

Informationsvisualisierung

Zusammenfassung

Benedikt Lüken-Winkels

July 9, 2019

Contents

1	Einführung	3
2	Infographiken	3
2.1	Diagramme	3
2.2	Metaphern und Symbole	4
2.3	Infographiken	4
2.3.1	Gegenüberstellung Infographik und Informationsvisualisierung . .	4
3	Visuelle Wahrnehmung	4
3.1	Verarbeitung visueller Informationen	4
3.2	Farbwahrnehmung	5
3.3	Mustererkennung	5
3.4	Dreidimensionale Wahrnehmung	6
3.5	Design Empfehlungen	6
4	Visualisierung von Hierarchien	7
4.1	Node-Link	7
4.2	Eingerückte Gliederungs Plots	7
4.3	Eiszapfen Plot (Icicle Plot)	7
4.4	Treemap	8
4.5	Verwendung in der Praxis	8
5	Visualisierung von Graphen	8
5.1	Layout Algorithmen	9
5.2	Matrix Visualisierung von Graphen	10
5.3	Visualisierung von dynamischen Graphen	10
6	Multivariate Daten	10
6.1	Statistische Analyse	10

6.2	Tabellen	11
7	Software Visualisierung	12
7.1	Code Diagramme	12
7.2	Architektur	12
7.3	Reverse Engineering	13
7.4	Verhalten von Software	14
7.4.1	Dynmaische Programm Visualisierung	14
7.4.2	Animierung von Algorithmen	14
7.4.3	Visuelles Debugging	15
7.5	Entwicklung von Software	15
7.5.1	Visuelles Data-Mining von Softwarearchiven	16

1 Einführung

Definition Informationsvisualisierung ist die Kommunikation von abstrakten Informationen durch interaktive visuelle Schnittstellen.

Abgrenzung zur Visualisierung Informationsvisualisierung ist nicht

- Wissenschaftliche Visualisierung: Darstellung nicht-abstrakter Informationen mit realen physischen Representationen. (Röntgenbild)
- Computer Graphik: Technischer und Mathematischer Aspekt von Visualisierung
- Graphik-Design: Ästhetische graphische Darstellung

Darstellungsmöglichkeiten von abstrakten Daten oder Informationen

- Text und Tabellen
- Hierarchien und Graphen
- Multivariate Daten (Mehrdimensionale Daten)
- Zeitreihen (Multivariate Daten, wobei die Zeit eine besondere Dimension darstellt)

2 Infographiken

2.1 Diagramme

Einfache Beispiele von Diagrammen

- Linien-Diagramm
- Balken-Diagramm
- Kuchen-Diagramm
 - Gut bei Part-Whole-Relationen
 - Tatsächliche Werte/Kategorien sind schwer zu vergleichen
- Zeitreihen
- Sparkreihen: Zeitreihen, reduziert um Trends darzustellen

2.2 Metaphern und Symbole

Metaphern sollen Konstrukte und Konzepte vereinfacht darstellen und zugänglicher machen. Beispiele:

- Städte (Cluster in Zusammenhängen)
- Bäume (Hierarchien)
- Tiere (Vererbung)

Symbole stellen stark vereinfachte Darstellungen von Sachverhalten oder Objekten dar, die unter anderem auch Metaphern darstellen können (Papierkorb Icon für gelöschte Elemente).

Isotyp Darstellung von statistischen Informationen durch Piktogramme (Symbole). Größe der Zahl wird durch Anzahl an Symbolen kodiert.

2.3 Infographiken

Infographiken sind graphische Representationen von Informationen, Daten oder Wissen, die komplexe Informationen schnell und leicht zugänglich machen sollen.

2.3.1 Gegenüberstellung Infographik und Informationsvisualisierung

Infographik

- Von Hand geschrieben
- Selbsterklärend
- Erzählt eine Geschichte
- Meistens Ad-Hoc
- Kann voreingenommen sein

- Automatisch generiert
- Auf verschiedene Datensätze anwendbar
- Nicht unbedingt selbsterklärend

Informationsvisualisierung

3 Visuelle Wahrnehmung

Visuelle Wahrnehmung ist selektiv, interpretierend und kann durch die Umgebung (Kontrast/Struktur) abgelenkt werden (nicht wie eine Kamera)

3.1 Verarbeitung visueller Informationen

Dreiphasenmodell

1. Primitive Mustererkennung

2. Aktionen/Reflexe, Komplexe Mustererkennung
3. Visuelles Arbeitsgedächtnis

Menschliches Auge Man kann verschiedene Eigenschaften der visuellen Wahrnehmung ausnutzen.

- Antizipation von Bewegungen: Vorhersehen von Ereignissen
- Mustererkennung: Verdeutlichen von Clustern

Periphere Schärfe Die Schärfe mit der Text lesbar oder Objekte erkennbar sind, nimmt abhängig vom Zentrum ab:

- Zentrum: Farbe und Helligkeit sind klar erkennbar.
- Rand/Peripherie: Unschärf und nur Helligkeit ist erkennbar.

3.2 Farbwahrnehmung

Opponent Color Theory. Gegenfarbtheorie Es ist einfacher und effizienter die Farben anhand der Unterschiede zwischen benachbarten Farben zu erkennen. 3 chemische Prozesse mit jeweils 2 Gegenfarben sorgen für Farbidentifizierung. \Rightarrow Farbe und Helligkeit sind relativ.

3.3 Mustererkennung

Kategorien von Mustern

- Kantenerkennung (Farben, Helligkeit)
- Einfache Muster (Verbindung durch Kanten oder Rotation)
- Komplexe Muster (Muster in Mustern, Cluster)
- Objekterkennung (Icons, Symbole)

Präattentive Elemente Ein Element in einer Gruppe ähnlicher Elemente mit herausstehenden Eigenschaften kann schnell erkannt werden. **Bewegung ist präattentiv**

Bewegungserkennung Bewegung kann auf einen Zusammenhang hinweisen.

Visuelle Suche Folgt einem Zyklus:

1. Erkenne Muster
2. Wähle einen Kandidaten aus
3. Schließe vorherige Ziele aus
4. Bewege das Auge

Gestalt Psychologie Gesetze, die helfen, Gruppierungen leichter erkennbar zu machen.

- Räumliche Nähe
- Ähnlichkeit in Gestalt oder Form
- Verbindungen (durch Kanten in einem Graphen)
- Fortläufigkeit (Continuity, zB nicht unterbrochene Linien)

3.4 Dreidimensionale Wahrnehmung

Problematisch bei der Visualisierung von abstrakten Informationen.

Hauptprobleme von 3D

- Schwierige Navigation
- Ausschluss von Information im Hintergrund

2,5D Darstellung Kombiniert die Vorteile von 2D und 3D. Dritte Dimension bietet Schatten oder Perspektive. Beispiele:

- Cushion treemap
- Cone Trees
- UML Geons
- Perspective Wall

3.5 Design Empfehlungen

- 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)
- Use yellow/blue variations for colorblinds

4 Visualisierung von Hierarchien

Hierarchische Daten/Informationen als Baumstruktur

- Keine Zyklen
- Es gibt genau einen Weg von der Wurzel zu einem beliebigen Knoten.
- \Rightarrow Verbundener, ungerichteter Graph

4.1 Node-Link

Darstellung eines Baums, mit Knoten als Punkte und Kanten als Linien.

Vorteile:

- Intuitiv
- Hierarchie schnell erkennbar
- Flexibles Layout

Nachteile:

- Kanten brauchen Platz
- Degenerierte Bäume sind schwer darzustellen.

4.2 Eingerückte Gliederungs Plots

Darstellung eines Baums in einer Liste

Vorteile:

- Leicht lesbar
- Allgemein bekannt
- Darstellen degenerierter Bäume möglich

Nachteile:

- Unflexibles Layout

4.3 Eiszapfen Plot (Icicle Plot)

Hierarchie als Teil-Ganzes Beziehung

Vorteile:

- Labelling ist einfach
- Effektive Nutzung von Bildschirmplatz

Nachteile:

- Nicht sehr intuitiv
- Breite der Kindelement wird durch Breite der Vater-elemente beschränkt

4.4 Treemap

Elemente als Flächen (Venn Diagramm - artig)

Vorteile:

- Kaum Platzverbrauch für Kanten
- Platz der Blattknoten kann verbraucht werden

Nachteile:

- Labelling schwierig
- Rand eines Elements muss berücksichtigt werden und ins Layout mit einbezogen werden. (2,5D)

4.5 Verwendung in der Praxis

Empfohlene Layouts: Node-Link, Einzapfen, Eingerückte Gliederung. Nicht empfohlen, weil unintuitiv und meistens schwer lesbar: Treemap und Radial Layouts.

Vergleich von Hierarchien Aufbau als Matrix, farbliche Hervorhebung, Linienverbindungen

5 Visualisierung von Graphen

Anwendung

- Karten: Multivariate Daten in einer Karte (London Underground: Verbindungen und geographische Lage)
- Ego/Personen-Netzwerke: Persönliche Verbindungen als Graph (Facebook)

Visuelle Kodierung

- Dicke einer Kante: Kantengewicht
- Pfeil: Kantenrichtung
- Verschachtelte Boxen: Hierarchie
- Knotenlabel: Knotennamen

Ästhetische Kriterien

- Minimale Kantenüberschneidung
- Minimale Bemalte Fläche
- Minimale Kantenlänge
- Minimale Anzahl an Biegungen in Kanten
- Maximale Symmetrie
- Verdeutlichung von Clustern
- Fortlaufende Winkel an den Kanten

5.1 Layout Algorithmen

Radial Layout

- Gleichverteilung im Knotengewicht
- Viele Kantenüberschneidungen
- Gut anwendbar, wenn keine Metadaten vorhanden sind

Force-Directed Layout

- Kanten werden wie Federn in eine bestimmte Länge geforcet
- Equilibrium, wo sich Kantenkräfte gegenseitig ausbalancieren

Hierarchisches Layout

- Problem beim Zeichnen von Zyklen:
 - Drehe die Kanten um, die den Zyklus schließen
 - Male dann den Graphen
 - Drehe die Kanten wieder um
- Tiefensuche ergibt die topologische Ordnung der Elemente
- Dummy Knoten, die nach dem Erstellvorgang entfernt werden ergeben ein sauberes Layout

Orthogonales Layout

- Kanten folgen einem Raster
- Metriken für die Form: zB Je weniger Wendungen in einem Pfad, desto besser

Kantenbündelung

- Strukturiertes Radial Layout
- Kanten, die in die selbe Richtung gehen werden gebündelt

5.2 Matrix Visualisierung von Graphen

Adjazenzmatrix $N \times N$ -Matrix, mit Einfärbungen, wenn eine Kante existiert. Clustererkennung ist schwer. \Rightarrow Sortierung der Knoten

5.3 Visualisierung von dynamischen Graphen

Zeitliche Abfolge von Graphzuständen

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

- TimeArchTrees, 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

6 Multivariate Daten

Daten mit mehreren Informationen/Informationstupeln pro Objekt. Leichte Darstellung in einer Tabelle.

6.1 Statistische Analyse

Maßstäbe zur Analyse

- Mittelwert
- Median: Mittelwert der Werte bei sortierter Reihenfolge
- Quartile: oberes/unteres Quartil 25%/75% der Werte
- Modus/Modalwert: häufigster Wert
- Standardabweichung
- Standardfehler: Standardabweichung der Mittelwerte

Skalenniveau/Scale of measure Skalen können verschieden gemessen und verglichen werden.

- Nominal Scale \neq : Geschlecht
- Ordinal Scale $\neq < >$: Noten
- Interval Scale $\neq < > + -$: Temperatur
- Ratio/Verhältnis Scale $\neq < > + - \cdot \div$: Länge/Geometrische Maße

Boxplots Erweiterung eines Barcharts, mit zusätzlichen Informationen über Median und Quartile und Ausreißern

Fan Chart Zeigt Median und Quartile als Fächer in einem Kreis.

Fan Chart Map 2D Matrix von Fan Charts, um Tendenzen in der Entwicklung und den Zusammenhang der Variablen zu verdeutlichen.

Histogramm Bar Chart der Werte. Die Werte werden in Intervalle aufgeteilt

Dimensionalitätsreduktion Bei einem Datensatz mit vielen verschiedenen Informationen pro Objekt wird die Visualisierung von Gemeinsamkeiten schwierig. Durch Dimensionalitätsreduktion werden verschiedene Eigenschaften kombiniert. Methoden, um dies zu erreichen sind:

- Finden von Korrelationen
- Multidimensionale Skalierung
- Faktorenanalyse. Verschiedene Eigenschaften, die die größte Variation im Datensatz ergeben werden zu einem Faktor zusammengeschlossen

Clustering Herausheben von Objekten, die zu einer Gruppe gehören.

- Räumliches Gruppieren
- Partitionierung des Datensatzes: k-means
- Hierarchisches Clustering

6.2 Tabellen

Scatterplot Zusammenhang zweier Variablen

Anscombe Quartett Verdeutlicht, dass die graphische Analyse von Datensätzen genauso wichtig ist, wie die Erhebung von Statistischen Analysen (Mittelwert/Standardabweichung). Das Quartett zeigt 4 Datensätze mit identischen Werten bei der deskriptiven statistischen Analyse, die allerdings sehr verschiedene Werte haben.

Parallele Koordinaten

Star Plots

Chernoff Faces

7 Software Visualisierung

Definition: Visualisierung des Aufbaus, Verhaltens und der Entwicklung von Software.

7.1 Code Diagramme

Jackson Diagramme Aufbau der Aktionen als Baumstruktur.

Struktogramme Verschachtelte Boxen verdeutlichen den Programmablauf (von oben nach unten)

Kontrollflussgraph CFG

- Knoten: Statement oder Bedingung
- Kante: Flussrichtung, Pfeil zur nächsten Operation

Kontrollflussdiagramm

7.2 Architektur

Häufige Architekturmodelle:

- Pipes und Filter: Input/Output Stream. Ausgaben werden in den nächsten Filter gepipet
- Layered Systems: Verschiedene Schichten verwenden die Funktionalitäten der oberen Schicht.
- Blackboard-driven: Verschiedene Prozesse greifen auf eine Quelle zurück

Ad-Hoc Diagramme Diagramme die einen bestimmten Sachverhalt oder Programmverlauf beschreiben. Folgen keiner Standardnotation und sind ohne Erklärung selten verständlich.

UML Diagramme Standardisierte Darstellungen für verschiedene Aspekte in der Softwarearchitektur

- Use Case Diagram
- Class Diagram
- Behaviour Diagram
- Interaction Diagram
- Implementation Diagram
- Model Management Diagram

7.3 Reverse Engineering

Definition: Analyse eines Systems, um Repräsentationen auf einer höheren Abstraktionsebene zu erstellen. Problem dabei ist, dass die Erkennung von Designpatterns sehr schwer ist.

Class Blueprint Aufsplitten einer Klasse in

- Initialisierung (init-Methoden)
- Interface-Methoden (Methoden, die von anderen Klassen verwendet werden)
- Implementierungen (private Methoden)
- Accessors (Getter/Setter)
- Attributen

Dependency Viewer Package Graph, der Dependencies zwischen Packages veranschaulicht, um Designfehler zu zeigen.

Dependency Structure Matrix Adjazenzmatrix, die die Dependencies einer Software darstellt. Zyklenerkennung möglich

Software Städte Fläche repräsentiert das System. Dritte Dimension kann verschiedene Metriken darstellen, wie Dependencies oder Größe von Paketen.

7.4 Verhalten von Software

7.4.1 Dynamische Programm Visualisierung

Erfassung von Daten Viele Methoden haben das Problem, dass sie direkt in den Kontrollfluss des Programms eingreifen und so möglicherweise den Ablauf oder die Laufzeit beeinflussen, die gemessen werden soll. Methoden zur Erfassung sind:

- Instrumentation: Vor/Nach jeder Operation, wenn eine Datenstruktur geändert wurde oder an bestimmten Programmpunkten wird eine Messung vorgenommen
- Messen in einem parallelen Thread, der den Speicher liest
- Besondere Hardware

Dynamische Datenerfassung Erfassung von Daten zur Laufzeit: Programm positionen, Werte von Variablen, Inhalt des Heaps

7.4.2 Animierung von Algorithmen

Anwendungen

- Verständnis der Vorgehensweise
- Design
- Optimierung
- Debugging

⇒ Problem: Finden des richtigen Darstellungsmodells (Korrekte Darstellung der Übergänge zwischen Zuständen)

Architekturen für Graph Animationen

- Ad Hoc: Ohne Verwendung von Libraries (from Scratch)
- Special Datatypes: Algorithmen verwenden Datentypen mit eingebauten Visualisierungen
- Interesting Events: Einfügen von Aktionen an bestimmten Punkten im Code
- Deklarativ: Teilung von Code und Annotation. Annotation bietet Code für einen Demon, der Datenänderungen anzeigt
- Post-Mortem Visualisierung: Aufnahme der Änderungen während der Ausführung in einem Log. Anschließend wird die Ausführung animiert.
- Semantik-gerichtet: Automatische Animation durch Interpreter der Programmiersprache

7.4.3 Visuelles Debugging

Memory Graph Darstellung des Speichers eines Programms. Kann sehr groß werden \Rightarrow Fokussierung auf Subgraphen

Memory Slices Forward Memory Graph:

Forward Reference Pattern

- Zugriffsgraph (Baum)
- Keine Erkennung von Zyklen

Static Slice How **can** a variable be changed by other code points. Slice is a small part

Dynamic Slice How **is** a variable actually changed by other code points. Slice is a small part

Execution Slices Sequence of program points.

Dice Difference of two Slices.

X-Slice (Heuristic) Compare a run with failing and compiling input. Only the failing program points are highlighted. Color coding coverage data by failure probability and evidence for failure.

Test Blueprint Highlight non-executed program points in the Class Blueprint.

7.5 Entwicklung von Software

Synonym für den Entwicklungsprozess einer Software.

Software Archive version control/collection of the history of a program of any kind.

Color-coding

- Line Representation: indentation/different metrics
- Code Age: when was a file/line changed
- Pixel Representation
- Version-specific Code: highlight eg platform specific code
- Depth of nested blocks
- CVSScan: different versions for a file with LOC as bar height.

Evolution Matrix Classes are represented as boxes. Box height and width encode a certain metric. \Rightarrow No insight on program structure

Call Graph Which function calls which function (low level info). Encode program structure. Edge splatting (the more often an edge is drawn the more intense its color gets) shows call clusters.

7.5.1 Visuelles Data-Mining von Softwarearchiven

Gesetze der Softwareentwicklung

- Lehman: Größe, Komplexität und Funktionalität sind antiproportional zur Wachstumsrate und der Codequalität eines Programms
- McCabe: Dichte des Callgraphen ist ein guter Indikator für Bugdichte
- Basili und Möller: Kleine Änderungen haben eine höhere Fehlerdichte

Data Mining Process Startgin with a version control program (git)

1. Analysis
2. Extraction
3. Data Mining
4. Visual Data Mining

Coupling

- Evolutionary Coupling artifact are related, when they are changed together.
- Logical Coupling artifacts are related, when they are programmatically calling each other.