

# Algorithmen zur Visualisierung von Graphen

## Lagenlayouts

INSTITUT FÜR THEORETISCHE INFORMATIK · FAKULTÄT FÜR INFORMATIK

Ignaz Rutter  
13. November 2014



# Einführung Graphenvisualisierung

Was ist ein Graph?

## Was ist ein Graph?

Tupel  $G = (V, E)$

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

Kantenmenge  $E = \{e_1, \dots, e_m\}$

Was ist ein Graph?

Tupel  $G = (V, E)$

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

Kantenmenge  $E = \{e_1, \dots, e_m\}$

Darstellungsformen?

## Was ist ein Graph?

Tupel  $G = (V, E)$

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

Kantenmenge  $E = \{e_1, \dots, e_m\}$

## Darstellungsformen?

**Mengenschreibweise**

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

## Was ist ein Graph?

Tupel  $G = (V, E)$

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

Kantenmenge  $E = \{e_1, \dots, e_m\}$

## Darstellungsformen?

Mengenschreibweise

Adjazenzliste

```
v1 : v2, v8
v2 : v1, v3
v3 : v2, v5, v9, v10
v4 : v5, v6, v9
v5 : v3, v4, v8
v6 : v4, v8, v9
v7 : v8, v9
v8 : v1, v5, v6, v7, v9, v10
v9 : v3, v4, v6, v7, v8, v10
v10 : v3, v8, v9
```

## Was ist ein Graph?

Tupel  $G = (V, E)$

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

Kantenmenge  $E = \{e_1, \dots, e_m\}$

## Darstellungsformen?

Mengenschreibweise

Adjazenzliste

Adjazenzmatrix

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

## Was ist ein Graph?

Tupel  $G = (V, E)$

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

Kantenmenge  $E = \{e_1, \dots, e_m\}$

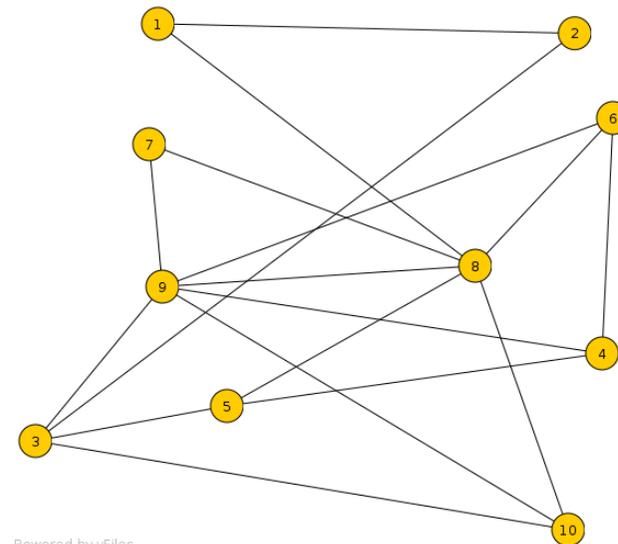
## Darstellungsformen?

Mengenschreibweise

Adjazenzliste

Adjazenzmatrix

**Zeichnung**



## Was ist ein Graph?

Tupel  $G = (V, E)$

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

Kantenmenge  $E = \{e_1, \dots, e_m\}$

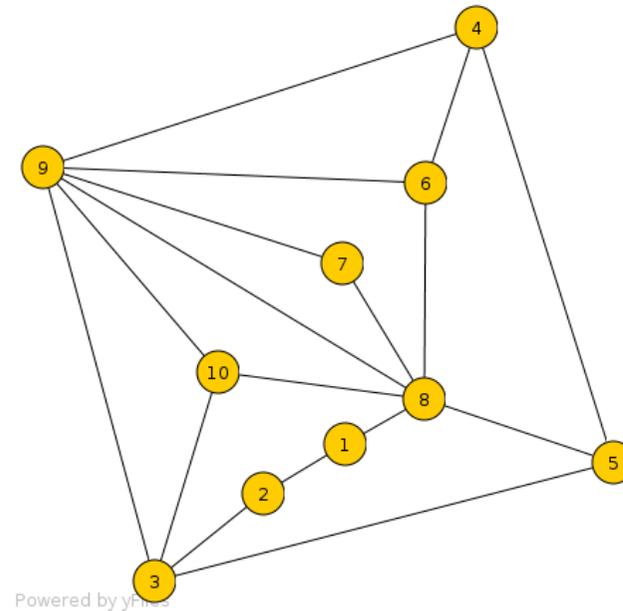
## Darstellungsformen?

Mengenschreibweise

Adjazenzliste

Adjazenzmatrix

**Zeichnung**

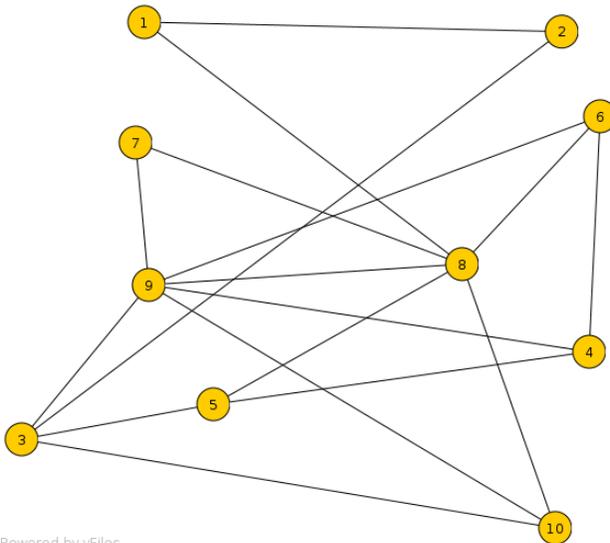


# Graphen und ihre Darstellung

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

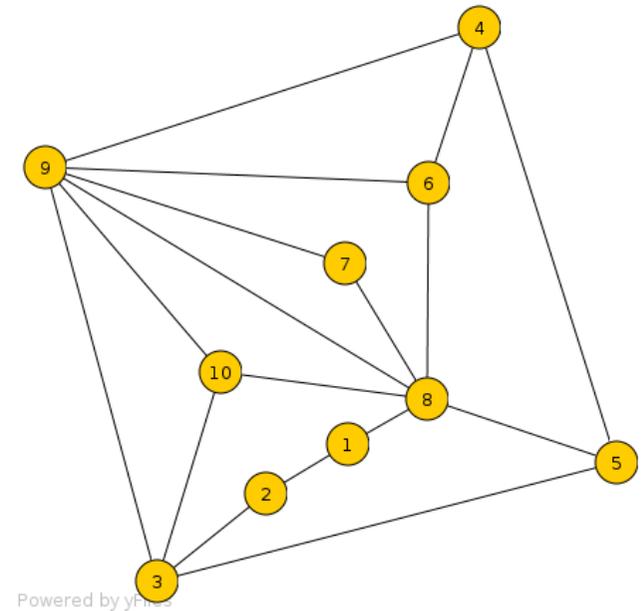
$$E = \{\{v_1, v_2\}, \{v_1, v_8\}, \{v_2, v_3\}, \{v_3, v_5\}, \{v_3, v_9\}, \{v_3, v_{10}\}, \{v_4, v_5\}, \{v_4, v_6\}, \{v_4, v_9\}, \{v_5, v_8\}, \{v_6, v_8\}, \{v_6, v_9\}, \{v_7, v_8\}, \{v_7, v_9\}, \{v_8, v_{10}\}, \{v_9, v_{10}\}\}$$

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



Powered by yFiles

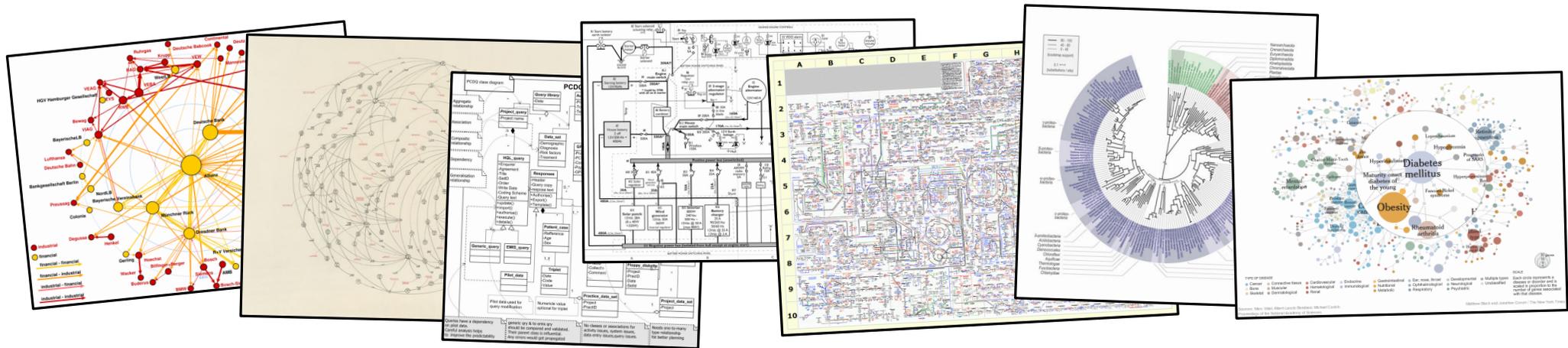
$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0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



Powered by yFiles

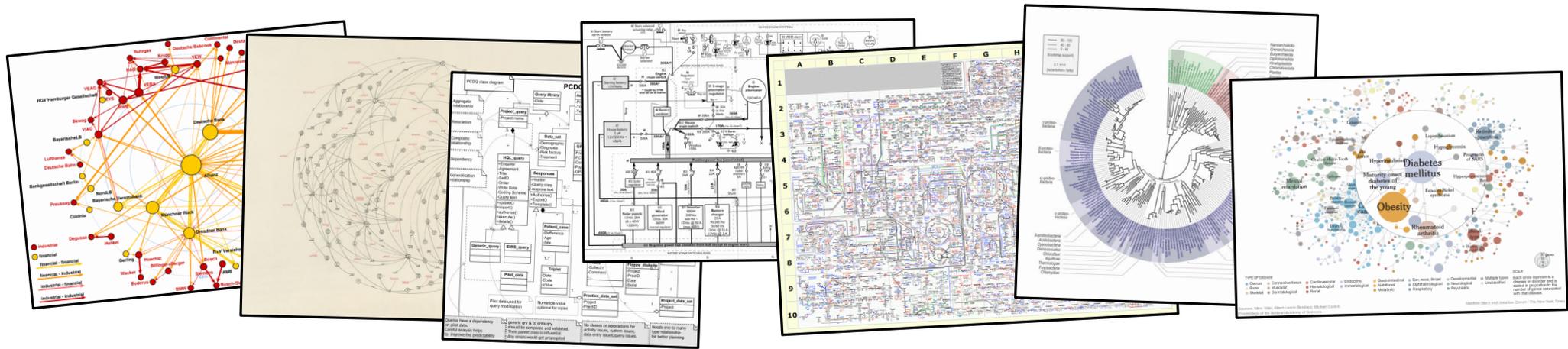
# Wozu Graphen zeichnen?

Graphen sind mathematische Modelle realer physischer und abstrakter Netzwerke (soziale Netze, metabolische Netze, VLSI-Layout, UML Diagramme, Infrastrukturnetze, ...)



# Wozu Graphen zeichnen?

Graphen sind mathematische Modelle realer physischer und abstrakter Netzwerke (soziale Netze, metabolische Netze, VLSI-Layout, UML Diagramme, Infrastrukturnetze, ...)

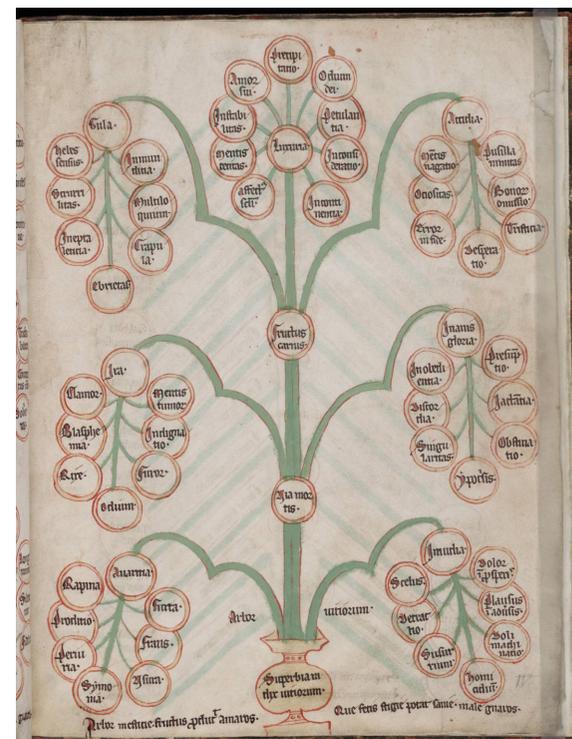
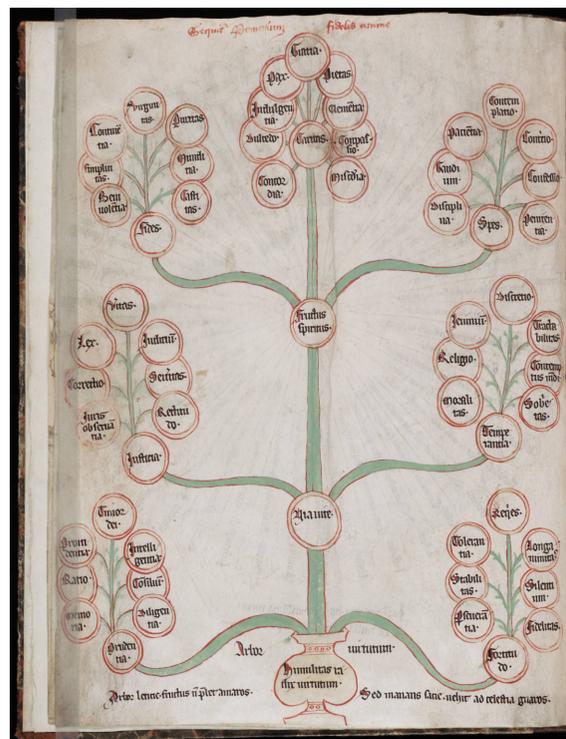


- **Menschen denken visuell** – ohne gute Visualisierung sind komplexe Graphen für uns unverständlich
- Visualisierungen helfen bei der **Kommunikation** und der **Exploration** von Graphen/Netzwerken
- wir brauchen **Algorithmen** zum Zeichnen von Graphen um Netzwerke dem Menschen zugänglich zu machen

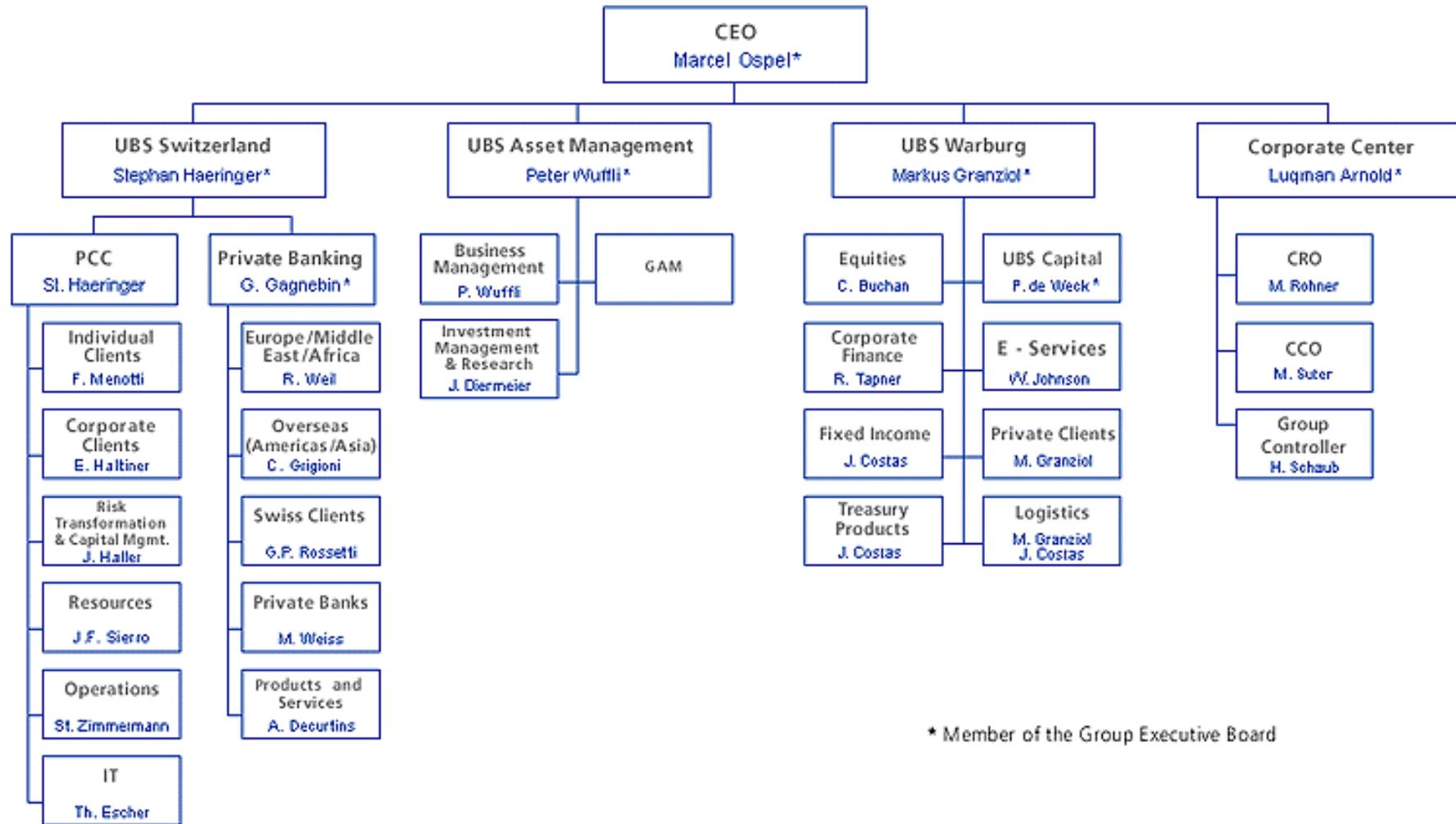
## Beispiele

eine kleine Diaschau

# Tugenden und Sünden – Mittelalter

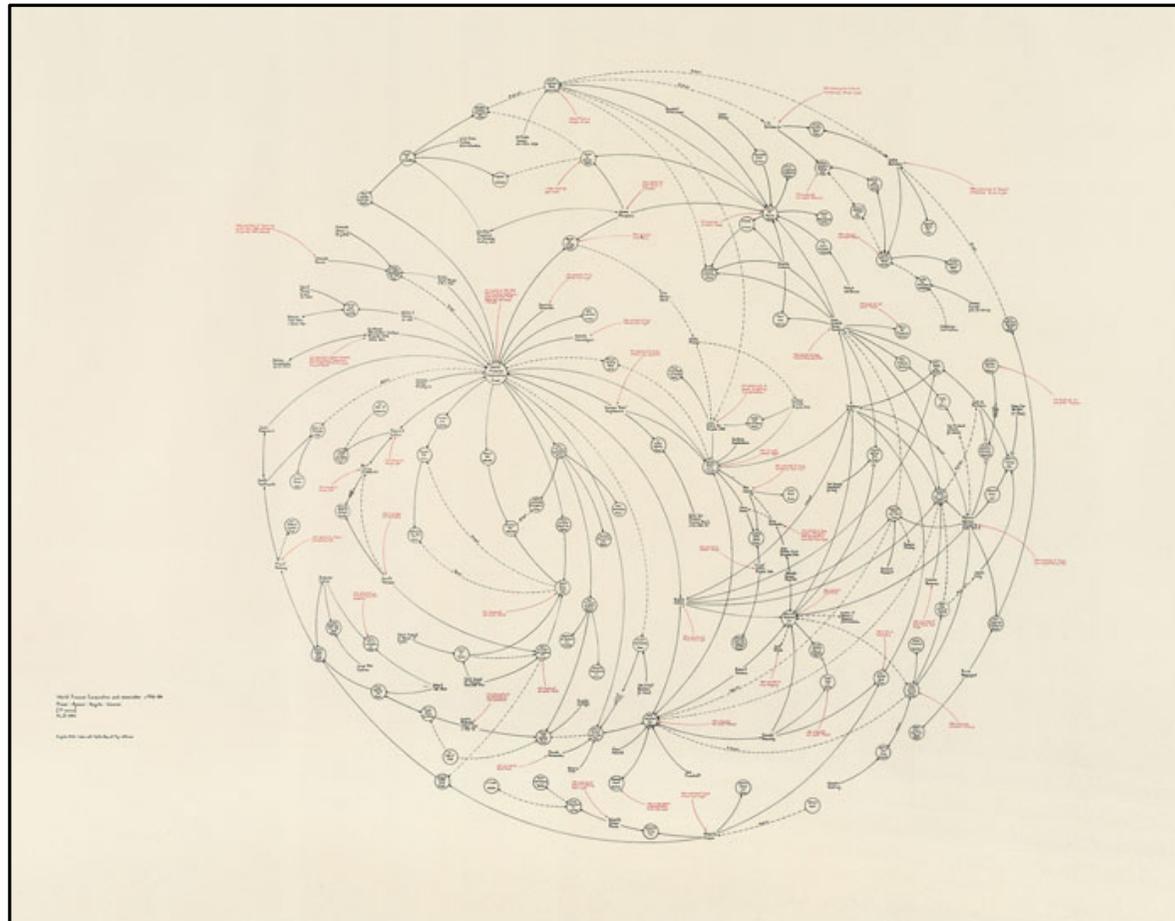


# Soziale Netze – Organigramm UBS



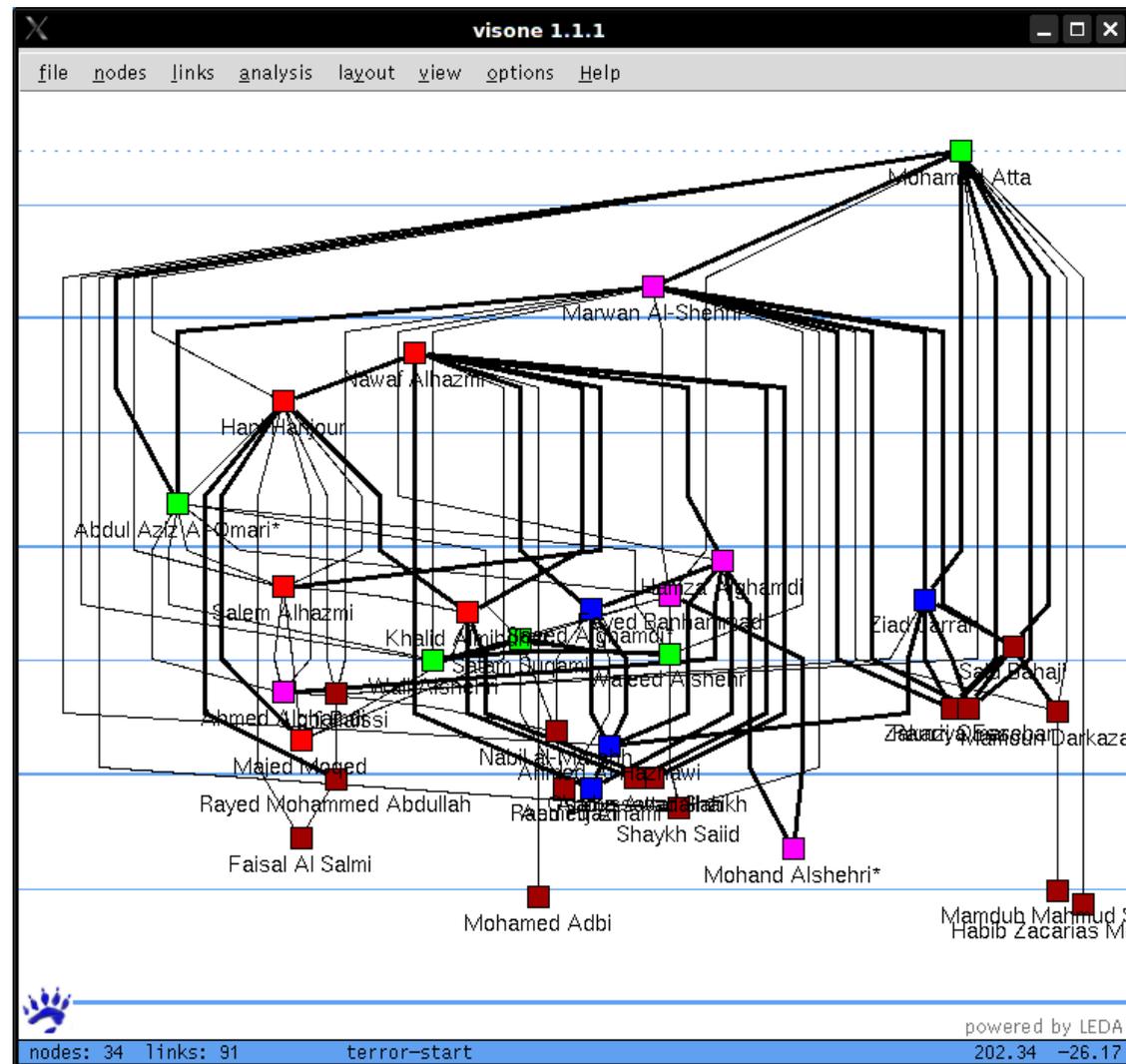
\* Member of the Group Executive Board

# Soziale Netze – Welt-Finanzsystem

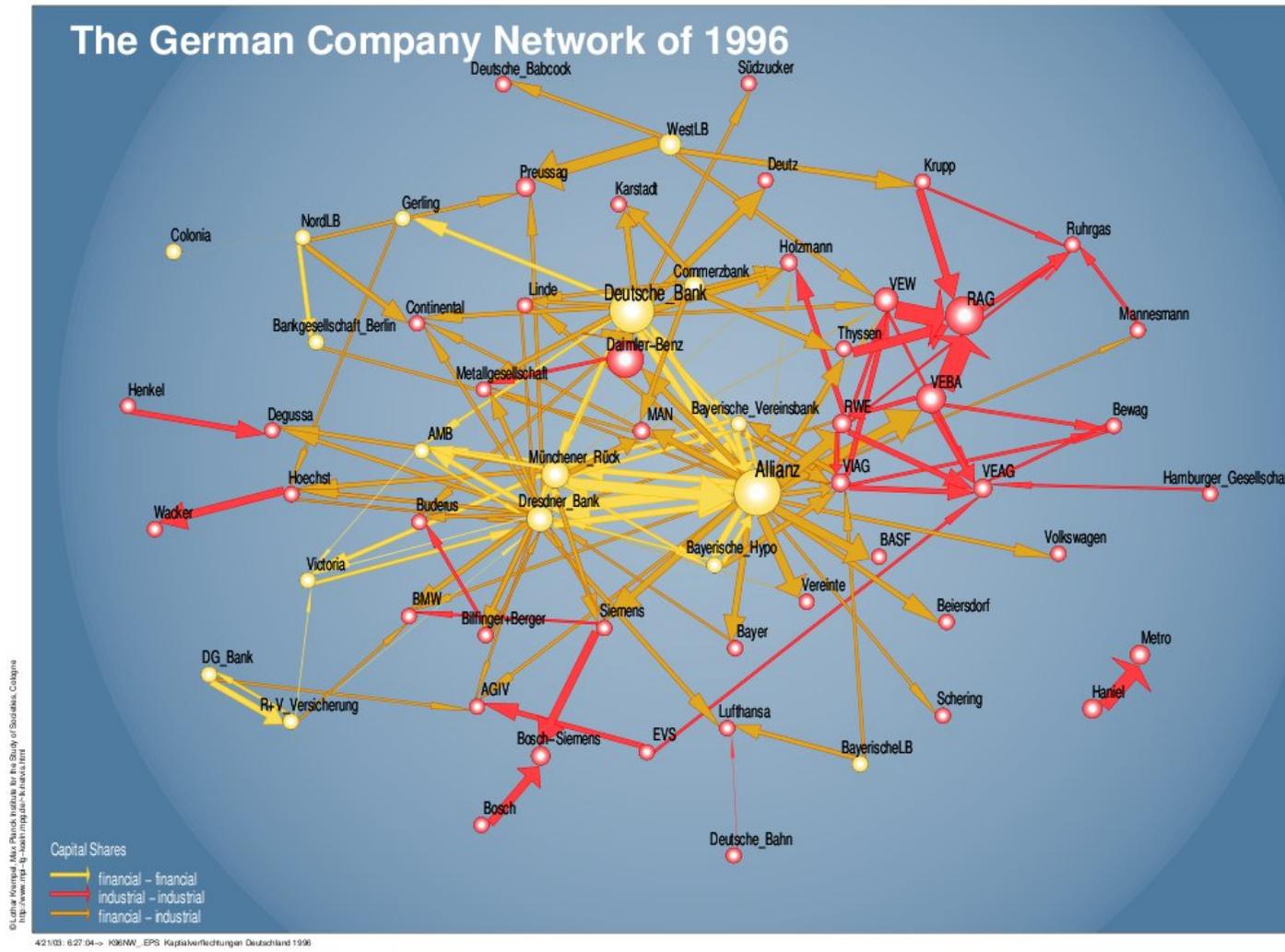


World Finance Corporation  
© Mark Lombardi

# Soziale Netze – Terrorzelle



# Soziale Netze – Firmenbeteiligungen



# Soziale Netze – Staatsfonds

## FOLLOW THE MONEY

### The New Global Wealth Machine

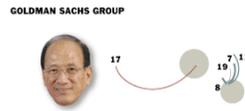
Sovereign wealth funds have emerged in recent months as the world's power brokers. They have used their tremendous wealth to make big cross-border investments and prop up some of Wall Street's best-known firms. The increased activity comes as other kinds of acquirers have been sidelined by the credit 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 in 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.



**Michael Klein, Chairman, institutional clients group**  
One of the firm's highest-profile investment bankers, he advised Cayle in its stake sale to Mitsubishi, as well as Citigroup in both of its deals with sovereign wealth funds.



**Richard Ong, Former managing director**  
Mr. Ong left Goldman early this year after the Chinese government refused to allow the firm to promote him to run its Beijing office. Mr. Ong's brother, Charles, was the chief investment officer of Temasek Holdings until 2006.



**Gary Parr, Deputy chairman**  
In addition to becoming the key adviser on many of the biggest sovereign wealth deals, Mr. Parr helped advise Bear Stearns on its distressed sale to JP Morgan Chase.



**Kate Richdale, Managing director**  
The head of Morgan Stanley's Asian general industries group, based in Hong Kong. She previously held a senior position in the investment bank's Southeast Asia group.

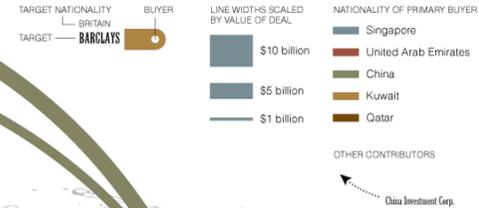
#### The Targets



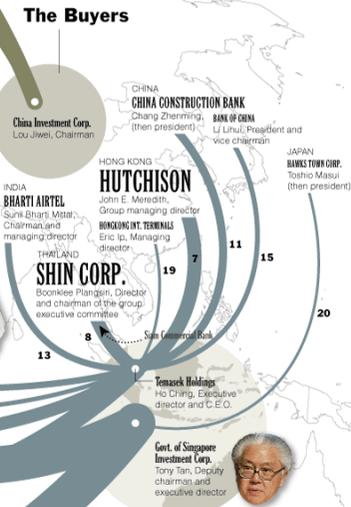
**BRITAIN**  
**J. SAINSBURY** (Justin King, Chief executive)  
**STANDARD CHARTERED BANK** (Peter Sands, Chief executive)

**SWITZERLAND**  
**UBS** (Marcel Rohrab, Chief executive)

### The 20 Biggest Cross-Border Sovereign Wealth Fund Deals Since 2005

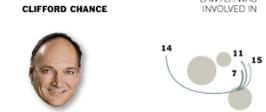


#### The Buyers

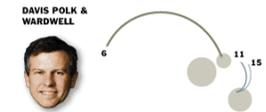


#### The Lawyers

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



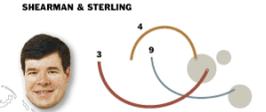
**James Baird, Partner and global head of private equity**  
Mr. Baird's firm, based in London, was one of the early firms to make a bet on Asia by staffing up there before some of the traditional white-shoe Wall Street firms ventured there.



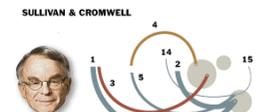
**Randall D. Guynn, Partner**  
As head of the firm's financial institutions group, he has advised on many international deals in Europe and Asia. He also worked on the team that advised Morgan Stanley in its \$5.5 billion stake sale to China's sovereign wealth fund.



**Richard Good, Partner**  
Based in Singapore, Mr. Good is the firm's man-on-the-ground in Asia. He has worked for Linklaters in Asia since 2000.



**Stephen M. Besen, Partner**  
A longtime hand in the Middle East, Mr. Besen's deep relationships have helped his firm carve out one of the strongest niches in the region.

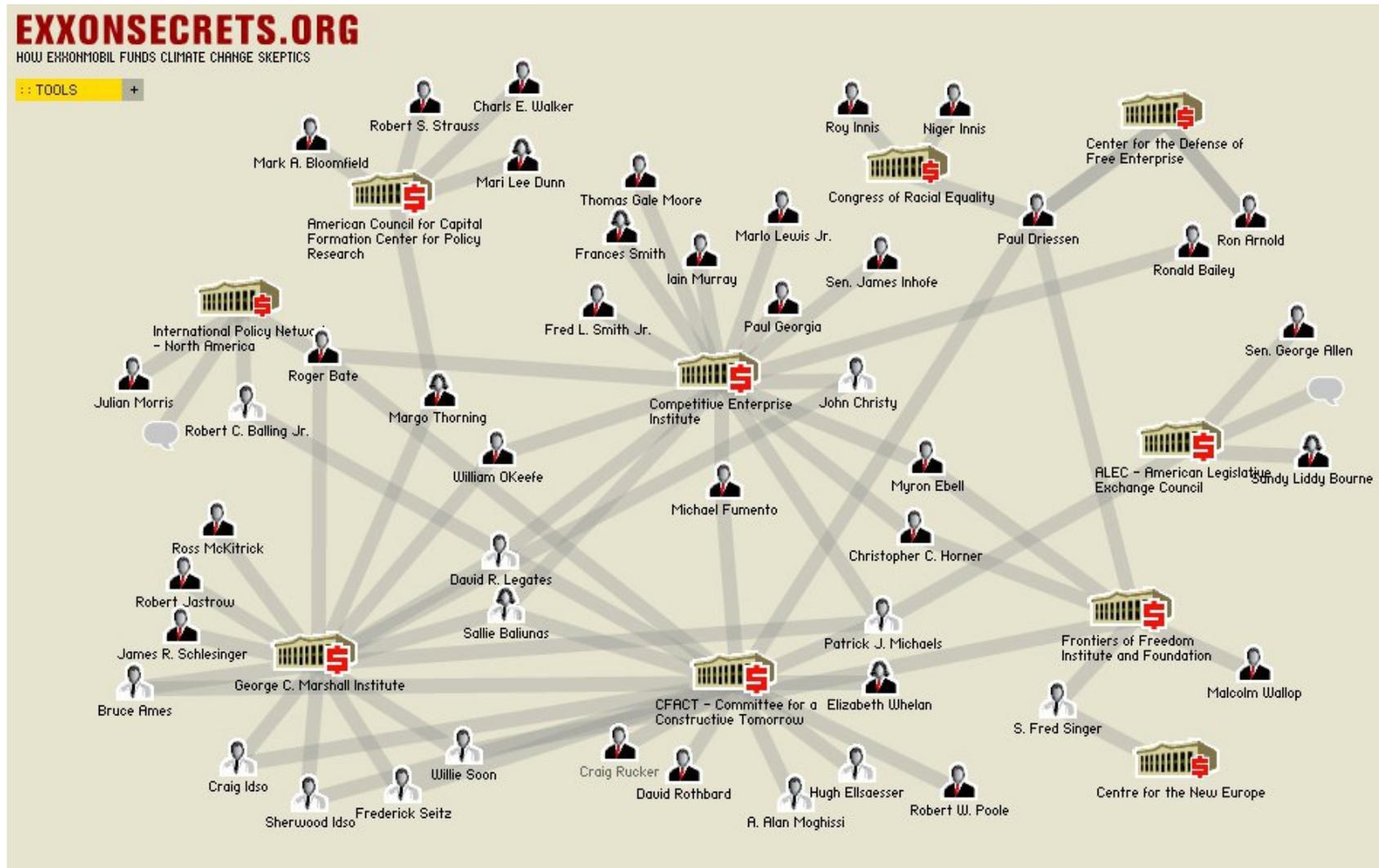


**H. Rodgin Cohen, Chairman**  
The world's go-to lawyer for sovereign wealth investments in financial services firms. He worked on twice as many sovereign wealth related deals than any other individual.

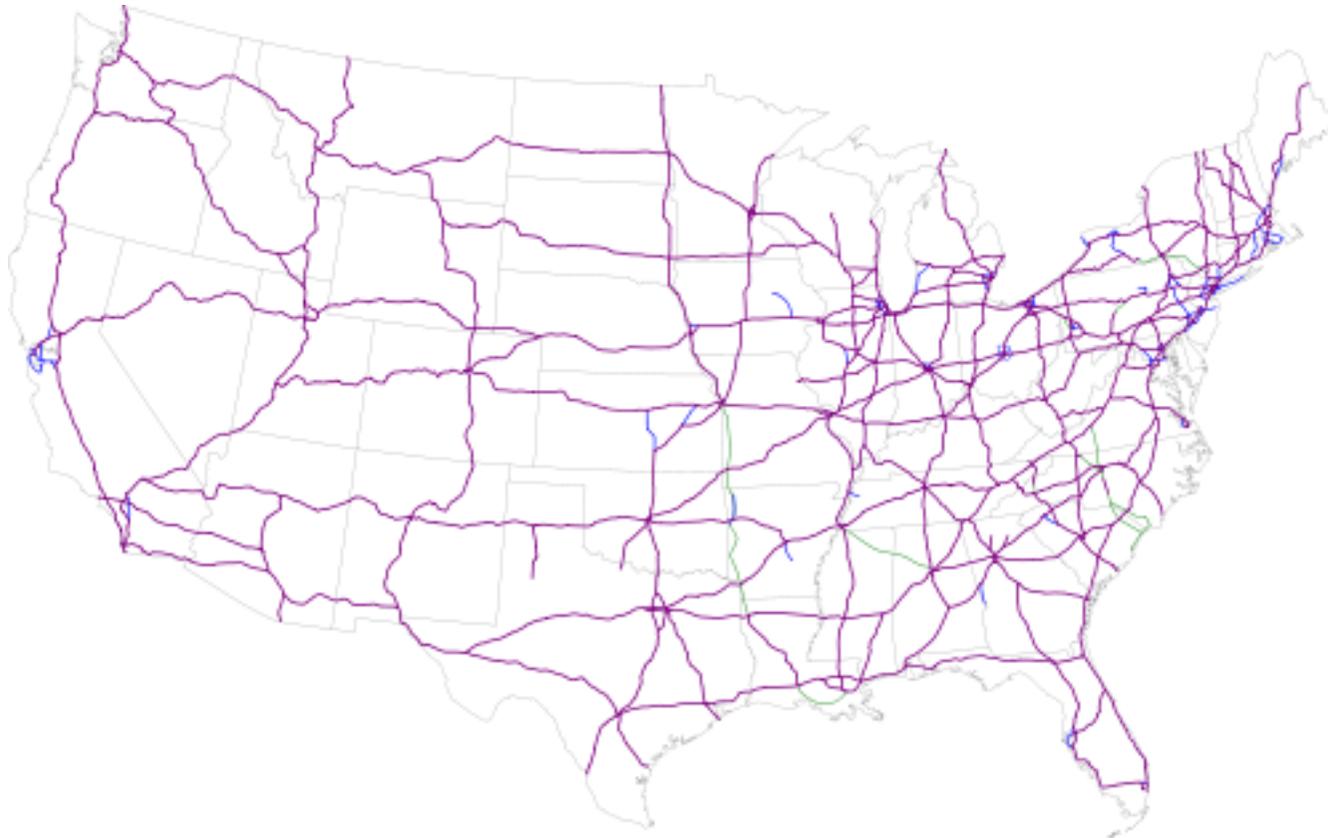
Source: Dealogic, the companies

RESEARCH BY MICHAEL DE LA MERCEZ, GRAPHIC BY GILBERT GATES FOR THE NEW YORK TIMES

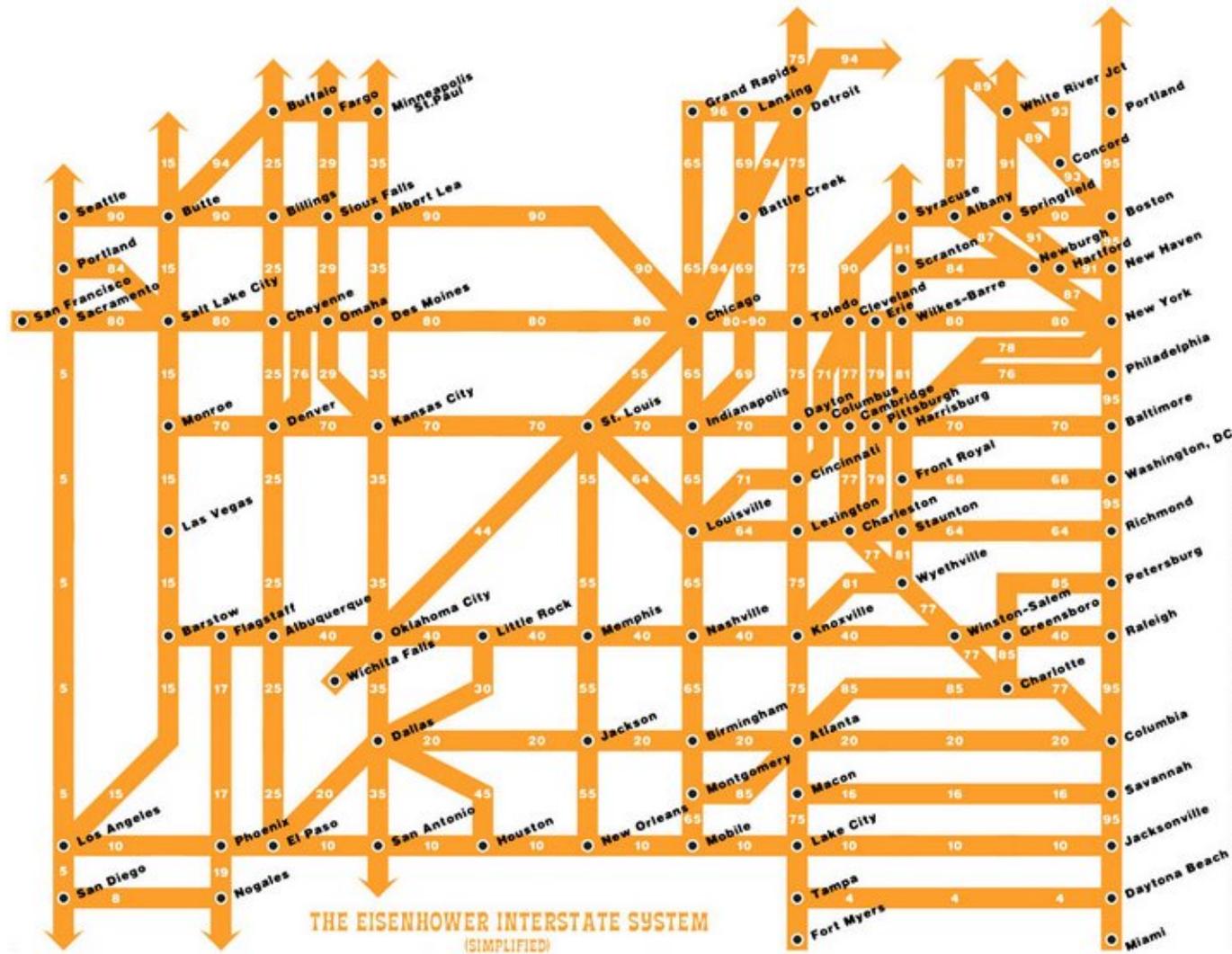
# Soziale Netze – Exxon Fördergelder



# Verkehrsnetze – Highways USA



# Verkehrsnetze – Highways USA

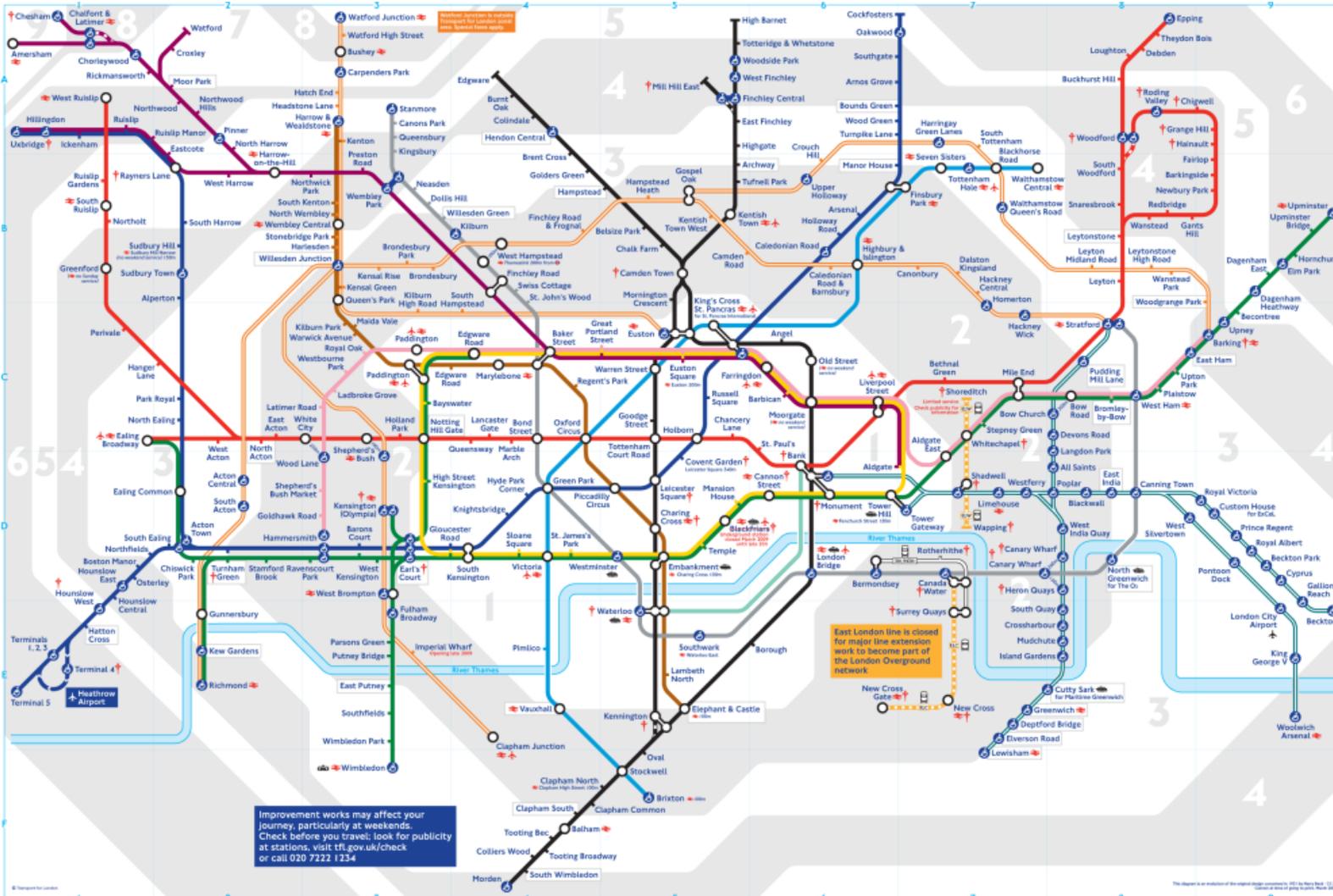


COPYRIGHT 2000-2007 CHRIS YATES WWW.CHRISYATES.NET

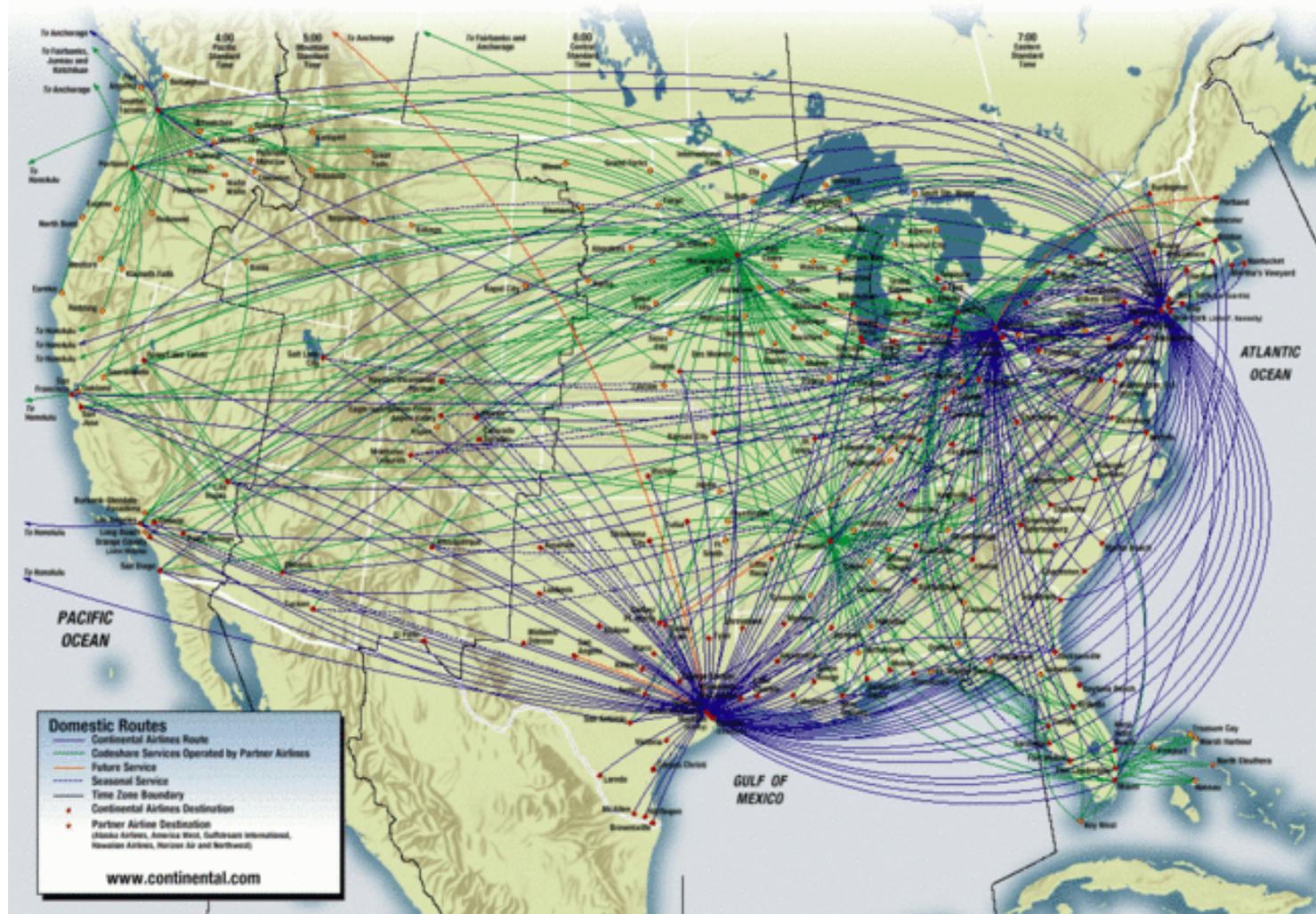
# Verkehrsnetze – U-Bahnen London

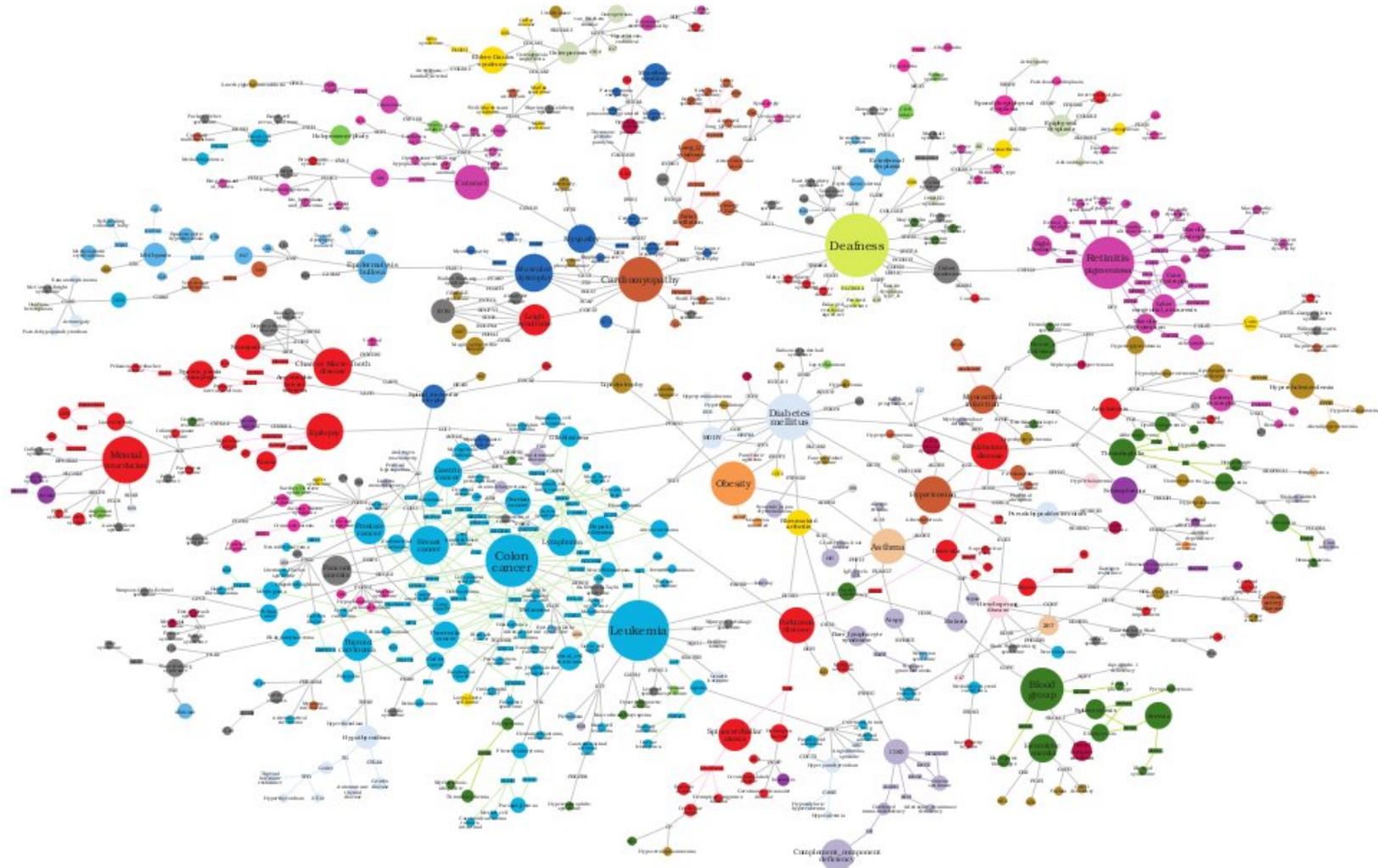


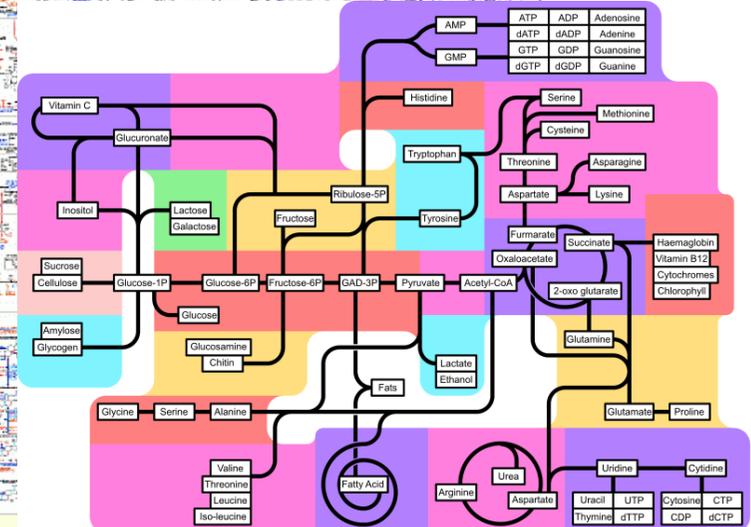
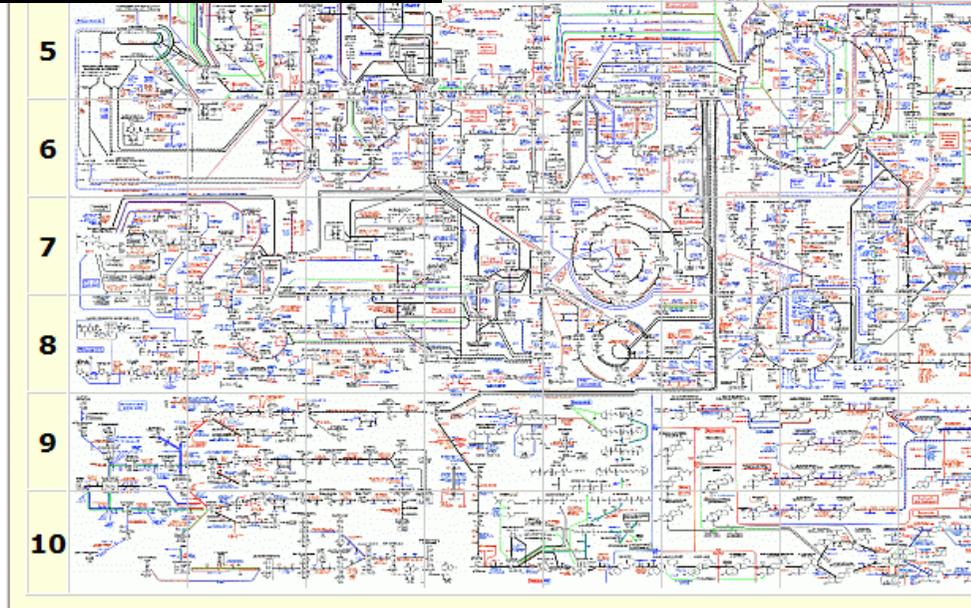
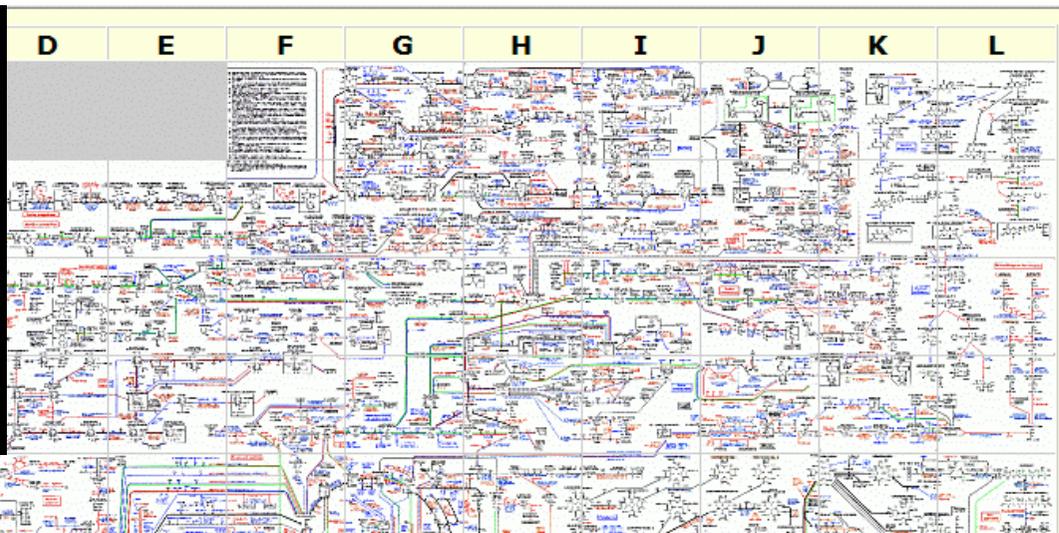
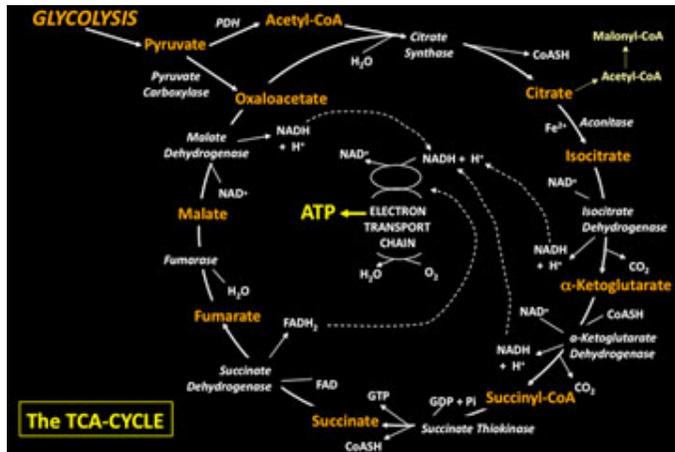
# Verkehrsnetze – U-Bahnen London

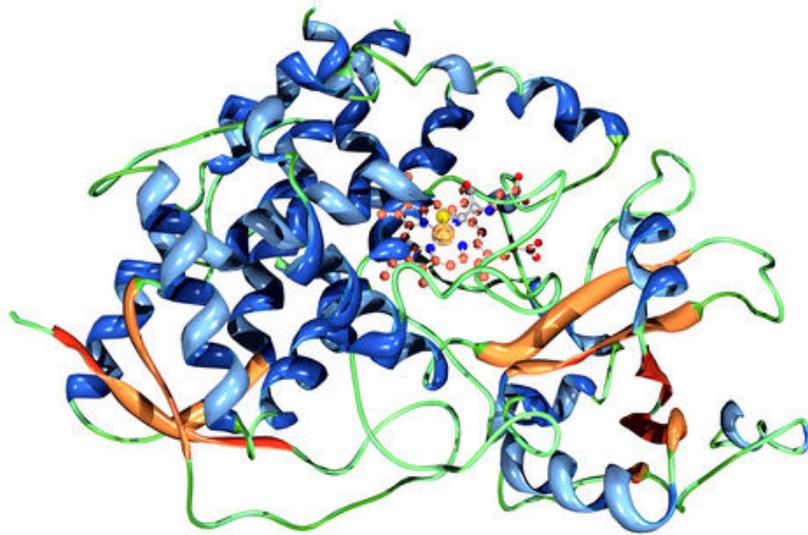


# Verkehrsnetze – Flugverbindungen Continental

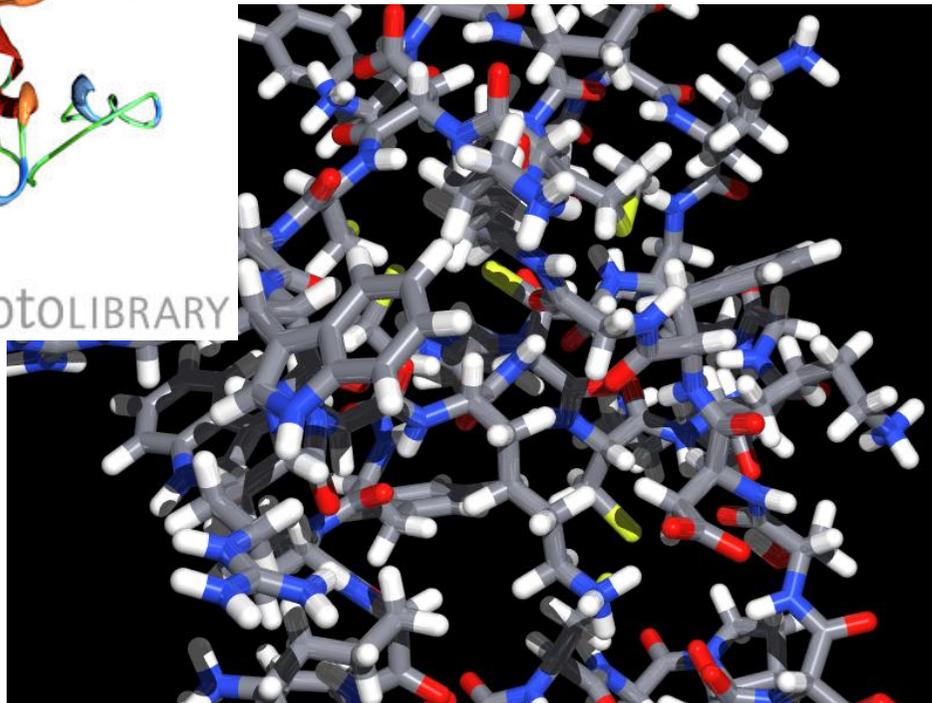






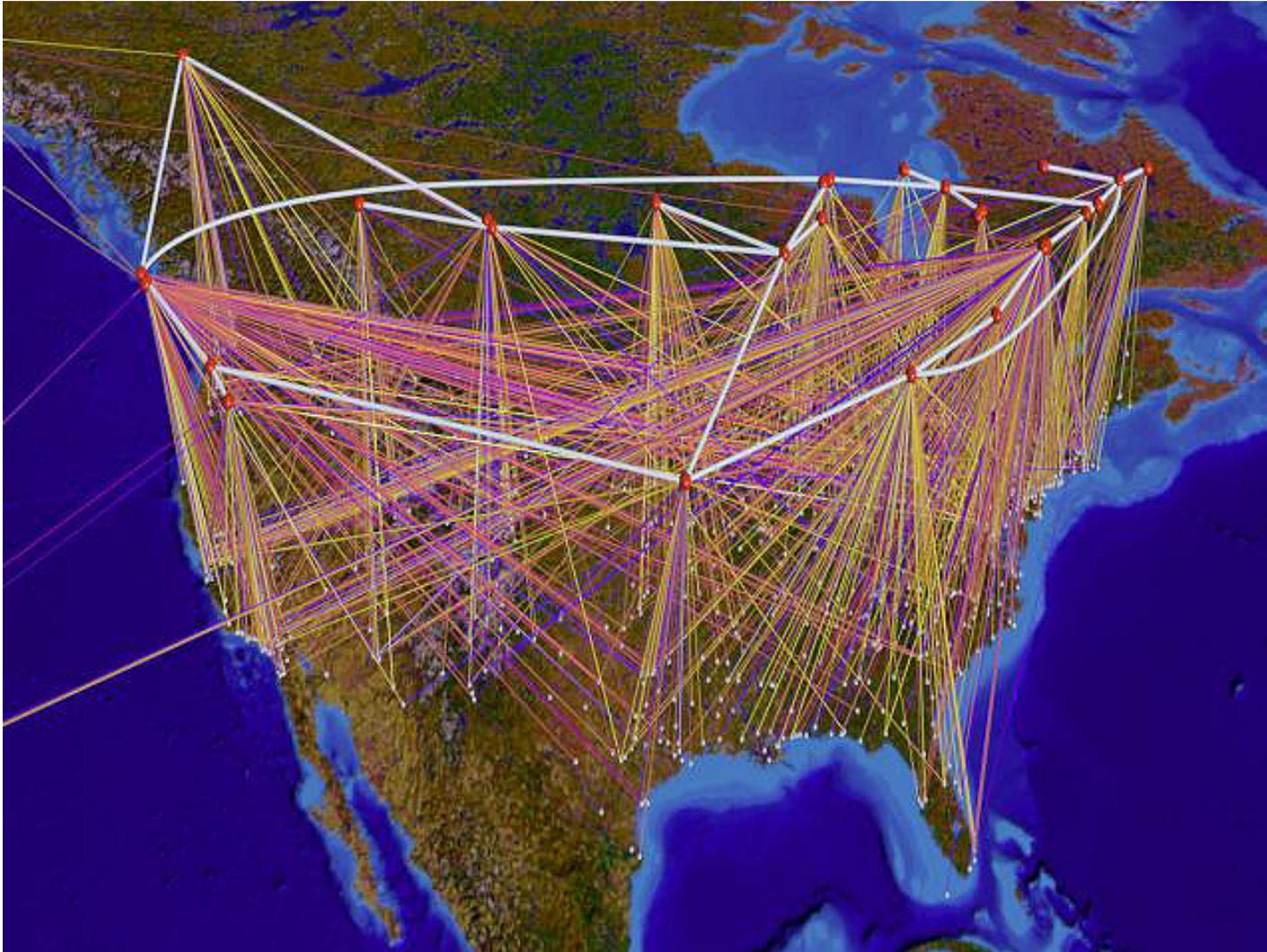


SCIENCEPHOTOLIBRARY

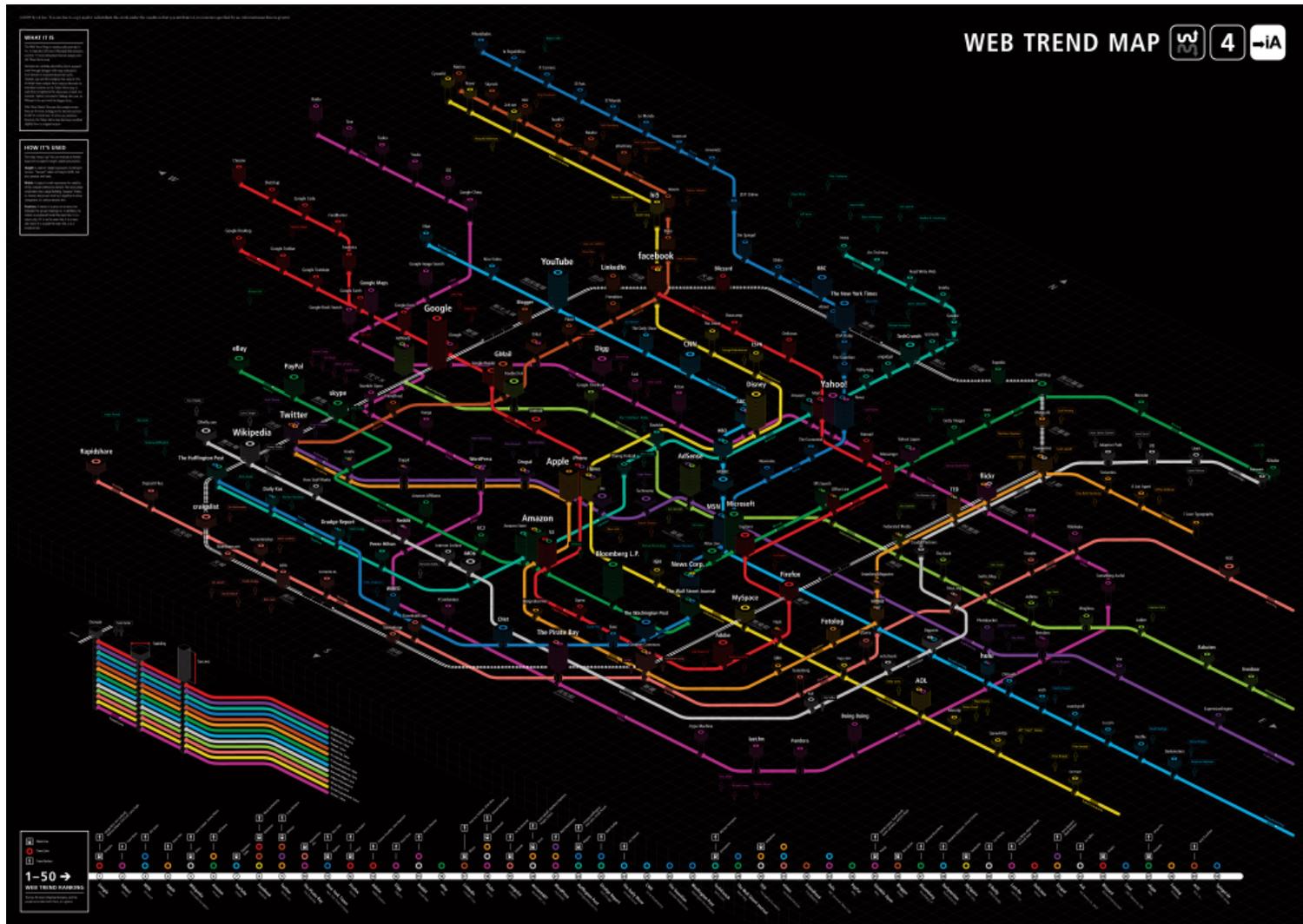




# Technische Netze – Internet USA

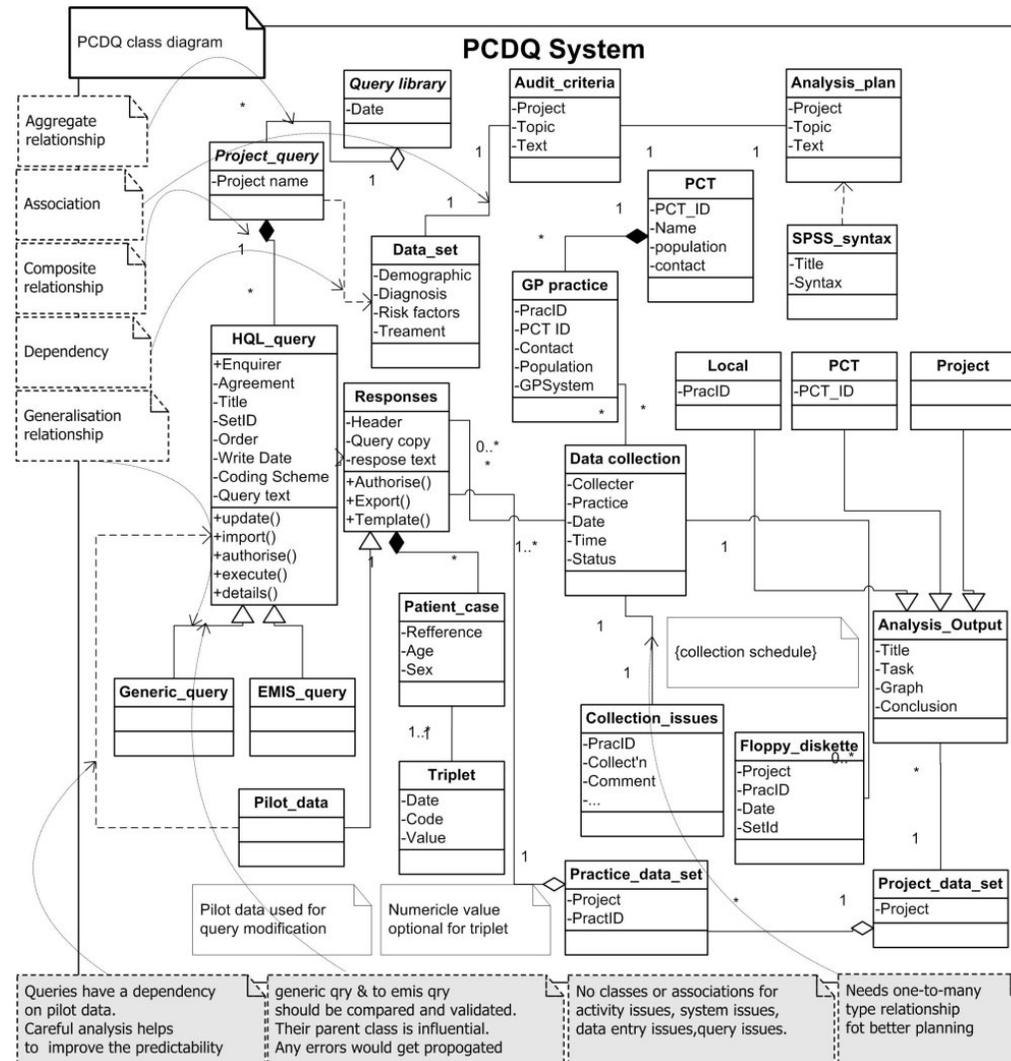


# Technische Netze – Webtrends





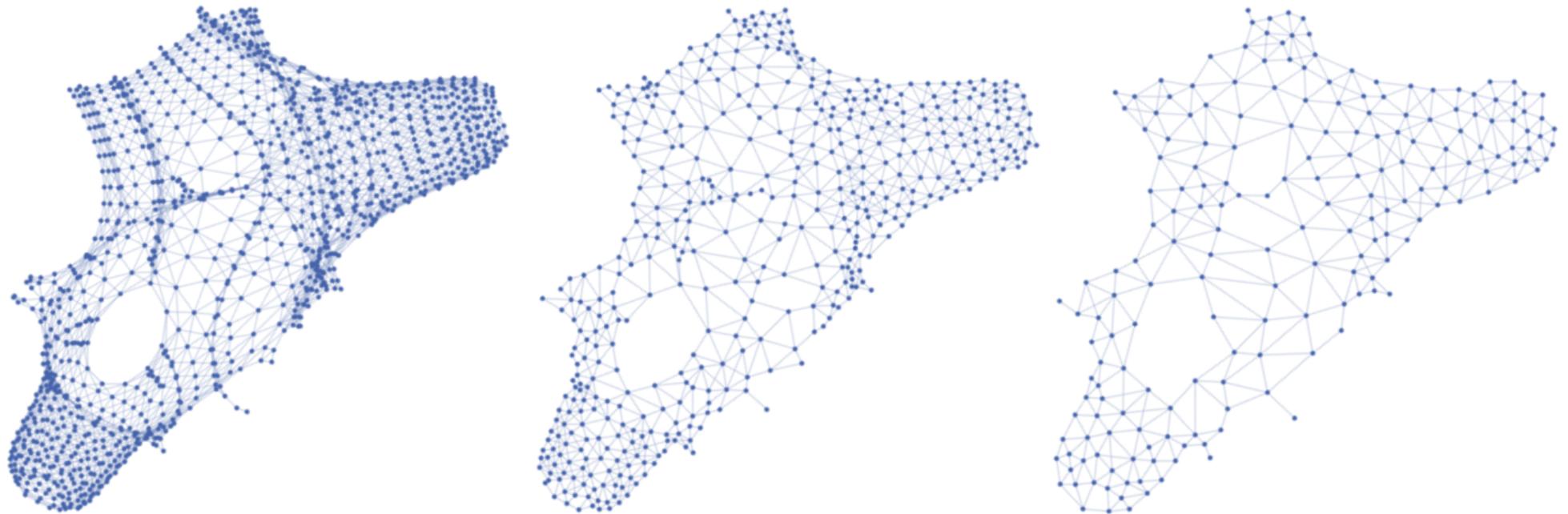
# Technische Netze – UML Diagramme



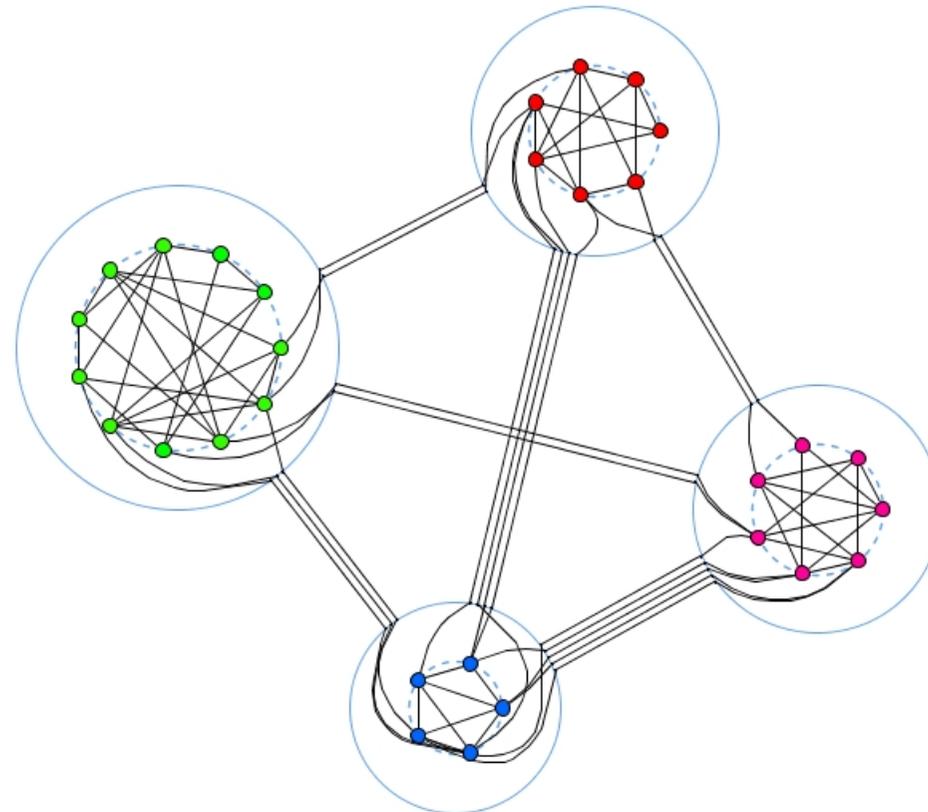
# Allgemeine Graphen – große Graphen



# Allgemeine Graphen – große Graphen



# Allgemeine Graphen – Mikro-Makro Layout



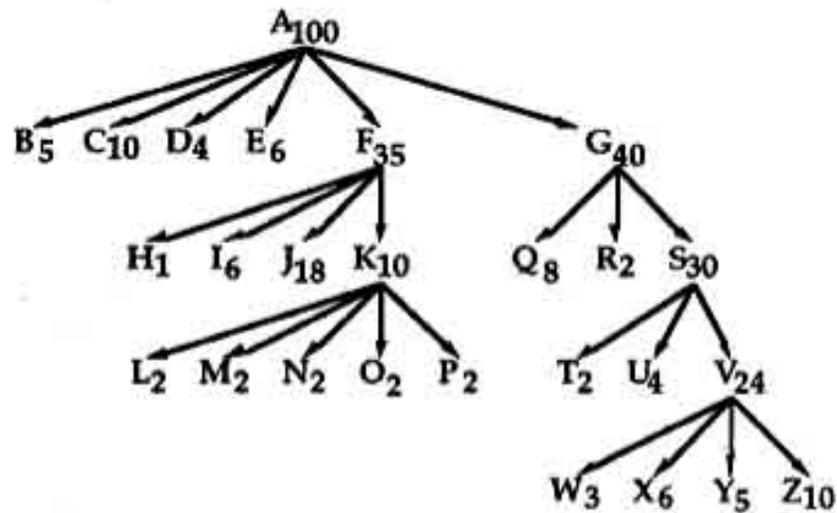


Figure 1: Traditional Tree Diagram Representation.

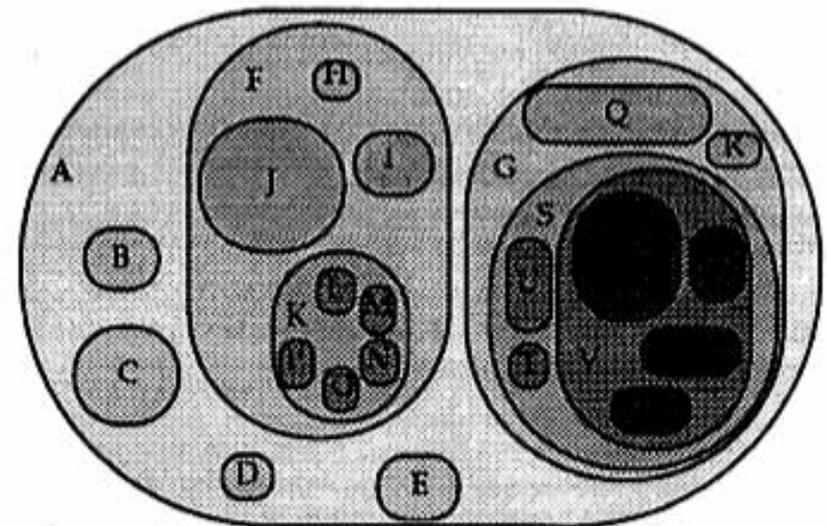
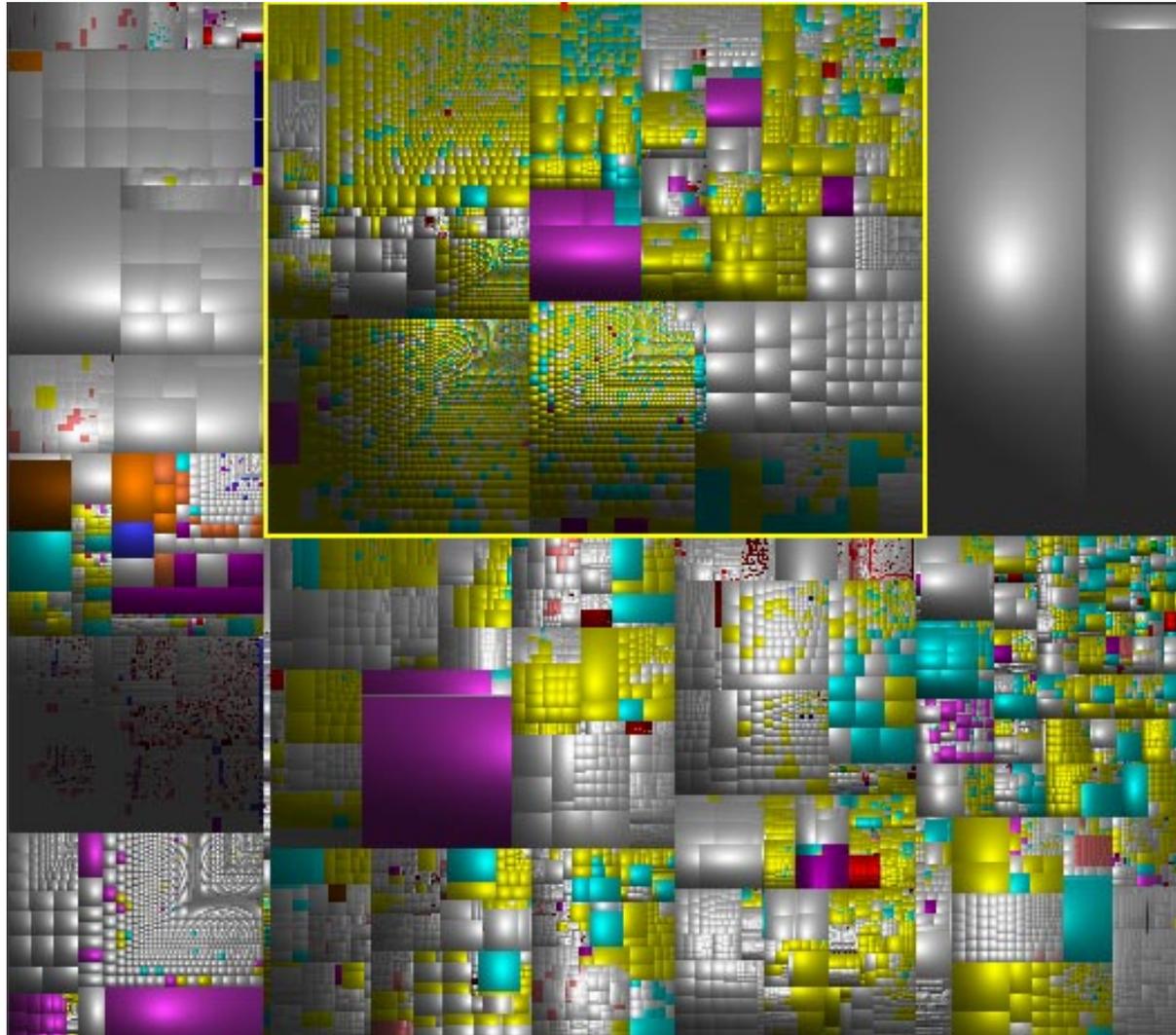
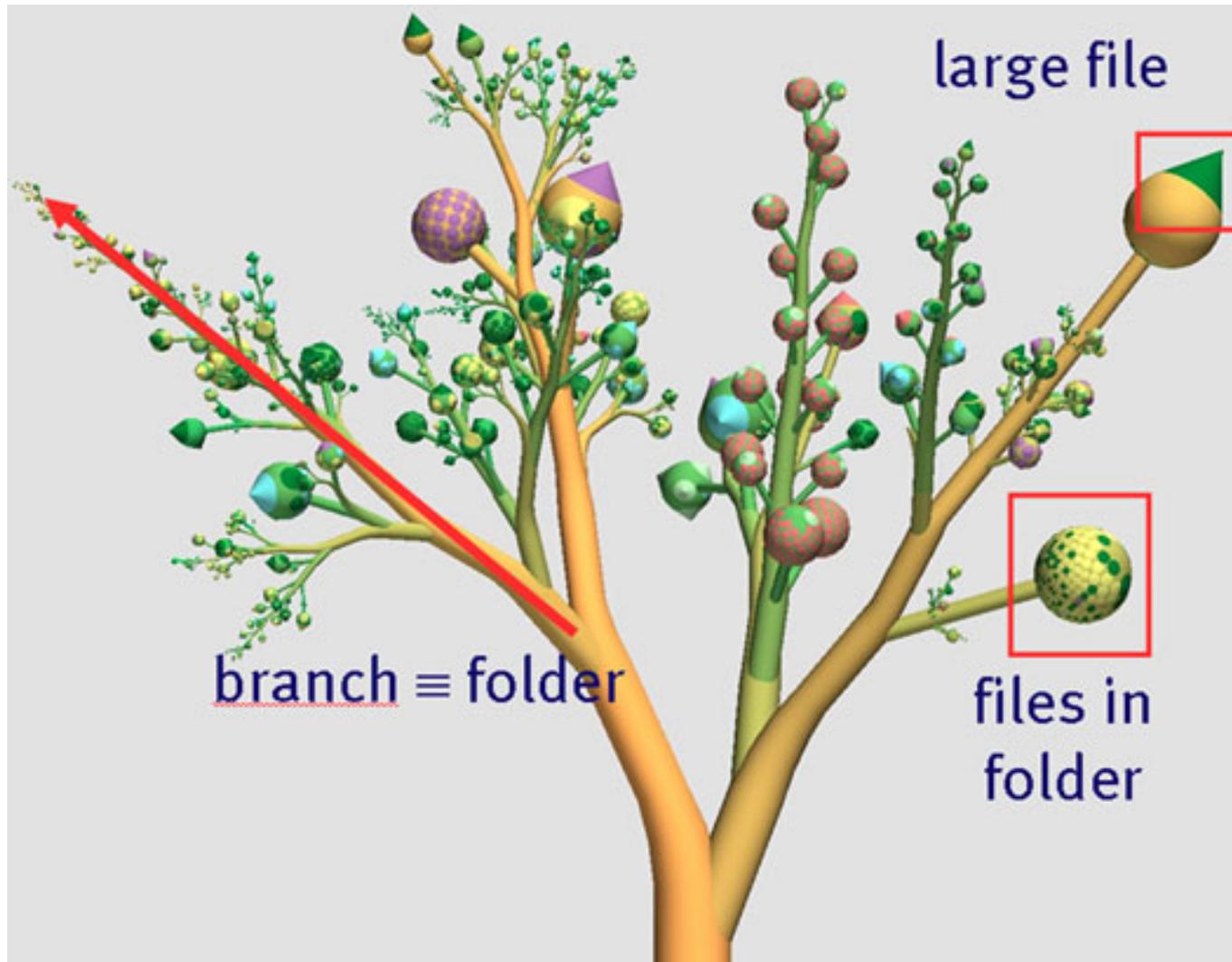


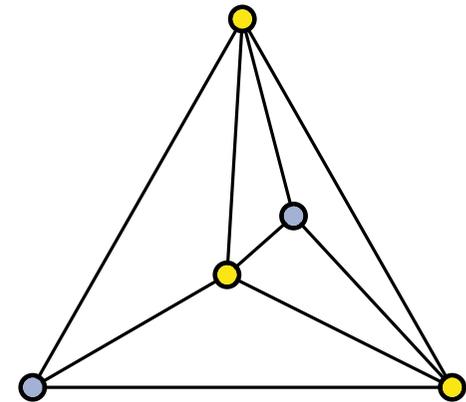
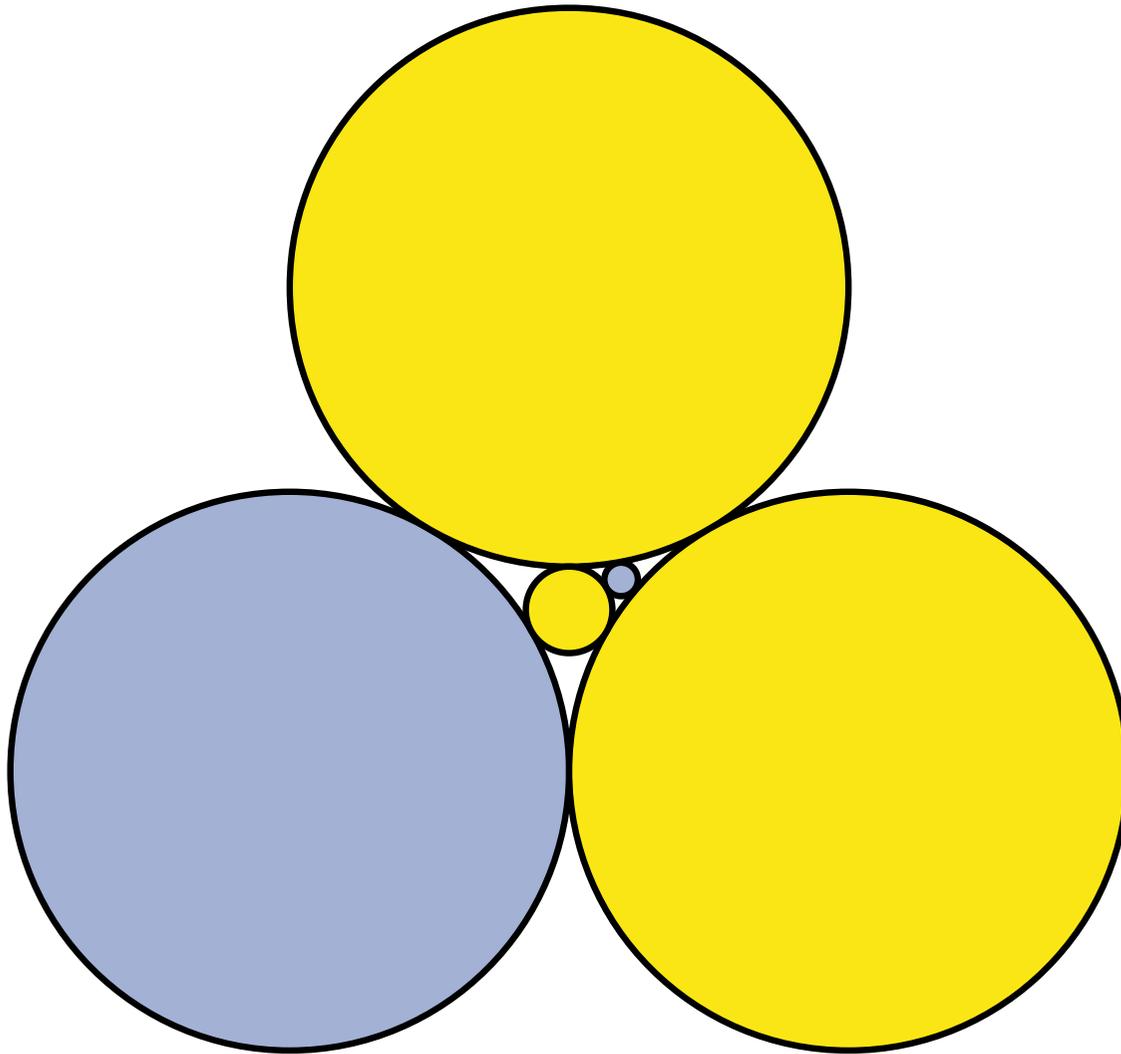
Figure 2: Venn Diagram Representation.  
Node size is proportional to weight.

# Alternative Darstellungen – Treemap



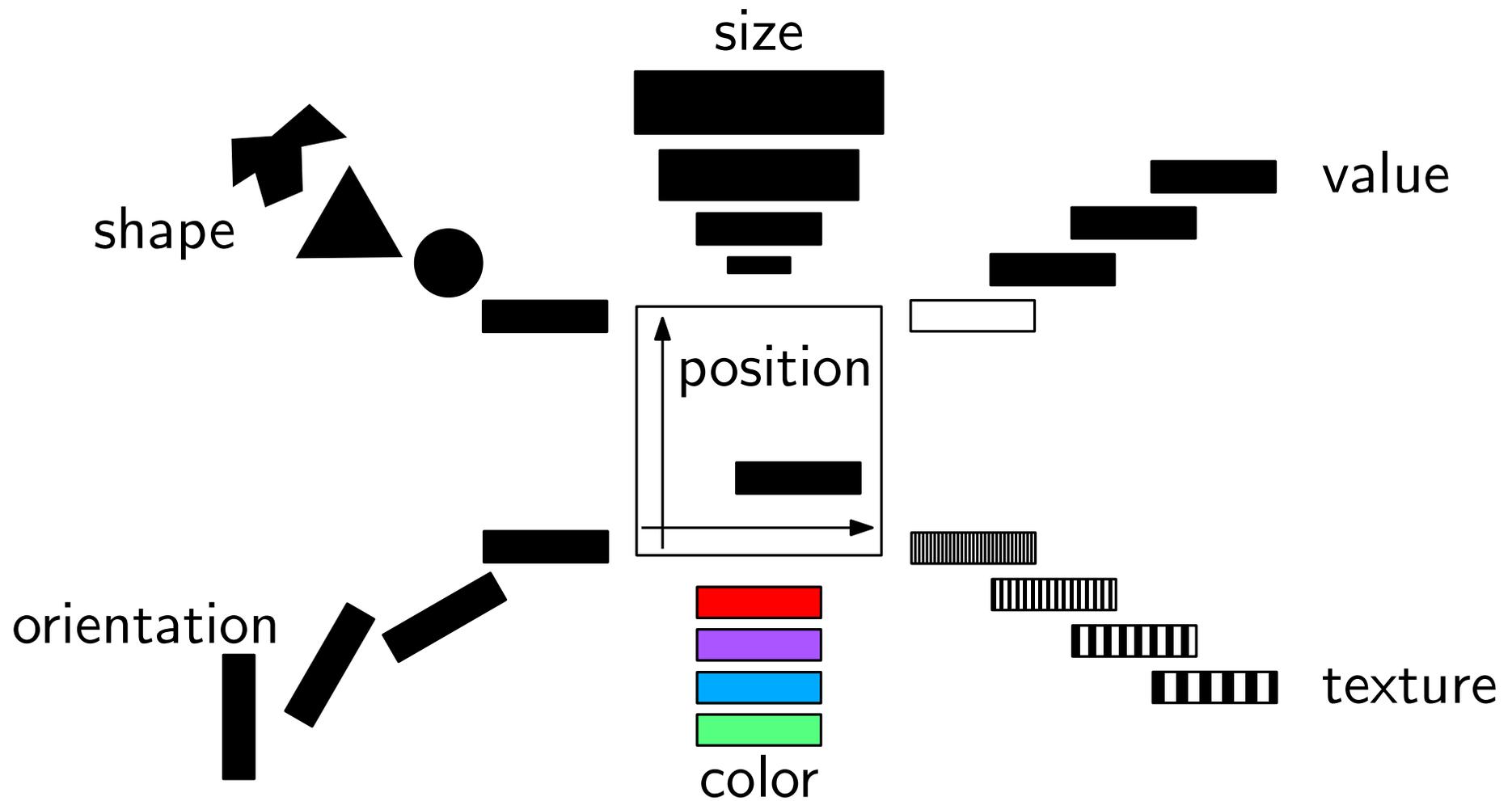
# Alternative Darstellungen – Baum 3D



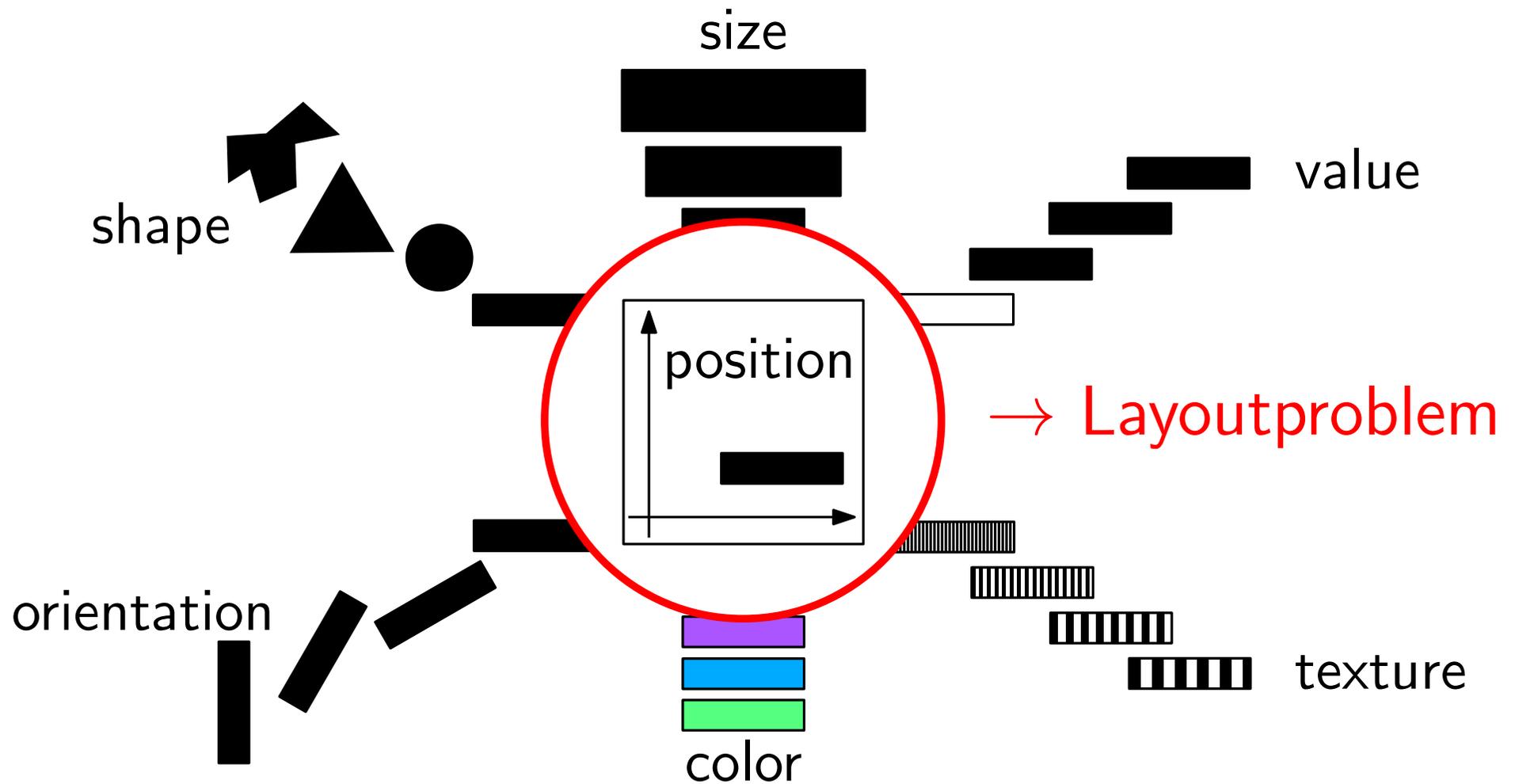


# Grundlegende Definitionen

# Visuelle Variablen nach Bertin (1967)



# Visuelle Variablen nach Bertin (1967)



# Das Layoutproblem

*hier:* Beschränkung auf die sog. **Standardrepräsentation**  
(node-link diagram)

## Graphvisualisierungsproblem

**geg.:** Graph  $G = (V, E)$

**ges.:** gute Zeichnung  $\Gamma$  von  $G$

- $\Gamma : V \rightarrow \mathbb{R}^2$ , Knoten  $v \mapsto$  Punkt  $\Gamma(v)$
- $\Gamma : E \rightarrow$  Kurven in  $\mathbb{R}^2$ , Kante  $\{u, v\} \mapsto$  einfache offene Kurve  $\Gamma(\{u, v\})$  mit Endpunkten  $\Gamma(u)$  und  $\Gamma(v)$

# Das Layoutproblem

*hier:* Beschränkung auf die sog. **Standardrepräsentation**  
(node-link diagram)

## Graphvisualisierungsproblem

**geg.:** Graph  $G = (V, E)$

**ges.:** gute Zeichnung  $\Gamma$  von  $G$

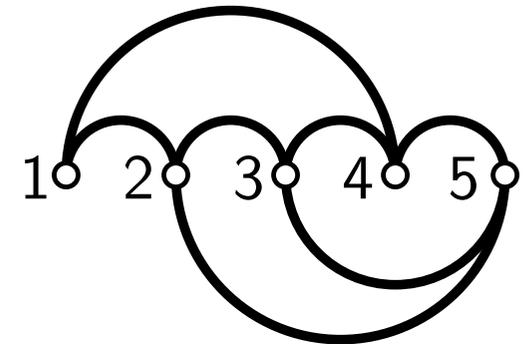
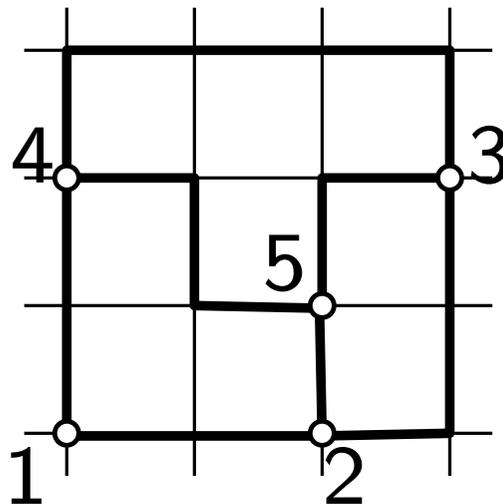
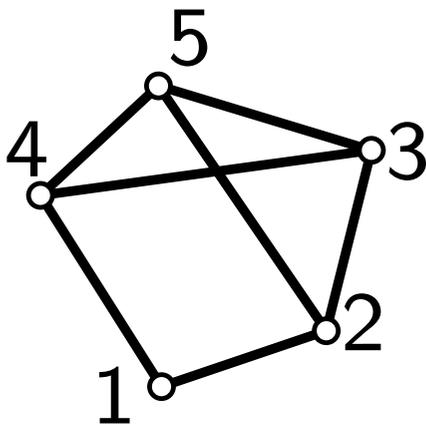
- $\Gamma : V \rightarrow \mathbb{R}^2$ , Knoten  $v \mapsto$  Punkt  $\Gamma(v)$
- $\Gamma : E \rightarrow$  Kurven in  $\mathbb{R}^2$ , Kante  $\{u, v\} \mapsto$  einfache offene Kurve  $\Gamma(\{u, v\})$  mit Endpunkten  $\Gamma(u)$  und  $\Gamma(v)$

Aber was ist eine gute Zeichnung?

# Anforderungen an ein Graphlayout

## 1) **Zeichenkonventionen**, erforderliche Eigenschaften, z.B.

- geradlinige Kanten mit  $\Gamma(uv) = \overline{\Gamma(u)\Gamma(v)}$
- orthogonale Kanten (i.A. mit Knicken)
- Gitterzeichnungen
- kreuzungsfrei
- ...



# Anforderungen an ein Graphlayout

1) **Zeichenkonventionen**, erforderliche Eigenschaften

2) **Ästhetikkriterien** (zu optimieren), z.B.

- Kreuzungsminimierung
- Knickminimierung
- gleichmäßige Kantenlängen
- minimale Gesamtlänge/Fläche
- Winkelauflösung
- Symmetrie / Struktur
- ...

→ führen häufig zu NP-schweren Optimierungsproblemen!

→ oft mehrere konkurrierende Kriterien

# Anforderungen an ein Graphlayout

- 1) **Zeichenkonventionen**, erforderliche Eigenschaften
- 2) **Ästhetikkriterien** (zu optimieren)
- 3) **Lokale Nebenbedingungen**, z.B.
  - Positionseinschränkungen für Nachbarknoten
  - Einschränkungen für Gruppen von Knoten/Kanten

## Graphvisualisierungsproblem

**geg.:** Graph  $G = (V, E)$

**ges.:** **gute** Zeichnung  $\Gamma$  von  $G$ , die

- Zeichenkonventionen erfüllt
- Ästhetikkriterien optimiert
- ggf. weitere Nebenbedingungen erfüllt

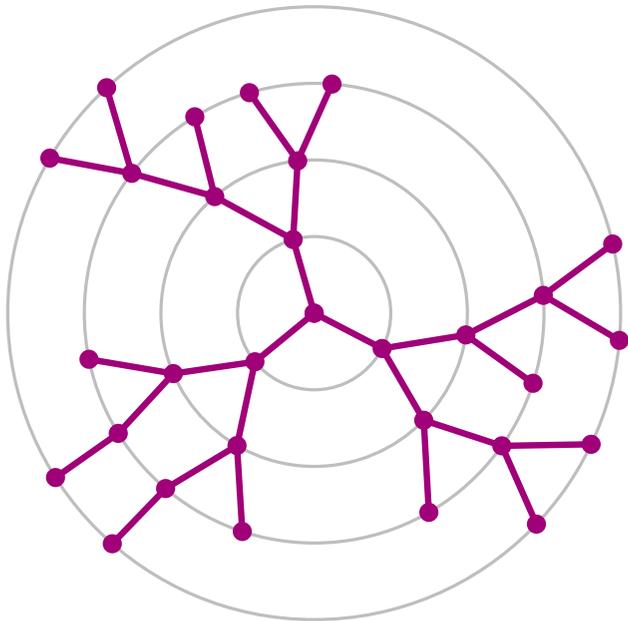
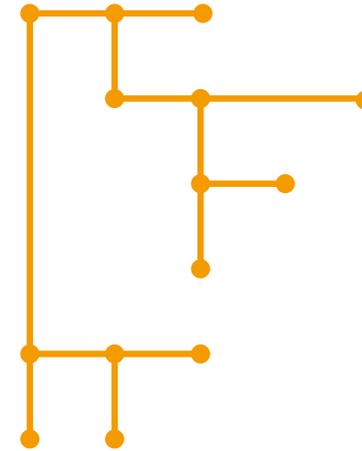
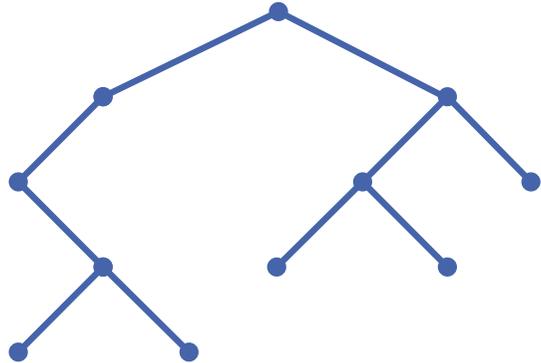
## Graphvisualisierungsproblem

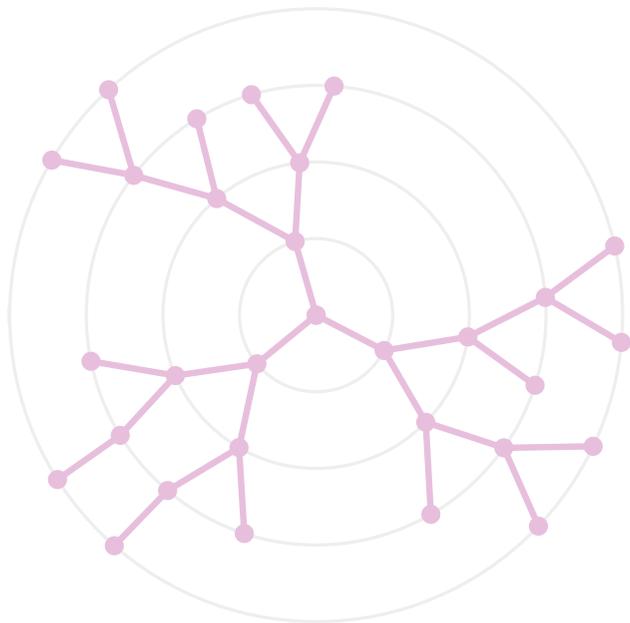
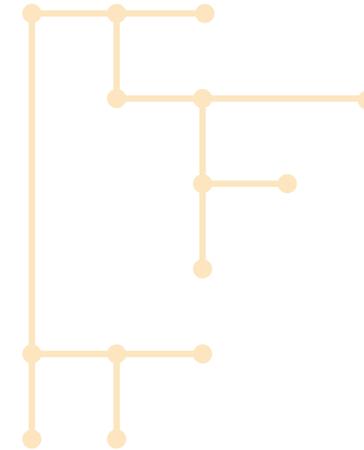
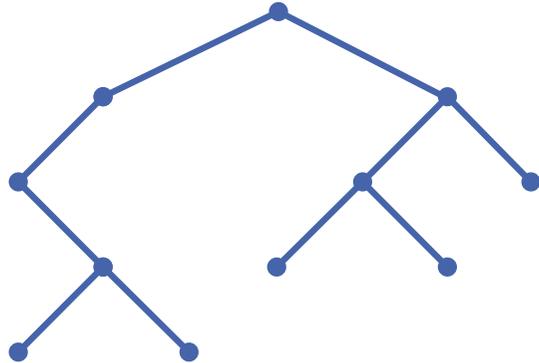
**geg.:** Graph  $G = (V, E)$

**ges.:** **gute** Zeichnung  $\Gamma$  von  $G$ , die

- Zeichenkonventionen erfüllt
  - Ästhetikkriterien optimiert
  - ggf. weitere Nebenbedingungen erfüllt
- 
- führt zu algorithmisch interessanten Fragestellungen
  - beweisbare Garantien  $\rightsquigarrow$  gute praktische Verfahren
  - nachgelagertes Renderingproblem bleibt außen vor!

# Baumlayouts



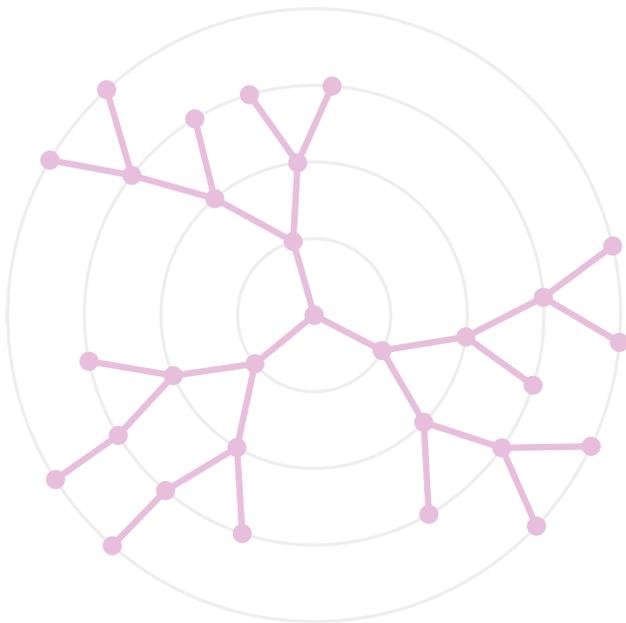
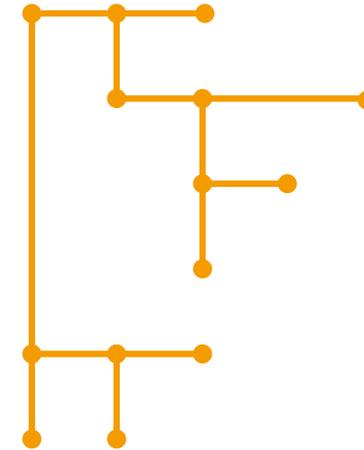
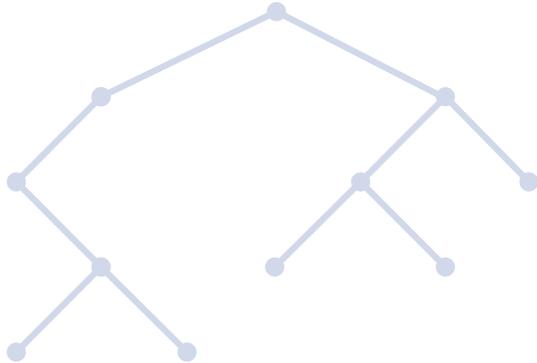


Zeichenstil:

- geradlinig
- Knoten auf Ebenen
- Vorgänger zentriert über Nachfolgern

Optimierungskriterien:

- Breite (Fläche)

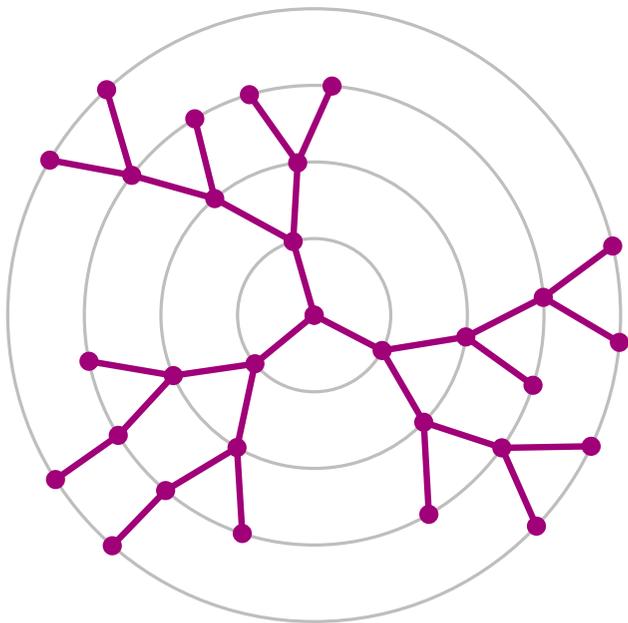
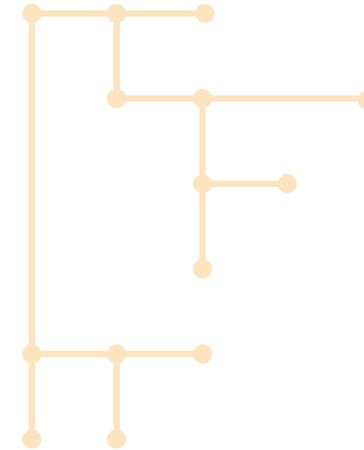
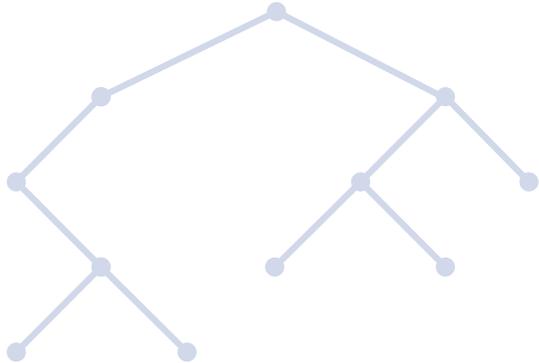


Zeichenstil:

- geradlinig, orthogonal
- Knoten auf Ebenen
- Nachfolger rechts unterhalb

Optimierungskriterien:

- Fläche / Umfang

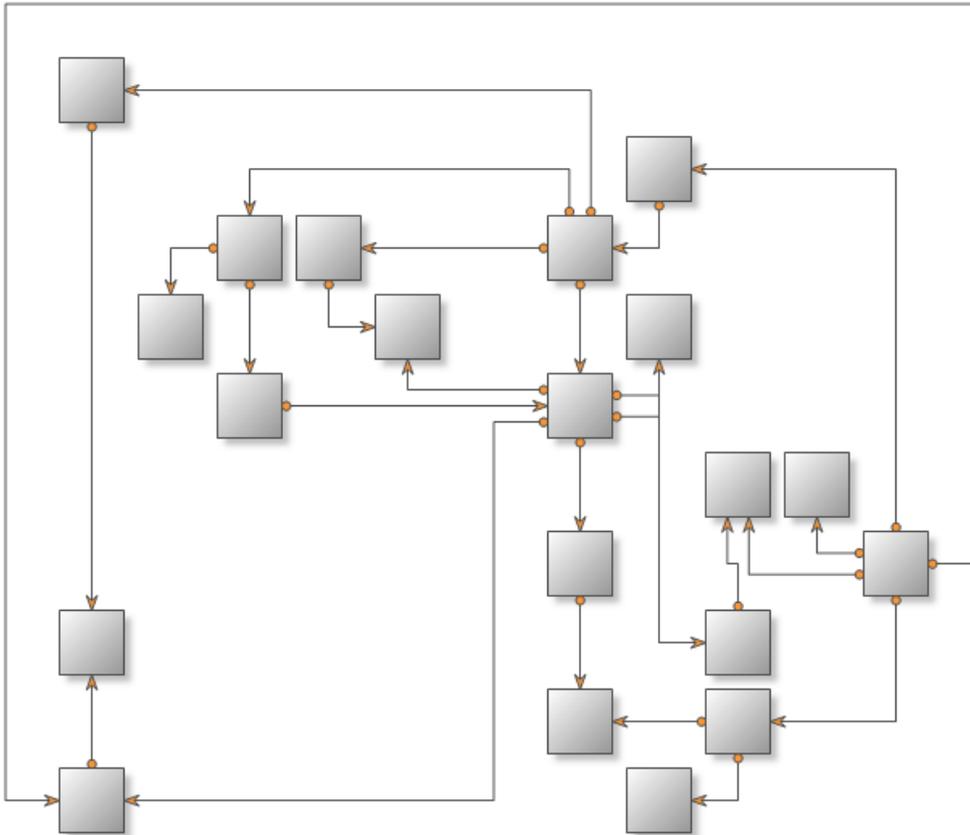


Zeichenstil:

- geradlinig
- Knoten auf radialen Ebenen
- Vorgänger zentriert über Nachfolgern

Optimierungskriterien:

- Winkelauflösung
- Fläche

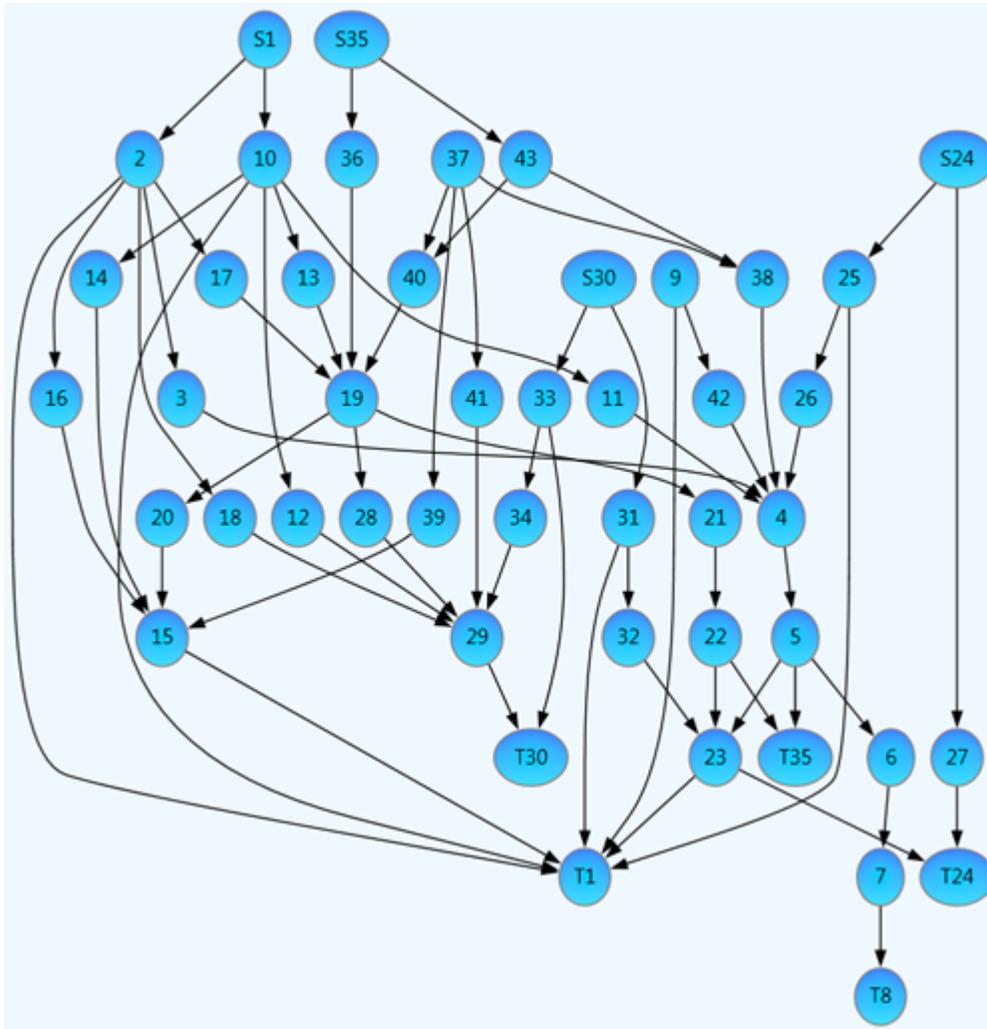


Zeichenstil:

- Knoten auf Gitterpunkten
- Kanten verlaufen auf Gitter

Optimierungsziele:

- Kreuzungen
- Knicke
- Fläche



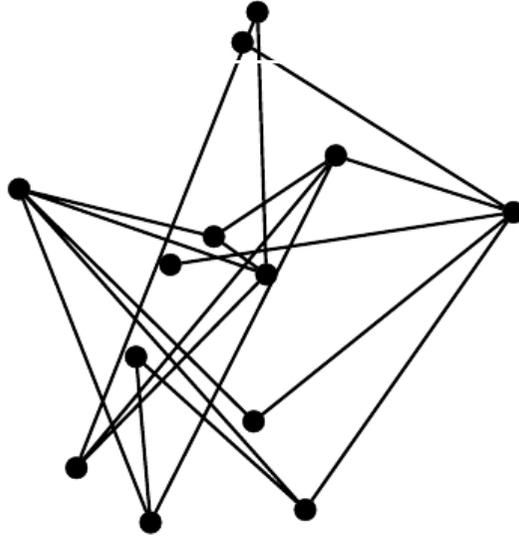
Zeichenstil:

- Knoten auf Ebenen
- Kanten verlaufen abwärts

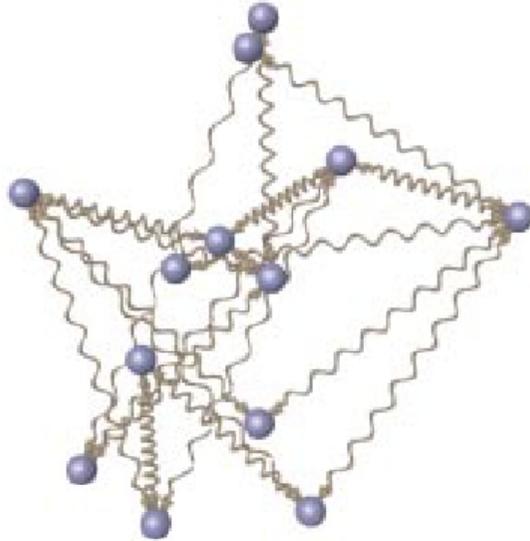
Optimierungsziele:

- Wenige Kanten rückwärts
- Breite
- Kreuzungen
- Kanten möglichst gerade

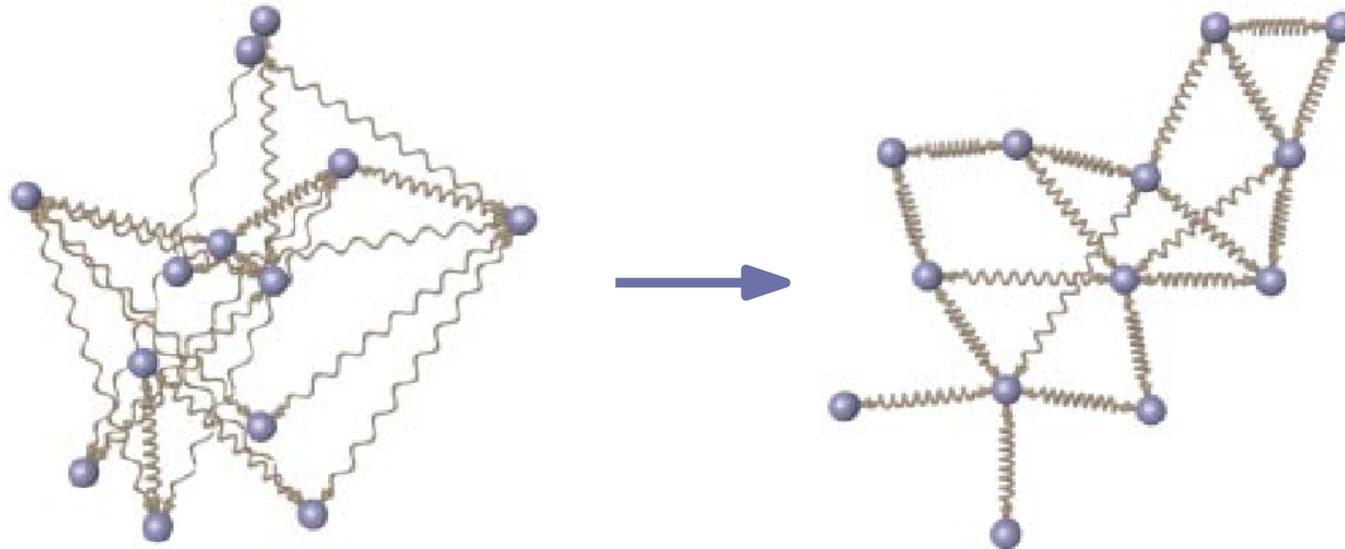
# Kräftebasiertes Layout



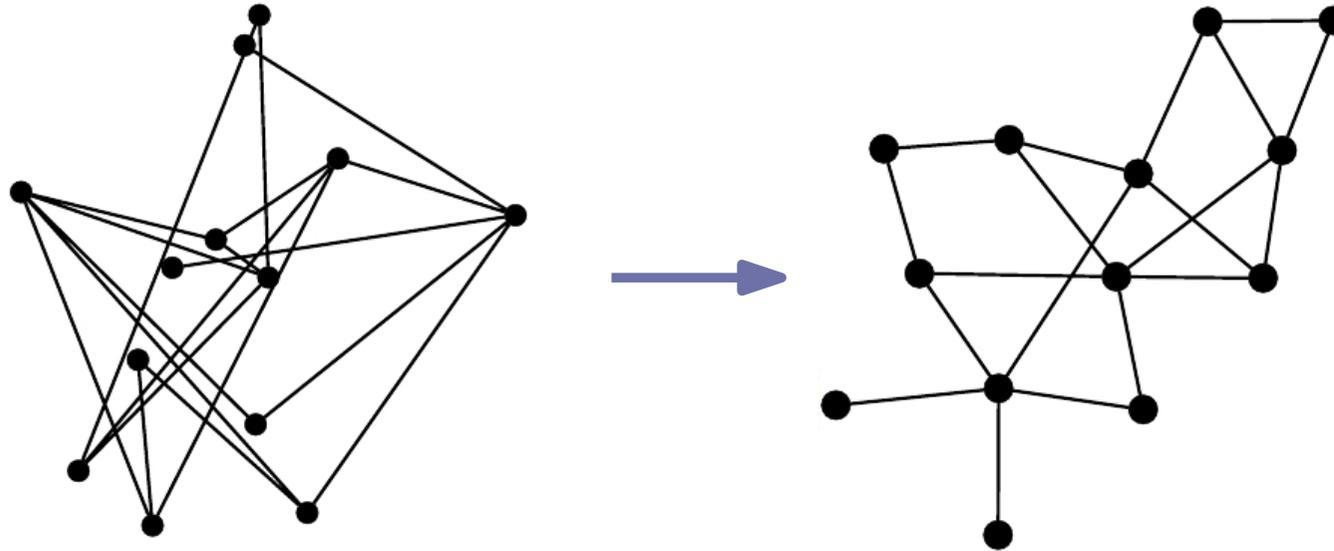
# Kräftebasiertes Layout



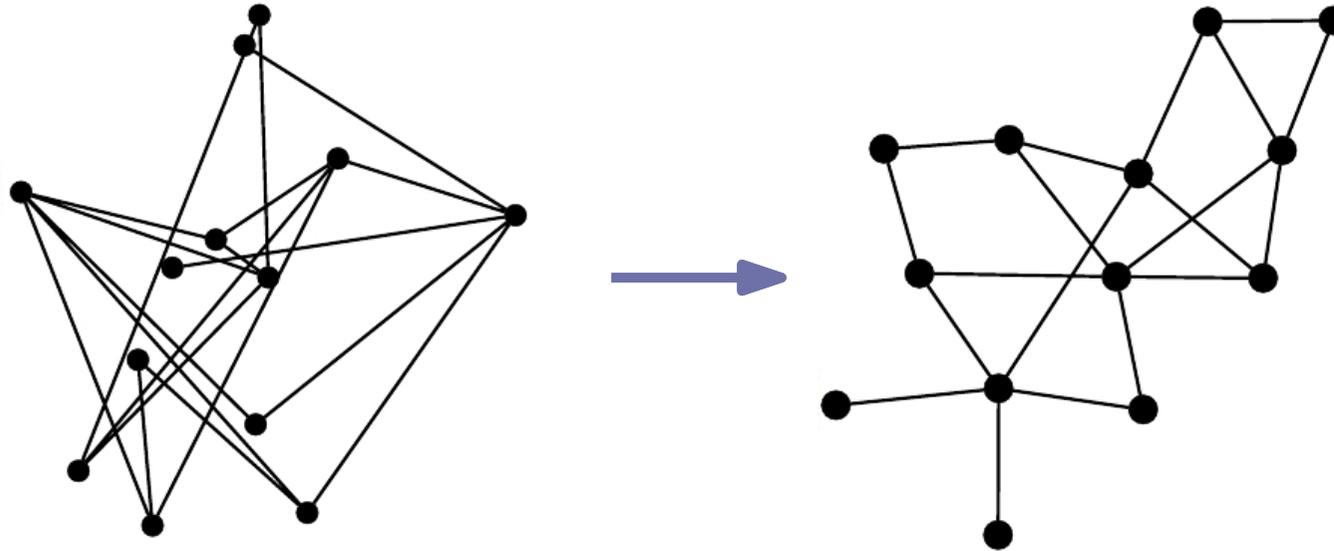
“To embed a graph we replace the vertices by steel rings and replace each edge with a spring to form a mechanical system . . .



“To embed a graph we replace the vertices by steel rings and replace each edge with a spring to form a mechanical system . . . The vertices are placed in some initial layout and let go so that the spring forces on the rings move the system to a minimal energy state.” [Eades, '84]



“To embed a graph we replace the vertices by steel rings and replace each edge with a spring to form a mechanical system . . . The vertices are placed in some initial layout and let go so that the spring forces on the rings move the system to a minimal energy state.” [Eades, '84]



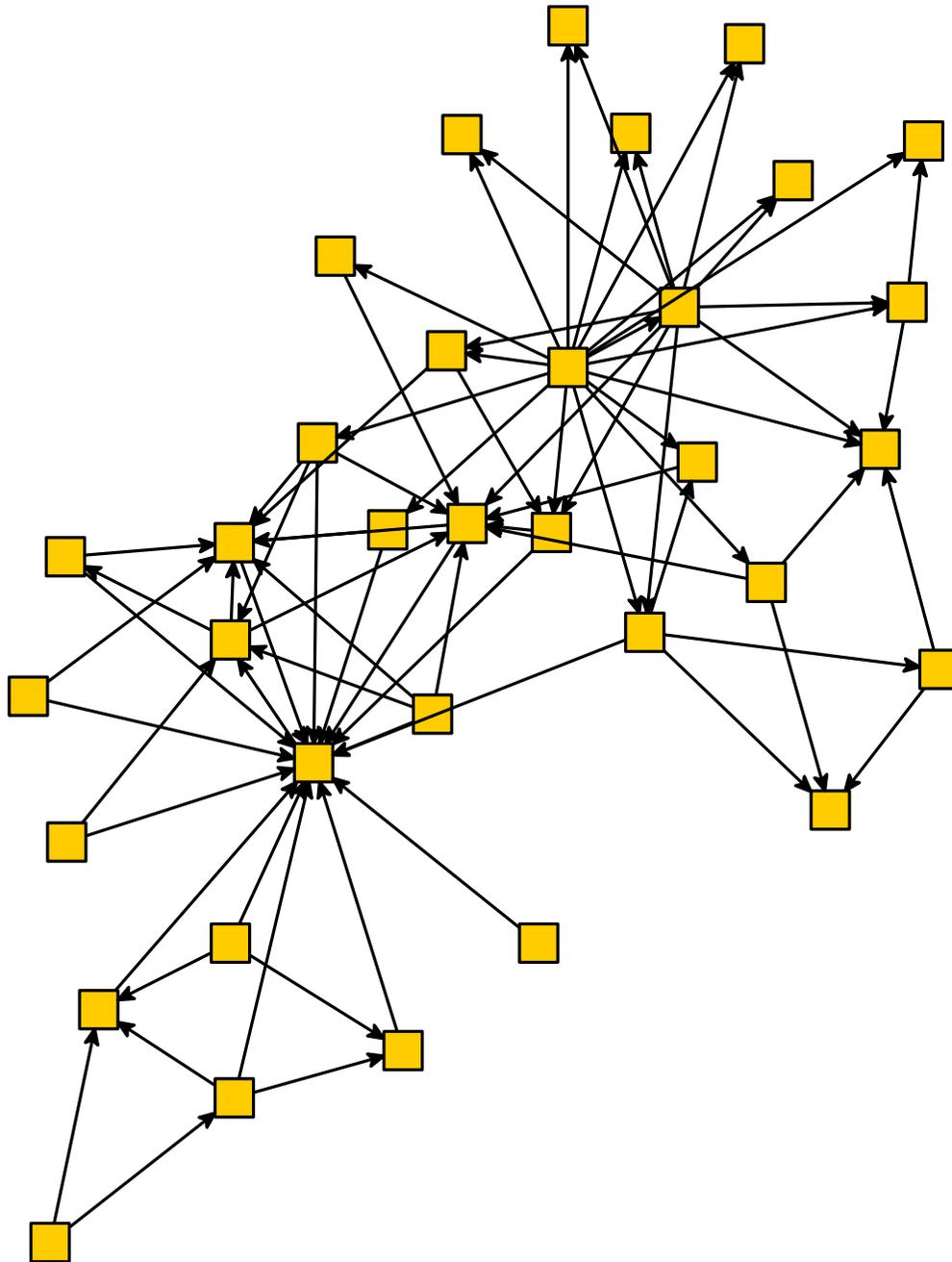
“To each place a ring are e

Sogenannte **spring-embedder** Algorithmen, die nach diesem oder ähnlichen Prinzipien arbeiten, gehören zu den häufigst verwendeten Graphenzeichenmethoden in der Praxis.

Aber: Layout nicht gut kontrollierbar

# Ein Beispielverfahren

# Ein Beispielgraph...

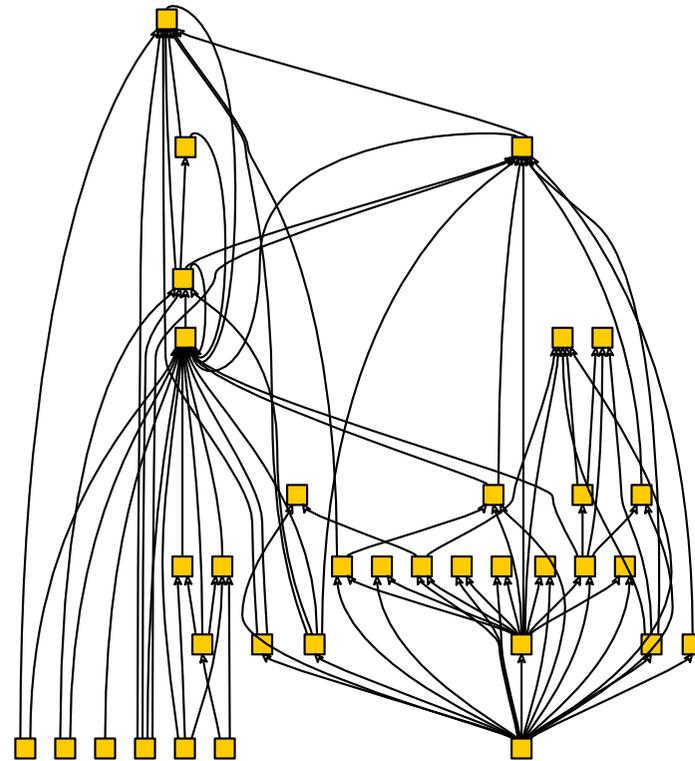
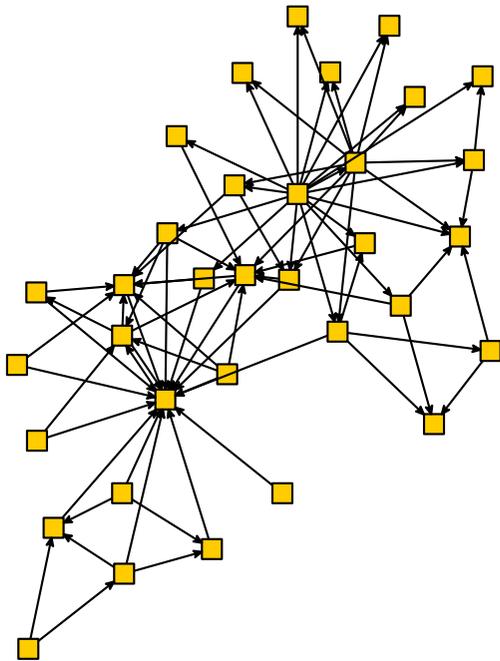


- Welche Eigenschaften haben wir?
- Welche Ästhetikkriterien sind sinnvoll?

# Lagenlayouts

**Geg.:** gerichteter Graph  $D = (V, A)$

**Ges.:** Zeichnung von  $D$ , die die Hierarchie verdeutlicht



# Lagenlayouts

**Geg.:** gerichteter Graph  $D = (V, A)$

**Ges.:** Zeichnung von  $D$ , die die Hierarchie verdeutlicht

## Kriterien:

- möglichst viele Kanten aufwärtsgerichtet
- Kanten möglichst geradlinig und kurz
- Zuordnung der Knoten auf (wenige) horizontale Ebenen
- möglichst wenige Kantenkreuzungen
- Knoten gleichmäßig verteilt

# Lagenlayouts

**Geg.:** gerichteter Graph  $D = (V, A)$

**Ges.:** Zeichnung von  $D$ , die die Hierarchie verdeutlicht

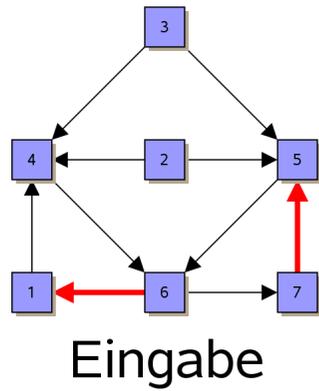
## Kriterien:

- möglichst viele Kanten aufwärtsgerichtet
- Kanten möglichst geradlinig und kurz
- Zuordnung der Knoten auf (wenige) horizontale Ebenen
- möglichst wenige Kantenkreuzungen
- Knoten gleichmäßig verteilt

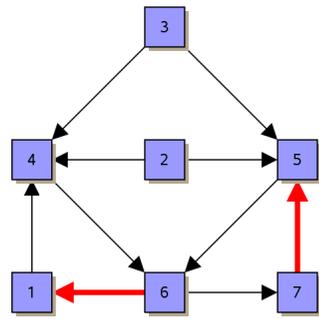


Optimierungskriterien widersprechen sich zum Teil

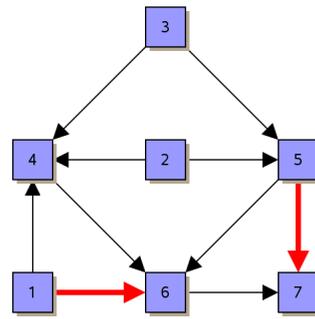
# Sugiyama Framework (Sugiyama, Tagawa, Toda 1981)



# Sugiyama Framework (Sugiyama, Tagawa, Toda 1981)

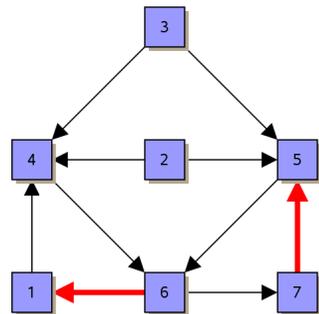


Eingabe

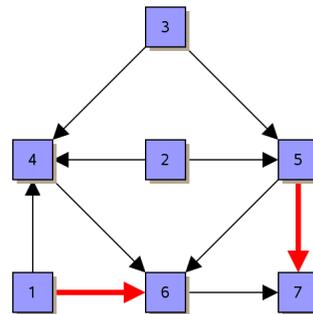


Kreise brechen

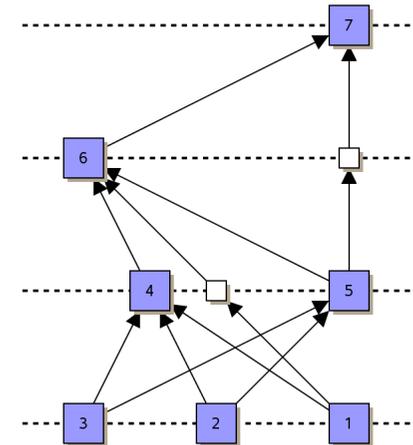
# Sugiyama Framework (Sugiyama, Tagawa, Toda 1981)



Eingabe

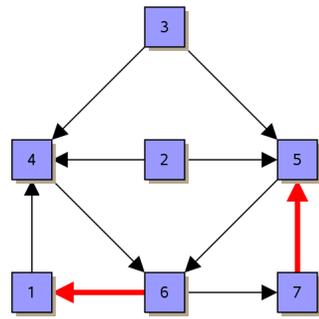


Kreise brechen

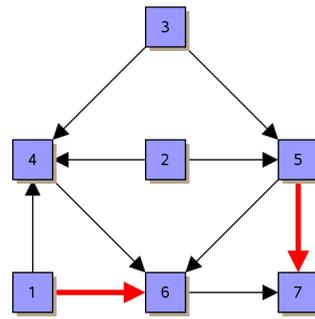


Lagenzuordnung

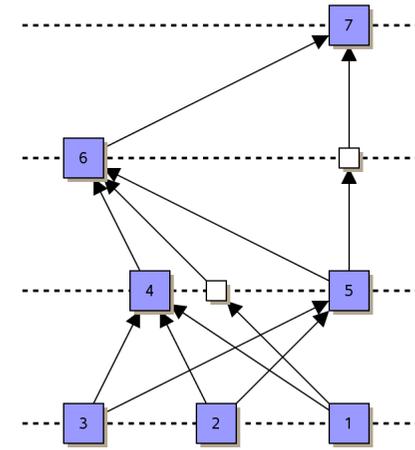
# Sugiyama Framework (Sugiyama, Tagawa, Toda 1981)



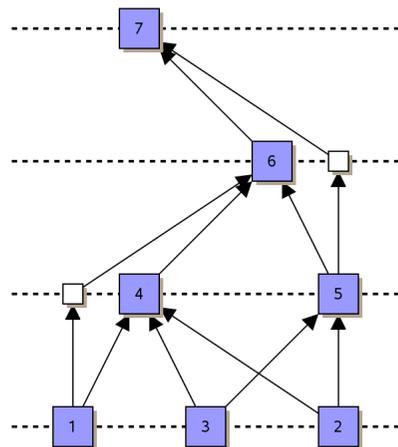
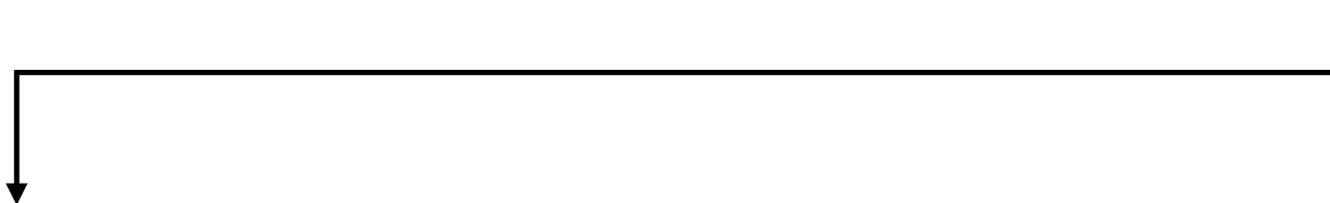
Eingabe



Kreise brechen

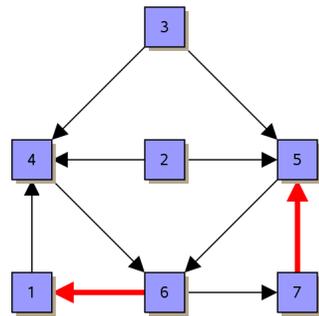


Lagenzuordnung

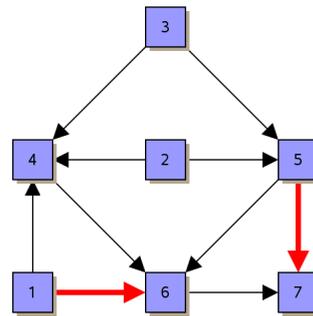


Kreuzungsminimierung

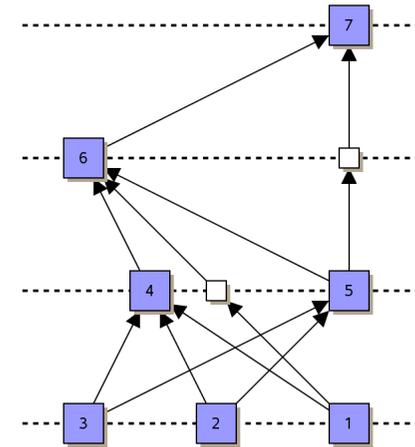
# Sugiyama Framework (Sugiyama, Tagawa, Toda 1981)



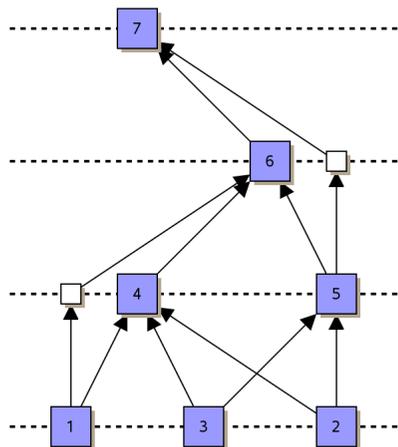
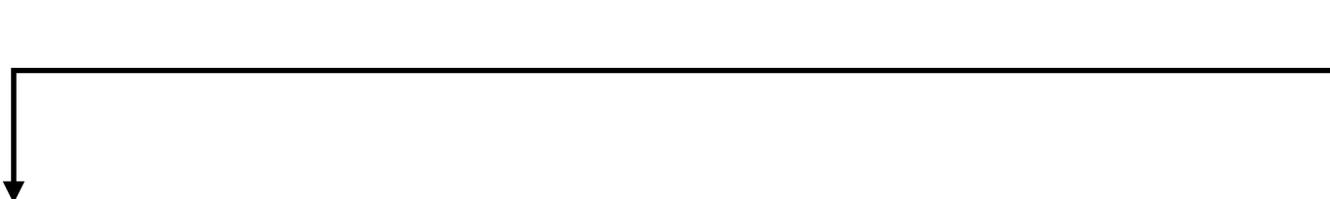
Eingabe



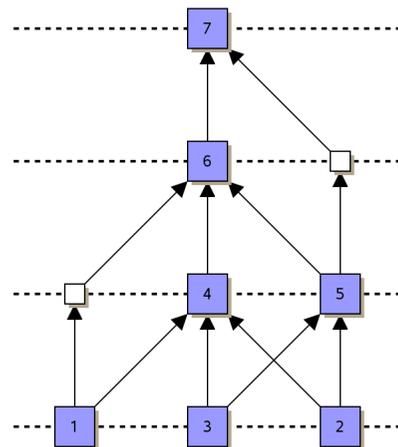
Kreise brechen



Lagenzuordnung

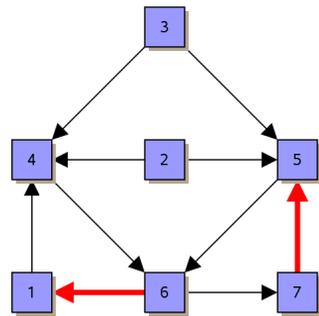


Kreuzungsminimierung

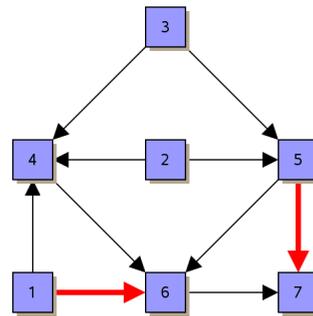


Knotenpositionierung

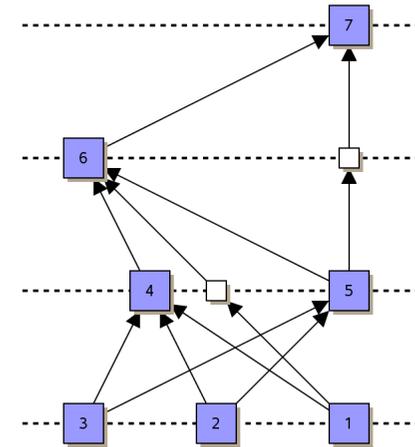
# Sugiyama Framework (Sugiyama, Tagawa, Toda 1981)



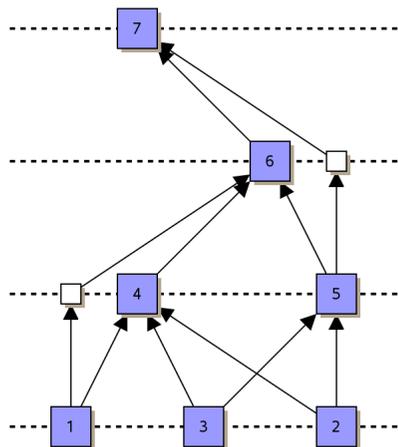
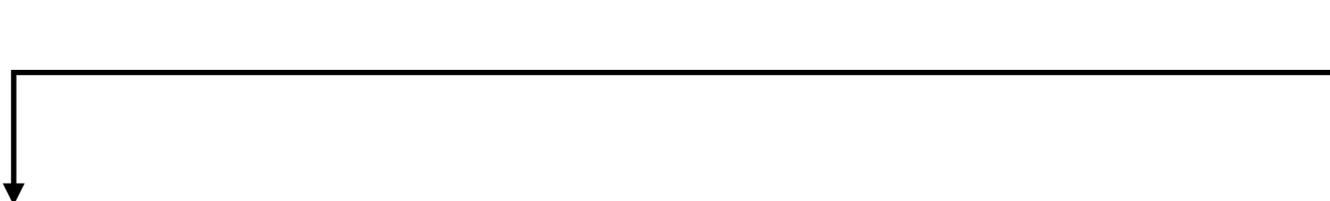
Eingabe



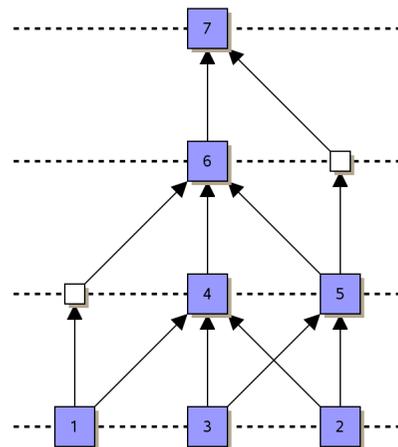
Kreise brechen



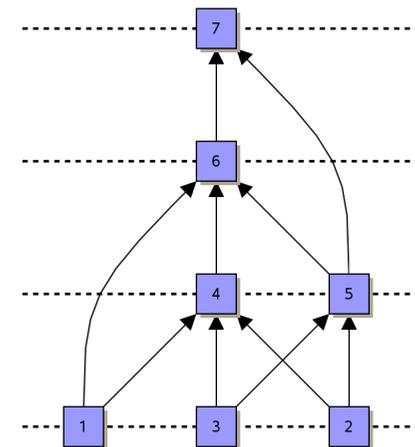
Lagenzuordnung



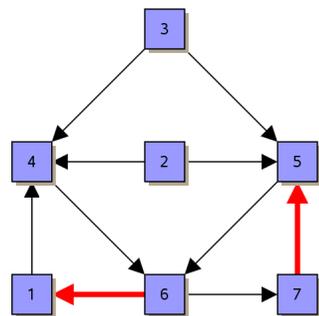
Kreuzungsminimierung



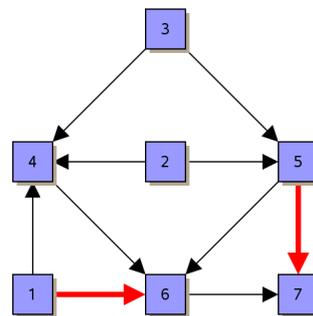
Knotenpositionierung



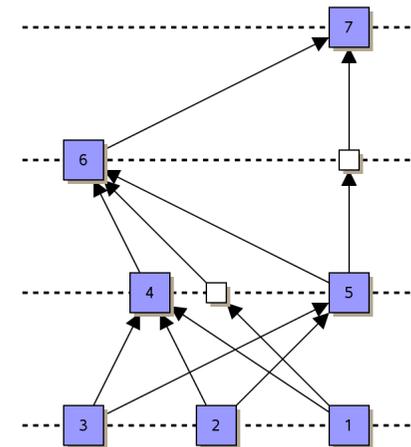
Kanten zeichnen



Eingabe



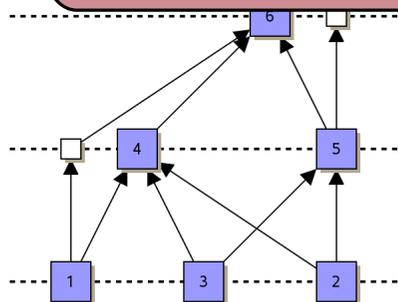
Kreise brechen



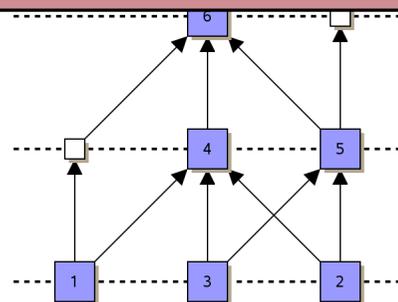
Lagenzuordnung

Jeder Schritt besteht aus algorithmischen Problemstellungen, die

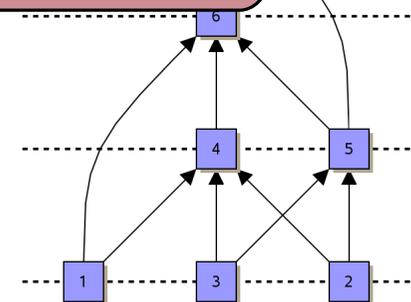
- verschiedene Lösungsansätze zulassen
- teilweise NP-schwer sind



Kreuzungsminimierung



Knotenpositionierung



Kanten zeichnen

1) Festlegung einer Visualisierungsaufgabe besteht aus:

- Festlegung eines Zeichenstils
- Festlegung von zu optimierenden Ästhetikkriterien
- Formulierung von (lokalen) Nebenbedingungen

2) Häufiges algorithmisches Vorgehen:

- Zerlegung in Teilschritte entlang der Festlegungen von Zeichenstil und Priorität der Ästhetikkriterien
- Lösung der einzelnen Schritte (heuristisch oder exakt)

Schritt 2) erfordert Schritt 1)

Schritt 1) erfordert häufig Hilfe von Anwendern