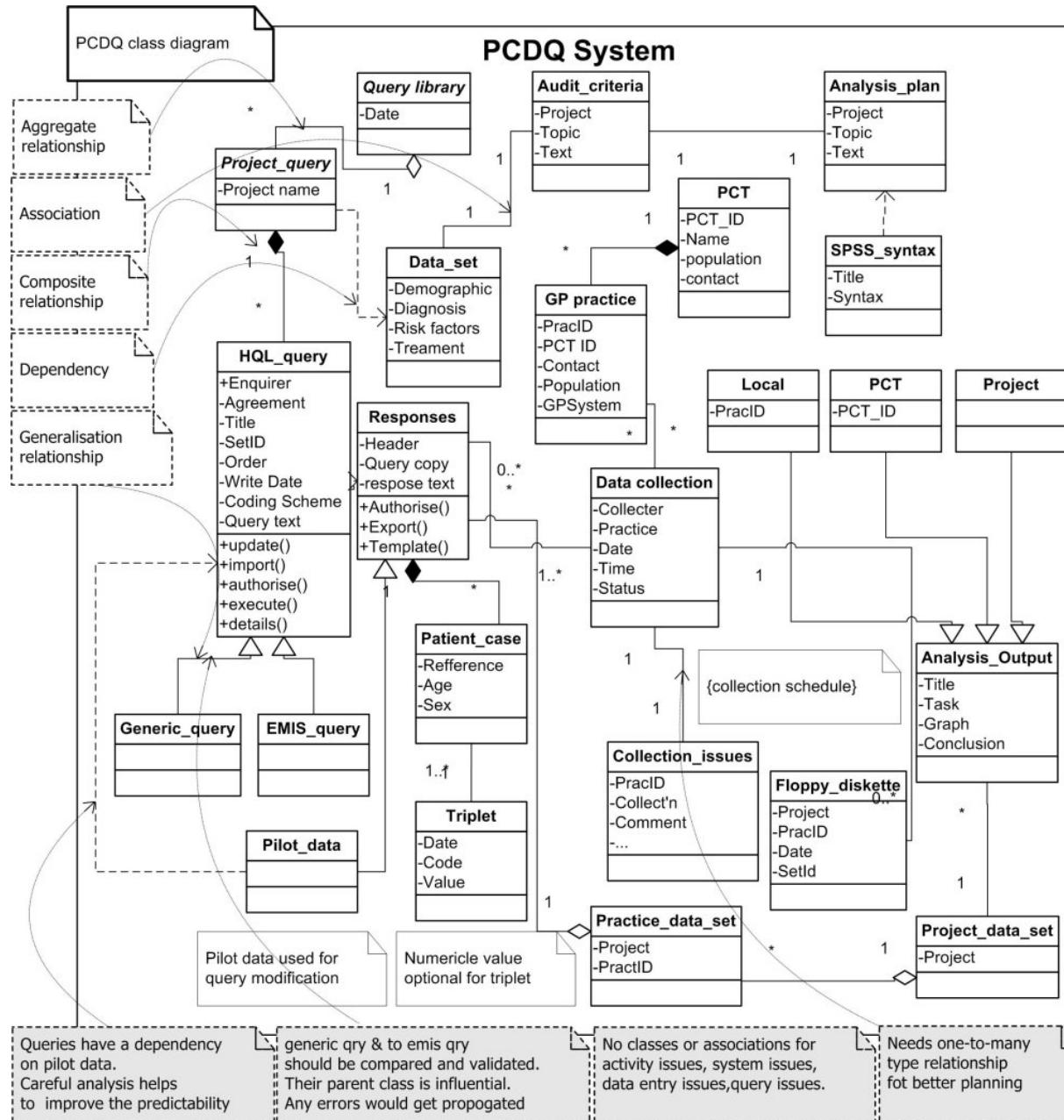


# Biology – Phylogenetic Trees

ac

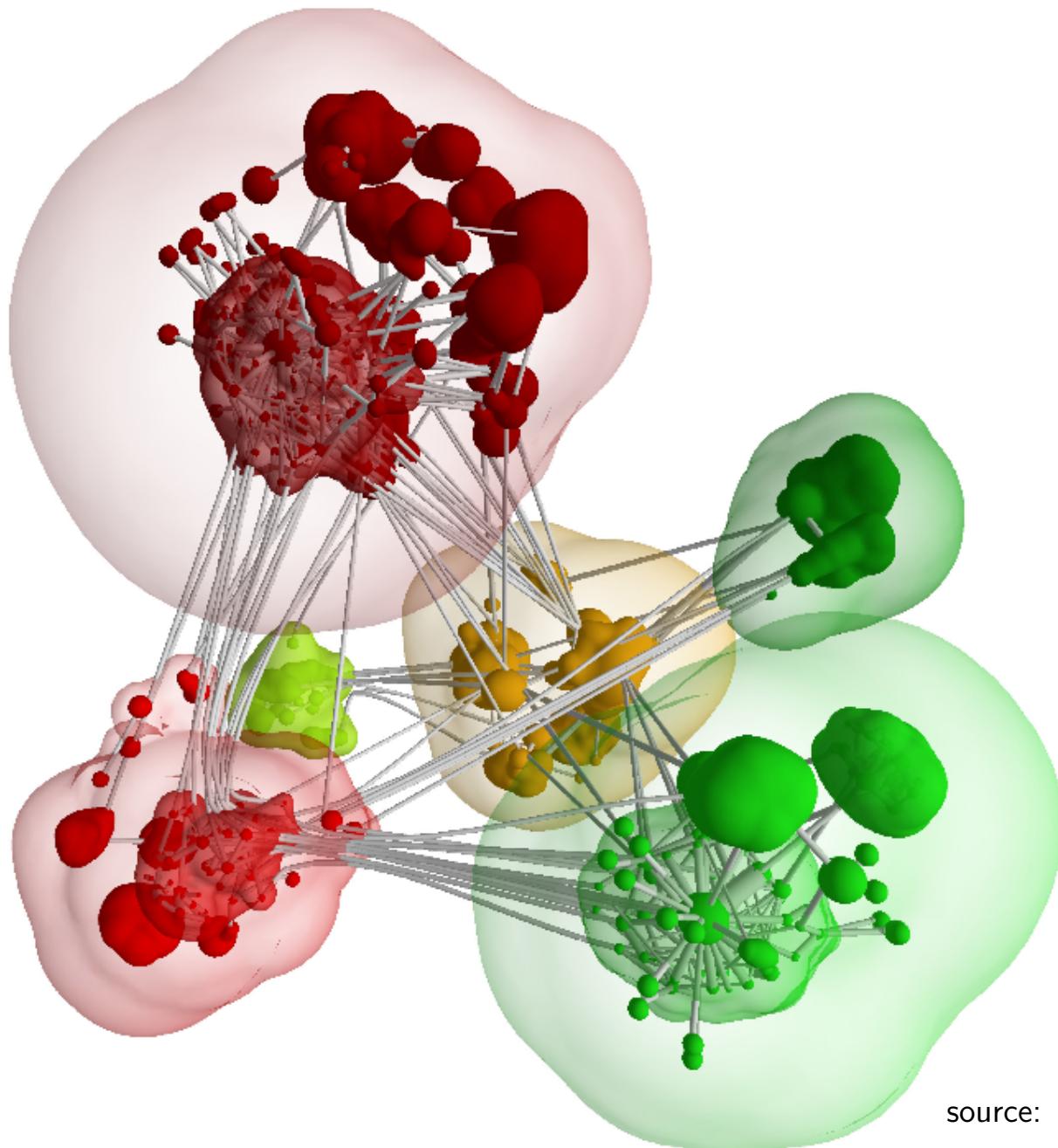


# Software Engineering – UML Diagrams



# Clustered Software Graph in 3D

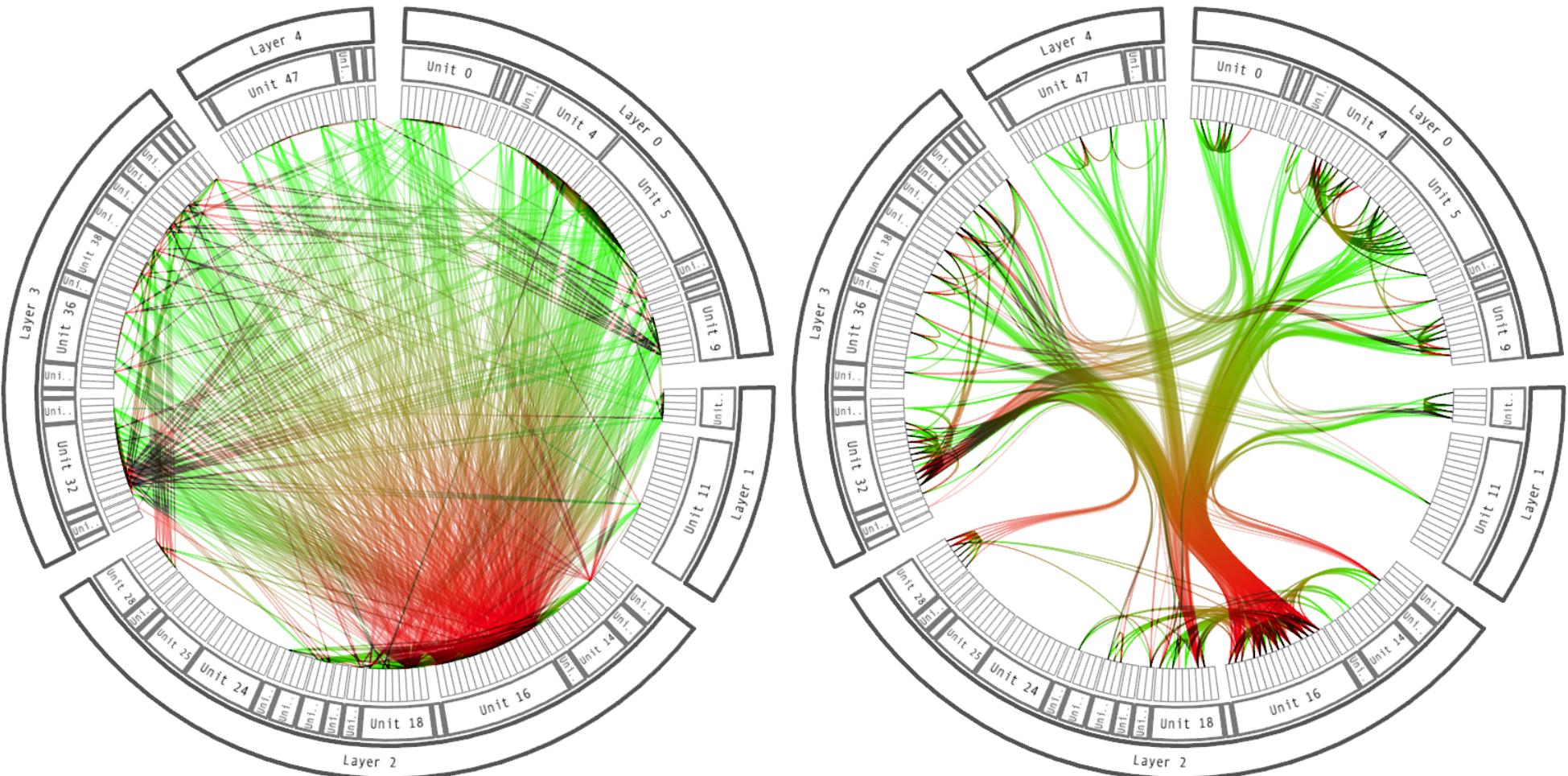
ac III



source: Balzer, Deussen, 2007

# Software Call Graph with Edge Bundling

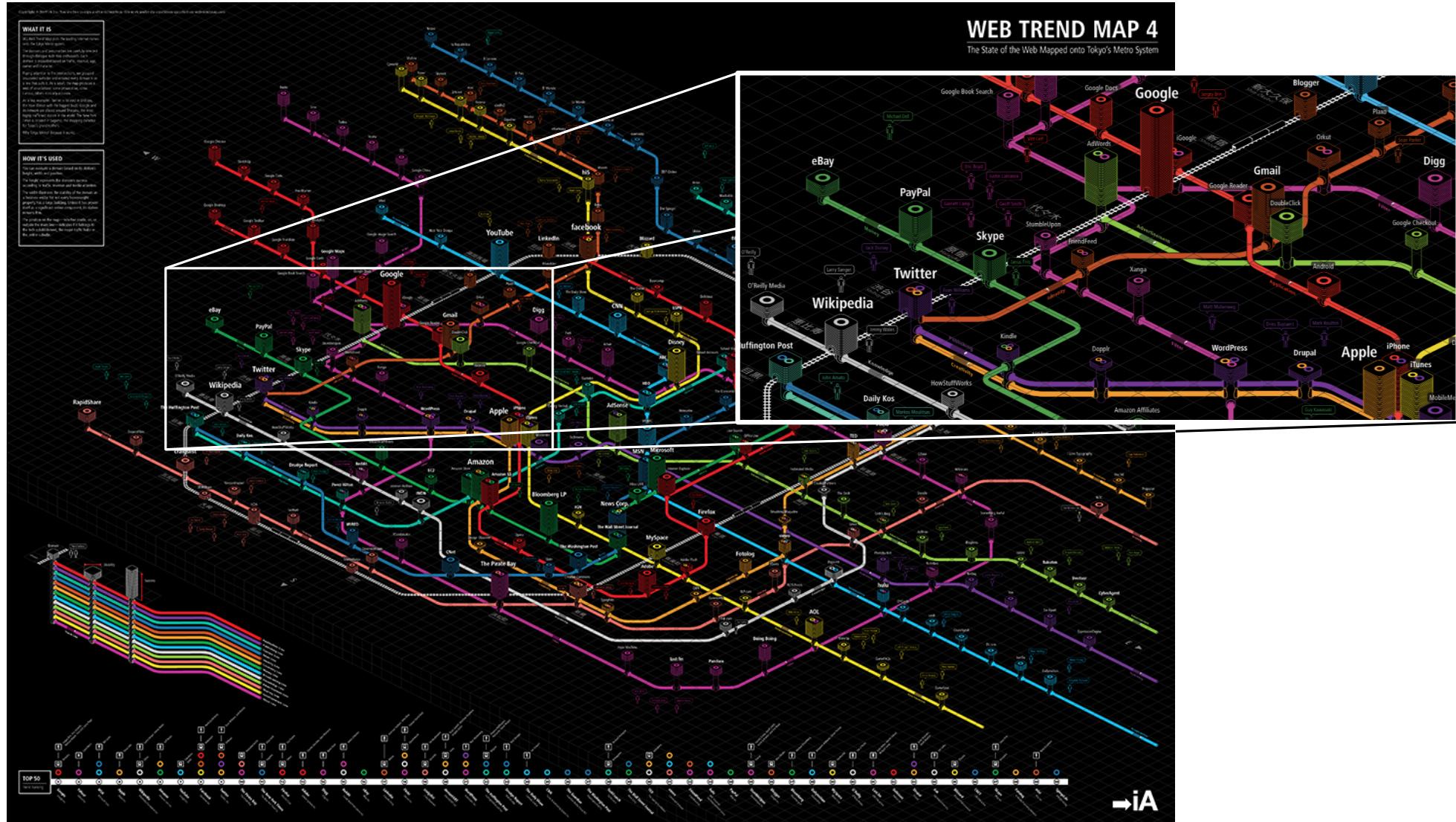
ac III



source: Danny Holten, 2011

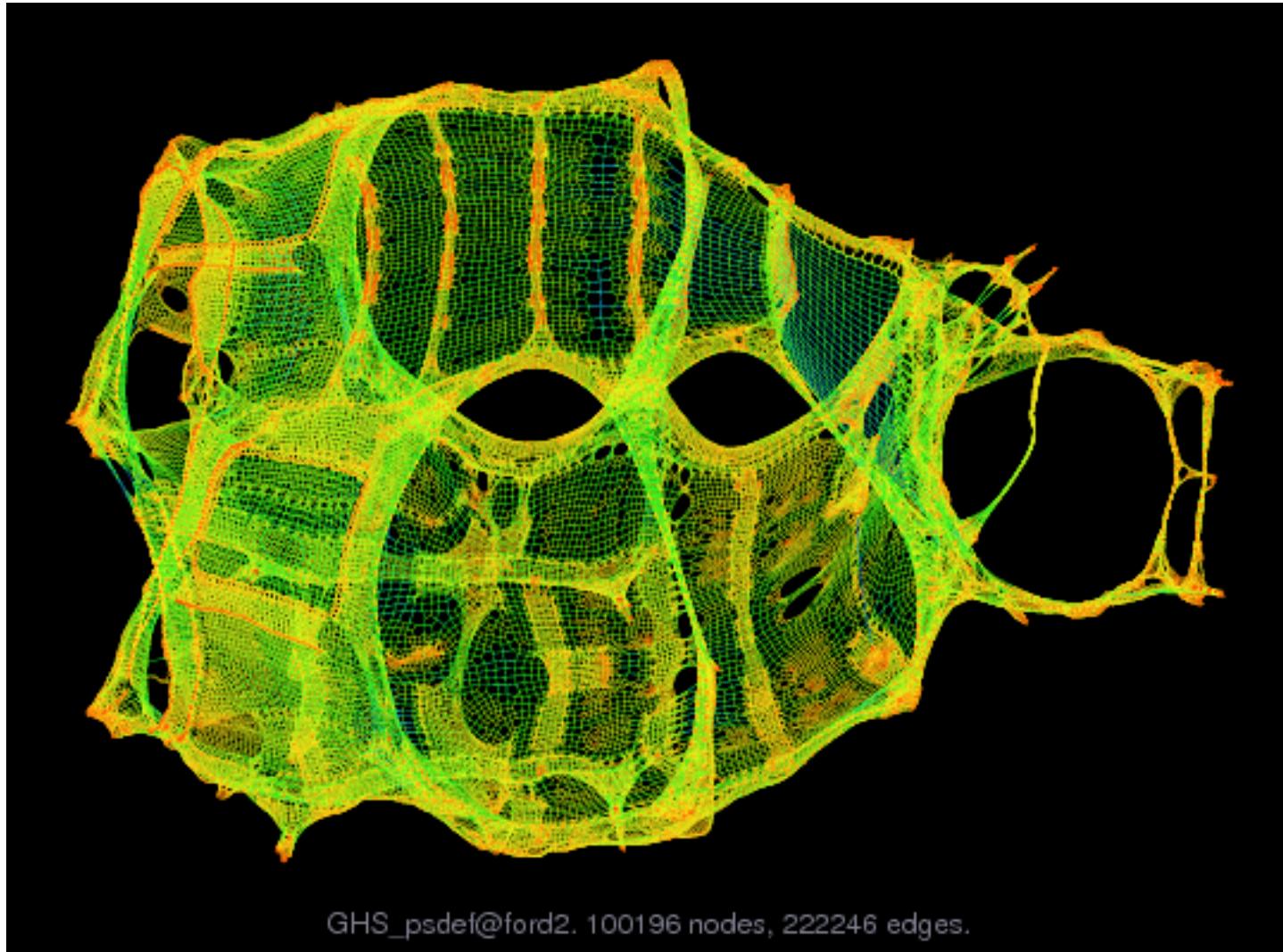
# Web Trend Map

ac III



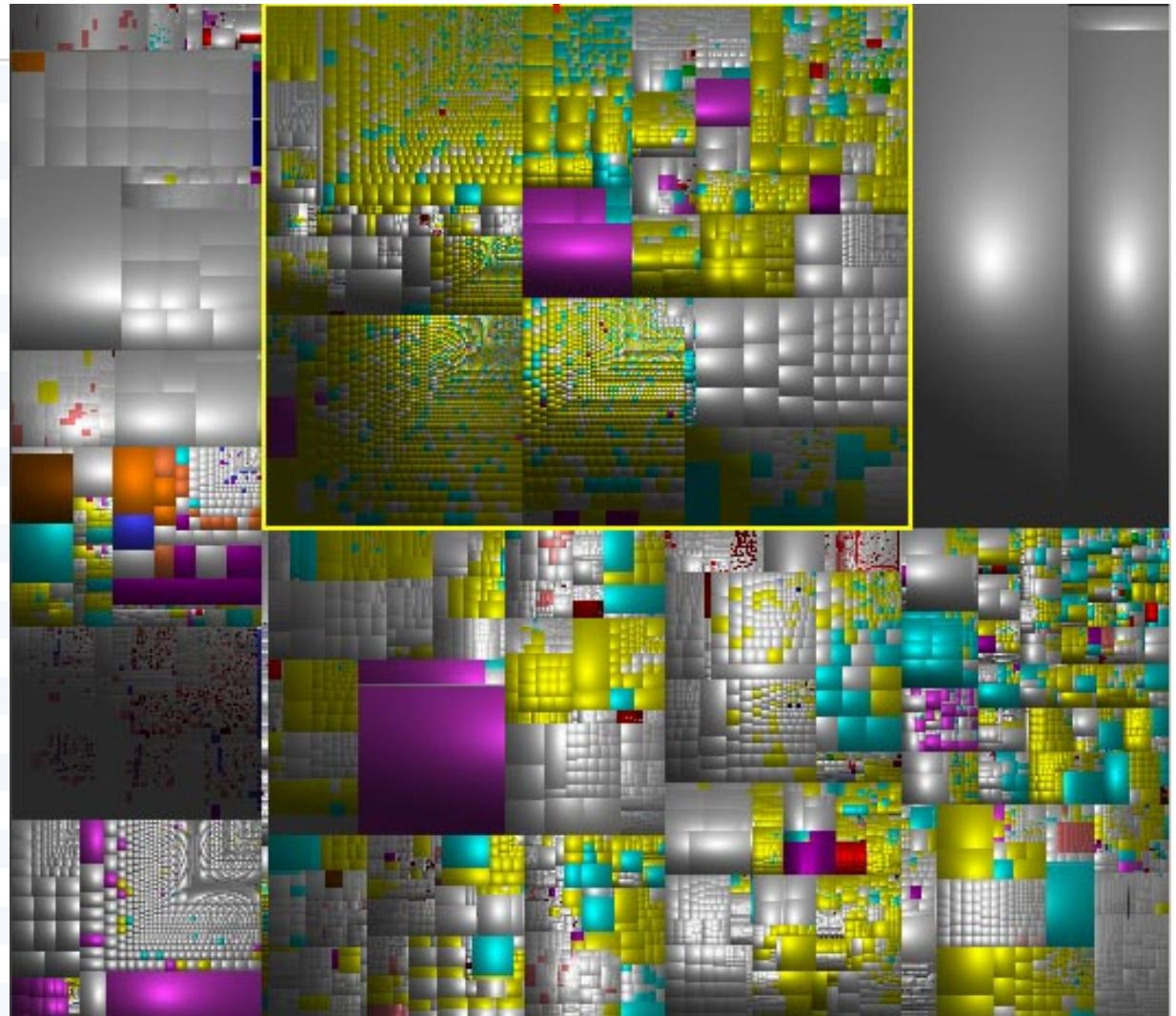
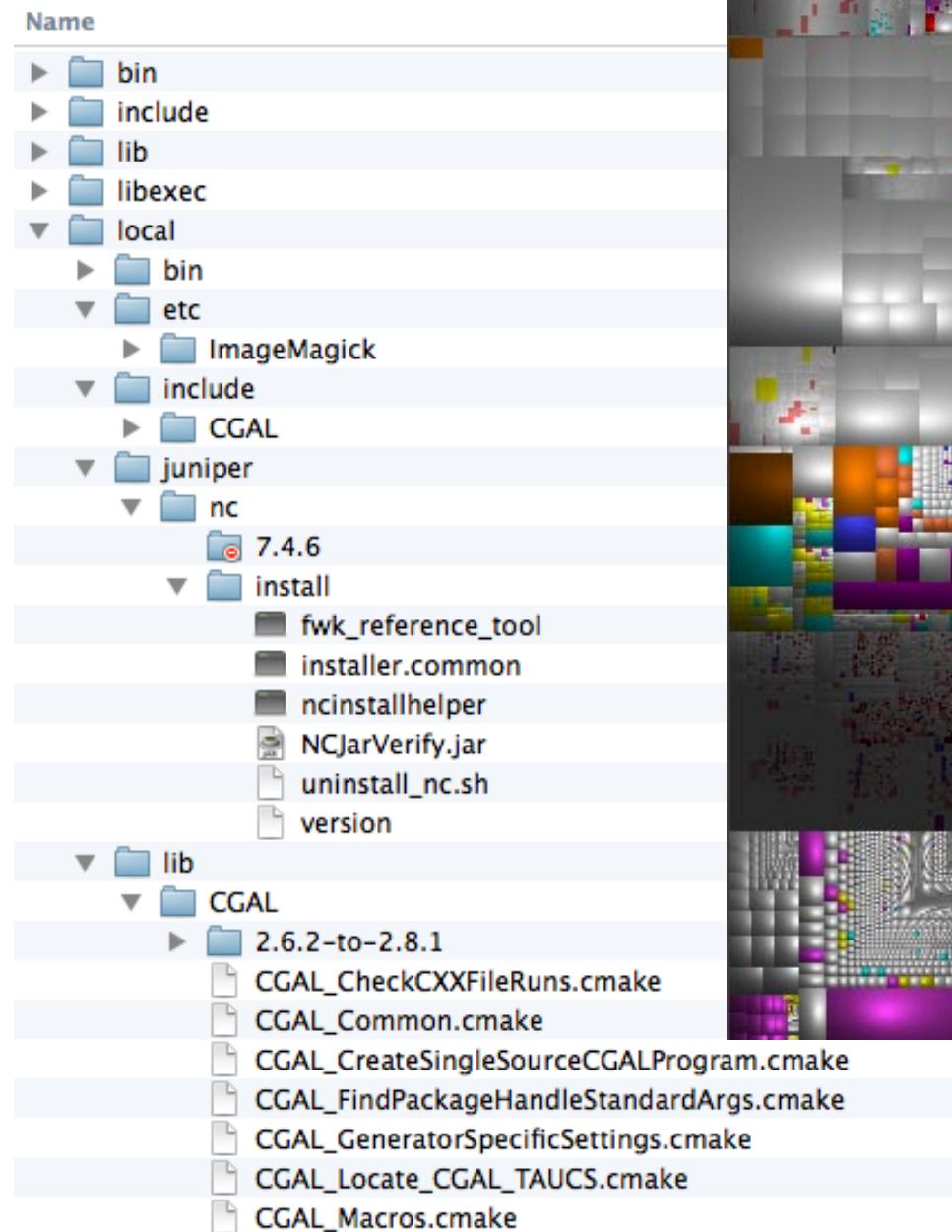
source: information Architects, 2009

# Large Graphs – Component Mesh

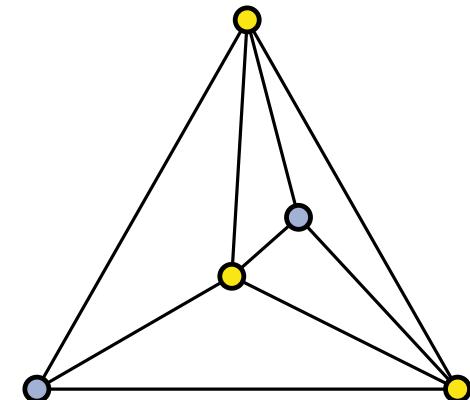
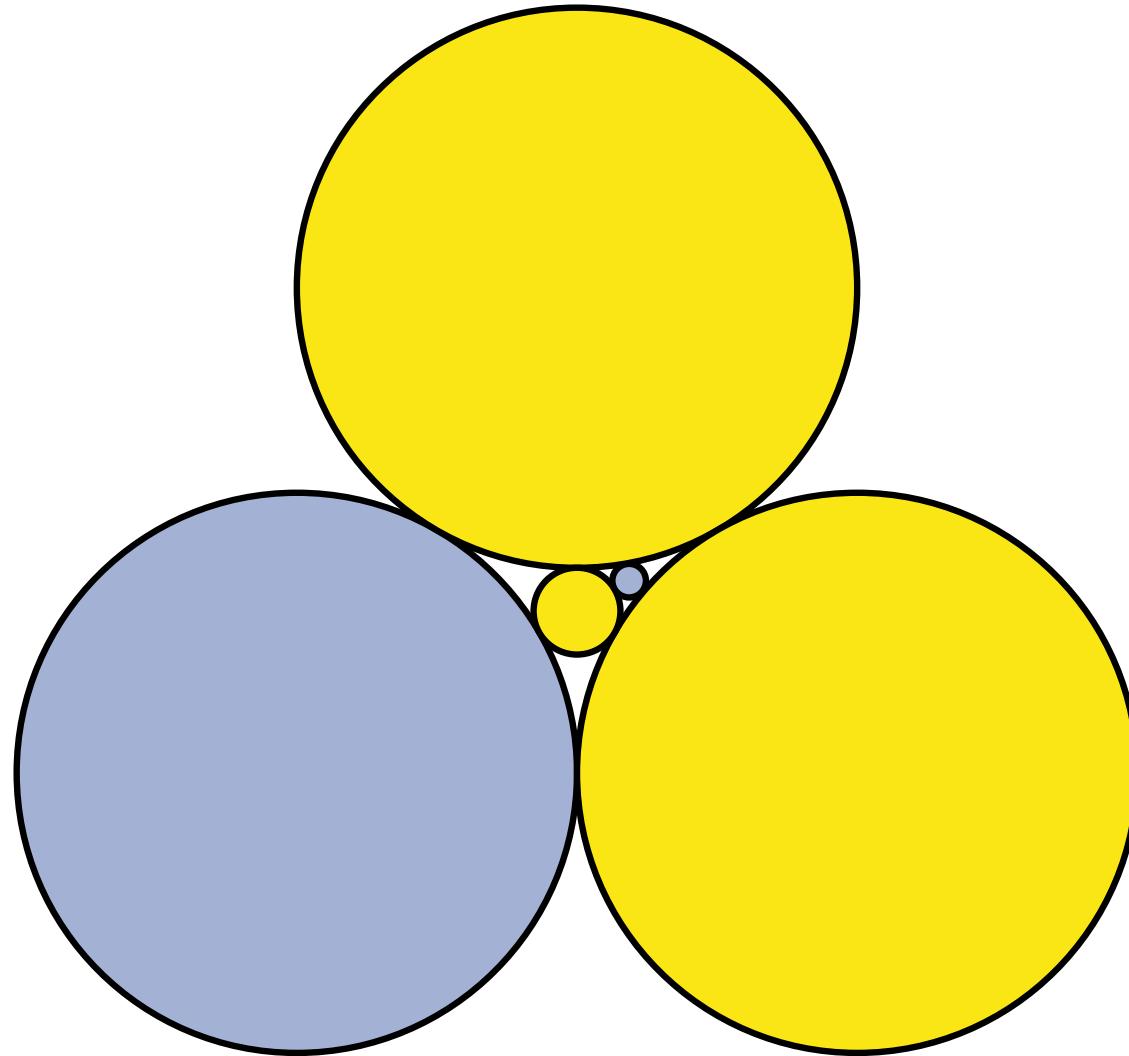


source: Yifan Hu

# File System: Explorer vs Treemap



# Alternative Representations – Contact Graph



## Graph visualization libraries

- JUNG [jung.sourceforge.net](http://jung.sourceforge.net) (Java)
- yFiles [www.yworks.com](http://www.yworks.com) (Java(FX), .NET, HTML)
- OGDF [www.ogdf.net](http://www.ogdf.net) (C++)

## Visualization tools

- visone [visone.info](http://visone.info)
- graphviz [www.graphviz.org](http://www.graphviz.org)
- yEd [www.yworks.com](http://www.yworks.com)
- Gephi [www.gephi.org](http://www.gephi.org)
- Tulip [tulip.labri.fr](http://tulip.labri.fr)
- Tom Sawyer Software [www.tomsawyer.com](http://www.tomsawyer.com)

## 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)

## 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?