

Informations Visualisierung

SoSe 19

Aaron Winziers & Benedikt Lücken-Winkels

July 2, 2019

Contents

1	1. Lecture	2
1.1	Orga	2
1.2	Visualisation-Basics	2
2	2. Lecture	3
2.1	Diagrams	3
2.2	Metaphors and Symbols	3
2.3	Symbols/Pictograms	3
2.4	Infographics	4
3	3. Lecture	4
3.1	Visual Memory	4
3.2	Visual Information Processing	4
3.3	Color Perception	5
3.4	Preattentive vision	5
3.5	Pattern Recognition	5
3.6	Motion recognition	6
4	Lecture	6
4.1	Layouting algorithms	7
4.2	Matrix visualization of Graphs	7
5	Lecture	8
5.1	Visualization of dynamic graphs	8
5.2	Multivariate data and time series	8
6	Lecture	8
6.1	Software Visualization: Architecture	8
6.1.1	Reverse Engineering	8
6.1.2	Enriched Node-Link Diagrams	8

1 1. Lecture

1.1 Orga

- Website: st.uni-trier.de/lectures/S19/IV/
- Tutorial: TBD (beginning: 22.-26.04.)
- Final exam: Do, 11.07. (elfths of July) 12-14 (H12)

1.2 Visualisation-Basics

Definition: Visualisation the use of computers or techniques for comprehending data or to extract knowledge from the results of simulations, computations, or measurements (not manually by humans)

Definition: Information Visualization the communication of abstract data through the use of interactive visual interfaces.

- Combine different kinds of information in one graphic (geographical, temporal, historical, numeric, etc.)
- Sharing and visualising abstract data, without physical representation
- Visualisation is not:
 - **Scientific Visualization** Visualization of data with a concrete physical representation (non-abstract data)
 - **Computer Graphics** Technical and mathematical aspects of visualization
 - **Graphic Design** Aesthetic graphical representation
- **Example** Treemap
 - representation of a hierarchy of a filesystem
 - no border used for a square (compression)
 - light effect shows curvature, indicating where the squares/areas end
 - \Rightarrow only 4 pixels needed instead of 9
 - Several drawbacks (alternative: tree view)

Abstract Data

- Text and Tables
- Hierarchies and Graphs
- Composed data (Multivariate data): Example Napoleon (Slide 1)
- Time series: multivariate data with time as a dimension

Visualisation process

- graphical user interface
- interaction to create and manipulate the visualisation (**Visual steering**)

2 2. Lecture

2.1 Diagrams

Simple Diagrams

- Line Charts
- Bar Charts
- Pie Charts

Pie charts

- applicable to part-whole relation
- Several issues
 - difficult to compare values within a chart
 - difficult to compare differences between pie charts

Other Diagrams

- **Timelines** align temporal information along an axis
- **Sparklines** Reduced to show trend and the change of values over time - a sparkline is a small intense, simple, word-sized graphic with typographic resolution.

2.2 Metaphors and Symbols

Make constructs/concepts more accessible/imaginable

2.3 Symbols/Pictograms

highly simplified representation of objects and activities. Very suitable for depicting metaphors

Isotype using pictograms to convey statistical information. Quantity is better represented by the number of pictograms than by the size of a pictogram.

2.4 Infographics

Definition Infographics Information graphics or infographics are graphic visual representations of information, data or knowledge. These graphics present complex information quickly and clearly, such as in signs, maps, journalism, technical writing, and education.

- Eyecatcher to get people interested in the presented data
- Contain few text
- Self-explanatory
- Should tell a **story** \Rightarrow express an opinion

Elements of an Infographic

- Story
- Graphics
 - Illustrative
 - Simplified
- Text
 - Keywords and short texts
- Diagrams
 - Connected to graphics.

3 3. Lecture

3.1 Visual Memory

- The brain fills empty gaps
- Distraction by environment (contrast/structure)
- \Rightarrow visual perception is selective

3.2 Visual Information Processing

3 Phases of processing

1. Simple patterns and colors are recognized
2. Action system: reflexes
3. Visual working memory/visual query

Human Eye

Usage of the properties of visual perception (Anticipation, pattern recognition)

- Eye Tracking (works by measuring the reflection from the eye's curvature)

3.3 Color Perception

3-Color-Theory

- Each color consists of rgb

Opponent-Color-Theory

- After image effect: color-receptors are getting exhausted, so white cannot be 'produced'
- three chemical processes with two opponent colors each
- Color is perceived by the difference between the opponent colors

⇒ Color and brightness are relative

Design Recommendations

- Emphasize with color
- Differences with brightness
- Coding of categories: max 6 to 12 different colors
- Color scales should vary in color and brightness
- Color perception depends on culture
- Motion to grab attention/indicate a relation
- Strong colors/contrast can cause inertia (ghost images)

3.4 Preattentive vision

- Detect patterns before an eye movement
- Motion is preattentive
- ⇒ Use preattentive patterns to encode information (spot an outlier)

3.5 Pattern Recognition

- Edge detection
- Simple patterns (detect small distortions)
- Complex patterns
- Object recognition (compare observation with learned patterns to recognise an object)

3.6 Motion recognition

Different elements perform similar motions

- Recognize patterns to identify object
- Recognize change after each frame
- Movements seem related, when they are in synch
- \Rightarrow Indicate a relation with a synchronous animation
- Motion can induce causality

4 Lecture

Visualization of Graphs: **Graph drawing**

Application

- Map-drawing: indicate multiple data sets in one map (London Underground)
- Ego(-centric) network: graph with personal connections

Visual Encoding

- Thickness, color of edges
- Color of nodes

Aesthetic Criteria Readability does not induce aesthetic

- min edge crossings
- min drawing
- min edge length
- min number of bends
- max symmetry
- uncover clusters
- max continuity amongst paths

4.1 Layouting algorithms

Radial Layout

- fair node weight, every node's representation is equal
- lots of edge crossings
- applicable, if there is no further info about the data

Force-Directed Layout

- force edges to a certain length
- reorder nodes
- try to find equilibrium, where the forces cancel out each other

Hierarchical Layout

- for cyclic structures: flip the edges that close the cycle while drawing the graph
- depth first search provides a topological ordering of the nodes
- sort nodes on the lower layer until the bottom is reached, then go back to start
- to have a clean layout, put in dummy nodes as a spacer

Orthogonal Layout

- edges follow grid (orthogonal paths)
- shape metrics
 - describe the path the edges take by turns
 - evaluate the paths

Edge Bundling

- structured radial layout
- bundle edges with the same direction

4.2 Matrix visualization of Graphs

Adjacency Matrix

- indicate an edge in a matrix
- uncovering clusters is hard

Layouting

Compound graphs

5 Lecture

5.1 Visualization of dynamic graphs

Dynamic graph: sequence of graph states

Animation see difference between layout and data changes to preserve the mental map of the graph. Examples:

- TimeLine, horizontal development of the graph, vertical orientation of the graph
- TimeSpiderTrees, circular layout, each ring is one graph
- TimeRadarTrees, circular layout, outer circles are a representation of the inner. The inner circle shows incoming edges, the outer shows outgoing

5.2 Multivariate data and time series

Boxplots box showing 50 percent of data, outer borders not standardized

Fan Chart wide part shows the mean (similar to the box plot)

Histogram bar represents a range of values (value range split into intervals)

6 Lecture

6.1 Software Visualization: Architecture

Pipes and Filters Input stream providing data, putting it into a pipe of filters

Layered Systems Layers provide functionality of upper layers (radial or stacked). Radial: small core, Pyramid: neutral representation

Blackboard-driven Different processes share info on one blackboard

6.1.1 Reverse Engineering

Create higher level of abstraction for a given system and automatically create architecture visualization. The detection of design patterns is non-trivial. To detect, the program is run and traced.

6.1.2 Enriched Node-Link Diagrams

Visualize/Encode software metrics. Aggregation of information to simplify.

Class Blueprint Categorize methods by name and access attributes (public/protected/private...)

Depenecies Viewer Visualize package graph and dependencies between packages and methods

Dependency Structure Matrix DSM Detect cycles and indirect cycles with highlighting

Software Citites and Maps 2D plane represents system. Hierarchy shown with trees/dimesions. 3rd dimension can be used to show other metrics, like evolution/age/dependencies

Summary Ad-hoc diagrams hard to understand without explanation. With reverse engineering automatic creation for specific techniques are possible