

Semantik-gestütztes Hilfesystem für ein komposites Informationsvisualisierungssystem

Diplomarbeit
Technische Universität Dresden
November 2013

Nikolaus Piccolotto

Betreuer: Dipl.-Medieninf. Martin Voigt
Hochschullehrer: Prof. Dr.-Ing. Klaus Meißner

Fakultät Informatik
Institut für Software- und Multimediatechnik
Seniorprofessur für Multimediatechnik



Aufgabenstellung

Diese Seite muss vor dem Binden der gedruckten Fassung der Arbeit durch die von Herrn Meißner und dem Studenten eigenhändig unterschriebene originale Aufgabenstellung ersetzt werden. Das zweite abzugebende gebundene Exemplar soll stattdessen eine Kopie dieser originalen Aufgabenstellung enthalten.

Erklärung

Hiermit erkläre ich, Nikolaus Piccolotto, die vorliegende Diplomarbeit zum Thema

Semantik-gestütztes Hilfesystem für ein komposites Informationsvisualisierungssystem

selbstständig und ausschließlich unter Verwendung sowie korrekter Zitierung der im Literaturverzeichnis angegebenen Quellen und Hilfsmittel verfasst zu haben.

Dresden, 30. November 2013

Unterschrift

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Problemstellung und Zielsetzung	1
1.3	Aufbau der Arbeit	1
2	Stand der Forschung und Technik	2
2.1	Grundlagen	2
2.2	Semantische Daten	2
2.3	Informationsvisualisierung	2
2.4	Software Support	2
2.5	Szenario	2
3	Konzept	4
3.1	Anforderungsanalyse	4
3.1.1	Funktionale Anforderungen	4
3.1.2	Nichtfunktionale Anforderungen	4

Abbildungsverzeichnis

1.1 VizBoard Workflow 1

Tabellenverzeichnis

1 Einleitung

1.1 Motivation

1.2 Problemstellung und Zielsetzung

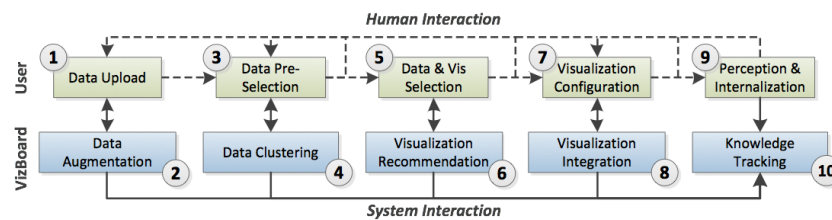


Abbildung 1.1: VizBoard Workflow

1.3 Aufbau der Arbeit

2 Stand der Forschung und Technik

2.1 Grundlagen

2.1.1 Semantische Daten

2.1.2 Informationsvisualisierung

2.1.3 Software Support

2.2 Szenario

Wie in Kapitel 1 erläutert, ist das komposite InfoVis-System Teil der Anwendung VizBoard. Sie leitet den Benutzer in mehreren Schritten von der Auswahl eines Datensatzes zur finalen, kompositen Informationsvisualisierung. Im vorletzten Schritt wählt dieser mit Hilfe eines Facettenbrowsers geeignete Visualisierungskomponenten aus, welche danach angezeigt werden. Um die Problemstellung noch einmal zu verdeutlichen, wird im folgenden ein mögliches Szenario beschrieben.

Alice möchte für ihr Biologiestudium mehr über die geografische Verteilung verschiedener Genvariationen herausfinden. Dazu sucht sie im Internet nach einem Datensatz, welchen sie auch findet. Leider ist er in einem für Alice unbekannten Format abgespeichert, nämlich OWL. Außerdem ist er ca. 30 MB groß, das ist zu viel um es manuell zu verarbeiten. Davon abgesehen sind geografische Breite und Länge als Zahlenkombination keine anschauliche Repräsentation von Orten. Alice hört von einem Freund, dass VizBoard gut geeignet ist um semantische Datensätze anzusehen und probiert es aus.

Nachdem Alice ihren Datensatz auch bei VizBoard gefunden und die gewünschten Visualisierungskomponenten (eine Karte, ein Balkendiagramm, eine Tabelle und eine Treemap) ausgewählt hat, werden diese ihr angezeigt. Alice benutzt VizBoard zum ersten Mal und macht außer Facebook und YouTube auch sonst nicht viel im Internet, das heißt sie ist zunächst von den vier unterschiedlichen Fenstern etwas überfordert.

VizBoard bietet Alice aber sofort eine einführende Übersicht und erklärt kurz die Darstellungsform und den Inhalt jeder Komponente. Eine denkbare Erklärung der Treemap wäre zum Beispiel:

Eine Treemap ist eine hierarchische Visualisierung um Größenverhältnisse anschaulich zu machen. In dieser werden die Anzahl von Genvariationen pro geografischer Region dargestellt.

Damit bekommt Alice einen Überblick über die verfügbaren Visualisierungen und weiß, welches Fenster welche Visualisierung enthält und für was diese gut sind. Nun möchte sie die Tabelle, in der die durch die Treemap visualisierten Daten stehen,

nach der Spalte »Anzahl« sortieren. Alice sieht aber nicht, wie sie das machen soll, da in der Tabelle kein offensichtliches Kontrollelement wie z. B. ein Button vorhanden ist. Sie bemerkt ein Fragezeichen in der Titelleiste des Fensters und klickt darauf. Der verfügbare Viewport wird abgedunkelt und es erscheint ein neues Fenster, welches die verfügbaren Aktionen mit Hilfe von Text, Bildern und Animationen erklärt. Alice lernt, dass sie mit einem einfachen Linksklick auf den jeweiligen Kopf einer Tabellenspalte nach dieser sortieren kann und außerdem eine oder mehrere Zeilen auswählen kann. Sie sortiert die Tabelle wie gewollt und wählt die ersten drei Zeilen aus. Plötzlich verkleinert die Karte ihr Zoomlevel und Alice ist verwirrt: Sie hat nur mit der Tabelle interagiert und es bestand keine sichtbare Verbindung zwischen den beiden Fenstern. Allerdings wurde nach der Zeilenauswahl ein Pfeil von der Tabelle zur Karte gezeichnet, welcher mit einem Icon in Form eines Briefes versehen ist. Alice vermutet, dass doch irgendeine Verbindung zwischen den beiden Visualisierungen besteht und klickt auf den Brief. Ähnlich wie vorhin bei der Hilfe zur Tabelle wird der Viewport abgedunkelt und ein neues Fenster wird eingeblendet. Es erklärt die Kommunikation zwischen den Visualisierungen mit Hilfe von Animationen, Text und Bildern. Nun weiß Alice auch, wie die verschiedenen Fenster zusammenhängen und kann sich ihrer eigentlichen Aufgabe widmen.

In der Tabelle findet sie auch eine Spalte »SNP«. Alice weiß zwar, dass sie die Abkürzung schon einmal gesehen hat, kennt aber im Moment ihre Bedeutung nicht. Praktischerweise ist der Spaltenkopf mit der Wikipedia verlinkt und sie wird sofort auf die entsprechende Seite weitergeleitet. Alice erinnert sich, dass »SNP« »Single-nucleotide polymorphism« bedeutet und sie bekommt auch gleich zusätzliche Informationen zu diesem Thema. Sie widmet sich weiter der Tabelle und stellt fest, dass die Ortsbezeichnung »Kopenhagen« nicht mit der Markierung in der Karte übereinstimmt. Außerdem ist sie erstaunt, wie hoch die Verbreitung eines bestimmten SNPs dort ist und würde gerne die Ursache dafür wissen. In der Hilfe zur Tabelle wurde sie auch über die Möglichkeit, Kommentare an den Daten vorzunehmen, aufgeklärt. Alice kommentiert sowohl die falschen Geokoordinaten als auch ihre Frage über die Verbreitung des SNPs, sodass sie später per Email über Antworten informiert wird. Nun kann Alice sich mit der vierten Visualisierung, dem Balkendiagramm, beschäftigen. Allerdings reagiert es auf keine Mausklicks und macht auch sonst nicht den Eindruck, die Daten akkurat darzustellen. Alice meldet die kaputte Visualisierung über die eingebaute Reporting-Funktion und schließt das Fenster, um sich den anderen drei Visualisierungskomponenten zuzuwenden.

3 Konzept

3.1 Anforderungsanalyse

Aus dem Szenario (Kapitel 2.5) lassen sich nun verschiedene Anforderungen an ein Hilfesystem für komposite Informationsvisualisierungssysteme ableiten.

3.1.1 Funktionale Anforderungen

Blabla

- **Überblick:** Das Hilfesystem soll einen kurzen Überblick über das InfoVis-System geben und Darstellungsform sowie Inhalt jeder Komponente kurz erläutern.
- **Bedienung:** Das Hilfesystem soll erklären können, wie eine Komponente bedient wird. Diese Informationen umfassen beispielsweise welche Operationen auf welchen DOM Elementen welche Aktionen (eventuell auf welchen Daten) ausführen.
- **Reporting:** Fehler in Komponenten sollen über ein Reporting-System gemeldet werden können.
- **Verlinkung:** Das Hilfesystem soll nicht-triviale Begriffe mit der Wikipedia verlinken, sodass nicht nur auf die Begriffsbedeutung hingewiesen werden, sondern dem Benutzer auch zusätzliche Informationen zur Verfügung gestellt werden können.
- **Kommunikation:** Das Hilfesystem soll erklären können, wie zwei gegebene Komponenten miteinander kommunizieren.
- **Kommentare:** Der Benutzer soll die Möglichkeit haben Daten zu kommentieren und Bereiche der Visualisierung zu markieren und mit ebenfalls mit einem Kommentar zu versehen, sodass auch auf fehlende Daten hingewiesen werden kann.

3.1.2 Nichtfunktionale Anforderungen

Bla bla

- **Korrektheit:** Eine gegebene Hilfestellung darf keine Fehlinformationen enthalten, weil sie sonst mehr verwirrt als hilft.

- **Vollständigkeit:** Eine gegebene Hilfestellung muss alle Informationen enthalten, die der Nutzer benötigt um danach seine gewünschte Aufgabe ausführen zu können.
- **Verständlichkeit:** Hilfestellungen müssen in einer Form präsentiert werden, die der Benutzer schnell und mit geringem mentalen Aufwand verarbeiten kann.
- **Einheitlichkeit:** Das Look & Feel von Teilen des Hilfesystems (z.B. Kommentare) muss komponentenübergreifend einheitlich sein, damit der Benutzer einmal gelerntes wiederverwenden kann.
- **Minimalität:** Der Komponentenentwickler soll seine Komponente mit möglichst wenig Aufwand zum Hilfesystem kompatibel machen können, ansonsten werden nur sehr wenige Komponenten – und damit der Benutzer – davon profitieren.
- **Universalität:** Das Hilfesystem soll für alle Komponenten und Visualisierungen in gleicher Qualität funktionieren.
- **Wiederverwendbarkeit:** Die Kommentare sollen möglichst in allen Visualisierungen wiederverwendet werden, damit viele Benutzer von den Erkenntnissen anderer profitieren können.