

Arrange Networks and Trees

Visual and Exploratory Data Analysis / Visualization of humanities data

May 16 / 23, 2023

Christian Knoll

Readings

Munzner, "Visualization Analysis and Design":

- Chapter 9 (Arrange Networks and Trees)
- Chapter 10 (Map Color and Other Channels)

Nobre, C., et al. "The state of the art in visualizing multivariate networks." Computer Graphics Forum. Vol. 38. No. 3. 2019.

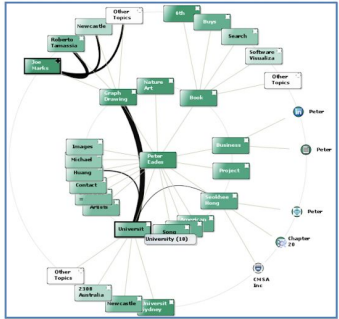
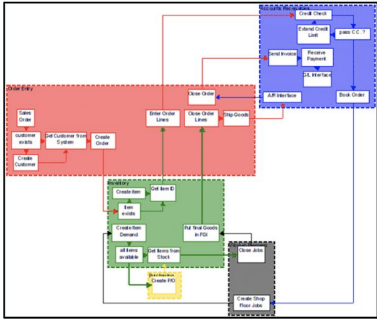
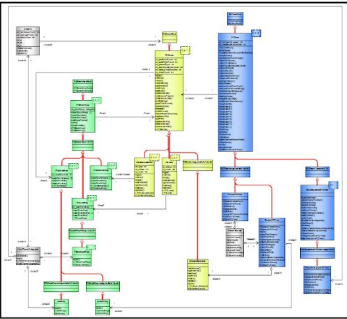
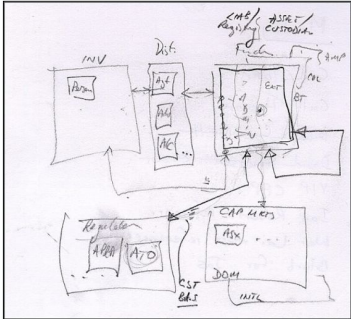
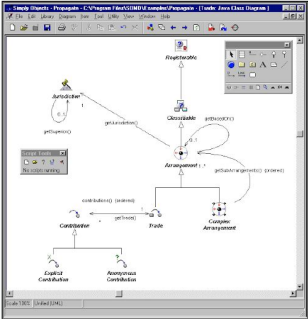
Beck, F., et al. "The State of the Art in Visualizing Dynamic Graphs." EuroVis (STARs). 2014.

Content

- **Introduction**
- Graph Theory
- Graph Drawing
- Dynamic Graphs
- Data Operations

Networks?
Network data?

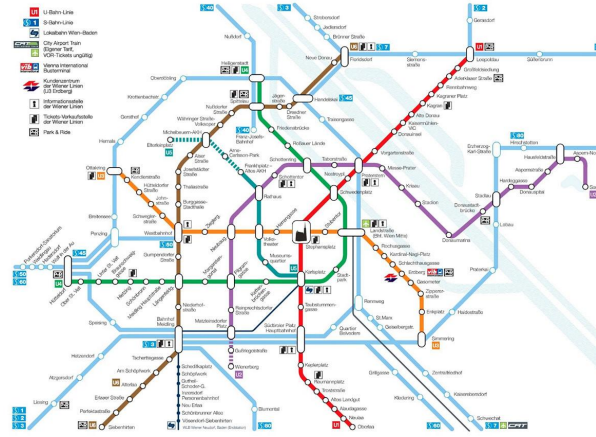
Networks and Graphs



etc.

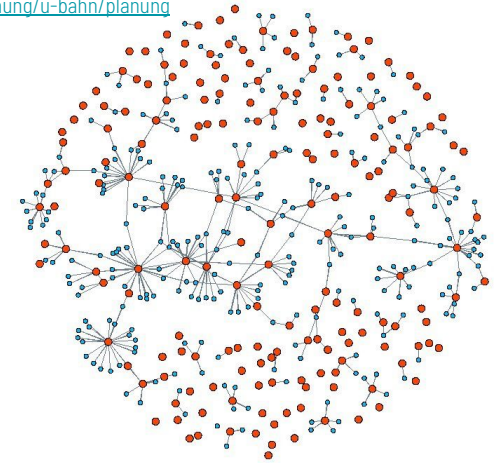
Applications

- Social network analysis
- Biological applications
- Software engineering
- Transportation networks
- Communication networks
- Security networks



SCHNELLVERBINDUNGEN IN WIEN

<https://www.wien.gv.at/stadtentwicklung/projekte/verkehrsplanung/u-bahn/planungsergebnis/> [January 6th, 2020]



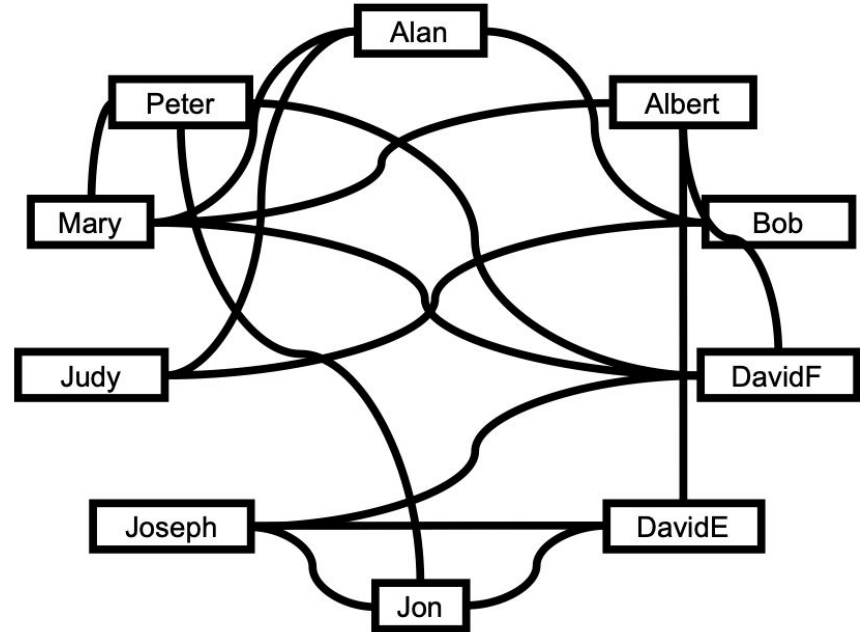
https://www.researchgate.net/figure/Social-Network-Graph-of-Festival-Host-Organizations-and-Partners_fig1_313389833 [January 6th, 2020]

Example

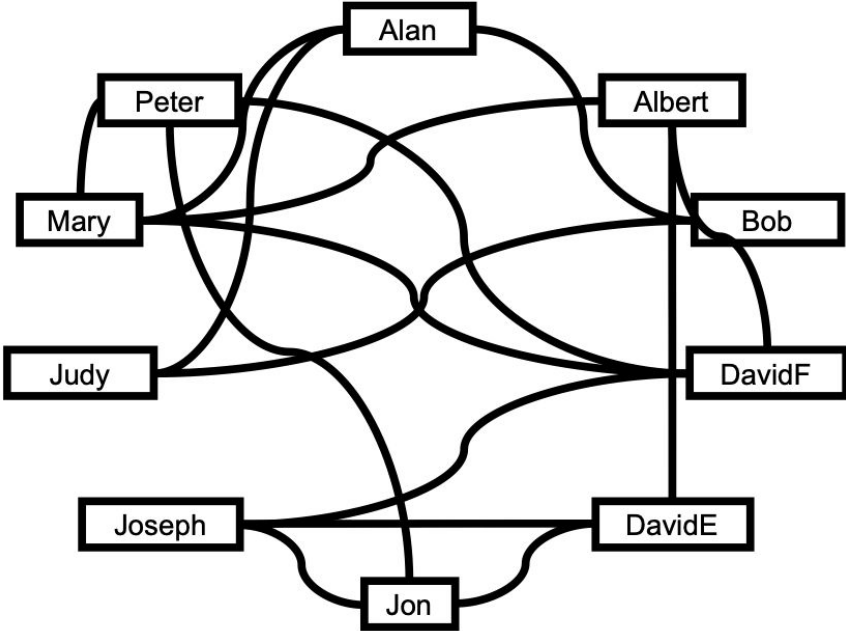
<i>x</i>	<i>Adjacent to X</i>
Mary	Peter, Albert, DavidF
Judy	Bob, Alan
Peter	Mary, DavidF, Jon
DavidF	Albert, Joseph, Peter, Mary
Jon	Peter, Joseph, DavidE
DavidE	Jon, Joseph, Albert
Joseph	DavidE, Jon, DavidF
Bob	Judy, Alan
Alan	Bob, Mary, Judy
Albert	DavidF, Mary, DavidE

Example

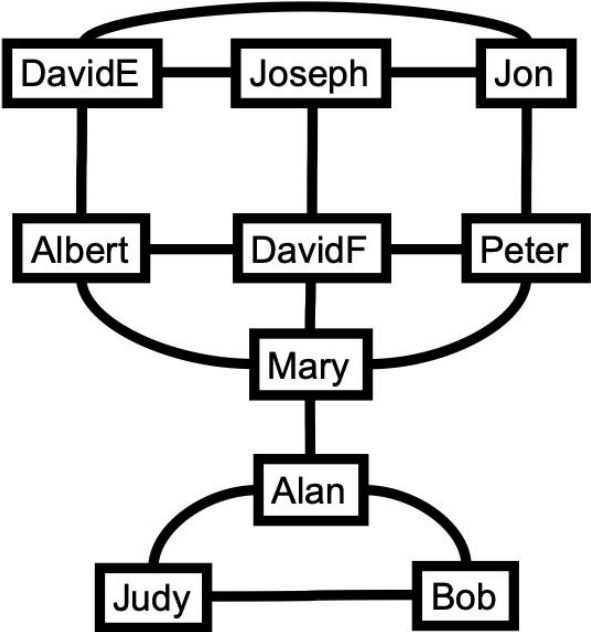
X	<i>Adjacent to X</i>
Mary	Peter, Albert, DavidF
Judy	Bob, Alan
Peter	Mary, DavidF, Jon
DavidF	Albert, Joseph, Peter, Mary
Jon	Peter, Joseph, DavidE
DavidE	Jon, Joseph, Albert
Joseph	DavidE, Jon, DavidF
Bob	Judy, Alan
Alan	Bob, Mary, Judy
Albert	DavidF, Mary, DavidE



Example



Confusing representation

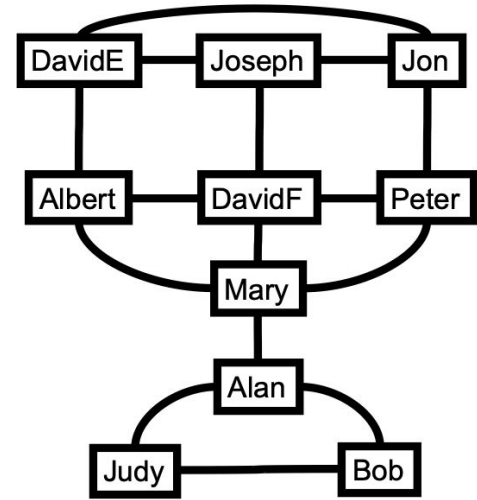


Much more clear!

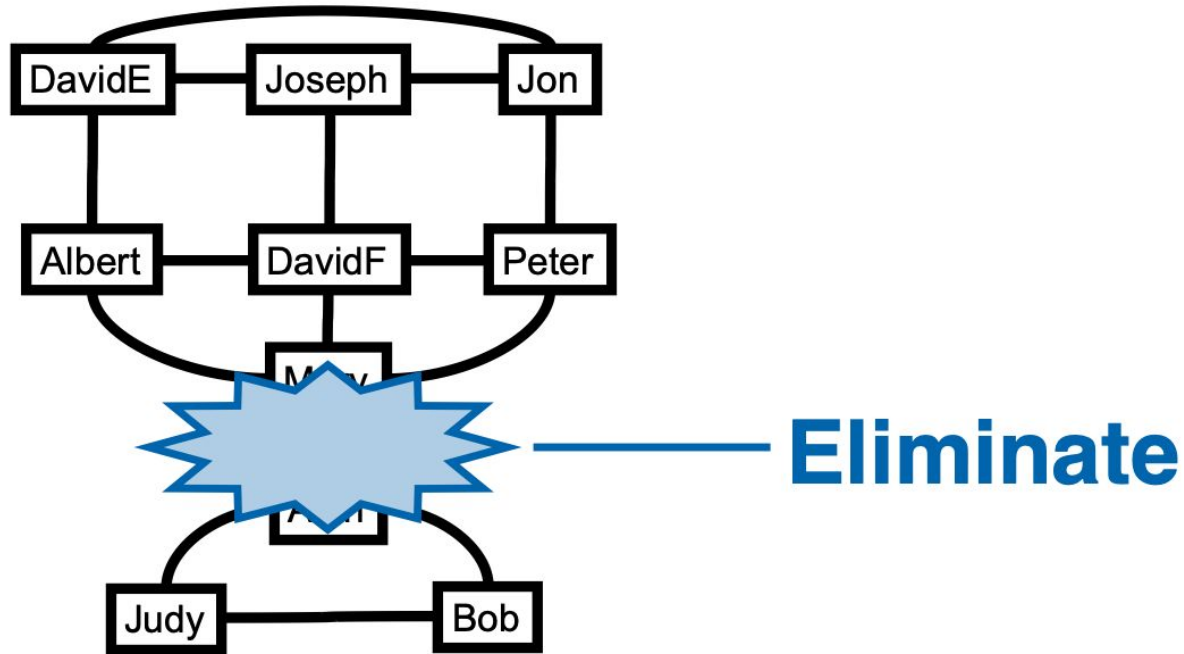
Topology Tasks: Examples

Finding

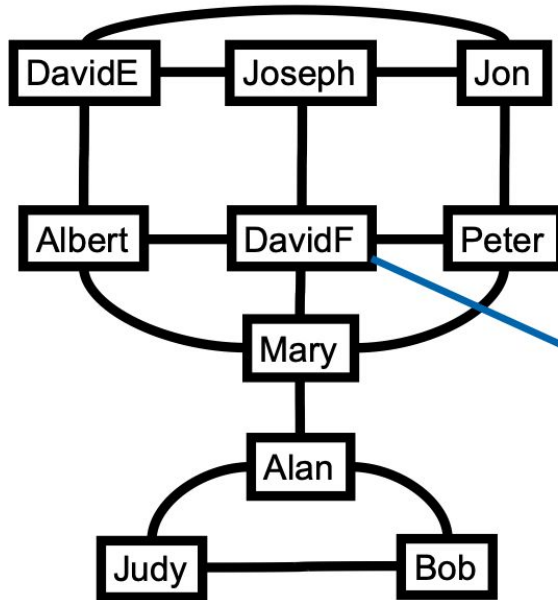
- all possible paths from one node to another
- the shortest path between two nodes
- the adjacent nodes one hop away from a target node
- all nodes that act as a bridge between two components that would otherwise be disconnected



Example: Terrorist network



Example: Mobile phone network

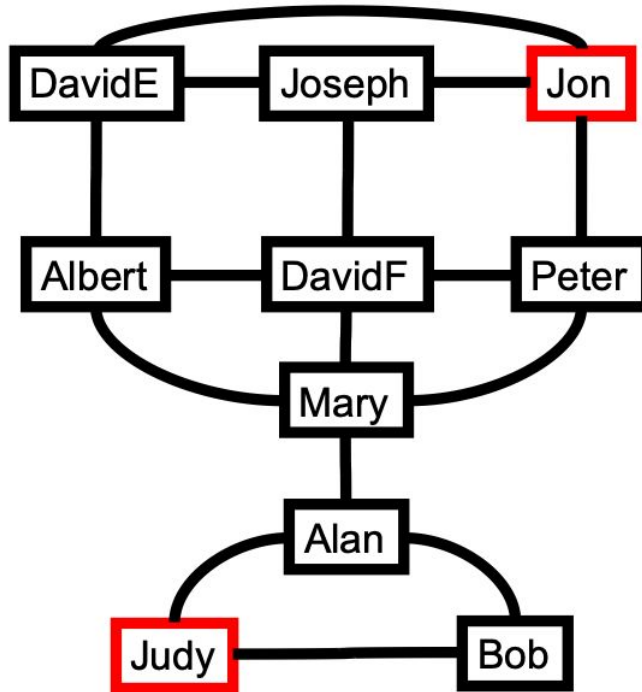


Nodes: people

Edges: phone calls

Good deal \$\$\$\$

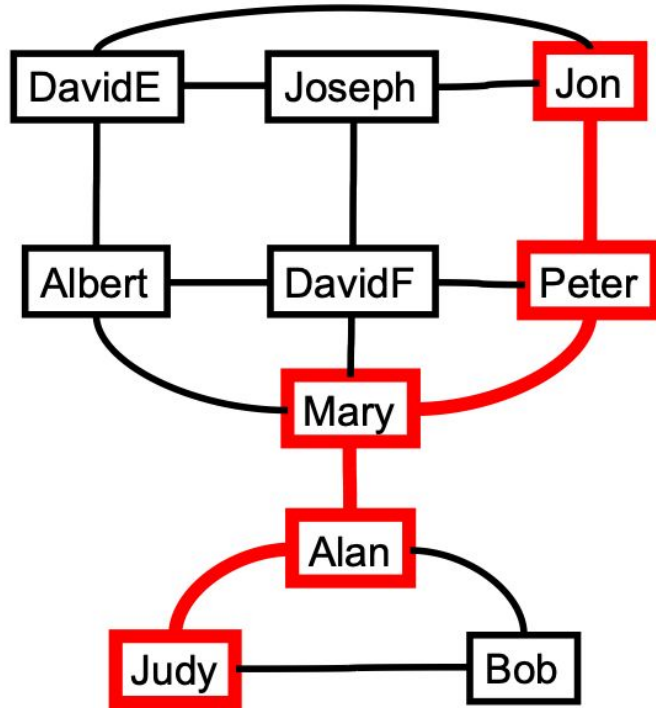
Example: Transport network



Nodes: places

Edges: train lines

Example: Transport network



Nodes: places

Edges: train lines

Shortest path?

Content

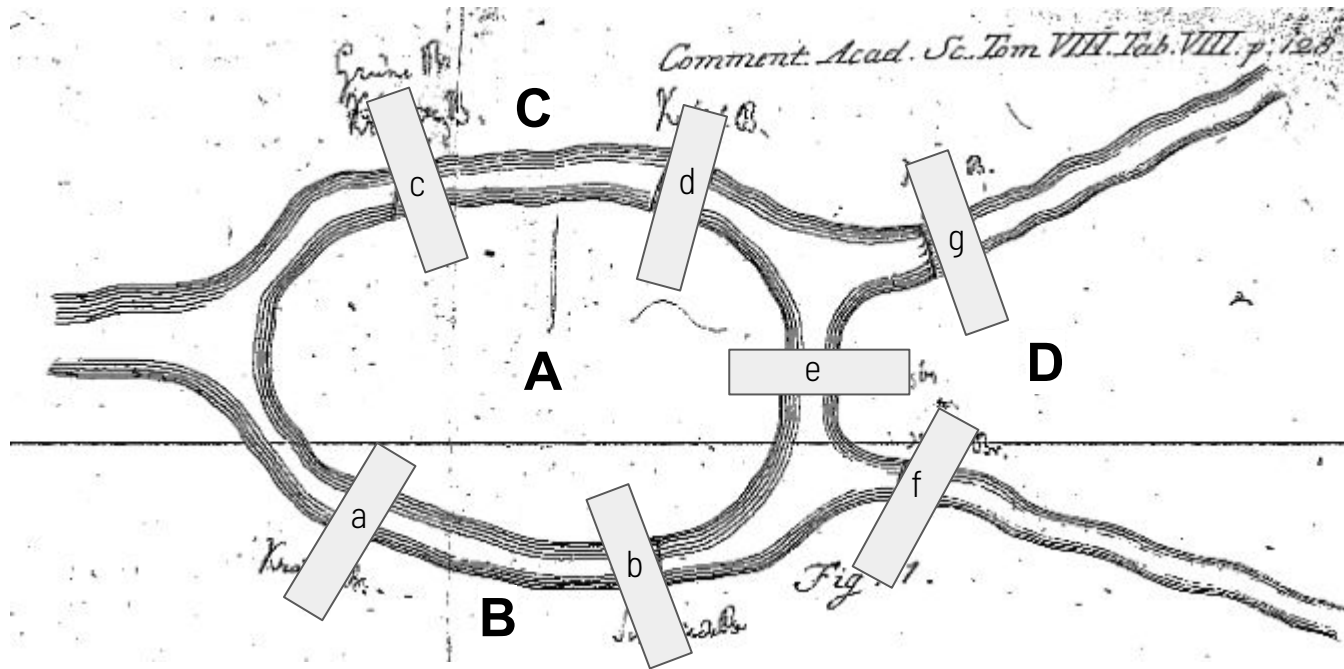
- Introduction
- **Graph Theory**
- Graph Drawing
- Dynamic Graphs
- Data Operations

Euler - The Seven Bridges of Königsberg

- The city has seven bridges
- Walk over each bridge
- Do not use a bridge twice
- A rather simple problem but both geometry and algebra could not solve it



Image of Königsberg <https://visualhistory.livejournal.com/39249.html> [January 6th, 2020]



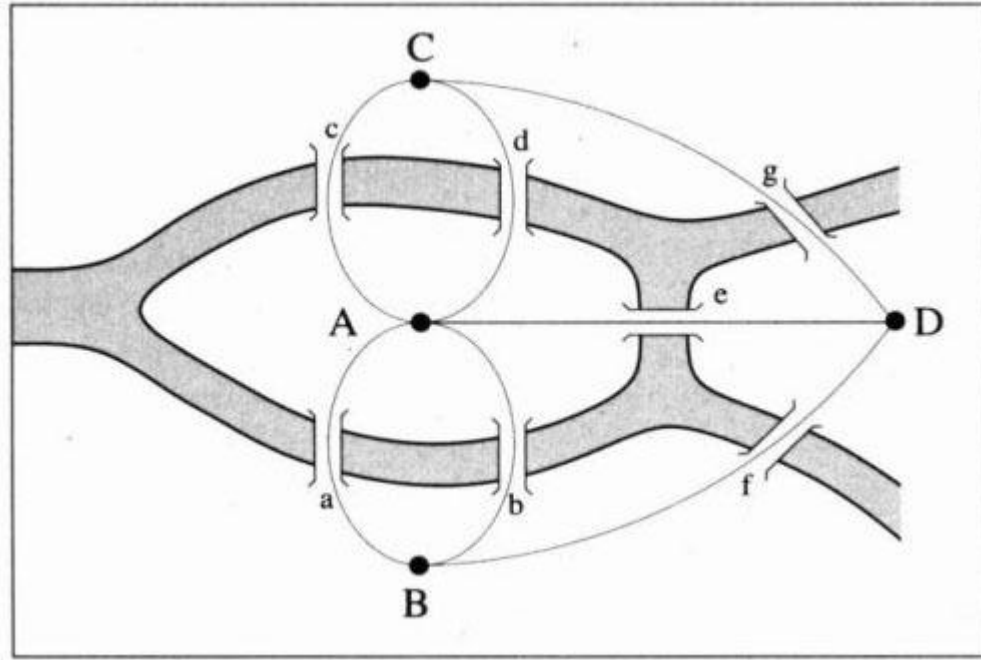
https://www.maa.org/sites/default/files/images/upload_library/46/1/old_convergence/Paoletti/Figure-2-perchance.png [January 6th, 2020]

- 4 areas (A, B, C, D)
- 7 bridges (a, b, c, d, e, f, g)

Graph Definition

- $G = (V, E)$
- **Vertices** (nodes) V
 - Geometric attributes e.g. position and size
 - Graphical attributes e.g. color
- **Edges** (lines) $E = (v_i, v_j)$ where $v_i, v_j \in V$
 - Geometric attributes e.g. route
 - Graphical attributes e.g. color and linestyle

- 4 vertices
- 7 edges



Degree of a node =
Number of edges touching it

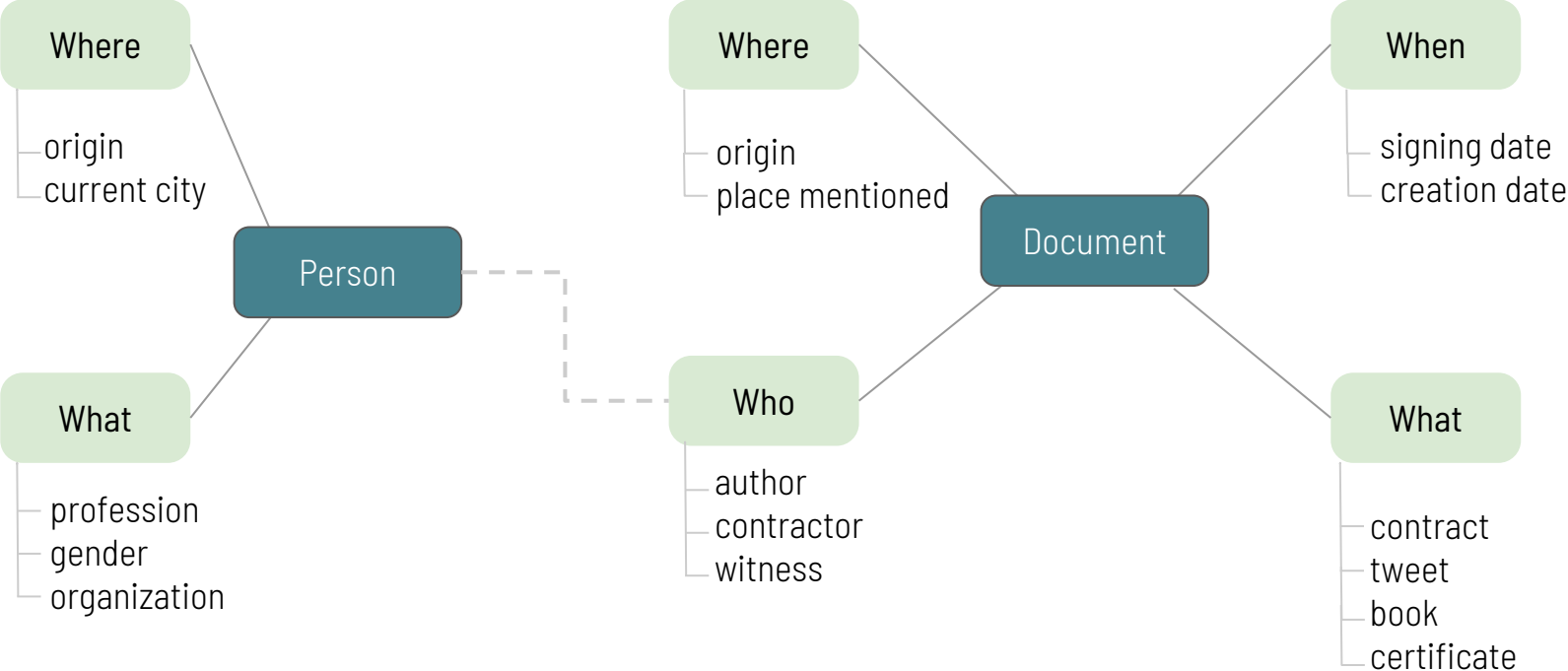
<https://maffsisphun.wordpress.com/2018/02/17/the-konigsberg-bridge-problem/> [January 6th, 2020]

Euler proved that there is no solution to this problem
but it laid the foundation for graph theory!

Content

- Introduction
- Graph Theory
- **Graph Drawing**
- Dynamic Graphs
- Data Operations

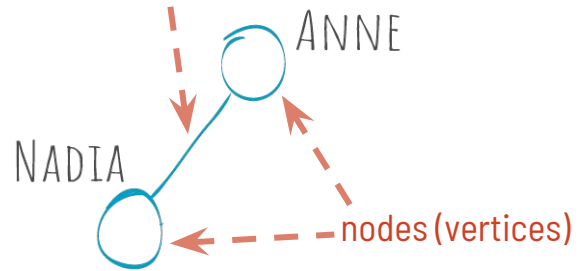
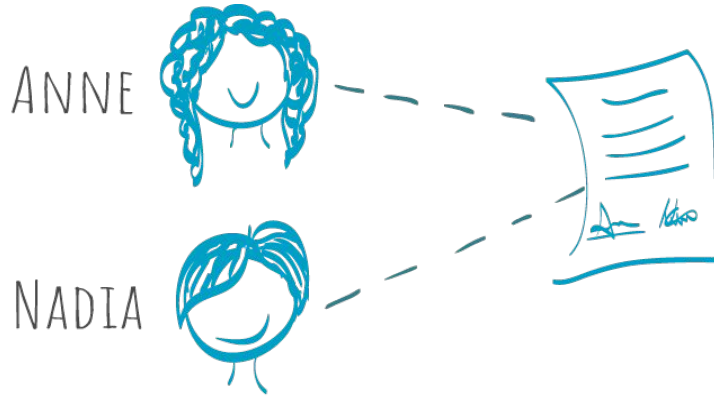
Example: Historical data



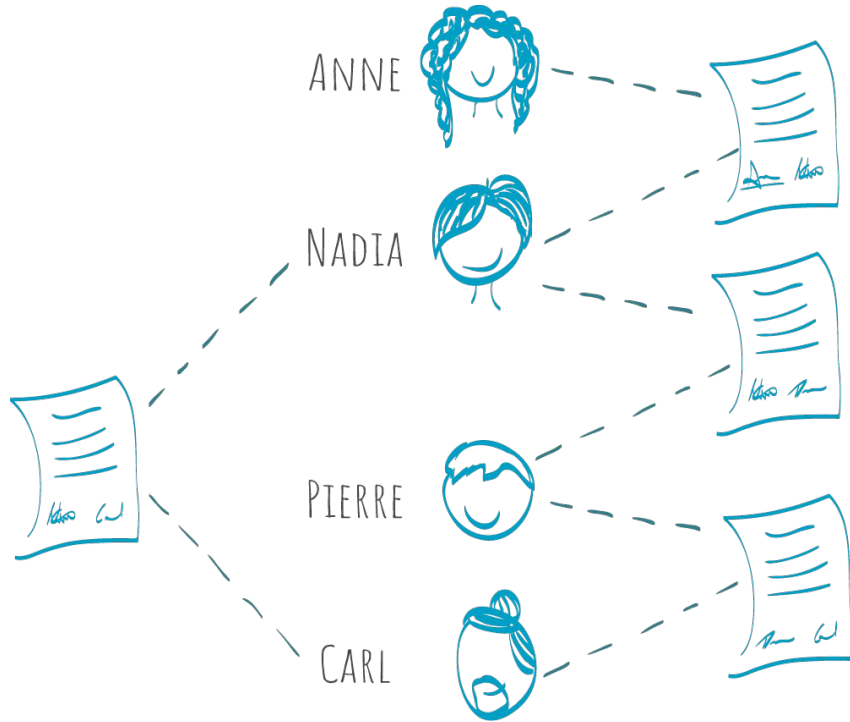
Contracts: who signed a contract with whom?

represent it as a graph

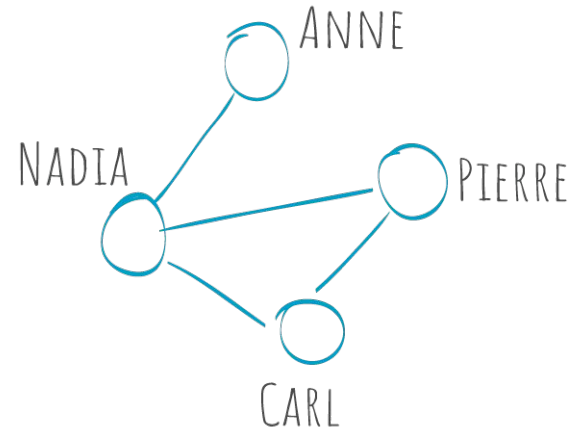
link (edge) between 2 nodes



Contracts: who signed a contract with whom?

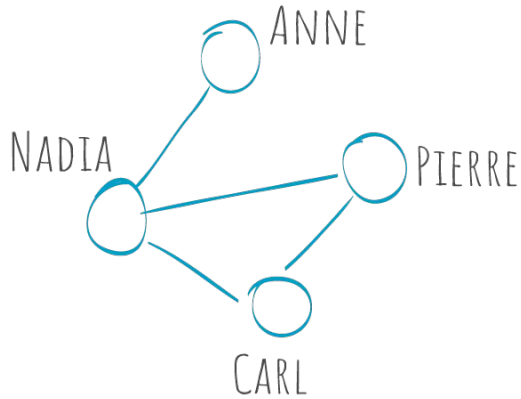


Node-link diagrams



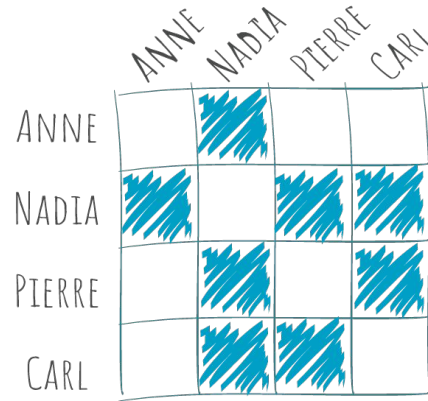
Contracts: who signed a contract with whom?

- Node-link diagrams



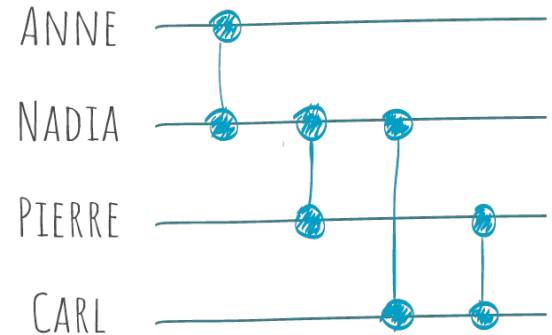
Node-link layout

- Matrix



Tabular layout

- Biofabric



Multivariate Networks



- gender: female
- profession: shopkeeper
- hometown: London
- age: 42



- gender: female
- profession: designer
- hometown: Paris
- age: 33



- gender: male
- profession: shopkeeper
- hometown: Rome
- age: 47



- gender: male
- profession: photographer
- hometown: Madrid
- age: 36

People have attributes -
how to visualize them?

Mapping people attributes to visual attributes of nodes (On-node encoding)

- Color intensity

- Node size



• age: 43



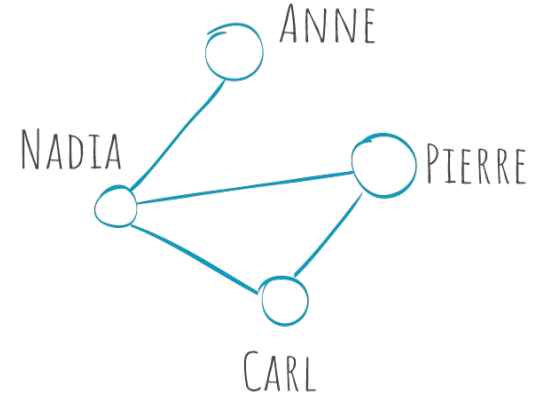
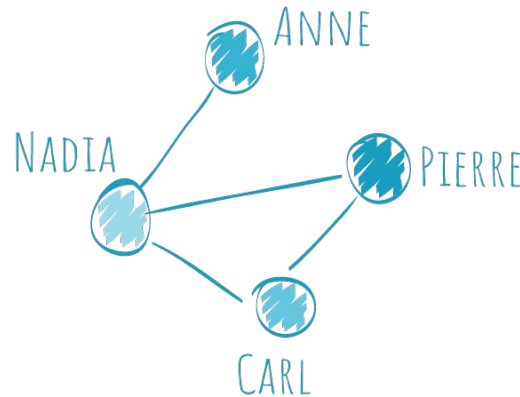
• age: 30



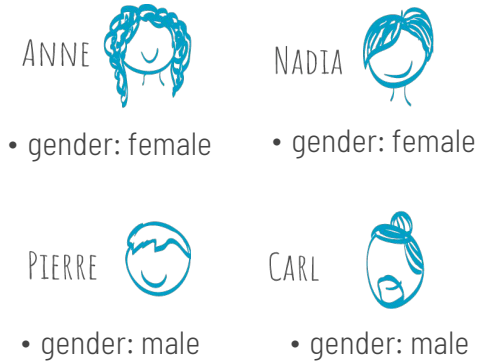
• age: 50



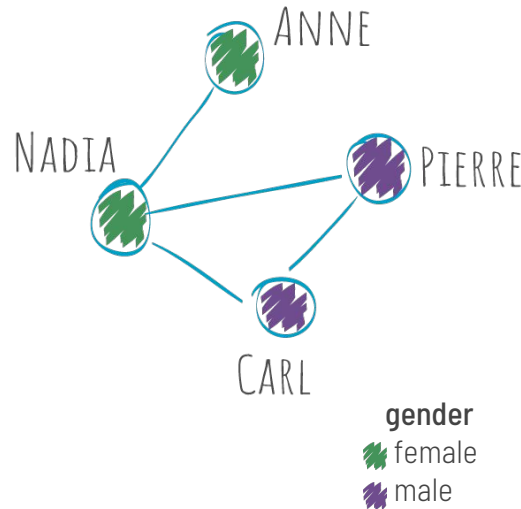
• age: 37



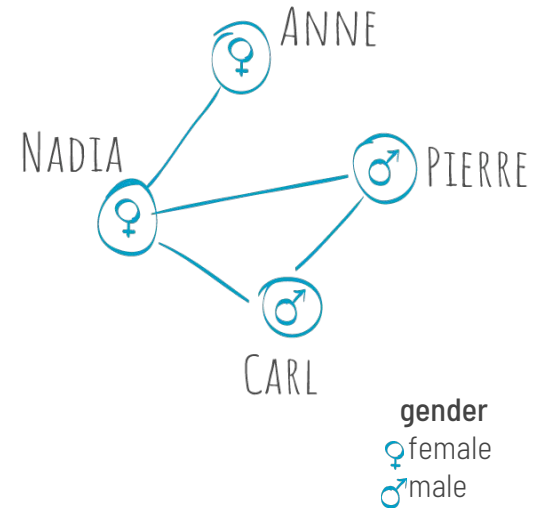
Mapping people attributes to visual attributes of nodes (On-node encoding)



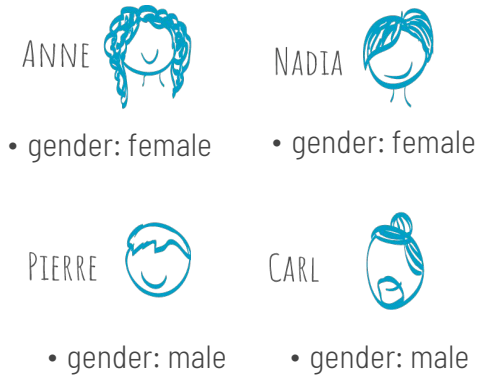
- Color



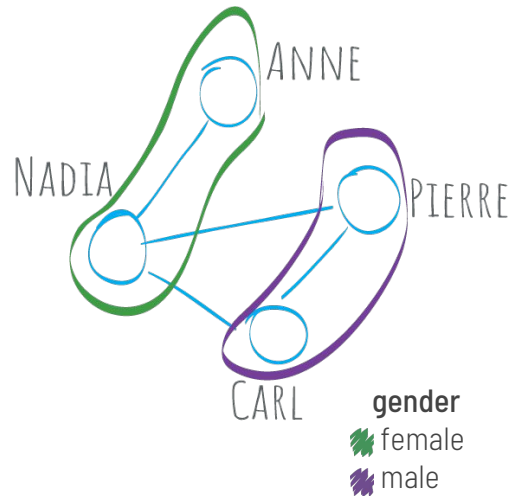
- Symbols



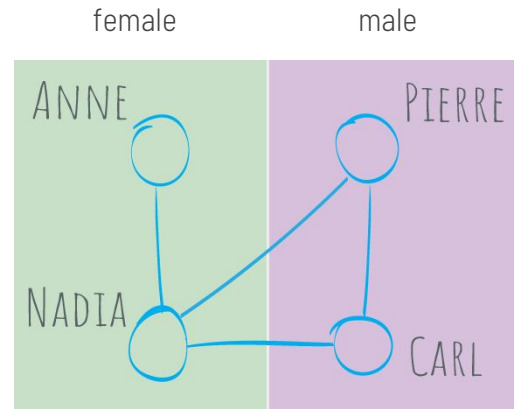
Mapping people attributes



- Contour overlay



- Attribute-driven faceting



Mapping people attributes

- Geographical attributes (Attribute-driven positioning)



• hometown: London

• hometown: Paris

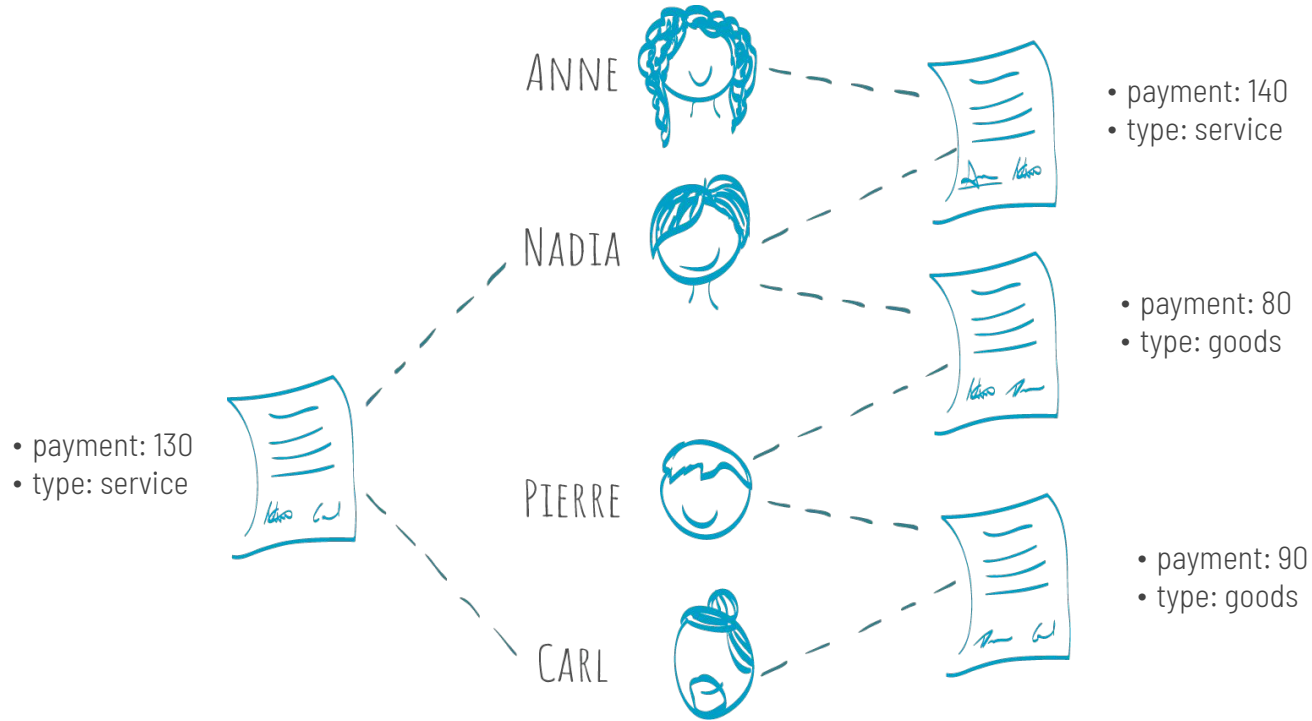


• hometown: Rome

• hometown: Madrid

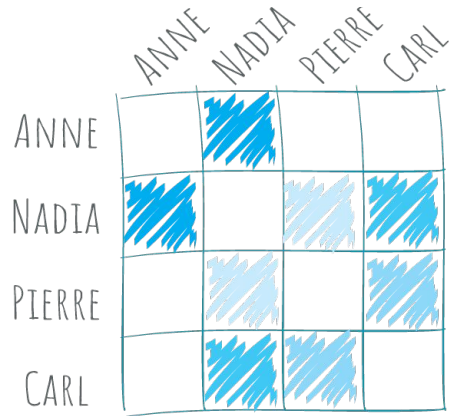
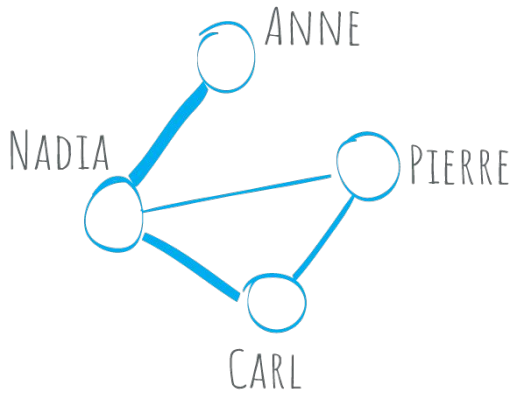


Contracts have attributes

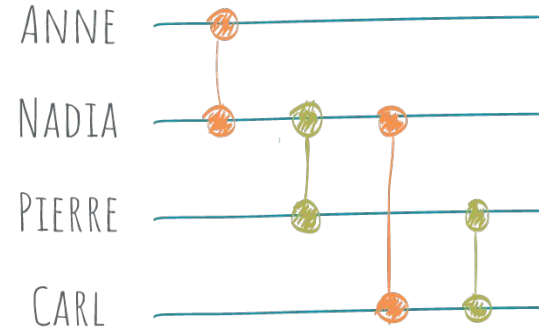


Contracts: how to visualize contract's attributes? (On-edge encoding)

- Edge attributes



payments

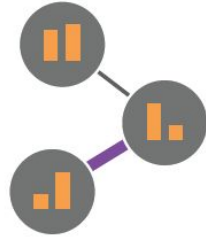


type
service
goods

type

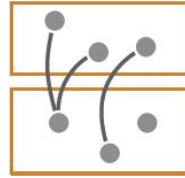
Comparison of Node-Link Layouts

Topology-Driven Layout

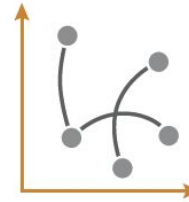


On-Node / On-Edge
Encoding

Attribute-Driven Layouts



Attribute-Driven
Faceting



Attribute-Driven
Positioning

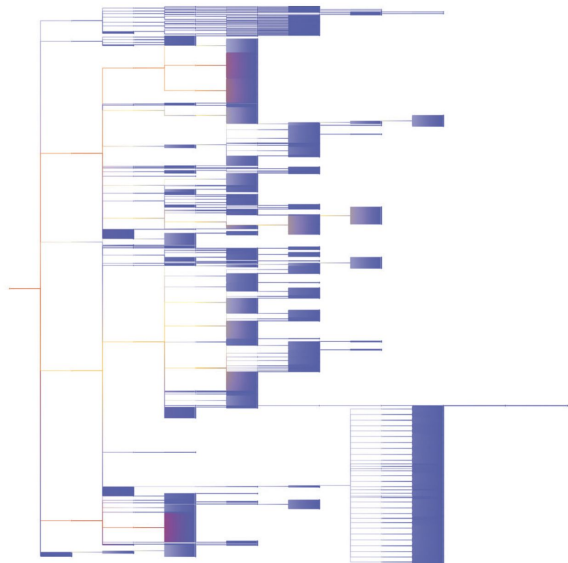
Summary:	Modifying visual appearance of a node or an edge or embedding marks in a node	Grouping nodes according to one or more attributes and places the elements of a group in a shared region	Assigning node or edge positions according to one or more attribute values
Layout:	<ul style="list-style-type: none"> • Free layout (e.g. force-directed or orthogonal) • Styled layout (position based on pattern, e.g. grid or radial) 	<ul style="list-style-type: none"> • Relaxed fixed layout 	<ul style="list-style-type: none"> • Fixed layout

Connection vs. Containment

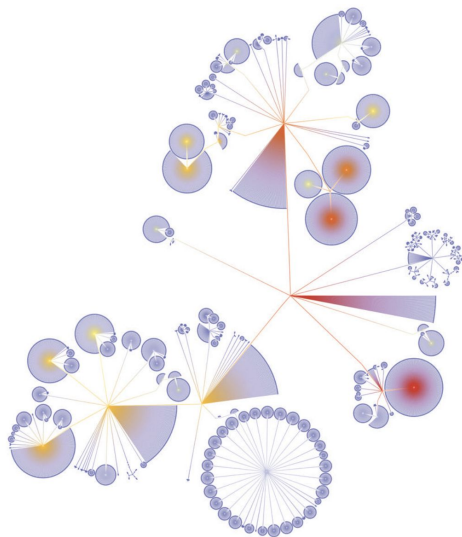
- Relevant for drawing trees
- Connection -- traditional view of graphs
- Containment -- essentially treemaps
- Type of information is different:
 - Connection marks: Only show pairwise relationships between two items
 - Containment marks: Show complete information about hierarchical structure

Connection vs. Containment

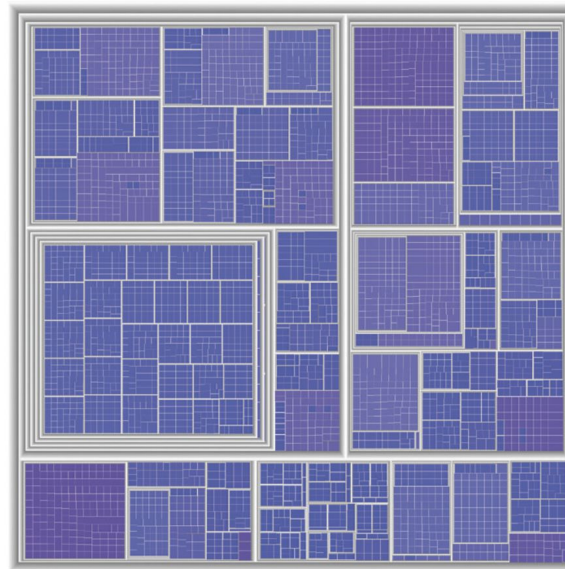
Dataset: 5161-node tree (computer file system)



Rectangular horizontal node-link layout
(connection)



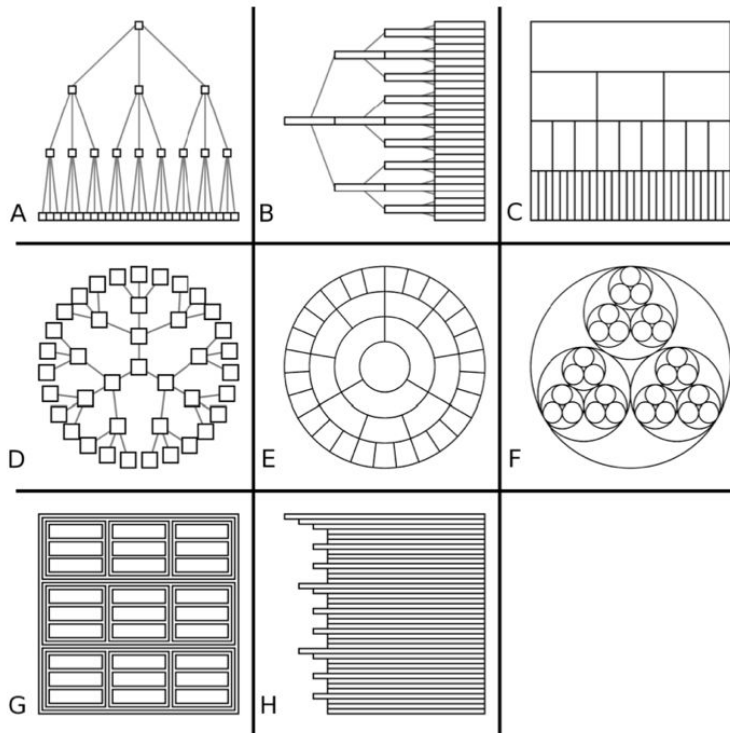
BubbleTree node-link layout
(connection)



Treemap showing hierarchical structure
with containment rather than connection

Connection vs. Containment

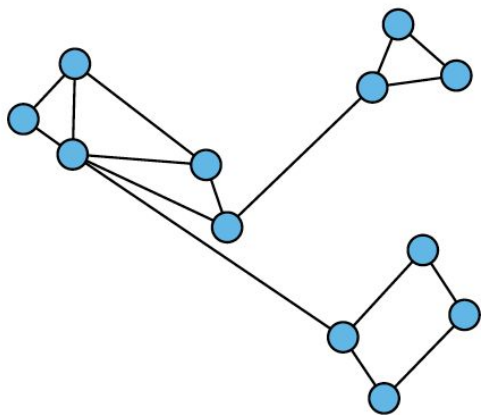
Eight visual encodings of the same tree dataset



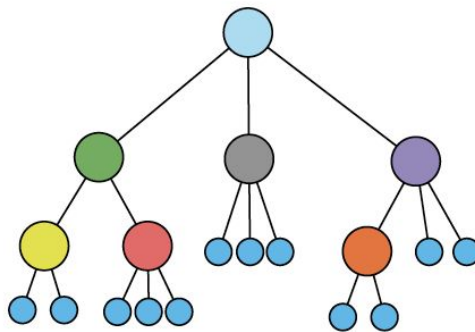
Connection vs. Containment

Compound network: Combination of a network and tree

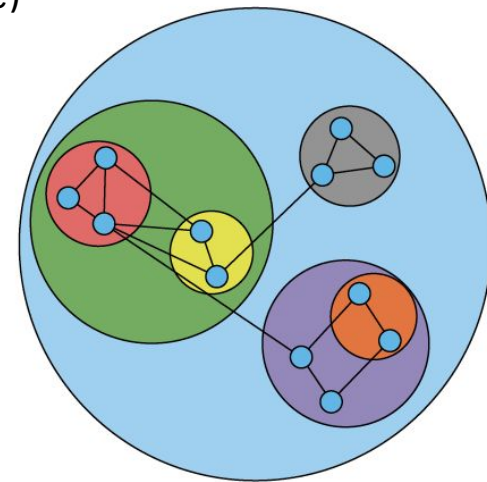
Example: GrouseFlocks (uses containment to show graph hierarchy structure)



(a) Original graph



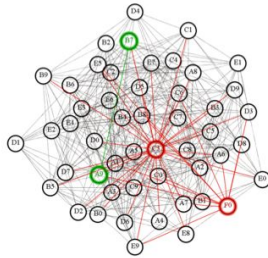
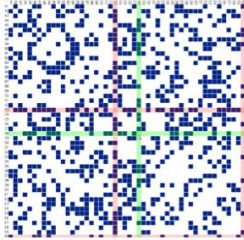
(b) Cluster hierarchy built atop the graph, shown with a node-link layout.



(c) Network encoded using connection, with hierarchy encoded using containment.

Node-link or Matrix?

Node-link or Matrix?



- Study
 - 36 users
 - 9 networks
 - 7 tasks → measure time & errors
- Results
 - **Node-link** only for:
 - small graphs (~20 nodes)
 - path finding tasks
 - Else **Matrix**
- Limitations

*Ghoniem et al. (InfoVis 2004).
A Comparison of the Readability of
Graphs Using Node-Link and
Matrix-Based Representations.*

Node-link or Matrix?

Okoe et al. (2019).

Node-Link or Adjacency Matrices:
Old Question, New Insights

Crowdsourced evaluation: 864 participants

“We found that NL is better than AM for questions about network topology, connectivity, and memorability tasks, while AM outperforms NL for group tasks.”

Ren et al. (2019).

Understanding node-link and matrix visualizations of networks:
A large-scale online experiment

Large-scale study: 600 participants

“[...] results show that node-link representations produced a better implicit understanding of the data, with higher response accuracy and faster completion times than the matrix representations (23% and 19% less time than MatrixDegree and MatrixGroup, respectively). This result is consistent with prior work (Henry & Fekete, 2007; Ghoniem et al., 2005). For the larger social network graph, the performance difference between matrix and node-link representations was reduced, with node-link still performing best overall. [...] our study results show that node-link representations are more efficient analysis tools than matrix representations for the majority of smaller-scale social network graph analysis tasks, with the exception of those that require explicit sorting of nodes.”

Node-link or Matrix?

	Pros	Cons
Node-link:	<ul style="list-style-type: none">● Most popular way of visualizing networks● Extremely intuitive for small networks● Especially useful for understanding topological structure	<ul style="list-style-type: none">● Limit of network size and link density: hard to read if exceeded (“hairball”)
Matrix:	<ul style="list-style-type: none">● Perceptually scalable for large and dense networks● Predictable screen space● Adding a new item will only cause a small visual change● Support for reordering	<ul style="list-style-type: none">● Lack of support for investigating topological structure● Unfamiliarity: Users may need training to interpret the matrix

Graph Drawing

The classical graph drawing problem is to develop **algorithms to draw graphs nicely**

graph

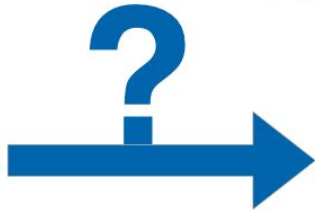
A - B, C, D

B - A, C, D

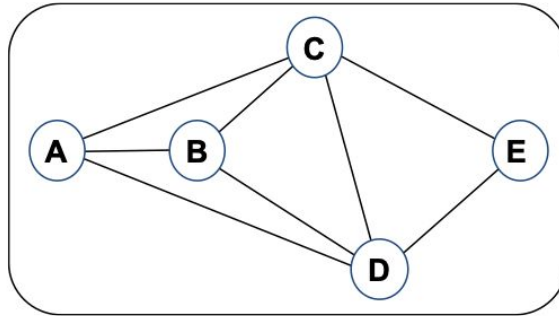
C - A, B, D, E

D - A, B, D, E

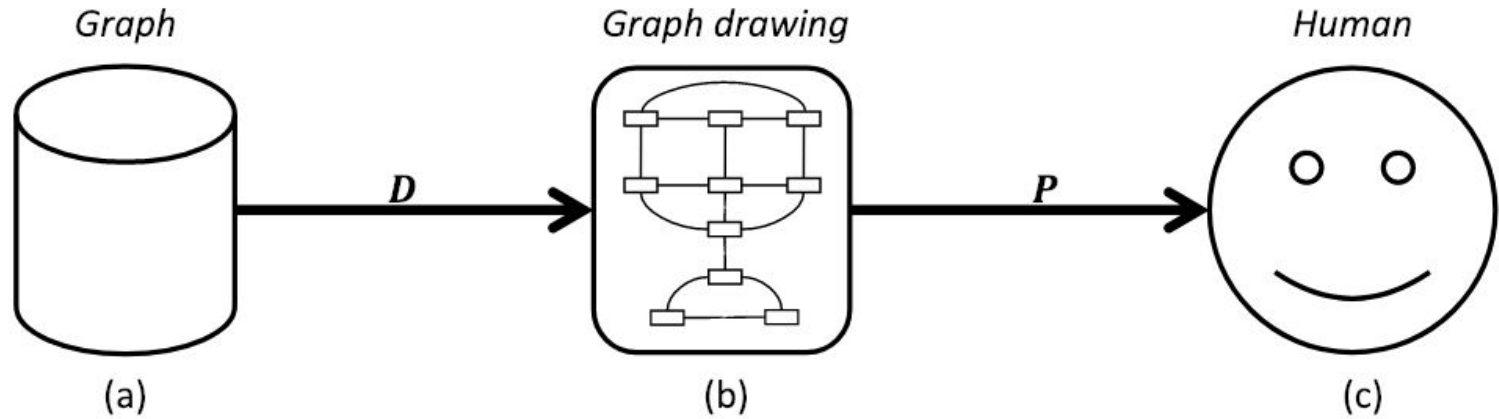
E - C, D



nice graph drawing



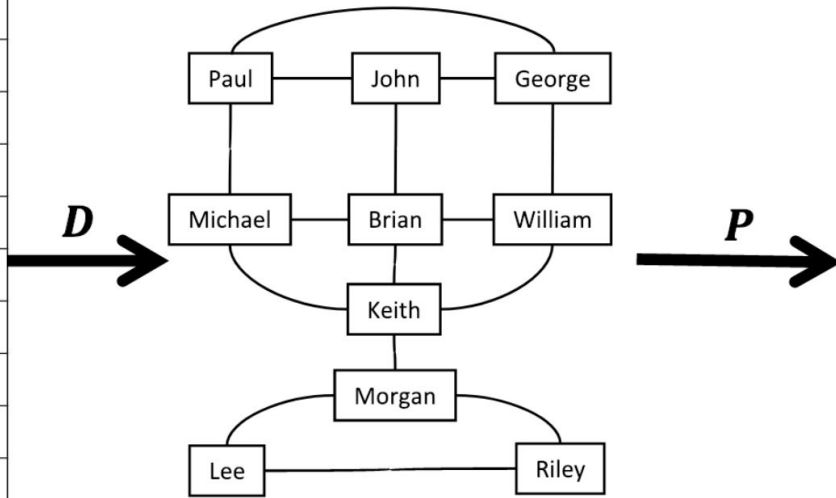
Graph Visualisation Pipeline



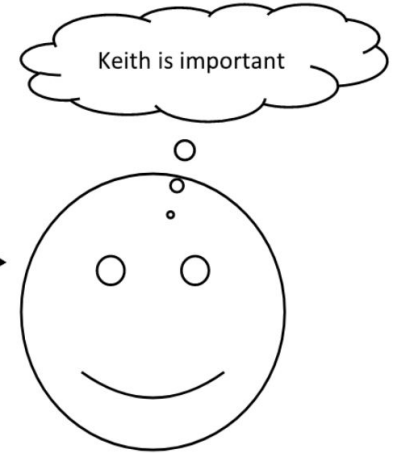
Graph Visualisation Pipeline: Example

Brian	Keith, John, Michael, William
George	John, Paul, William
John	Brian, George, Paul
Keith	Brian, Michael, Morgan, William
Lee	Morgan, Riley
Michael	Brian, Keith, Paul
Morgan	Keith, Lee, Riley
Paul	George, John, Michael
Riley	Lee, Morgan
William	Brian, George, Keith

(a)



(b)



(c)

Faithfulness

- **Quality** of the **drawing function D**
- The drawing function D is **faithful** if it **uniquely represents** the graph G
- Significant for large graphs as faithfulness is usually given in small graphs
- If **edge bundling** occurs the drawing function no longer uniquely represents the graph

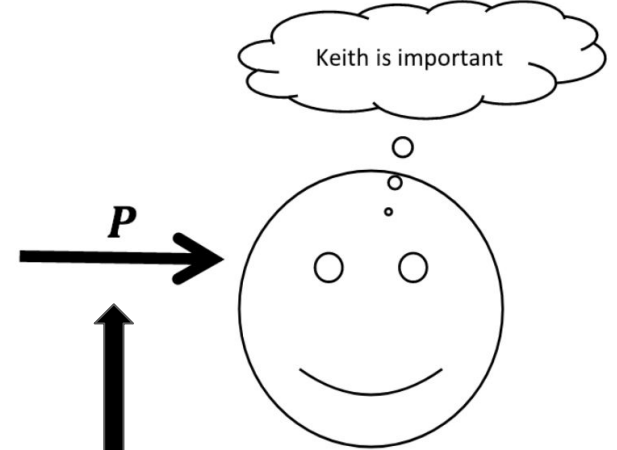
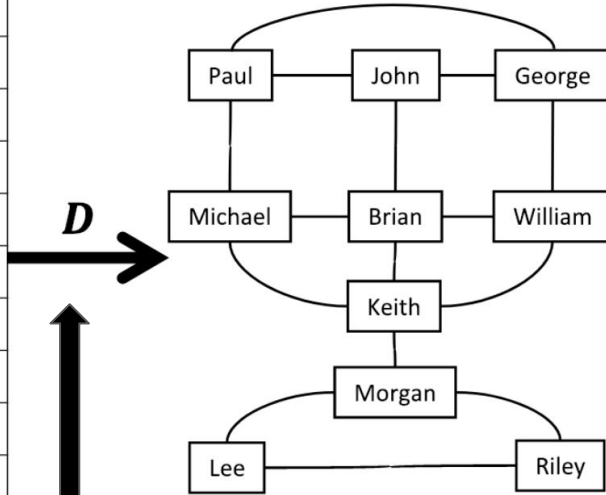
Readability

- **Quality** of **perception function P**
- How well can we as a **human understand** the picture of the graph
- **Minimize** the number of edge **crossings** and edge **bends**
- First based on intuition later measured through HCI-style experiments
 - Measure task completion time e.g. tracing a shortest path

Graph Visualisation Pipeline: Example

Graph G

Brian	Keith, John, Michael, William
George	John, Paul, William
John	Brian, George, Paul
Keith	Brian, Michael, Morgan, William
Lee	Morgan, Riley
Michael	Brian, Keith, Paul
Morgan	Keith, Lee, Riley
Paul	George, John, Michael
Riley	Lee, Morgan
William	Brian, George, Keith



(a)

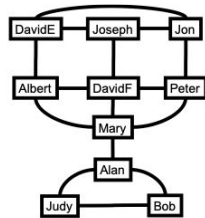
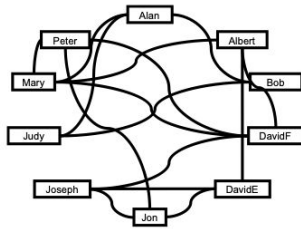
Drawing function D
Faithfulness

(b)

Perception function P
Readability

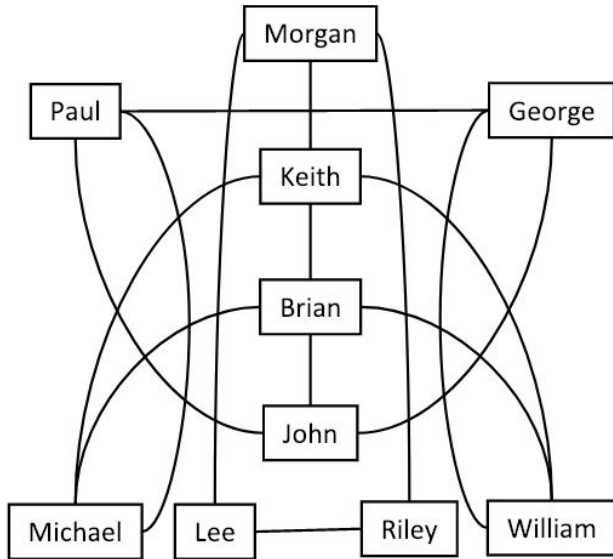
(c)

Quality measures for networks

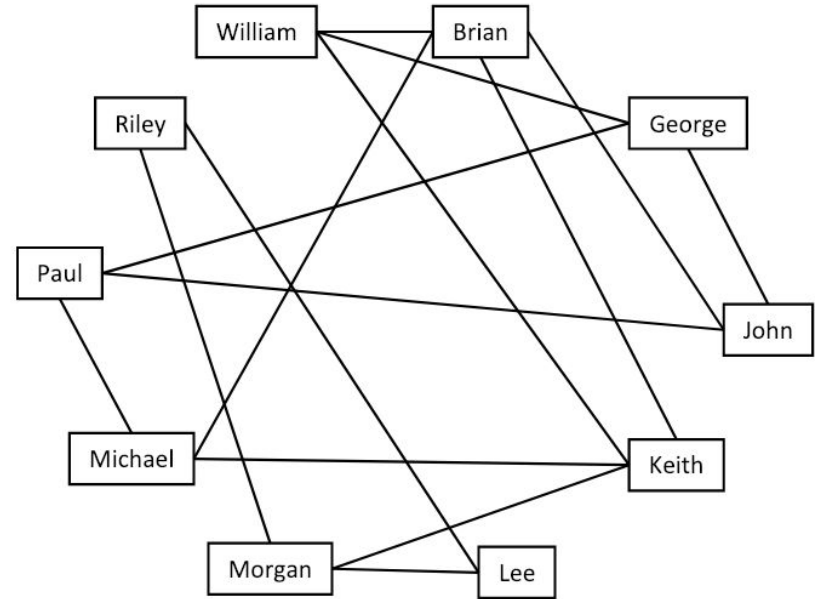


- Classical quality measures
 - minimize edge crossings
 - minimize bends
 - ...
- Human subject experiments found crossings & bends to be most important *wrt* readability
 - Purchase et al., 1997
 - Ware et al 2002
 - Huang et al 2004

Poor Readability (bends & crossings)

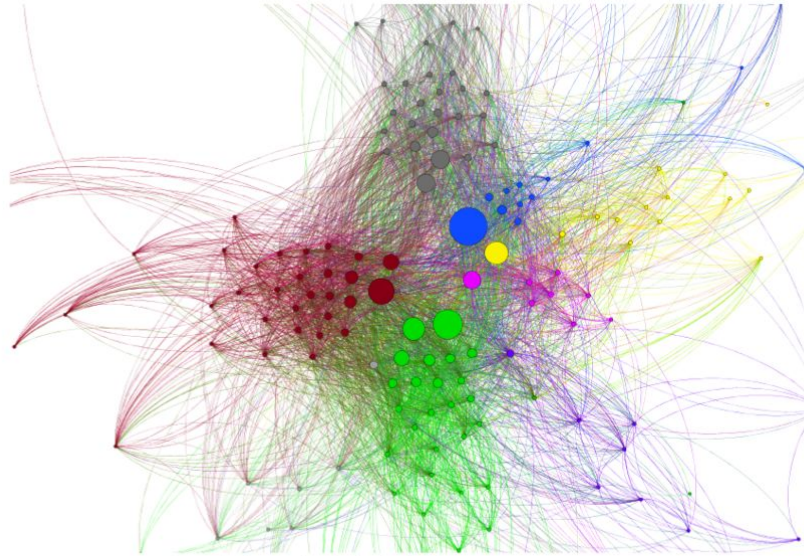


(a)



(b)

Showing all the data?



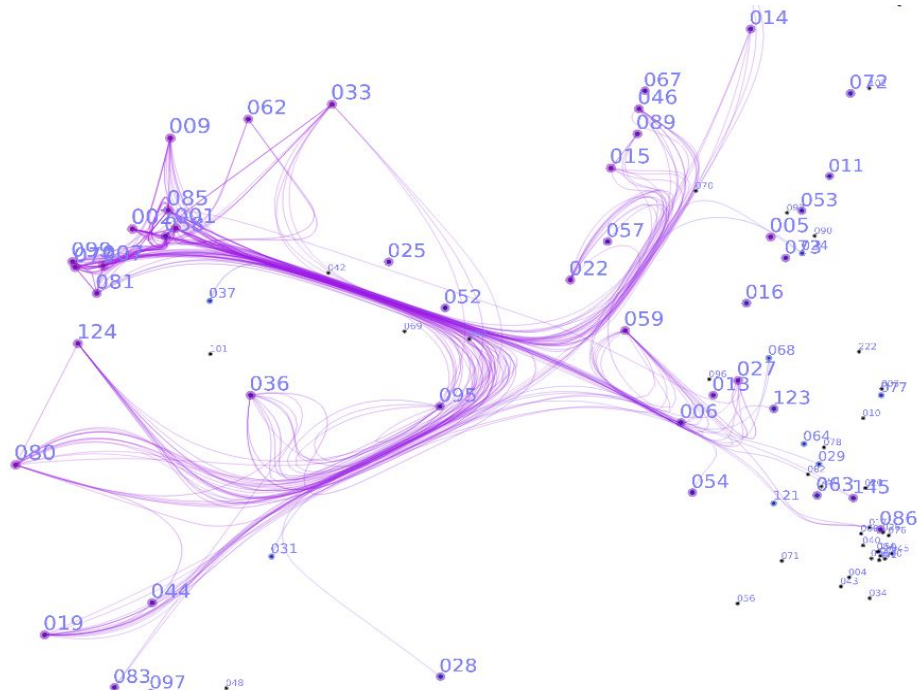
<https://twitter.com/axelmaireder/media>



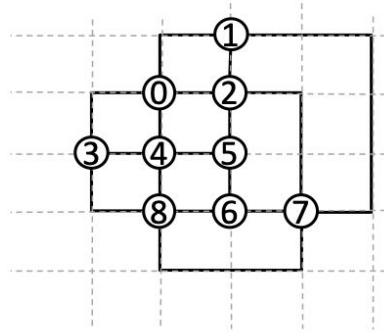
Hairball

Unfaithful Graph

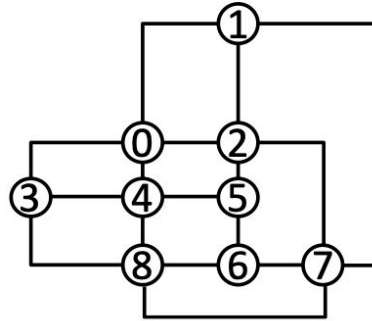
- The graph drawing to the right is unfaithful because multiple graphs G could be represented by this graph drawing
- The use of edge bundling makes it impossible to see individual edges and their distinct connections
- [Example](#)



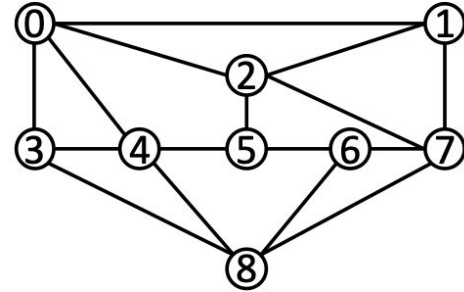
Different kinds of layouts



grid-based



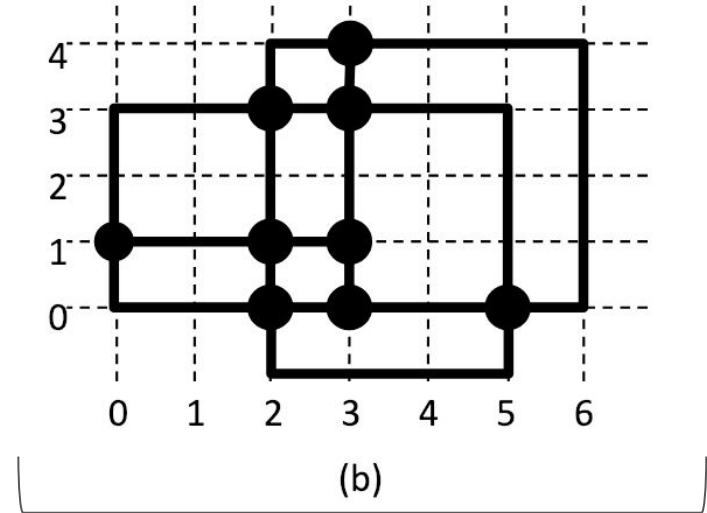
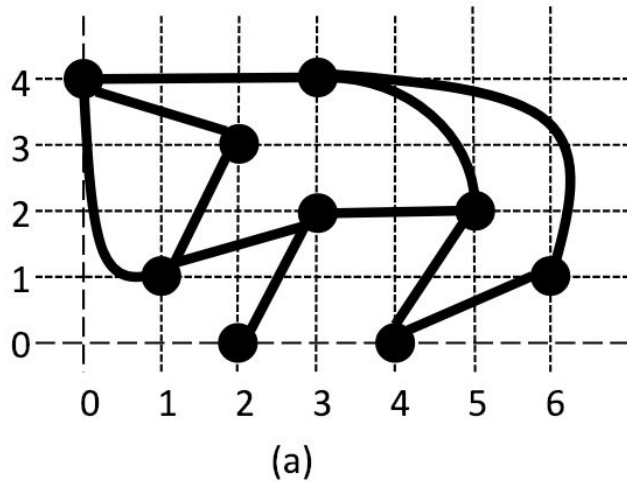
orthogonal



straight-line

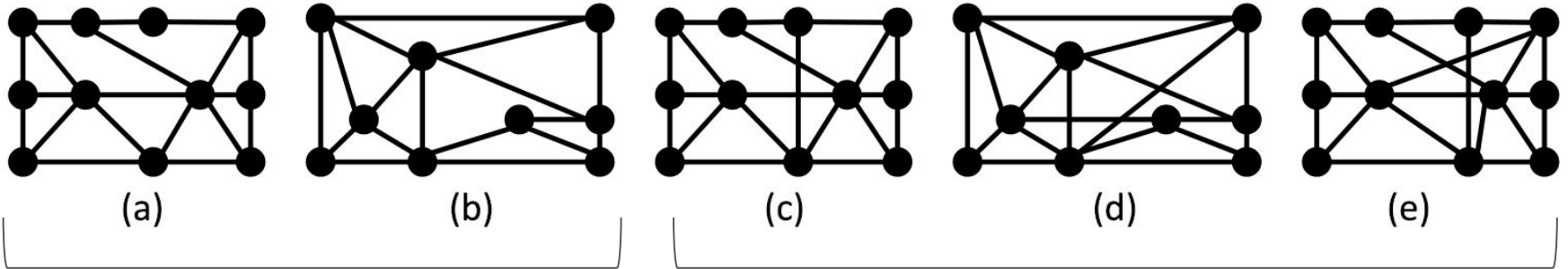
Orthogonal graph drawing

- A graph drawing is **orthogonal** if each edge is a polyline consisting of vertical and horizontal line segments



Planar graphs

- A drawing of a graph is **planar** if it has no edges crossing; a graph G is planar if there exists a planar drawing of G .
- Adding vertices at edge crossings can convert a non-planar graph to a planar graph.

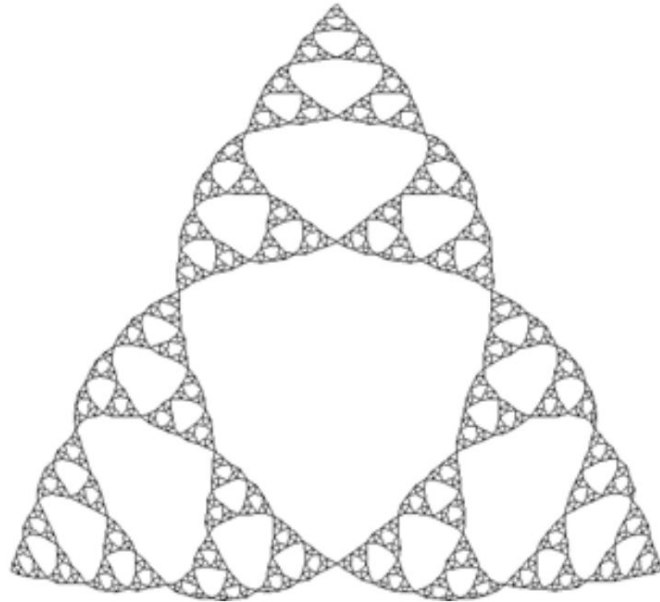


Planar drawing

Non-planar drawing
((c) is planar, (d) and (e) are non-planar)

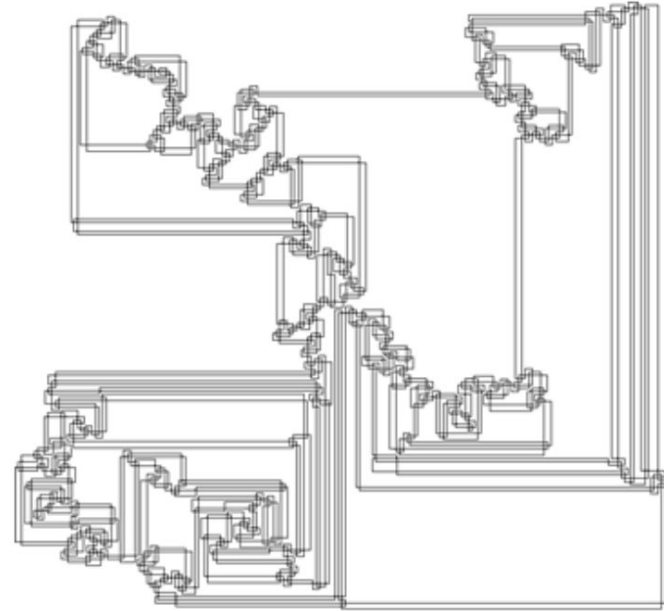
Energy-based vs Topology-shape-metrics

Drawing of a Sierpinski triangle, a fractal defined as a recursively subdivided triangle.



(a)

Energy-based method



(b)

Topology-shape-metrics approach

Energy-based

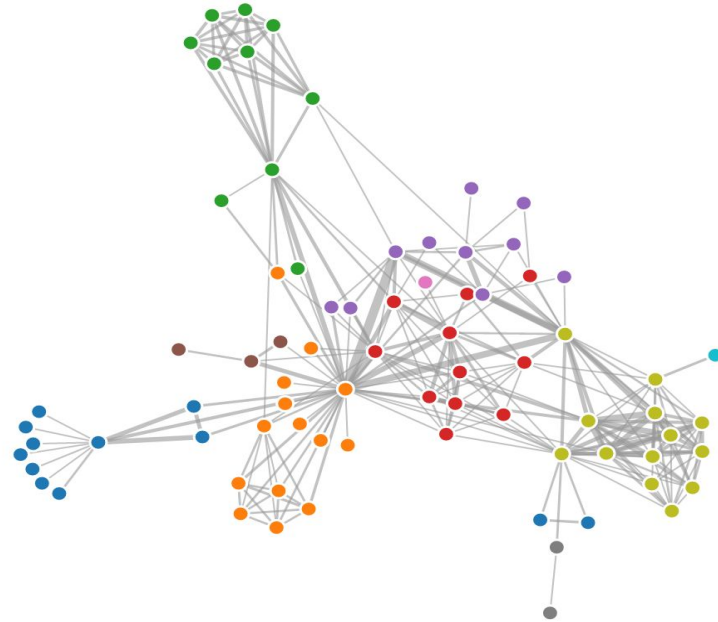
- Model the graph as a **system of objects** that contribute to the overall **energy** of the system (e.g. edges act like springs)
- A **low energy state** of the system corresponds to a good drawing
- Two components:
 - Model of objects and their interactions (**virtual physical model**)
 - Algorithm to compute energy minimum (**energy minimization method**)
- Often aesthetically pleasing

Topology-shape-metrics

- **1. Topology:** Compute topological arrangement of vertices, edges and faces (aim for small number of edge crossings)
- **2. Shape:** Then compute the general shape of each edge (aim for small number of edge bends)
- **3. Metrics:** Finally compute the precise location of each vertex, each edge bend and each edge crossing (aim for drawing with high resolution)

Force directed models

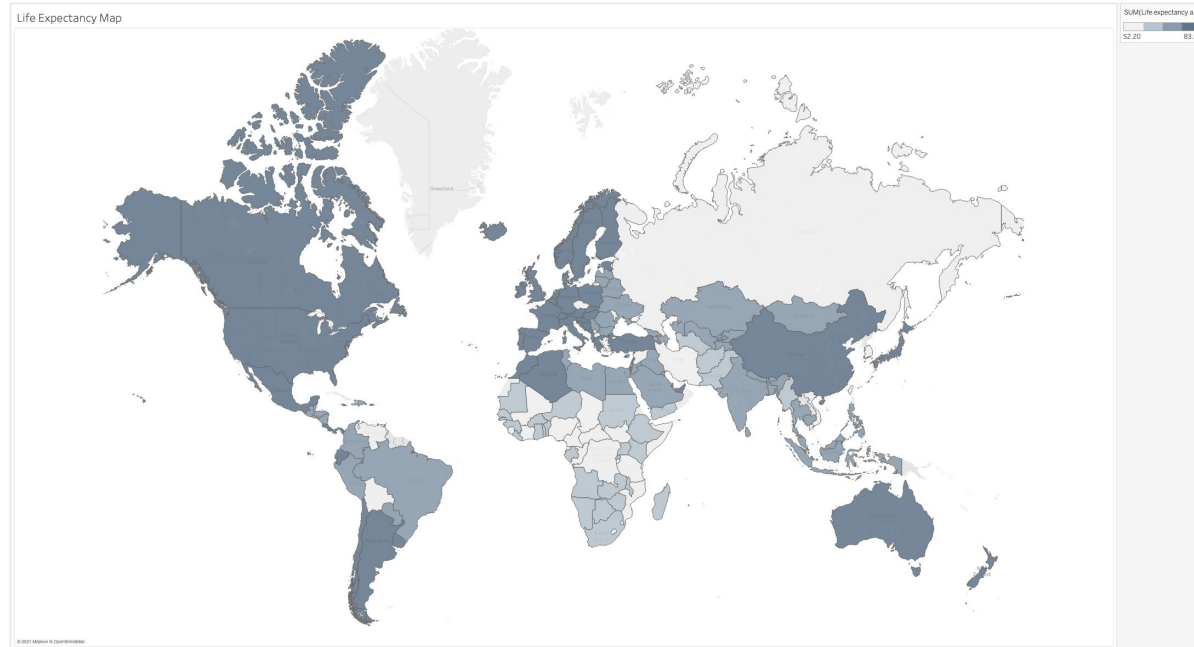
- Energy-based approach
- Graph objects are modeled as physical objects that mutually exert forces on each other (e.g. spring / electric charge)
- Vertices linked by edges attract each other
- Vertices not linked repel each other
- Optimize the euclidean distance between the vertices to minimize the overall energy of the system
- [Example](#)



Map-based Visualization

Geometric spatial data (geospatial data):

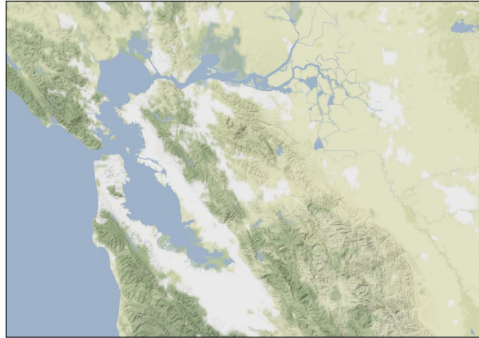
Conveys shape information directly through spatial position of its elements



Map-based Visualization

Geospatial data visualization examples:

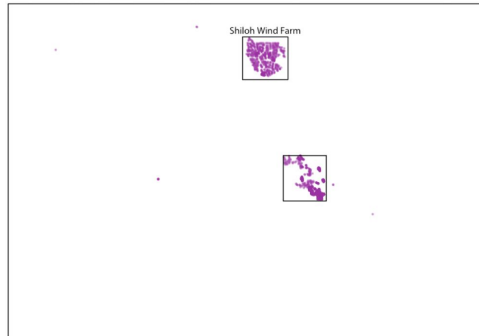
terrain



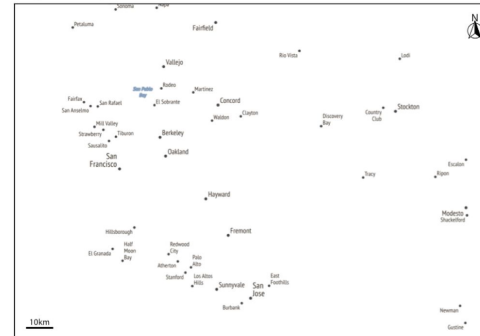
roads



wind turbines

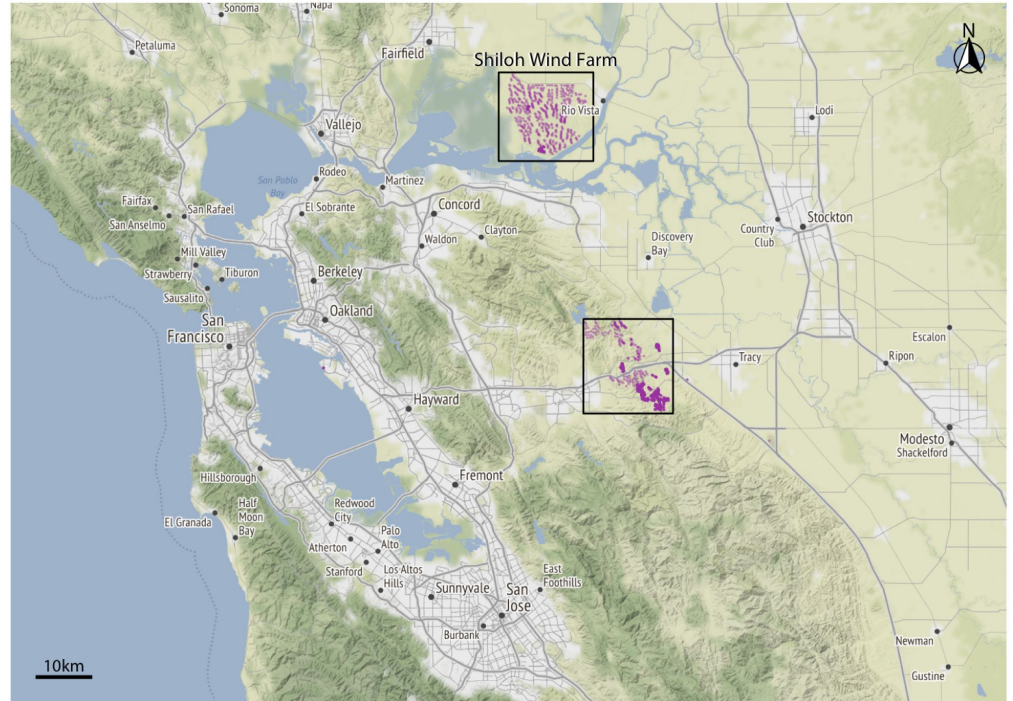
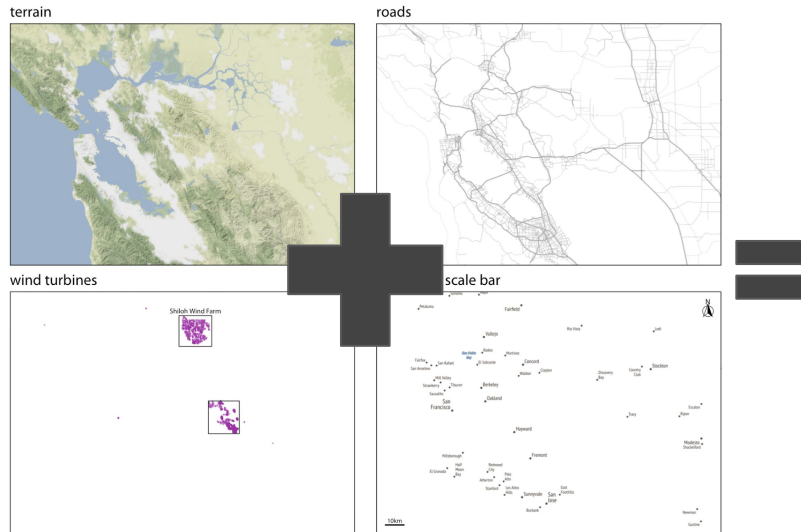


city labels, scale bar



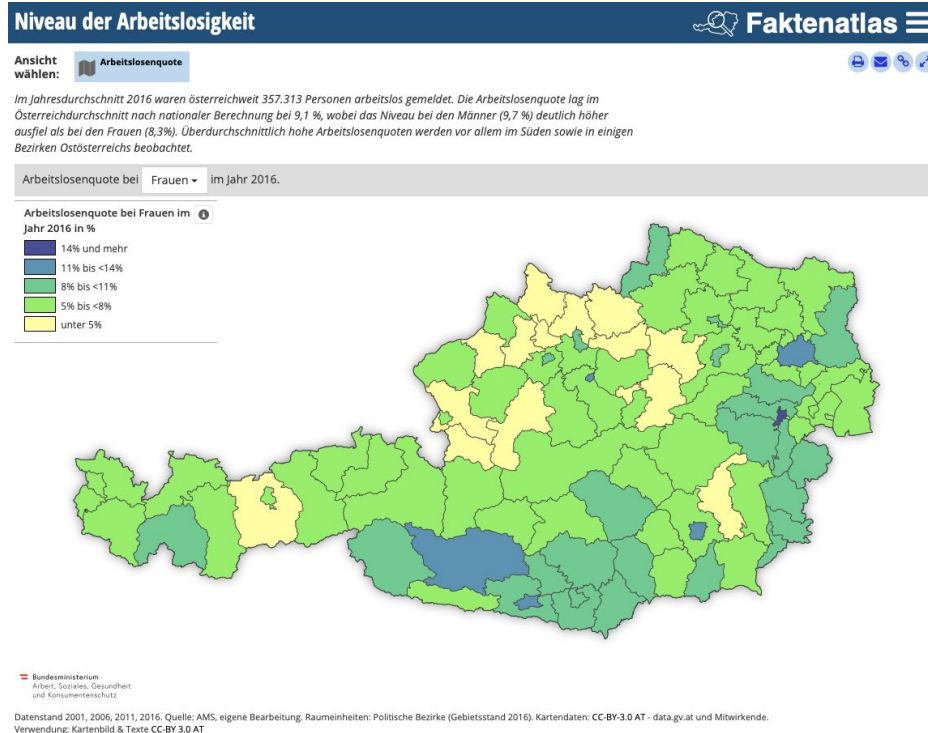
Map-based Visualization

Geospatial data visualization examples:



Map-based Visualization

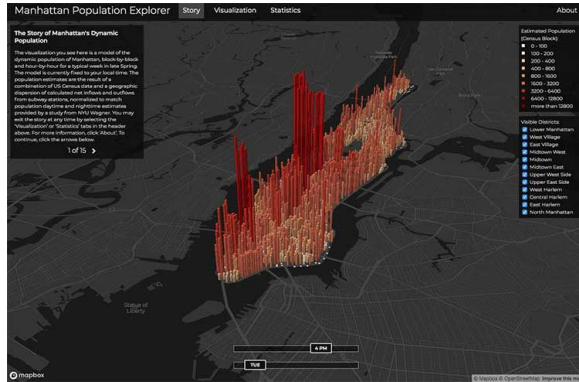
Faktenatlas by the Austrian government (discontinued)



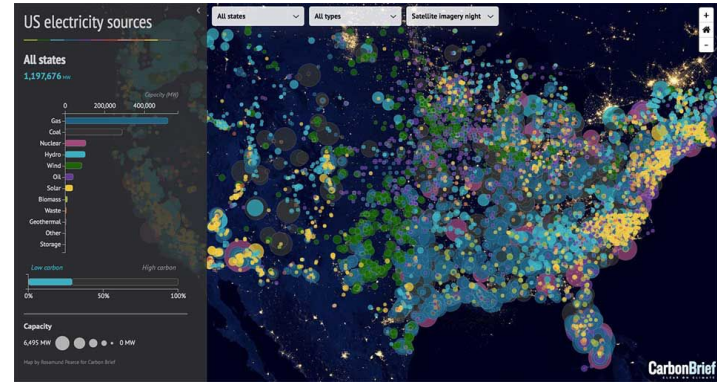
http://www.faktenatlas.gv.at/articles/niveau_arbeitslosigkeit.php

Map-based Visualization

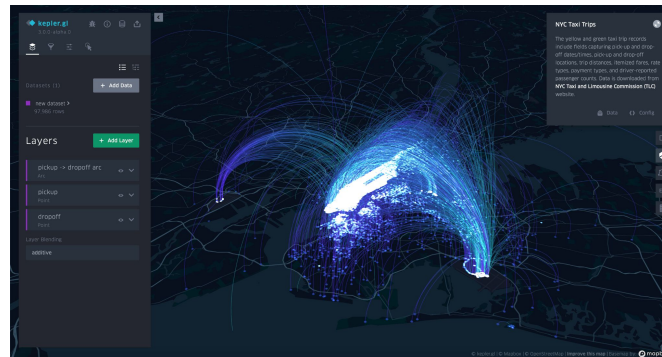
Geospatial data visualization examples:



<http://manpopex.us>



<https://www.carbonbrief.org/mapped-how-the-us-generates-electricity>



<https://kepler.gl/>

Content

- Introduction
- Graph Theory
- Graph Drawing
- **Dynamic Graphs**
- Data Operations

How can graphs change over time?

How can graphs change over time?

Graph change	Relevance	Example
Node addition and removal	Perhaps nodes were relevant in your graph for only a short duration of time.	In graphing IP traffic, it's useful to see if an IP address had a brief flurry of activity and then went silent, possibly indicating the device was turned off.
Link addition and removal	Perhaps two nodes were connected for only a short duration of time and unrelated otherwise.	If you're looking at criminal networks, two criminals may have communicated directly at first but now go through intermediaries.
Property changes on nodes or links	Perhaps properties that are relevant are increasing or decreasing over time.	In graphing financial networks, perhaps a bank account started out small but grew significantly over time and is now worthy of more scrutiny.
Community formation or dissolution	Perhaps groups of nodes are tightly connected at times but unrelated otherwise.	When graphing social networks, social groups often start off homogenous with lots of links between them but quickly split off into isolated groups.

How can we visualize/map time in graphs?

t_1

<i>X</i>	<i>Adjacent to X</i>
Alan	Bob, Mary
Bob	Alan, Nicole
Mary	Alan, Nicole
Nicole	Bob, Mary

t_2

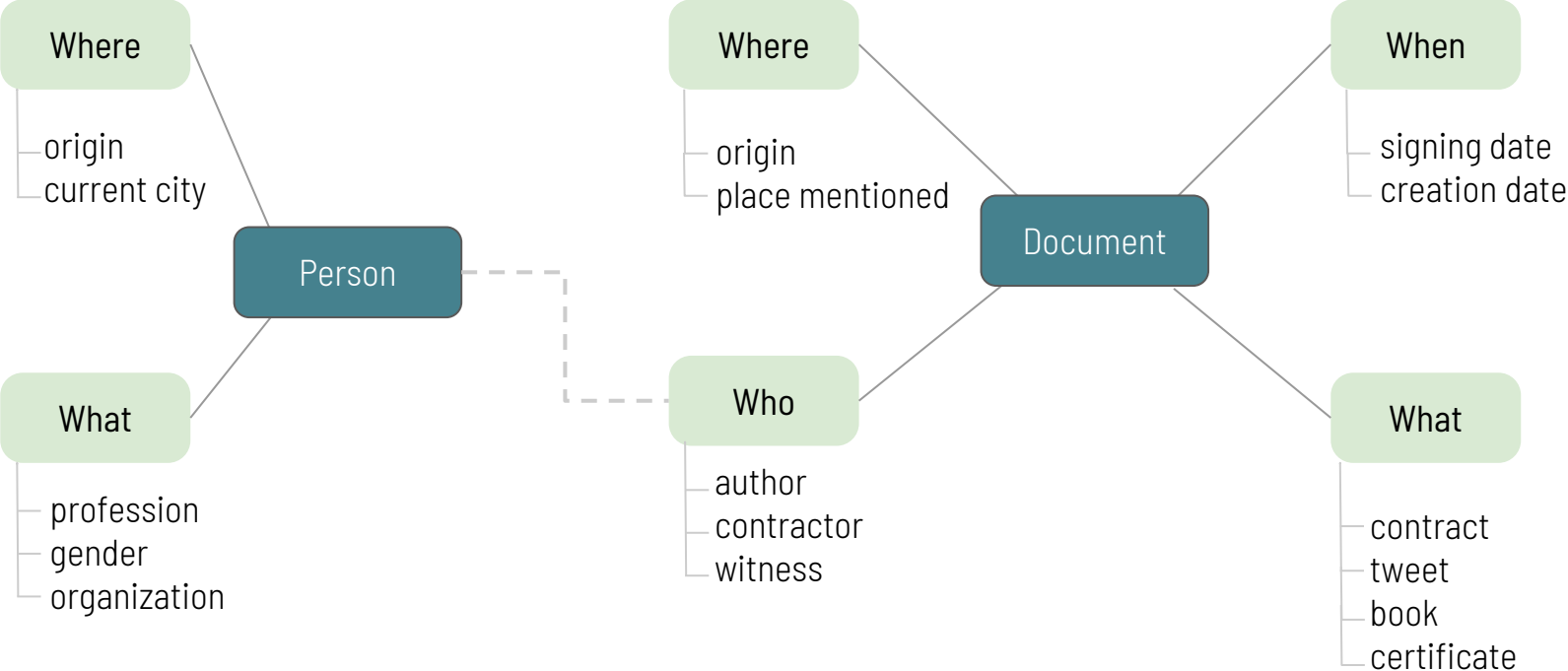
<i>X</i>	<i>Adjacent to X</i>
Alan	Bob, Mary
Bob	Alan
Mary	Alan

t_3

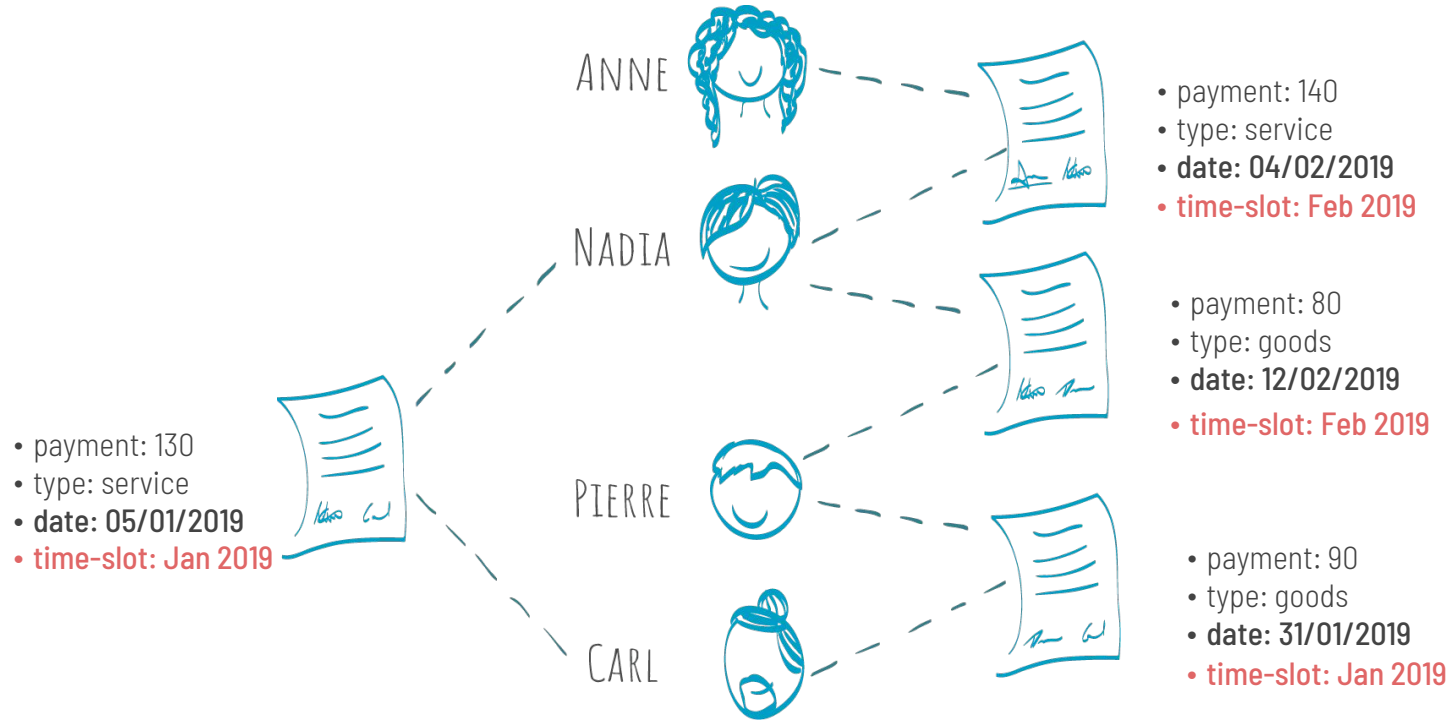
<i>X</i>	<i>Adjacent to X</i>
Alan	Mary
Bob	-
Mary	Alan, Lisa
Lisa	Mary

<https://whiteboard.fi/p9f46>

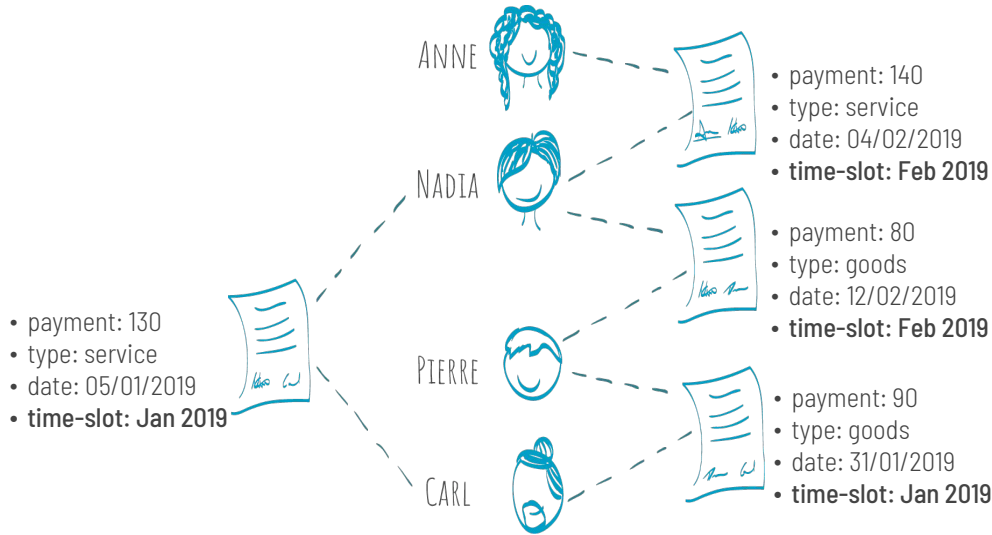
Example: Historical data



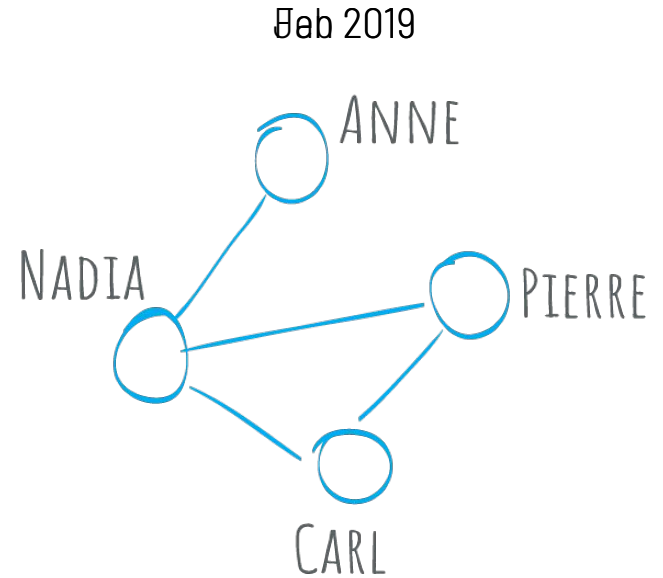
Time can also be a contract attribute



Map time to time

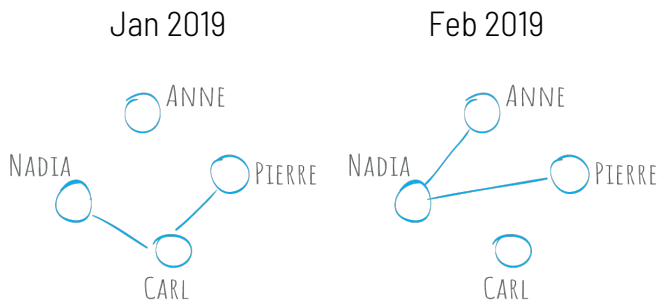


- Animation

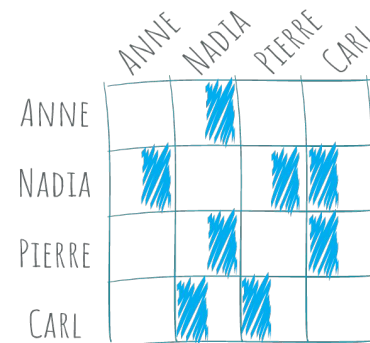
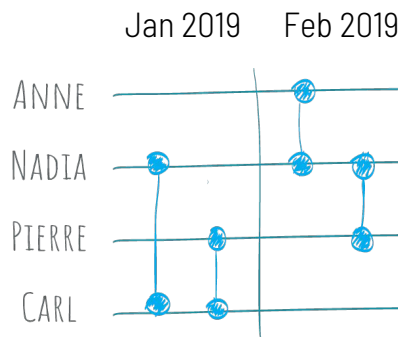


Map time to space

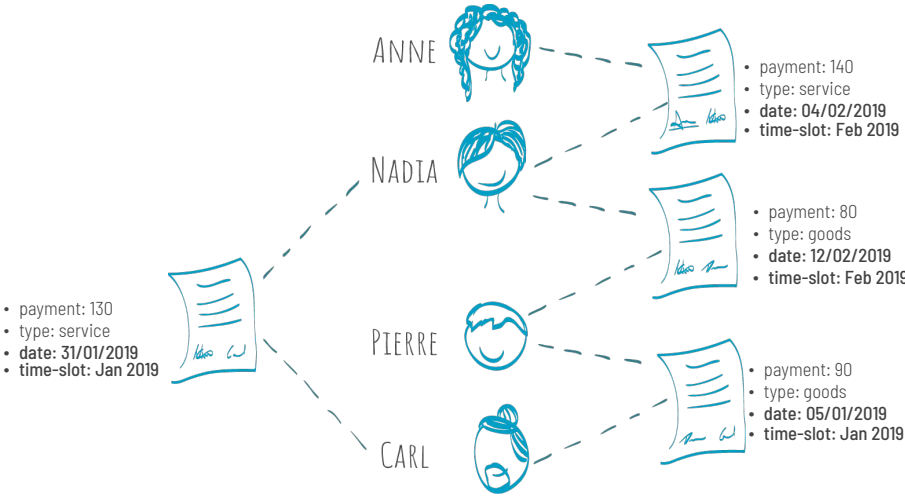
- Juxtaposed



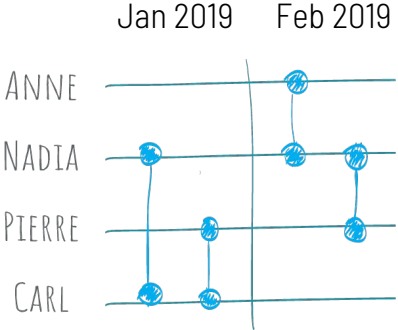
- Integrated



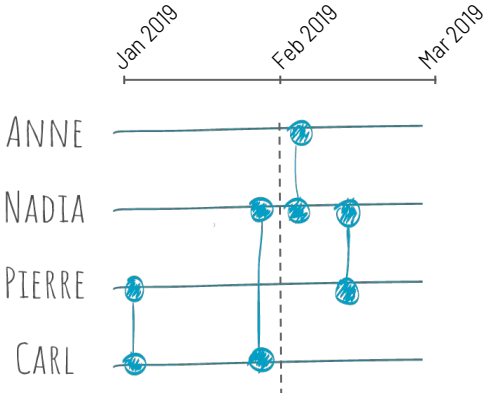
How to handle time?



- multiple networks with time slots



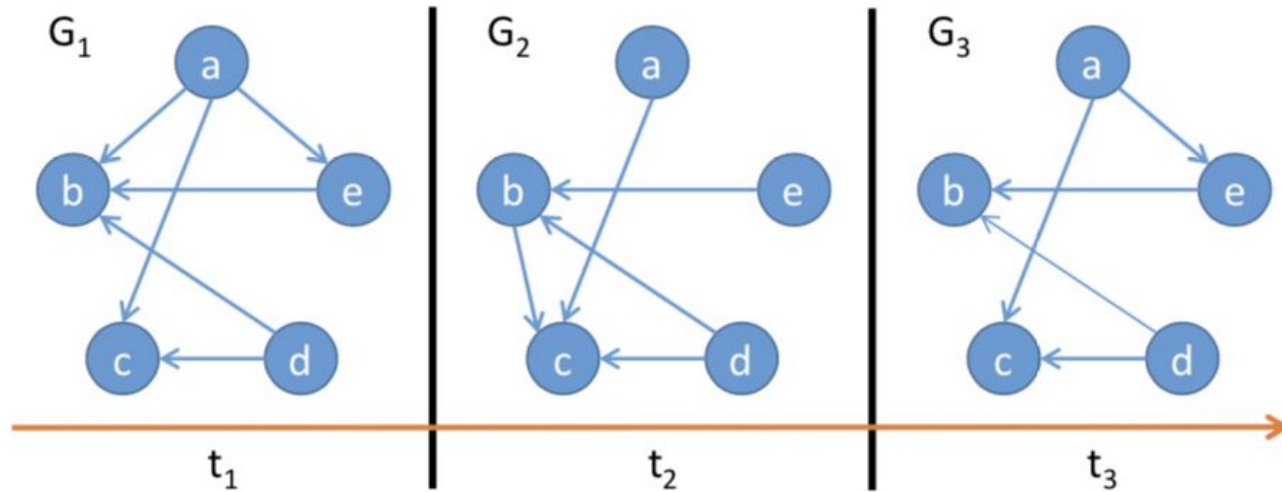
- continuous (link streams)



Dynamic Graphs

- Dynamic graph visualisation focuses on the challenge of **representing** the **evolution of relationships** between entities in readable, scalable and effective diagrams.
- We look at how the structure of the vertices and edges as well as their attributes change over time.
- Types of dynamic graphs
 - Animated (Time-to-Time Mapping)
 - Timeline (Time-to-Space Mapping)
- $G = (V, E)$
- $\Gamma = (G_1, G_2, G_3, \dots G_n)$
 - Γ - Gamma is the dynamic graph

Dynamic Graphs: example

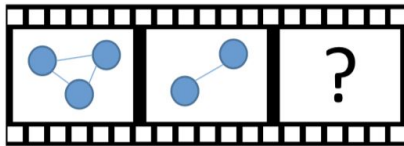


Animated (Time-to-Time Mapping)

- A mapping of **timestamps** are assigned to the sequence of graphs.
- Animated dynamic graphs are **mainly** used for **node-link** diagrams.
- Mental map should be preserved (changes to this map should be minimal)
- General-purpose Layout
 - Online - only considering past steps
 - Offline - considering both past and future time steps
- Special-purpose Layout
 - Compound graphs

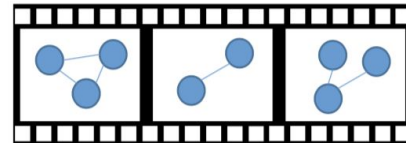
Online Problem

- Only considers past time steps
- In general more **flexible** as the complete evolution of the graph does not need to be known at the animation start
- Good for **interactively changing** graphs



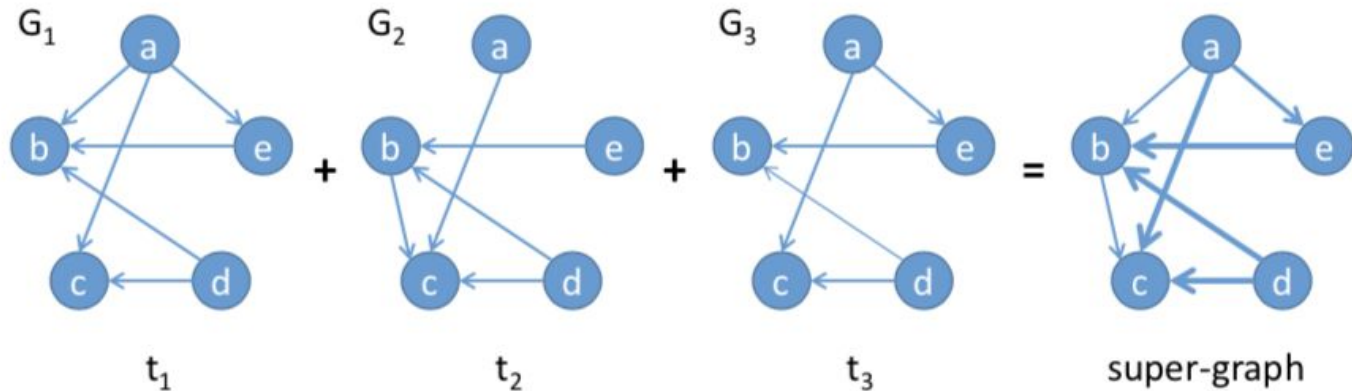
Offline Problem

- Considers both past and future time steps
- Full **evolution** is **known** at **visualisation time**
- Allows for **better optimizations** of the layout and maintaining the mental map because all changes are known



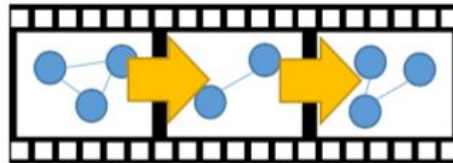
Super Graph

- A **super graph** is constructed from a dynamic graph where the **edges** from multiple time-steps are **aggregated** and represented in the super-graph through **weighted edges** (line thickness).



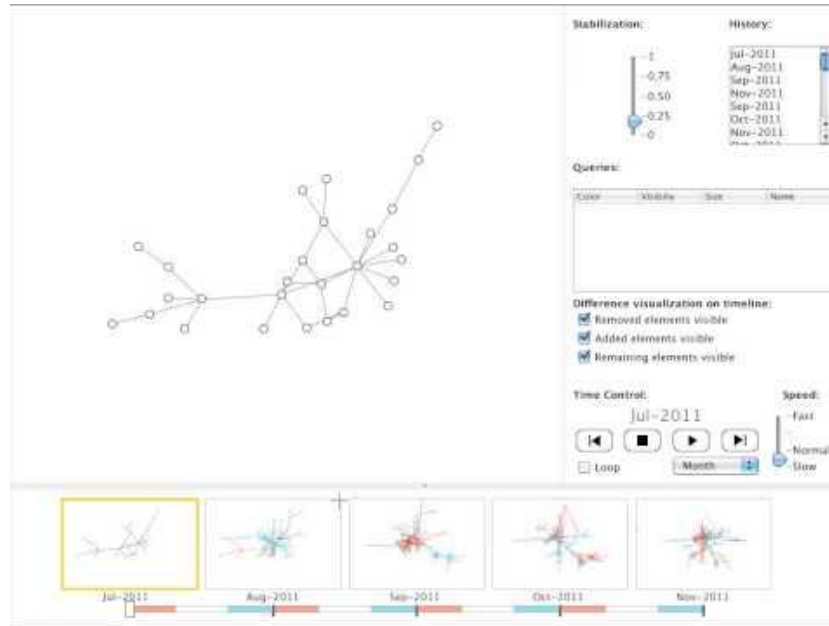
Transition Problem

- Transitions between consecutive layouts need to be modeled
- **Too many changes** may happen at the same time
- Transitions must be **traceable** by the user to not lose the mental map of the graph
- Split the transition into four steps
 - a. Nodes and edges are removed
 - b. The graph is translated towards the new layout
 - c. Nodes are moved individually
 - d. The new nodes and edges are displayed


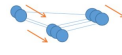


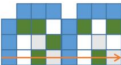


Example: GraphDiaries

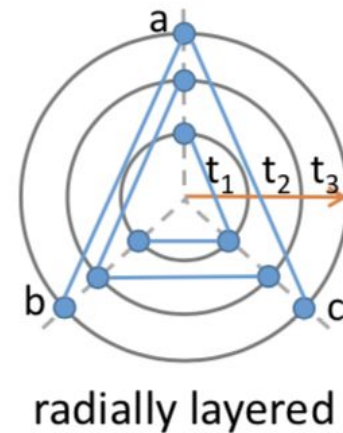
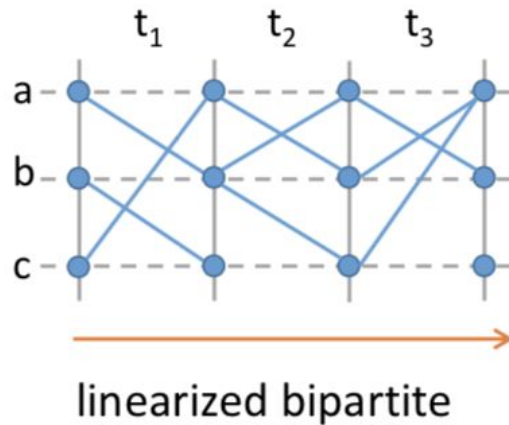
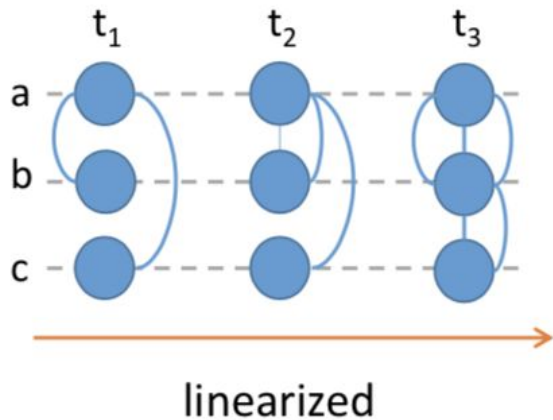
"Identifying, tracking and understanding changes in dynamic networks are complex and cognitively demanding tasks. [...] GraphDiaries, a visual interface designed to improve support for these tasks in any node-link based graph visualization system." [1, p. 740]



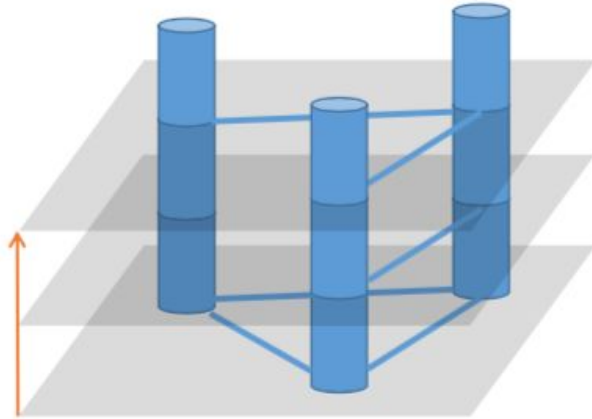
Timeline (Time-to-Space Mapping)

- The complete sequence of graphs is displayed on a **timeline**
- A challenge with this approach is the **visual scalability** since each timestep takes up additional space
- There are **node-link** and **matrix** based approaches
- Approaches node-link
 - Juxtaposed 
 - Superimposed 
 - Integrated 
- Approaches Matrix
 - Intra-Cell 
 - Layered Matrices 

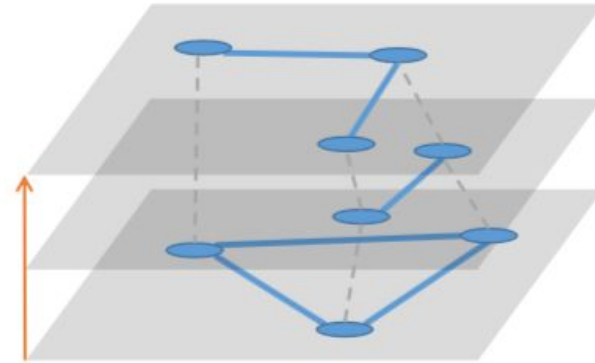
Juxtaposed



Superimposed

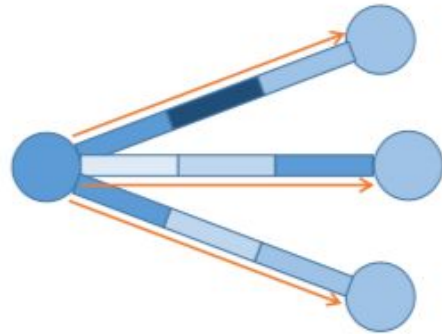


fixed positions

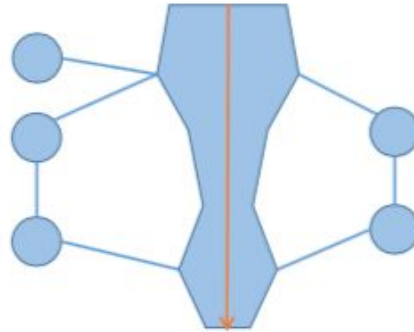


relaxed positions

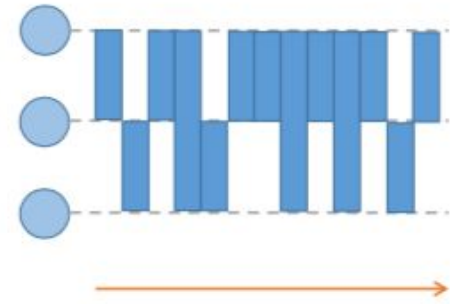
Integrated



timeline edges



ego timeline



parallel edges

Integrated: Timeline Edges (Example)

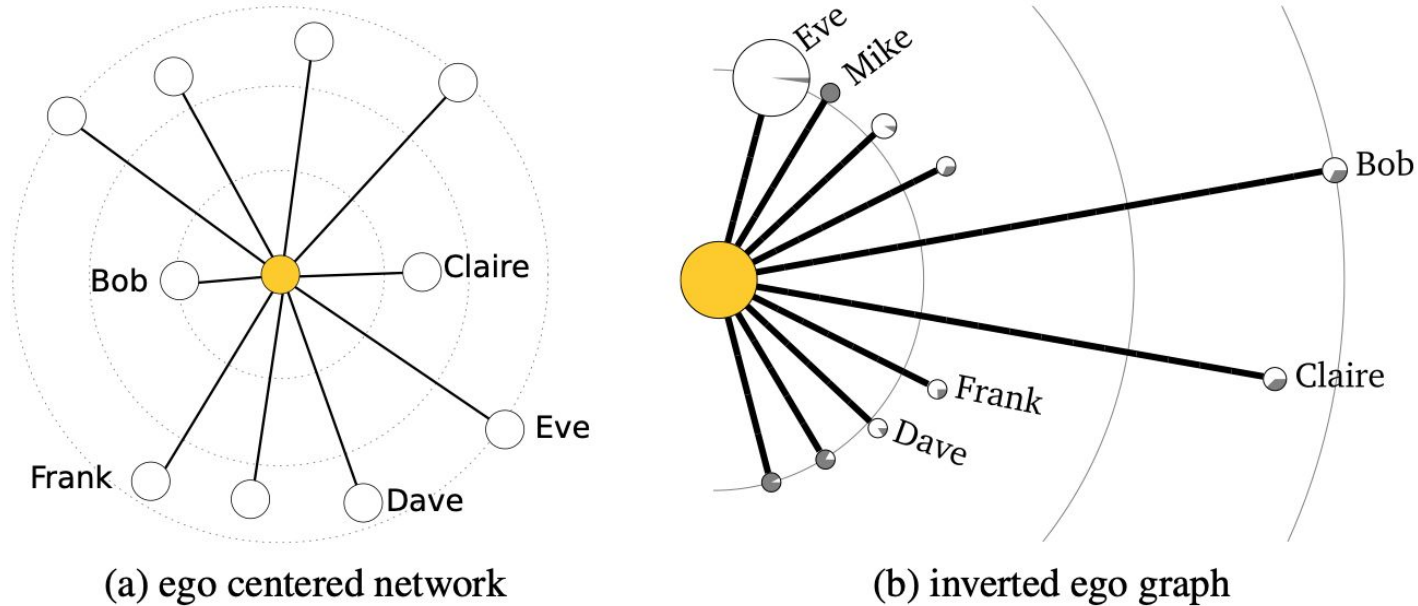


Fig. 2: *Adam's* coauthor network as ego graphs. *Adam's* node is painted in yellow.

Integrated: Timeline Edges (Example)

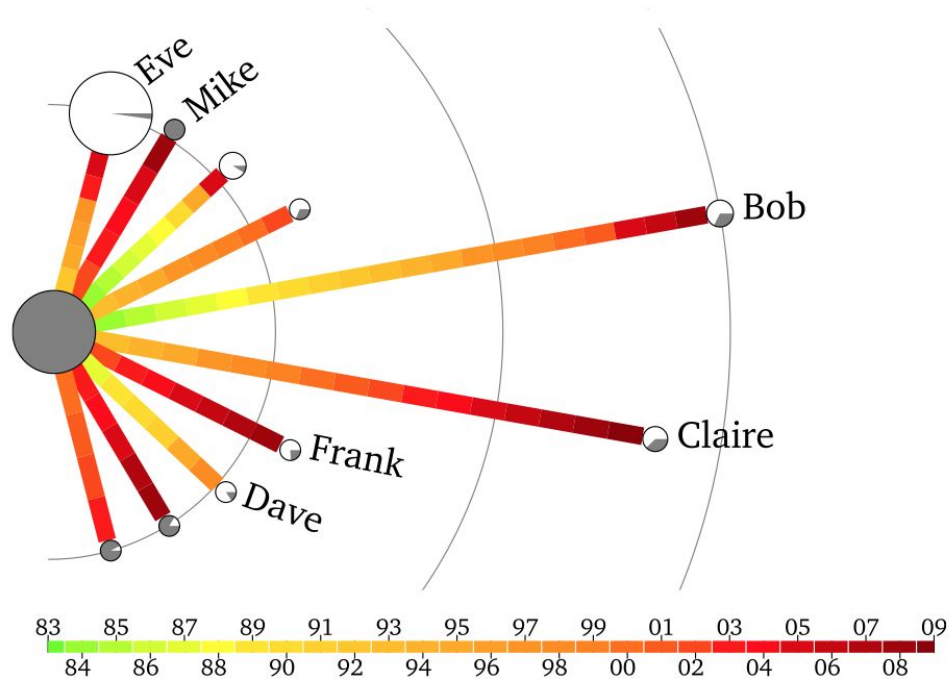
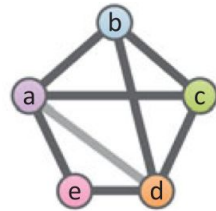


Fig. 3: The time-color view for *Adam's* coauthors

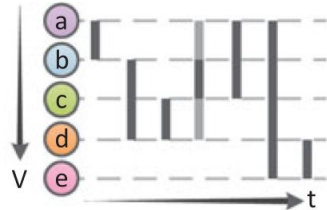
Integrated: Parallel Edges (Example)

From	To	Time	...
a	b	1	...
b	d	2	...
d	c	3	...
c	b	4	...
d	a	4	...
c	a	5	...
a	e	6	...
e	d	7	...

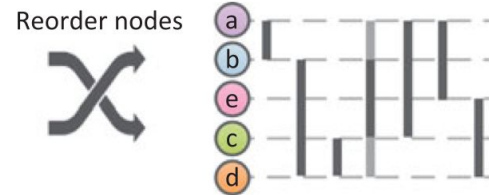


(a) Table.

(b) Node-link diagram.



(c) MSV: Lexicographic node order.



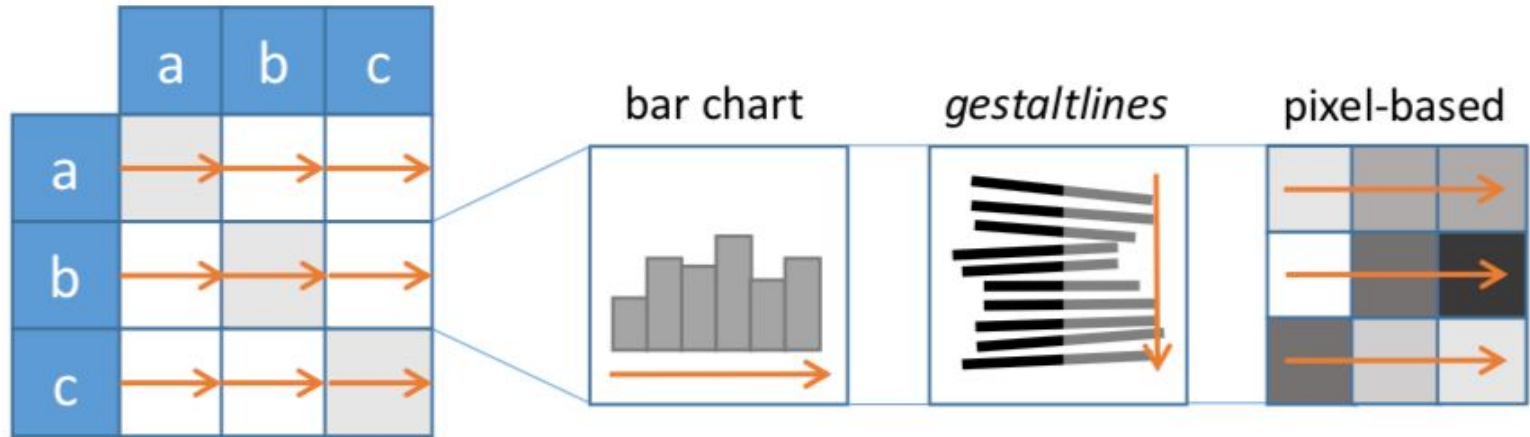
(d) Maximal edge length order.



(e) Minimal edge length order.

Fig. 1. Different visualizations of dynamic network data. For the node-link diagram (b) time is flattened and not visible. The msv (c) shows the relations over time. Note that there are no overlapping edges in the minimal edge length ordering (e).

Intra-Cell



Intra-Cell: Example 1

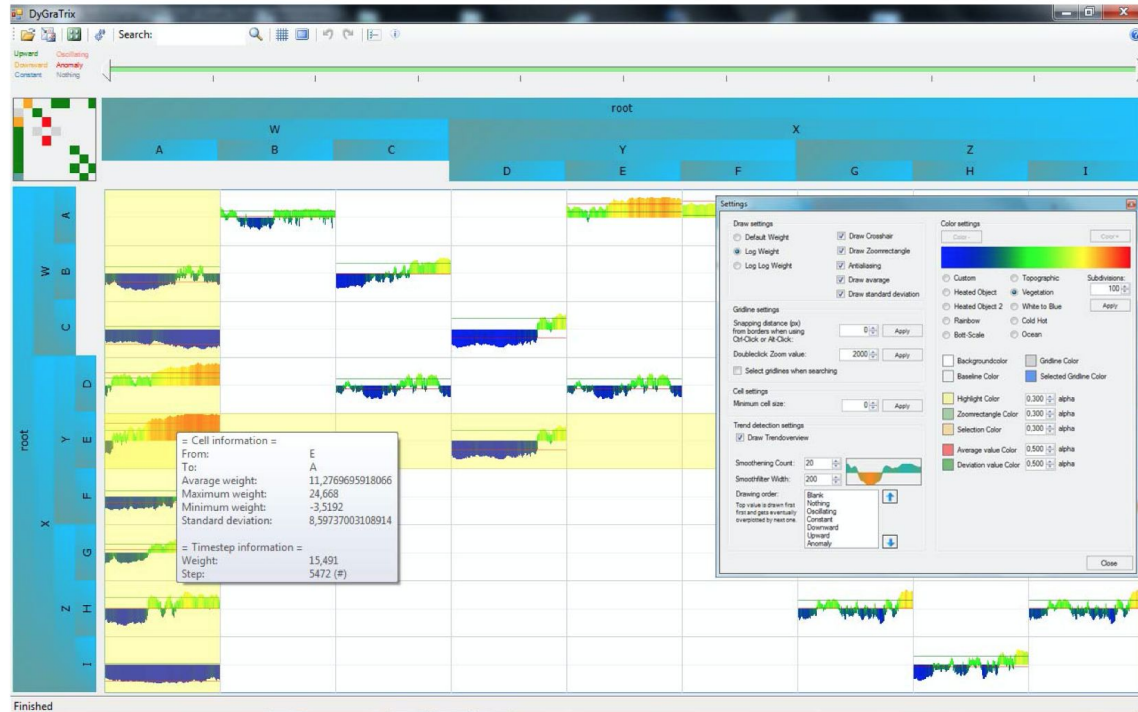
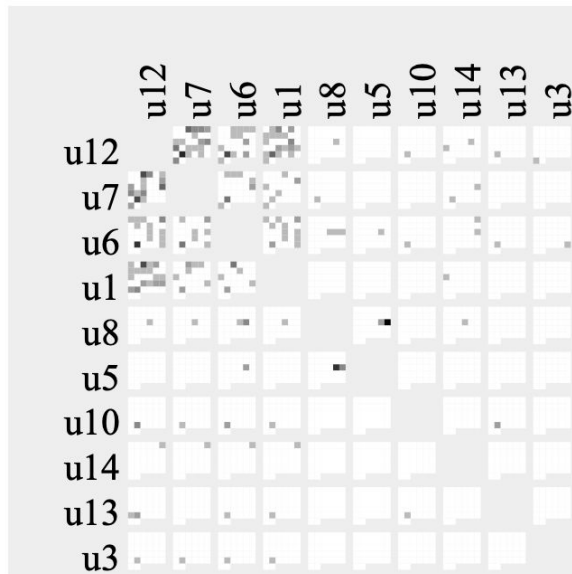


Figure 1. The graphical user interface showing several cells side-by-side for time-varying edge weights as well as additional panels for details-on-demand and parameter settings.

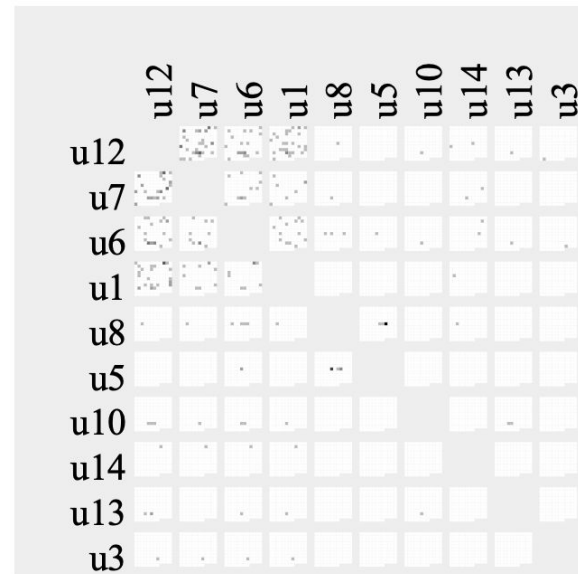
Intra-Cell: Example 2

	u12	u7	u6	u1	u8	u5	u10	u14	u13	u3
u12		43	20	29	1	0	1	2	1	1
u7	39		11	7	1	0	0	2	0	0
u6	22	11		13	3	1	1	2	1	1
u1	30	9	9		0	0	0	1	0	0
u8	1	1	4	1		16	0	1	0	0
u5	0	0	2	0	12		0	0	0	0
u10	3	1	2	1	0	0		0	2	0
u14	1	1	1	1	0	0	0		0	0
u13	3	0	1	1	0	0	1	0		0
u3	1	1	1	1	0	0	0	0	0	

(a) matrix



(b) 4 week steps



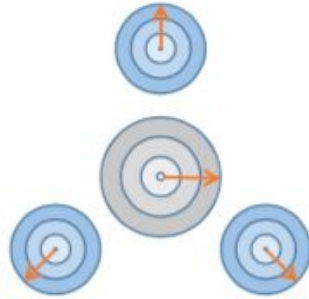
(c) weekly steps

Fig. 3. *workgroup* wiki network matrix

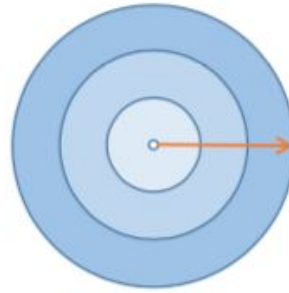
Layered



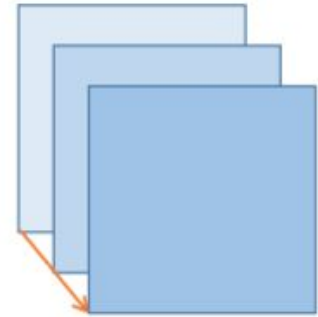
small multiples



radially distributed



radially layered



stacked

Layered: Radially Layered/Distributed Example

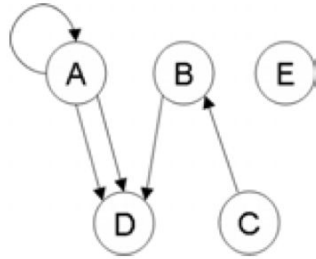


Figure 1: Node-link diagram of a graph.

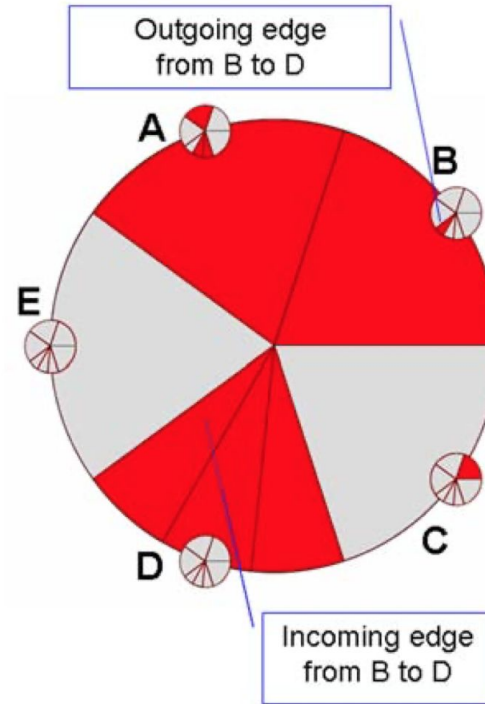


Figure 2: TimeRadarTree of a single graph.

Layered: Radially Layered/Distributed Example

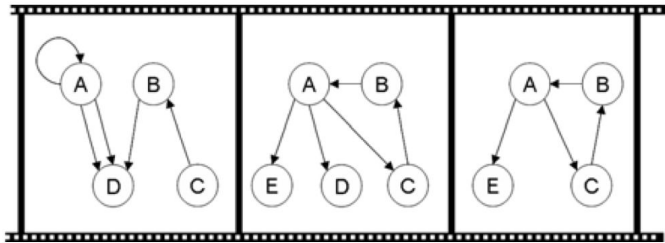


Figure 3: Node-link diagrams of a sequence of graphs.

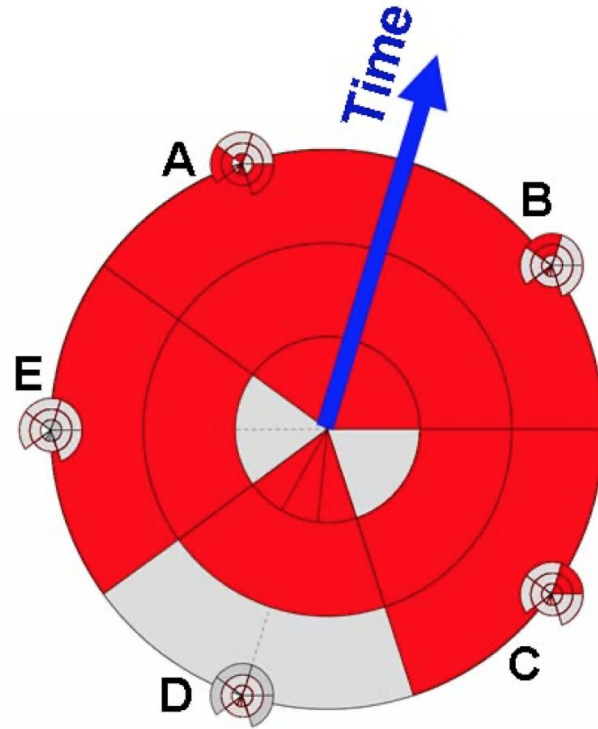


Figure 4: TimeRadarTree of a sequence of graphs.

Layered: Stacked Example

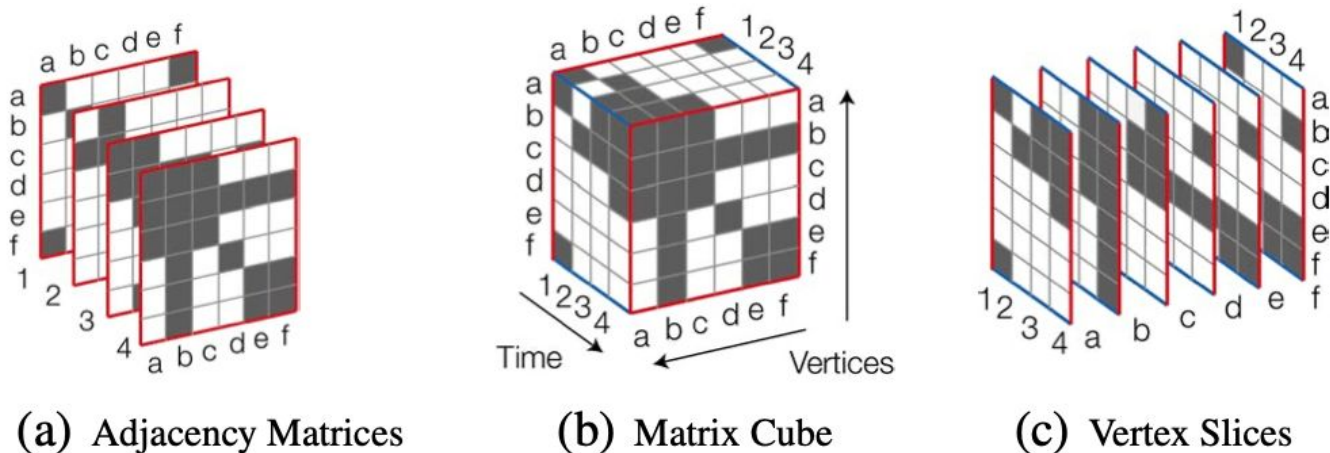


Figure 1. The Matrix Cube. (a) Each time step of the network (1,2,3,4), is represented as an adjacency matrix. (b) The Matrix Cube results from stacking those matrices. Red edges of the cube hold nodes and correspond to the rows and columns of the constituent adjacency matrices; blue edges of the cube hold time steps. (c) Slicing the cube along one of the vertex dimensions yields *vertex slices*.

Content

- Introduction
- Graph Theory
- Graph Drawing
- Dynamic Graphs
- **Data Operations**

Data Operations



Aggregating Nodes/Edges

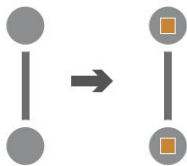
- Common strategy to **increase the scalability** and **summarize nodes/edges** with shared characteristics (e.g. clusters)
- Aggregation **supports overview tasks** on the network
- Useful for **high-level overviews**
- Can be preprocessed but is even more powerful when it is invoked interactively



Querying and Filtering

- Allows users to **focus on nodes or edges of interest by reducing visual clutter**
- Enables analysts to work with large networks
- Can be defined based on topological structures, attributes, or combination of both
- Especially useful for **query-first strategies**

Data Operations



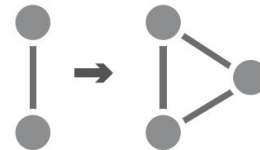
Deriving New Attributes

- Can be derived from **topological features** (e.g. node degree, clusters) **or** from **other attributes**
- Useful for specific analysis questions, e.g. finding the most connected node in a network or quickly identifying clusters of interest
- Once computed, they can be visualized just as other attributes



Clustering

- Grouping nodes **based on similar features**
- Mostly based on **network topology**
- Can be used to derive attributes, highlight topological features, to improve layouts or as a precursor step for aggregation



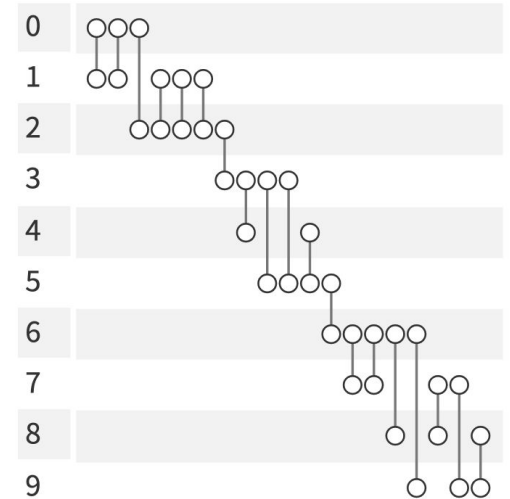
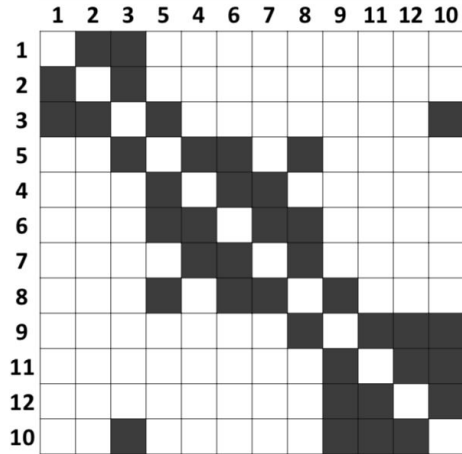
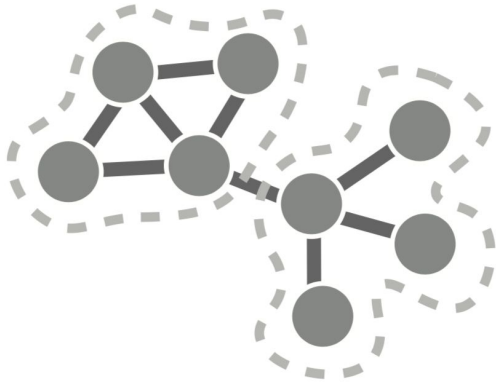
Converting Attributes/Edge to Nodes

- Many visualization techniques are **better suited to visualize node attributes** than edge attributes
- **Significant reshaping operation** which risks confusing analysts
- Most useful as preprocessing operation

How to represent groups/clusters in graphs?
Static vs. dynamic groups?

How to represent groups? Nodes

- Position

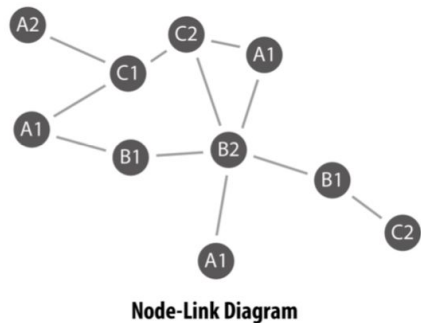


Nobre, C., et al. "The state of the art in visualizing multivariate networks." *Computer Graphics Forum*. Vol. 38. No. 3. 2019.

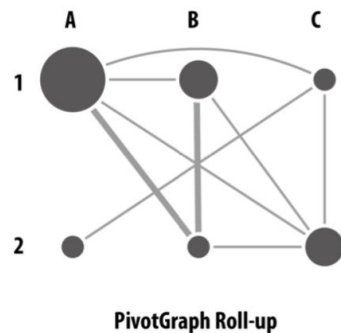
Vehlow, Corinna, Fabian Beck, and Daniel Weiskopf. "The State of the Art in Visualizing Group Structures in Graphs." *EuroVis (STARs)*. 2015.

How to represent groups?

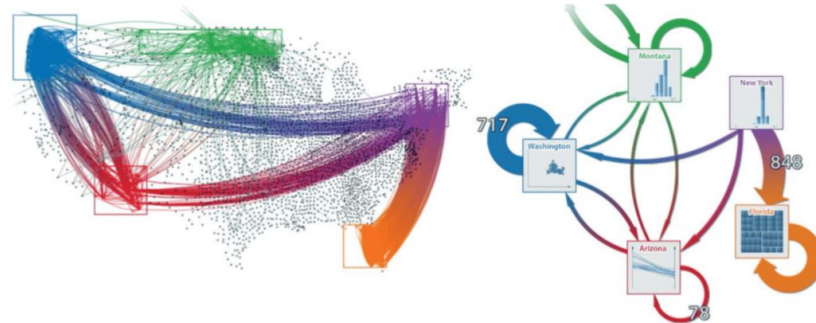
- Aggregation



(a)



(b)



How to represent dynamic groups?

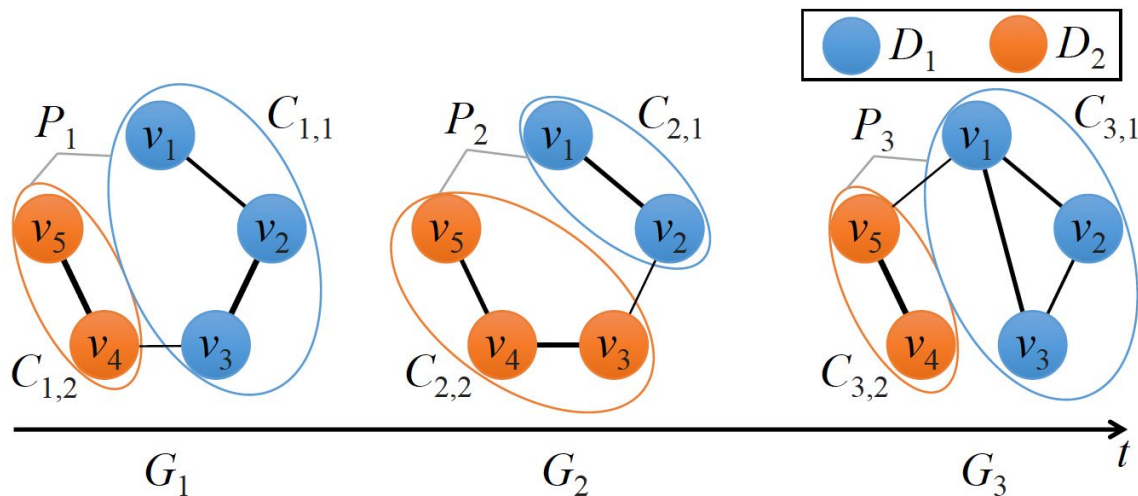
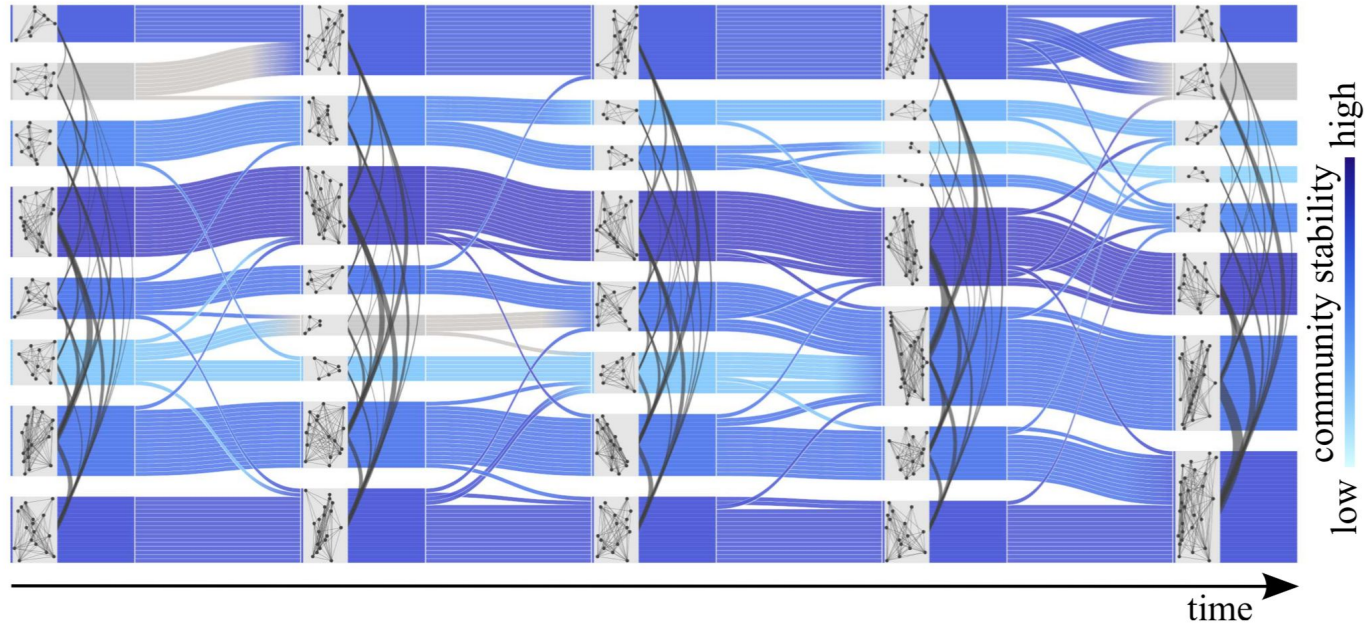


Figure 1: Example of a dynamic graph \mathcal{G} with $T = 3$ time steps and $L = 2$ dynamic communities. The edge weights w_{e_j} are mapped to the width of the links.

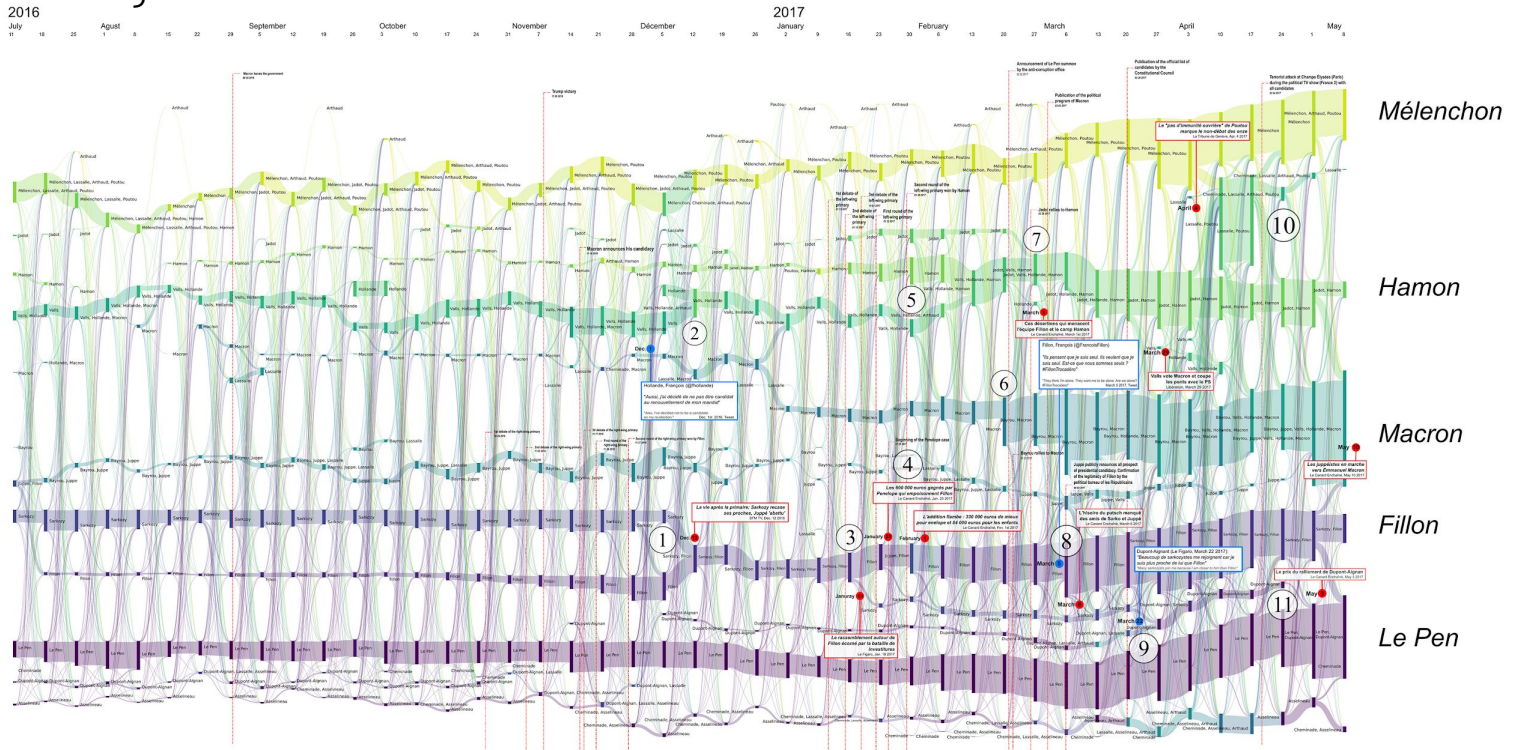
How to represent dynamic groups?

- Alluvial diagrams



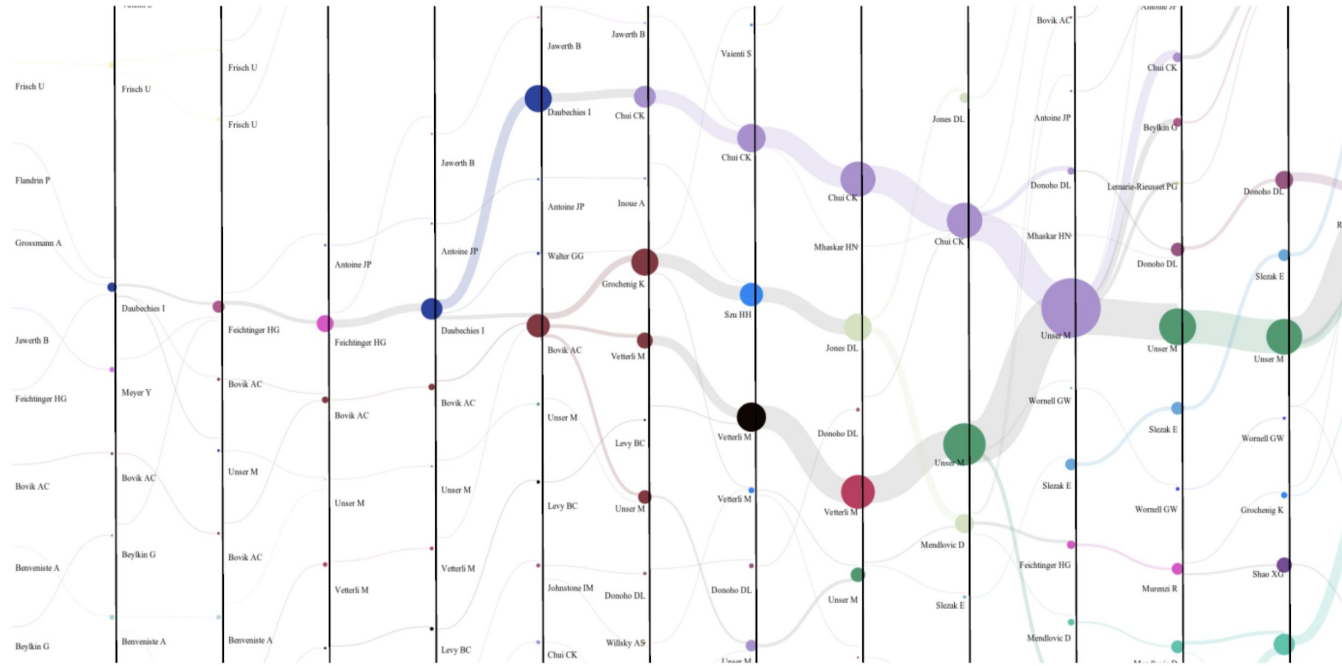
How to represent dynamic groups?

- Alluvial diagrams



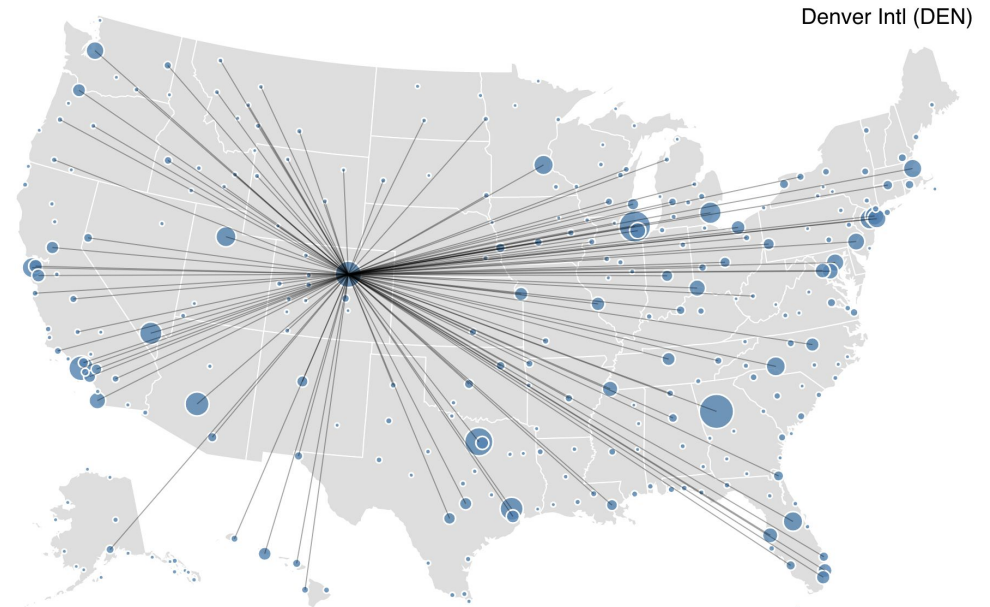
How to represent dynamic groups?

- Alluvial diagrams



Filtering: Edges on demand

- Hide edges when they are not needed
- Especially valuable in dense networks



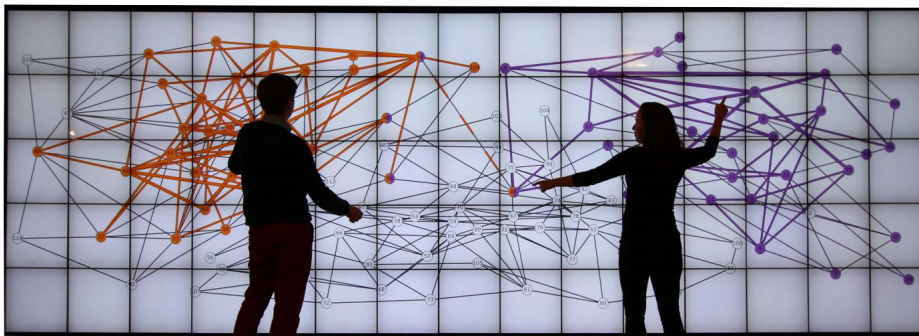
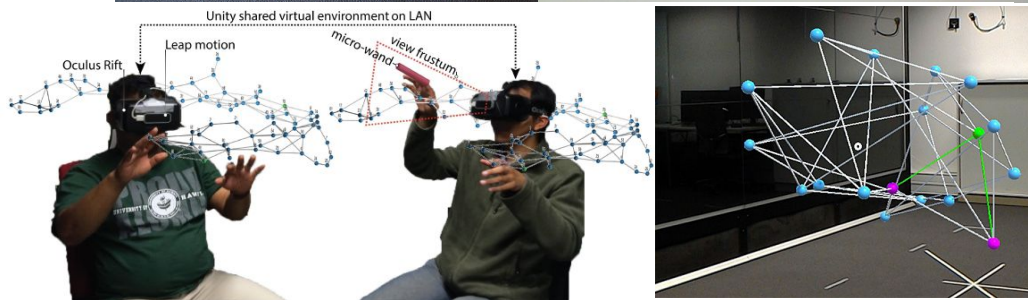
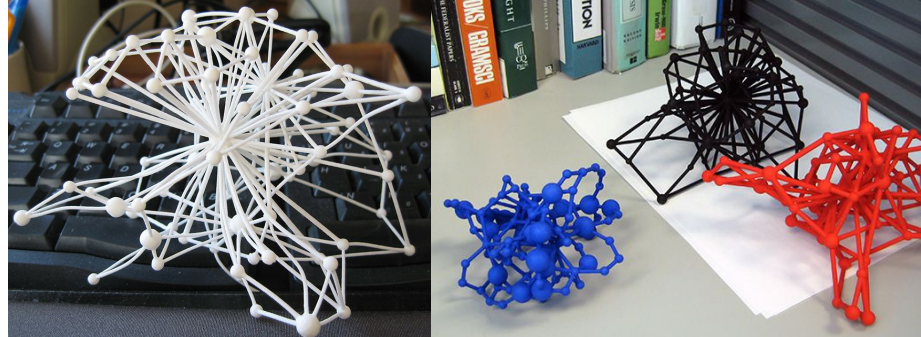
<https://vega.github.io/vega/examples/airport-connections/>

Content

- Introduction
- Graph Theory
- Graph Drawing
- Dynamic Graphs
- Data Operations

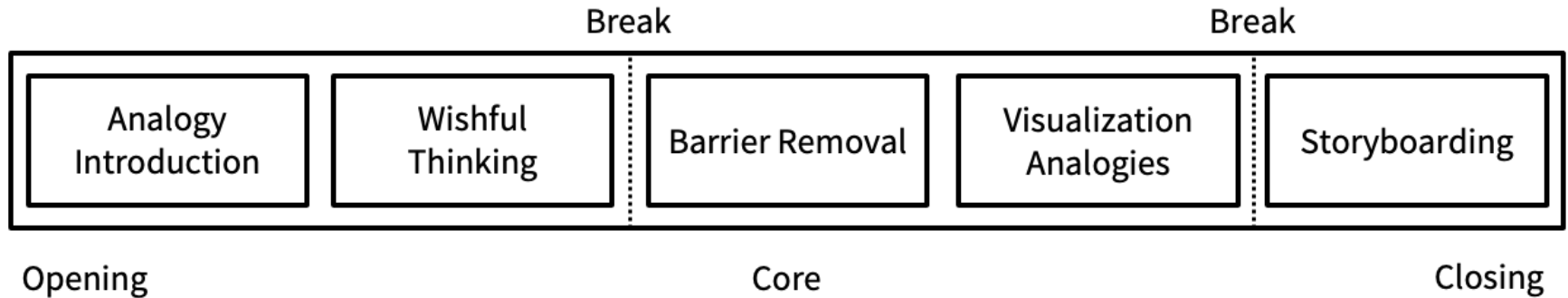
Beyond the Desktop

- Physical visualization of networks
 - <http://dataphys.org/list/tag/network/>
- Augmented Reality
 - E.g. <https://imld.de/en/research/research-projects/ar-aph-vis/>
- Wall-Sized Display
 - E.g. Arnaud Prouzeau, Anastasia Bezerianos, Olivie Chapuis. Evaluating Multi-User Selection for Exploring Graph Topology on Wall-Displays.

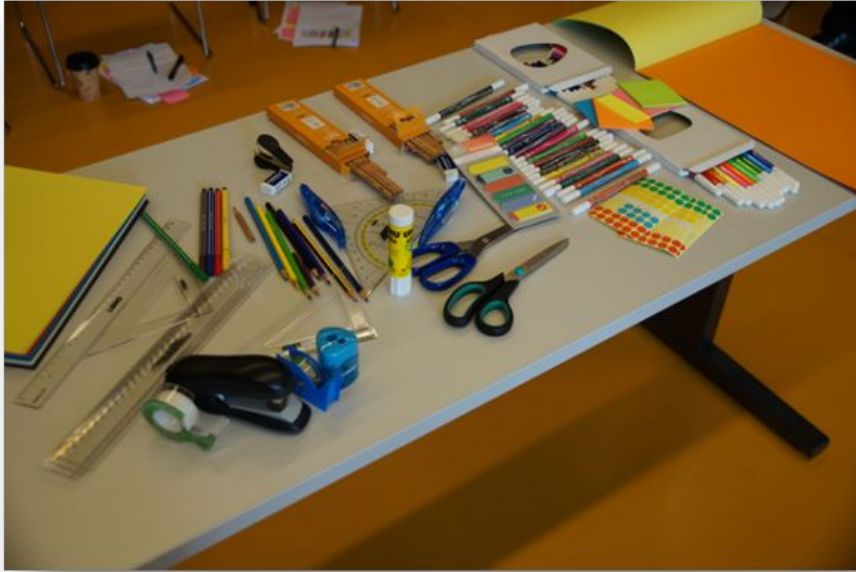


Digression: Creative Visualization-Opportunities Workshops

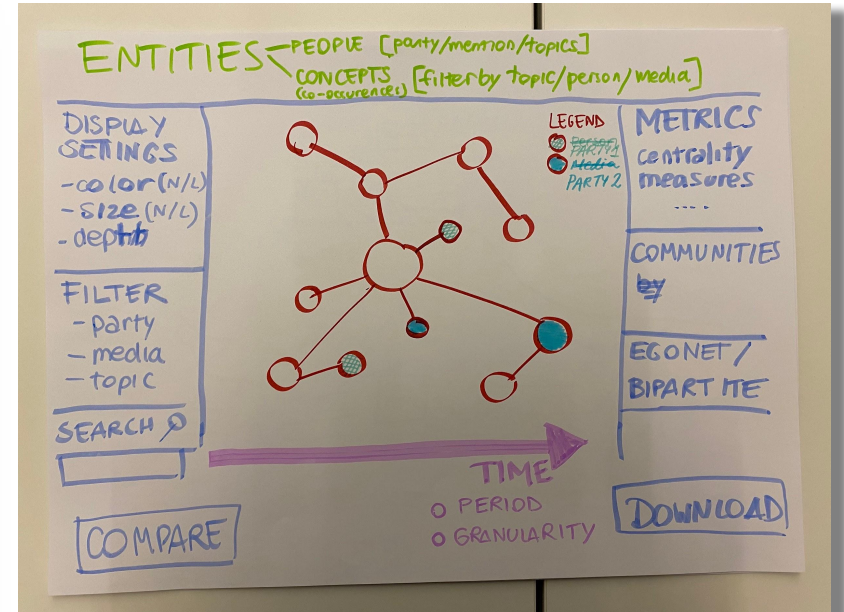
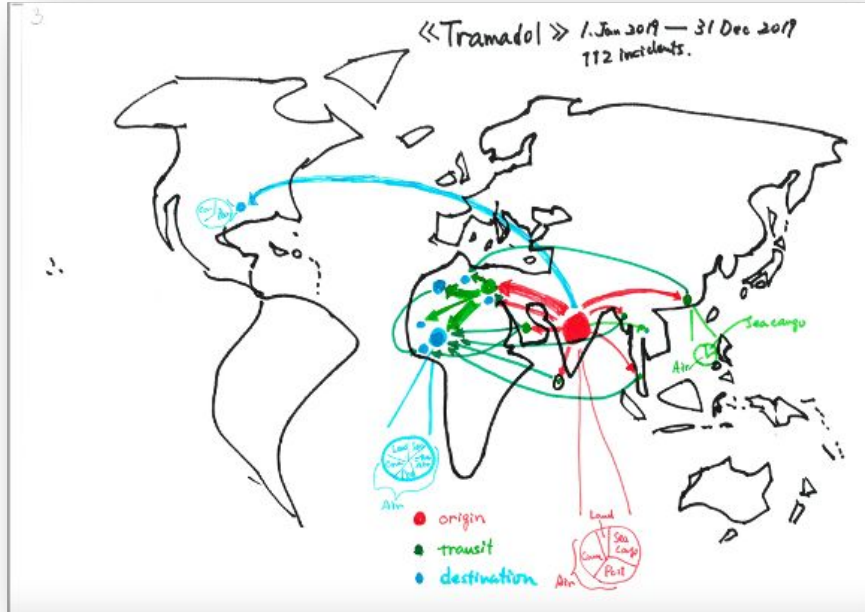
Minimize time for initial requirement analysis through a mix of active and passive activities



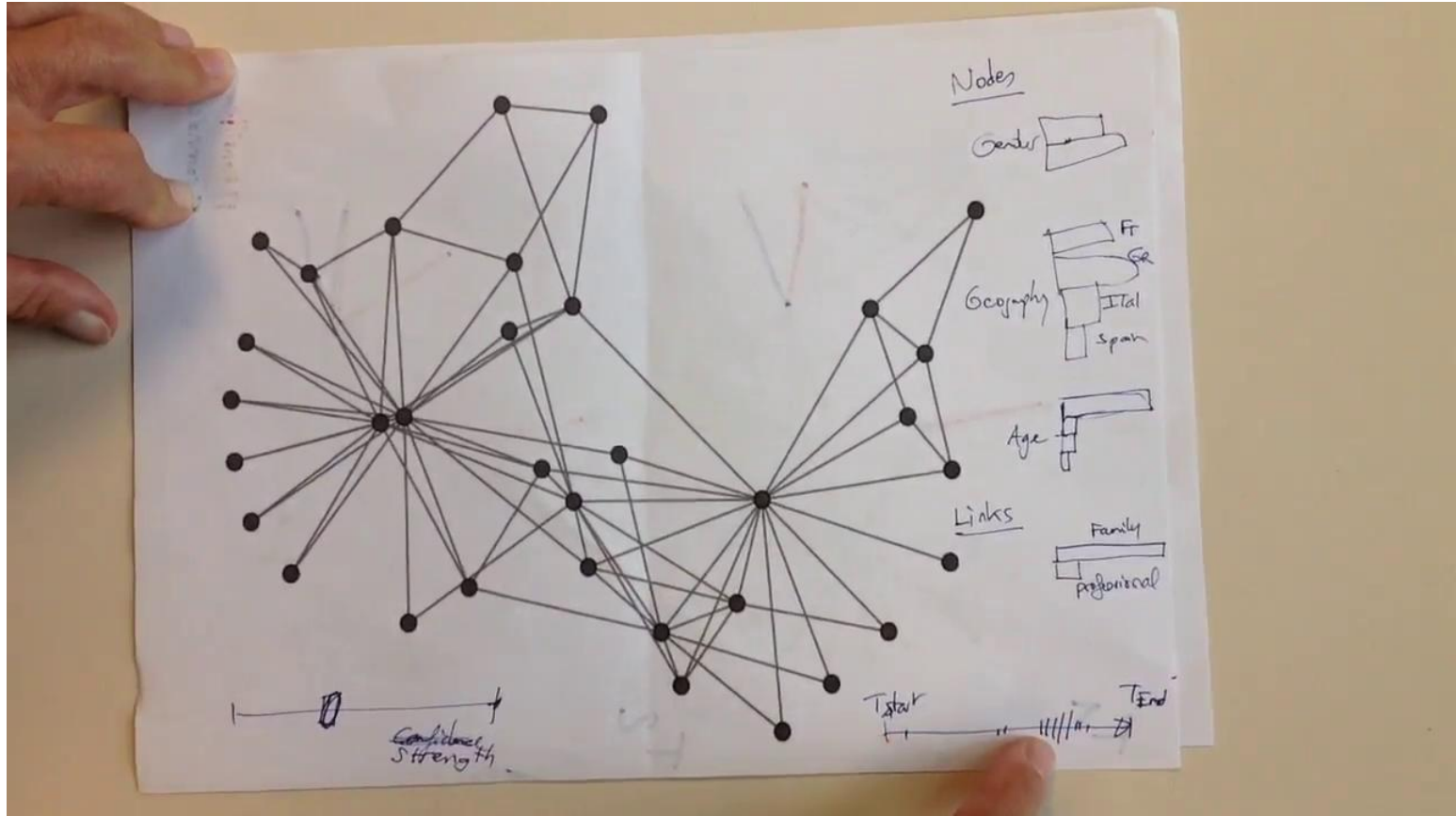
Storyboarding



Storyboarding



Video Prototyping



Thank you for your attention!

References

Tamara Munzner. "Visualization Analysis and Design. A K Peters Visualization Series", CRC Press, 2014.

Eaeds P., Klein K.. "Graph Visualization". In: Fletcher G., ed. and others. Graph Data Management : Fundamental Issues and Recent Developments. Springer, pp. 33-70. 2018.

Nobre, C., et al. "The state of the art in visualizing multivariate networks." *Computer Graphics Forum*. Vol. 38. No. 3. 2019.

Beck, Fabian, et al. "The State of the Art in Visualizing Dynamic Graphs." *EuroVis (STARs)*. 2014.

Vehlow, Corinna, Fabian Beck, and Daniel Weiskopf. "The State of the Art in Visualizing Group Structures in Graphs." *EuroVis (STARs)*. 2015.

Contributions to the slides from:

- Stefan Starflinger
- Paola Valdivia