

# Der CircleSegmentView: Design und Evaluation einer Query Preview

Alina Bey

Bachelorarbeit  
zur Erlangung des Bachelorabschlusses in Information Engineering  
Universität Konstanz  
Fachbereich Informatik und Informationswissenschaft  
Arbeitsgruppe Mensch–Computer–Interaktion

1. Gutachter: Prof. Dr. Harald Reiterer
2. Gutachter: Prof. Dr. Oliver Deussen

Januar 2004

## Abstract (deutsch)

Die vorliegende Bachelorarbeit befasst sich mit der Implementierung und Evaluierung einer kreisdiagrammbasierten Visualisierung, dem CircleSegmentView.

Der CircleSegmentView ist Bestandteil des visuellen Metadatenbrowsers VisMeb, der an der Universität Konstanz im Fachbereich Informatik und Informationswissenschaften in der Arbeitsgruppe Human-Computer-Interaction unter Leitung von Prof. Dr. Harald Reiterer entwickelt wurde.

Mit dem CircleSegmentView soll eine vielseitig einsetzbare Query Preview basierte Visualisierung zur Verfügung gestellt werden, die den Benutzer bei der Suche in großen Datenbanken unterstützen soll.

Zunächst werden grundlegenden Konzepte der Dynamic Queries und Query Previews erläutert und der Bezug zum CircleSegmentView hergestellt, gefolgt von der Beschreibung des Projekthintergrunds und der Entwicklungsgeschichte des CircleSegmentView.

Die Benutzeroberfläche und die Funktionsweise des CircleSegmentView werden detailliert erklärt.

Bezüglich der Implementierung des CircleSegmentView wird der Aufbau des Source-Code erklärt und die Umsetzung der wichtigsten Leistungsmerkmale vorgestellt.

Der Schwerpunkt der Erörterungen liegt auf der Evaluierung der Oberfläche. Es wurde ein Benutzertest mit 20 Testpersonen durchgeführt, mit welchem gezeigt werden sollte, dass die Arbeit mit dem CircleSegmentView im Vergleich mit einem herkömmlichen Suchformular einen Zeitvorteil bei der Suche bringt. Bei einigen Testaufgaben konnte nachgewiesen werden, dass die Testpersonen die jeweiligen Aufgaben statistisch signifikant schneller lösen konnten.

Abschließend wird ein Konzept für eine zweite Benutzerstudie vorgestellt, von deren Durchführung neue Ergebnisse erwartet werden.

## Abstract (english)

The thesis is focussed on the implementation and evaluation of the CircleSegmentView, a visualization based on piecharts.

The CircleSegmentView is part of the visual metadata browser VisMeb. VisMeb is a project of the department of Human-Computer-Interaction at the University of Konstanz.

First the fundamental concepts of dynamic queries and query previews are explained, followed by a description of the project background. An overview over previous works is given afterwards.

The functionality of the CircleSegmentView and its main interface components is explained.

Concerning the implementation of the CircleSegmentView the structure of the source code and the implementation of its main features are described in detail.

Central to the thesis is the evaluation of the user interface. A usability test with 20 testpersons was conducted. The aim of the usability test was to prove a statistically significant faster user performance when comparing the CircleSegmentView with a form fillin interface. Some of the test tasks were solved statistically significant faster with the help of the CircleSegmentView.

To sum up, a concept for a second usability test is presented. The second test is expected to bring further results.

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>8</b>
<b>2</b>	<b>Entwicklung von Query Previews</b>	<b>11</b>
2.1	Überblick . . . . .	11
2.2	Probleme bei der Interaktion zwischen Benutzer und Informationssystem . . . . .	12
2.3	Dynamic Queries . . . . .	15
2.4	Query Previews . . . . .	17
<b>3</b>	<b>Der CircleSegmentView</b>	<b>22</b>
3.1	Projekthintergrund . . . . .	22
3.2	Entwicklungsgeschichte . . . . .	24
3.3	Grundlegende Einordnung des CircleSegmentView als Visualisierung	26
3.4	Beschreibung der Benutzeroberfläche . . . . .	27
3.5	Analyse des CircleSegmentView als Visualisierung . . . . .	35
3.6	Zusammenfassung . . . . .	43
<b>4</b>	<b>Implementierung</b>	<b>44</b>
4.1	Pflichtenheft . . . . .	44
4.2	Systemarchitektur . . . . .	46
4.3	Codearchitektur . . . . .	47
4.4	Leistungsmerkmale . . . . .	51
<b>5</b>	<b>Evaluation des CircleSegmentView</b>	<b>56</b>
5.1	Motivation für die Durchführung des Benutzertests . . . . .	56
5.2	Testmethoden . . . . .	57

---

5.3	Testbeschreibung . . . . .	58
5.4	Ergebnisse . . . . .	65
5.5	Diskussion der Testergebnisse . . . . .	78
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>80</b>
<b>A</b>	<b>Test auf Normalverteilung</b>	<b>86</b>
<b>B</b>	<b>Testmaterialien</b>	<b>90</b>
B.1	Begrüßung . . . . .	90
B.2	Pretest Fragebogen . . . . .	91
B.3	Posttest Fragebogen . . . . .	94
B.4	Testfragen . . . . .	97
<b>C</b>	<b>CD-Rom</b>	<b>98</b>

# Abbildungsverzeichnis

2.1	Schema für die Interaktion des Nutzers mit einer Datenbank . . . . .	13
2.2	Benutzeroberfläche des Dynamic Home Finder, [WS92] . . . . .	16
2.3	Ansicht des Query Preview Interface von NASA Eosdis, [TPS00] . .	18
3.1	Benutzeroberfläche von VisMeb . . . . .	23
3.2	Die Architektur von VisMeb [RLP <sup>+</sup> 03] . . . . .	24
3.3	Ansicht der CircleSegments . . . . .	25
3.4	Ansicht des Einstiegsdialogs von VisMeb . . . . .	28
3.5	Die Metadaten der Filmdatenbank vom Typ Kategorie (Ausschnitt)	29
3.6	Ansicht der Combobox des linken Kreisdiagramms . . . . .	30
3.7	Metadatenauswahl für den vertikalen Slider . . . . .	32
3.8	Radius und Winkel eines Segments . . . . .	33
3.9	Anzeige der Anzahl der links und rechts selektierten Datensätze, sowie der Ergebnismenge . . . . .	34
3.10	Für den Test verwendete Ergebnistabelle . . . . .	35
3.11	Beispiel einer frühen Visualisierung . . . . .	36
3.12	Segmentfarben des CircleSegmentView . . . . .	39
4.1	UML-Klassendiagramm des CircleSegmentView . . . . .	48
4.2	Ansicht eines Segments . . . . .	52
4.3	Listing des Codes für die Platzierung der DataButtons . . . . .	53
4.4	Listing des Codes für die Verknüpfung . . . . .	55
5.1	Testversion des CircleSegmentView . . . . .	62
5.2	Für den Test verwendete Ergebnistabelle . . . . .	63
5.3	Das verwendete Formular . . . . .	64

---

5.4	Ansicht der Versuchsanordnung (nachgestellt). . . . .	65
A.1	Test auf Normalverteilung - Aufgabentyp A . . . . .	87
A.2	Test auf Normalverteilung - Aufgabentyp B . . . . .	88
A.3	Test auf Normalverteilung - Aufgabentyp C . . . . .	89

# Tabellenverzeichnis

5.1	Zeitunterschiede $\Delta t$ bei der Aufgabenerfüllung mit Formular und CircleSegmentView. Ein 'F' steht für nicht oder falsch gelöste Aufgaben.	67
5.2	P-Werte für den Shapiro-Wilk-Test . . . . .	69
5.3	Kritische Werte und Rangsummen für den Wilcoxon-Test . . . . .	74
5.4	P-Werte beim Wilcoxon-Test für die Aufgaben . . . . .	75
5.5	P-Werte für den Mann-Whitney U-Test . . . . .	76



# Kapitel 1

## Einführung

Eine wachsende Fülle an Informationen und Daten aller Art überflutet die Informationsgesellschaft über verschiedenartige und immer leistungsfähigere Medien. Gemäß [LV03] wurden allein im Jahr 2003 weltweit insgesamt 1.9 Millionen Terabyte an elektronischer Information auf Festplatten gespeichert. Diese Informationsflut wird teilweise in Informationssystemen gesammelt oder über das Internet frei zugänglich gemacht [Tan01].

Informationssysteme stellen eine Ansammlung von Prozessen und Abläufen dar, die eine technisch einwandfreie Ausstattung und sowohl menschliche Arbeitskraft als auch die Rechenleistung von Computern involvieren [RRH00].

Durch die weite Verbreitung des Internets ist die Zahl der Computer- und Internetnutzer stark angestiegen. Die bloße Speicherung von Informationen reicht nicht mehr aus, um das wachsende Informationsbedürfnis der modernen Wissens- und Informationsgesellschaft zu befriedigen. Adäquate Information Retrieval Systeme sind für die Strukturierung, Analyse, Suche und Organisation von Informationen erforderlich, um sie als in geeigneter Form gespeicherte Informationseinheiten der Nutzergemeinde zur Verfügung zu stellen. Retrieval Systeme können neben reinen Daten auch die verschiedenartigsten Dateien enthalten [BYRN99].

Somit besteht ein wachsendes Bedürfnis nach einer geeigneten Darstellung der archivierten Daten und Informationen, um den unterschiedlichen Anforderungen an die Art der Informationspräsentation gerecht werden zu können.

Bei der Suche ist der Nutzer des Informations- oder Retrievalsystems auf Unterstützung angewiesen, um Zeit und Kosten einzusparen. Unterstützung kann in

Form der Bereitstellung geeigneter Visualisierungen oder Benutzeroberflächen erfolgen.

Visualisierungen, die sich auf die grafische Repräsentation der zugrunde liegenden Daten beschränken, existieren in großer Zahl. Es fehlt jedoch an Systemen, die dem Benutzer durch die Interaktionsmöglichkeit mit der Benutzeroberfläche die direkte Kontrolle über die Art und Weise der grafischen Repräsentation der Daten zur Verfügung stellen. Wenn der Nutzer beispielsweise versucht, eine Anfrage an eine Datenbank zu stellen, wird er dabei selten durch eine grafische Repräsentation seiner Anfrage oder seines Anfrageergebnisses unterstützt, sondern ist darauf angewiesen ein Formular auszufüllen oder sich einer speziellen Anfragesprache zu bedienen. Diese Vorgehensweise hat meistens zur Folge, dass die Anfrage entweder zu viele oder keine Ergebnisse bringt.

Um derartige Probleme für den Nutzer zu lösen, wurde an der Universität Konstanz am Fachbereich Informatik und Informationswissenschaft in der Arbeitsgruppe Mensch–Computer–Interaktion unter Leitung von Prof. Dr. Harald Reiterer, eingebettet in einen visuellen Metadatenbrowser, der CircleSegmentView nach einem vorgegebenen Konzept entwickelt.

Beim Entwurf des CircleSegmentView wurde der Umstand ausgenutzt, dass Benutzer nach [CD00] Informationen leichter finden, wenn diese in topologische Kategorien eingeteilt präsentiert werden. Vor diesem Hintergrund wird mit Hilfe einer geeigneten manipulierbaren Oberfläche eine grafische Vorschau auf Anfrageergebnisse bereitgestellt, die verhindern soll, dass der Benutzer bei der eigentlichen Anfrage zu viele oder gar keine Anfrageergebnisse bekommt. Der CircleSegmentView ist eine kreisdiagrammbasierte Query Preview. Gegenstand dieser Arbeit ist die vom Autor durchgeführte Implementierung und Evaluierung des CircleSegmentView.

Kapitel 2 beschreibt dabei die Entwicklung Query Preview basierter Systeme und bietet einen Überblick über die Entwicklung von visuellen Systemen von Dynamic Queries bis hin zu Query Previews. Funktionsweise und Leistungsmerkmale des CircleSegmentView werden in Kapitel 3 detailliert erläutert, während in Kapitel 4 auf die Implementierung des CircleSegmentView näher eingegangen wird.

In Kapitel 5 wird der durchgeführte Benutzertest beschrieben, dessen Schwerpunkt darin liegt, zu zeigen, dass der Benutzer mit Hilfe des CircleSegmentView die gewünschten Informationen über die Daten schneller findet als bei der Benutzung eines Suchformulars. Das konnte durch die Ergebnisse des Benutzertests bestätigt

werden. In Kapitel 6 werden die durch den Benutzertest gewonnenen Ergebnisse zusammengefasst.

## Kapitel 2

# Entwicklung von Query Previews

### 2.1 Überblick

Beim Entwurf und bei der Beurteilung der Nützlichkeit einer Benutzeroberfläche, deren primäre Funktionalität darin besteht, aus einer Menge von beliebig strukturierten Informationen oder Daten, die für den Benutzer relevanten Informationen zu extrahieren, steht die Beantwortung folgender Fragen im Vordergrund: welches Informationsbedürfnis soll befriedigt werden und welche dahingehenden Eigenschaften weist das zugrunde liegende Informationssystem auf [BYRN99]. Denn die Art und Weise der Interaktion des Nutzers mit der Datenbank wird durch sein Informationsbedürfnis und die Eigenschaften des Informationssystems bestimmt, wobei das Informationsbedürfnis meist mehr oder weniger spezifisch ist. Bei einem Nutzer mit einem präzise spezifizierten Informationswunsch kann davon ausgegangen werden, dass dieser Nutzer eher dazu bereit sein wird, einen vergleichsweise größeren Aufwand zu betreiben, um sein Ziel zu erreichen [BYRN99]. Darunter fällt beispielsweise das Erlernen einer bestimmten Anfragesprache.

In Abbildung 2.1 wird die Interaktion des Nutzers mit einem Informationssystem unter Verwendung einer Anfragesprache dargestellt.

Der Nutzer, der versucht mittels einer Anfragesprache die gewünschten Informationen zu erlangen, muss diese Anfragesprache benutzen, was voraussetzt, dass er die Anfragesprache zumindest in Grundzügen beherrscht. Außerdem ist er gezwungen,

seine Anfrage korrekt zu formulieren; wobei eine Anfrage genau dann als korrekt formuliert bezeichnet wird, wenn sie frei von syntaktischen und semantischen Fehlern ist.

Für den Nutzer mit einem weniger ausgeprägten Informationsbedürfnis, der sich in den meisten Fällen nicht im Klaren darüber ist, welche Information er benötigt und wo sich diese befindet, kommt Browsing als bedarfsorientierte Vorgehensweise in Frage.

Browsing wird als interaktive Tätigkeit definiert, bei welcher der Nutzer eher daran interessiert ist, die Dokumenten- oder Datensammlung zu erkunden, als ein spezifisches Informationsbedürfnis zu befriedigen [BYRN99].

Beim Browsing werden dem Benutzer zwar keine spezifischen Kenntnisse abverlangt, allerdings stellt sich hier das grundlegende Problem ein, dass der Nutzer unter Umständen nur unwichtige Informationen oder überhaupt nichts findet.

Somit stellen beide Vorgehensweisen den Benutzer vor unterschiedliche Probleme und Schwierigkeiten. In Kapitel 2.2 soll vor allem näher auf die Probleme der Benutzer mit sehr speziellen Informationswünschen eingegangen werden. In Kapitel 2.3 und 2.4 werden die Konzepte der Dynamic Queries und Query Previews als Lösungsansätze vorgestellt, sowie deren jeweilige Vor- und Nachteile dargestellt.

## 2.2 Probleme bei der Interaktion zwischen Benutzer und Informationssystem

Aufgrund der weiten Verbreitung des Internets ist es inzwischen möglich, auf viele große Datenbanken online zuzugreifen. Diese sind entweder allgemein oder nur für einen autorisierten Kreis von Benutzern zugänglich. Eine Tatsache, die zur Folge hat, dass eine große Anzahl von Nutzern mit den unterschiedlichsten Informationsbedürfnissen und Erwartungen auf Onlinedatenbanken zugreifen kann.

### 2.2.1 Unterschiedliches Vorwissen der Benutzer des Informationssystems

Die Nutzer bilden keine homogene Gruppe, sondern unterscheiden sich teilweise stark hinsichtlich ihres Wissenstandes, ihrer Erfahrung im Umgang mit Daten und den von ihnen gewünschten Anfrageergebnissen.

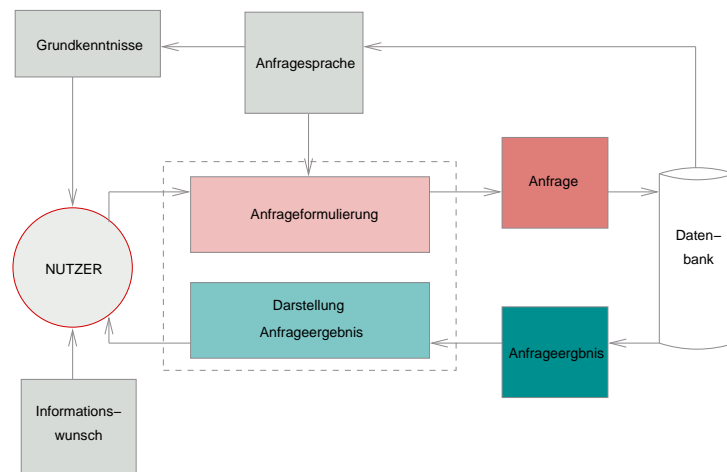


Abbildung 2.1: Schema für die Interaktion des Nutzers mit einer Datenbank

Diese Ausgangssituation erschwert die Entwicklung von geeigneten Werkzeugen für die Abfrage, da normalerweise bei einer Datenbank von einer einheitlichen Benutzergruppe ausgegangen werden kann. Hier wird jedoch von Nutzern mit unterschiedlichstem Vorwissen und beruflichen Hintergrund verlangt, die gleiche Oberfläche zu benutzen, was in Abhängigkeit von den Kenntnissen über die enthaltenen Daten zu verschiedenen Problemen führen kann:

- Laiennutzer

Laien können in der Regel keine Aussagen über die Struktur der Datenbank machen und beherrschen die Anfragesprache entweder schlecht oder gar nicht. Es liegt auf der Hand, dass dies die Formulierung von fehlerhaften und uneffektiven Anfragen zur Folge hat, wobei eine Anfrage im Allgemeinen dann als uneffektiv bezeichnet wird, wenn sie nicht das beabsichtigte Ergebnis liefert. Das kann auf zwei Arten geschehen:

Die Anfrage liefert eine lange Trefferliste, deren Einträge mühsam auf Relevanz und Nützlichkeit überprüft werden müssen. Ein Phänomen, welches als Megahit-Problem bezeichnet wird. Meistens bekommt der Nutzer dabei aufgrund der Tatsache, dass die meisten Informationssysteme die Trefferlisten trunkieren, nicht einmal die genaue Anzahl von Treffern geliefert [PVN<sup>+</sup>99]. Des weiteren besteht die Möglichkeit, dass die Anfrage kein einziges Resultat liefert, weil die Anfrage nicht zur Struktur und dem Inhalt der Datenbank

passt oder dem Nutzer bei der Anfrageformulierung bewusst oder unbewusst ein Fehler unterlaufen ist.

- **Expertennutzer**

Für Expertennutzer besteht oft das Problem, dass sie zwar die Anfragesprache beherrschen, aber oft die gleiche Anfrage mehrmals in leicht abgewandelter Form stellen müssen, um nicht Gefahr zu laufen, aufgrund einer semantisch ungünstigen Anfrageformulierung wichtige Informationen zu übersehen.

Somit bleibt die Formulierung einer effektiven Anfrage selbst für Expertennutzer eine schwierige Aufgabe – vor allem, wenn die gewünschte Information in einer absehbaren Zeitspanne aufgefunden werden soll. Dies stellt jedoch nicht das einzige Problem dar.

### **2.2.2 Größe und Beschaffenheit des Informationssystems**

Oft bekommen die Nutzer bei der Arbeit mit großen Datenbanken Probleme durch das Vorhandensein von vielen verschiedenen Datentypen. Teilweise sind die Datensätze in einer Datenbank als Text-, Film-, Audio- und Bilddaten in den verschiedensten Dateiformaten gespeichert. Darüber hinaus existieren häufig sehr spezielle anwendungsspezifische Dateiformate, so dass die meisten Werkzeuge mit der Kategorisierung und Identifizierung der Datenarten überfordert sind [DPSB97]. Viele Nutzer sind auch einfach nur mit der Anzahl der Datensätze und der Größe des Dokumentenraumes überfordert. Vor allem wenn die Suche auf ganz bestimmte Daten konzentriert ist, entsteht das Bedürfnis nach der Möglichkeit, die Suche benutzerdefiniert auf bestimmte Themenbereiche oder Dateiformate einzuschränken.

### **2.2.3 Art der Interaktion mit dem Informationssystem**

Unter Berücksichtigung des unterschiedlichen Vorwissens der Benutzer scheint die Verwendung einer Anfragesprache für die Interaktion mit dem Informationssystem dann nicht geeignet zu sein, wenn das Informationssystem auch von Laiennutzern und Nutzern mit geringem Vorwissen verwendet wird.

Um eine effektive und benutzerfreundliche Suche gewährleisten zu können, muss folglich auf eine alternative Form der Interaktion mit dem Informationssystem zurückgegriffen werden. Im Folgenden werden diese Alternativen kurz beschrieben.

Shneiderman [Shn98] unterscheidet folgende Möglichkeiten der Interaktion: kommandozeilenbasierte Interaktion, Formulareingaben, direkte Manipulation der Oberfläche und natürlich-sprachliche Interaktion.

Bezüglich der Interaktion mit einem Informationssystem soll die Möglichkeit der formularbasierten Interaktion näher betrachtet werden: Formularbasierte Benutzeroberflächen werden bis heute vorzugsweise für die Abfrage von Datenbanken verwendet. Um eine solche Oberfläche effektiv nutzen zu können, sind allerdings detaillierte Kenntnisse der zugrunde liegenden Datenbank erforderlich, um auch hier Abfragen mit zu vielen oder gar keinen Suchresultaten zu vermeiden, die den Benutzer lediglich verwirren und frustrieren. Die gleichen Probleme stellen sich bei der natürlich-sprachlichen Interaktion ein, während bei der kommandozeilenbasierten Interaktion Befehle gelernt werden müssen. Sobald Nutzer ohne Vorwissen mittels spezieller Befehle mit dem Informationssystem interagieren, treten bei dieser Art der Interaktion ähnliche Probleme auf, wie bei der Verwendung einer speziellen Anfragesprache.

#### 2.2.4 Lösungsmöglichkeiten

Als Lösung wird von [AS94] die Verwendung von direkt manipulierbaren Benutzeroberflächen vorgeschlagen. Auf dieser Grundlage können flexible Werkzeuge entworfen werden, die die Informationssuche von Benutzern mit unterschiedlichem Vorwissen und Hintergrundinformationen unterstützen sollen. Hierbei bietet sich die Verwendung von Dynamic Queries und Query Previews an. Beide Techniken basieren auf der direkten Interaktion mit der Benutzeroberfläche und werden inzwischen für verschiedenste Anwendungsbereiche immer weiter entwickelt. Sie bieten sowohl für Experten- als auch für Laiennutzer eine Möglichkeit zur Suche auf Daten, die ohne Anfragesprache auskommt. In Kapitel 2.3 werden die Vor- und Nachteile dieser Techniken erörtert.

### 2.3 Dynamic Queries

Nach [AWS92] erfüllen Dynamic Queries den Zweck, die Anfrage grafisch zu repräsentieren sowie eine grafische Darstellung der Anfrage und des Anfrageergebnisses bereitzustellen. Dabei erfolgt die grafische Darstellung der Anfrage auf der





Abbildung 2.2: Benutzeroberfläche des Dynamic Home Finder, [WS92]

Benutzeroberfläche mit Hilfe geeigneter Interfacekomponenten, wie Slider, Toolbar, Menüs und Auswahlboxen. Mit den Interfacekomponenten können Auswahlkriterien spezifiziert werden. Der Benutzer kann beobachten, wie sich die Menge der Daten, die diesen Kriterien entsprechen, kontinuierlich den vorgenommenen Einstellungen gemäß ändert. Als Beispiel soll hier der Dynamic Home Finder dienen, der in [WS92] vorgestellt wurde und eines der ersten Systeme ist, die mit Dynamic Queries arbeiten. Mittels der Interfacekomponente Slider kann der Benutzer Eigenschaften seiner Wunschimmobilie einstellen, wie beispielsweise Lage, Kosten und Anzahl der Räume. (Siehe Abbildung 2.2.) Die Immobilien, die diese Eigenschaften aufweisen, werden als gelbe Punkte visualisiert und den Slidereinstellungen gemäß aktualisiert. Sie befinden sich innerhalb des rot markierten Bereiches in Abbildung 2.2. Somit sind durch geeignete Interaktion mit den Interfacekomponenten schnelle, umkehrbare und aufeinander aufbauende Interaktionsschritte möglich, die dem Benutzer ein großes Maß an Kontrolle über den Anfrageprozess geben. Daten, die für den Benutzer nicht von Interesse sind, werden gar nicht erst angezeigt.

### Vorteile

Dadurch, dass nach jeder Änderung der Anfrage, z.B. durch Änderung der Einstellungen, die mit Hilfe der Interfacekomponenten gemacht wurden, ein sofortiges visuelles Feedback erfolgt, können die Nutzer besser mit großen Datenmengen arbeiten. Somit stellen Dynamic Queries sowohl für Laien als auch für Expertennutzer ein geeignetes und mächtiges Werkzeug dar, dessen Hauptvorteil in der sofortigen und unmittelbaren Visualisierung der Daten liegt. Der Benutzer bekommt einen Überblick über die vorhandenen Daten und kann dadurch den Datenbestand schnell und gezielt erkunden. Das führt zu einer Verbesserung und Beschleunigung der Interaktion und erhöht die Benutzerzufriedenheit [Shn94].

Darüber hinaus bietet die direkte Interaktionsmöglichkeit den Vorteil, dass der Benutzer keine Zeit mit dem Erlernen einer bestimmten Anfragesprache verschwenden muss. Zusätzlich spart der Benutzer Zeit dadurch, dass der Fall, dass zu viele oder gar keine Anfrageergebnisse zurückgegeben werden, weniger häufig eintritt.

### Nachteile

Problematisch kann die Verwendung von Dynamic Query basierten Systemen bei großen Datenbanken werden, da Dynamic Queries direkt auf den Datensätzen arbeiten und somit jeder Interaktionsschritt einen neuen Zugriff auf die Daten bedeutet. Aufgrund des hohen Ressourcenverbrauchs sind Dynamic Queries bei sehr großen Datenbanken und Onlinedatenbanken nicht anwendbar. Alternativ dazu wurde deshalb das Konzept der Query Previews entwickelt, welches die Vorteile der Dynamic Queries ausnutzt, aber gleichzeitig den ständigen Zugriff auf die Datensätze vermeidet.

## 2.4 Query Previews

Mit Query Previews lassen sich mögliche Anfrageergebnisse ressourcenschonend, flexibel und benutzerfreundlich visualisieren [BYRN99]. Sie bieten einen visuellen Überblick über die in der Datenbank vorhandenen Daten mit Hilfe von Metadaten. Mit dem Begriff Metadaten bezeichnet man im Allgemeinen Eigenschaften von Daten oder Dokumenten, die oft beschreibender Natur sind, wie z.B. Autor oder Inhalt, und in Kategorien aufgeteilt werden können. Als besonders bedeutsam sind in die-

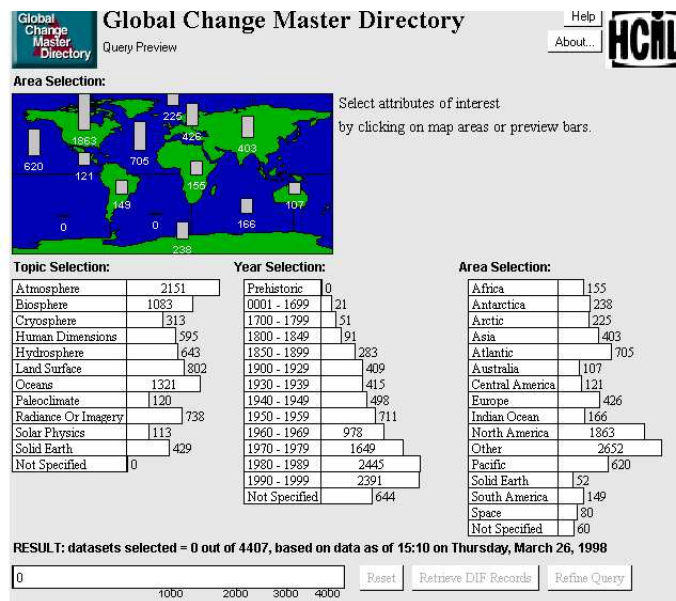


Abbildung 2.3: Ansicht des Query Preview Interface von NASA Eosdis, [TPS00]

sem Zusammenhang der Metadatenstandard MARC oder der Dublin Core Standard für die Erstellung von Metadaten zu erwähnen [BYRN99].

Query Previews basieren auf der Annahme, dass ein Großteil der Nutzer nur an einem kleinen Bereich der Daten interessiert ist [TPS00]. Deshalb bieten Query Previews primär einen visuellen Überblick über die Verteilung der Daten, meist auf der Basis von geeigneten Diagrammen, wie z.B. Balken- oder Kreisdiagrammen.

Ein gutes Beispiel bietet die Query Preview, die am Human-Computer-Interaction Lab der University of Maryland für das NASA Global Change Master Directory entwickelt wurde [TPS00]. Die für den Nutzer relevanten Bereiche oder Kategorien können durch Mausklick auf die Landkarte in Abbildung 2.3 ausgewählt werden. Die Anzahl der Datensätze pro Kategorie wird durch die Barcharts angezeigt. Die Gesamtanzahl der Ergebnisdatsätze für eine Abfrage wird im Anzeigefeld links unten in Abbildung 2.3 dargestellt.

Durch die Möglichkeit der direkten Manipulation der Oberfläche mit Hilfe von geeigneten Interfacekomponenten kann der Benutzer Bereiche, die für ihn von Interesse sind, mit Hilfe von Maus oder Tastatur auswählen. Nach jedem Interaktionsschritt werden die auf der Oberfläche visualisierten Datensätze entsprechend den Einstel-

lungen des Benutzers aktualisiert. Auf diese Weise können interessante Bereiche definiert werden. Gleichzeitig verkleinert sich die Menge der möglichen Anfrageergebnisse. In einem zweiten Schritt kann der Benutzer nähere Informationen über die Datensätze aus dem ausgewählten Bereich anfordern, die dann visuell dargestellt werden [TPS00]. Ist die Ergebnismenge aus der Sicht des Benutzers ausreichend überschaubar, können die relevanten Datensätze vollständig dargestellt oder zur weiteren Bearbeitung heruntergeladen werden.

### Vorteile

Der Hauptvorteil bei der Verwendung von Query Preview basierten Visualisierungen besteht in der überblicksartigen Repräsentation der vorhandenen Datensätze sowie in der Bereitstellung von direktem visuellem Feedback [TPS00]. Darüber hinaus kann der Benutzer die Anzahl der angezeigten Datensätze durch die Einstellung von Auswahlbereichen drastisch reduzieren. Die gesamte Auswahl erfolgt per Mausklick, somit sind keine Grundkenntnisse erforderlich. Die direkte Kontrollmöglichkeit führt dazu, dass sich der Nutzer durch die Menge der Daten nicht mehr überfordert fühlt. Dadurch, dass die durch Manipulation der Oberfläche ausgewählten Datensätze benutzerfreundlich dargestellt werden, ist deren Verteilung leichter erkennbar, und neue Erkenntnisse über die Daten können in verständlicher und kompakter Form gewonnen werden. Das bringt eine nicht unerhebliche Reduktion des Arbeitsaufwands bei der Suche nach bestimmten Daten mit sich. Insbesondere bei der Nutzung von Onlinedatenbanken ergeben sich daraus erhebliche ökonomische Vorteile aufgrund der Verkürzung der Anfragezeiten und geringerer Netzwerkbelastung. Der Zeitaufwand bei der Abfrage setzt sich gemäß [TLH<sup>+</sup>00] wie folgt zusammen:

Unter der Anfragezeit  $T_q$  versteht man die gesamte Zeit, die der Nutzer für die Anfrage benötigt. Diese besteht aus der Interaktionszeit  $T_i$ , der Antwortzeit  $T_r$  und der Analysezeit  $T_a$ , wobei die Antwortzeit  $T_r$  die Zeitspanne bezeichnet, die der Nutzer auf das Ergebnis seiner Anfrage warten muss, während unter der Analysezeit  $T_a$  die Zeitspanne verstanden wird, die für die gedankliche Auswertung der Ergebnisse verwendet wird.

$$T_q = T_i + T_r + T_a$$

Abschließend kann festgestellt werden, dass Query Previews schnelle und umkehr-

bare Aktionen erlauben [TLH<sup>+</sup>00]. Dadurch wird das explorative Verhalten des Benutzers bei gleichzeitiger Verkürzung der Anfragezeit gefördert.

### Nachteile

Nachteilhaft erweist sich bei der Verwendung von Query Previews lediglich die Tatsache, dass dem gesamten Prozess der Anfrageformulierung ein Interaktionsschritt hinzugefügt wird. Dadurch kann sich unter Umständen, vor allem in Verbindung mit einem ungünstigen Interfacedesign, die Schnelligkeit und Leichtigkeit der Benutzung verschlechtern.

Bei der Beurteilung von Query Previews sollte auch der Tatsache Beachtung geschenkt werden, dass die Auswahlmöglichkeiten für die Einschränkung der angezeigten Datensätze teilweise vorher aus einer ganzen Reihe von möglichen, durch die Attribute der Metadaten vorgegebenen Kriterien herausgesucht werden müssen.

Eine Einschränkung der Optionen im Vorfeld führt dazu, dass das Query Preview basierte Interface nicht die volle Mächtigkeit des zugrunde liegenden Informationssystems widerspiegelt. In diesem Zusammenhang wird auch die Frage relevant, ob Query Preview basierte Visualisierungen im Vergleich zu einer Anfragesprache ein ähnlich mächtiges Werkzeug darstellen.

Die Antwort darauf findet sich im Entwurfsprinzip der Query Previews: prinzipiell dient eine Query Preview dazu, den Benutzer bei der Formulierung seiner Anfrage zu unterstützen, was meistens durch die Visualisierung der vorhandenen Datensätze geschieht. Danach folgt in der Regel ein Zwischenschritt, der den Benutzer in die Lage versetzt, aufgrund des durch die Query Preview gewonnenen Vorwissens eine exaktere Anfrage zu formulieren, als es ihm ohne die graphische Vorschau möglich wäre. [TLH<sup>+</sup>00] beobachten allerdings statistisch relevante zeitliche Verzögerungen für den Fall, dass die Benutzer genau wissen, welche Datensätze für sie von Bedeutung sind.

Somit kann festgestellt werden, dass Query Previews häufig nicht dazu konzipiert sind, die gleichen Möglichkeiten wie eine Anfragesprache zu bieten, sondern um den Benutzer mit Informationen über die Daten zu versorgen, die ihn bei der Formulierung seiner Anfrage unterstützen.

Wie in Kapitel 2.2 am Beispiel einer normalen Anfrage erörtert, gilt auch bei der Verwendung von Query Preview basierten Oberflächen, dass das Maß für die Nütz-

lichkeit im konkreten Fall immer von der Art der durchzuführenden Anfragen und dem Vorwissen des Anwenders abhängt.

Doch muss zudem auch auf den Entwurf einer benutzerfreundlichen visuellen Darstellung geachtet werden. Eine Anforderung, auf die in Kapitel 3.5 anhand des CircleSegmentView als Beispiel für ein Query Preview basiertes System näher eingegangen wird.

## Kapitel 3

# Der CircleSegmentView

### 3.1 Projekthintergrund

#### 3.1.1 Der CircleSegmentView als Bestandteil von VisMeb

Der CircleSegmentView ist Bestandteil von VisMeb, einem visuellen Metadatenbrowser. Es stellt ein visuelles Suchsystem zur Erkundung großer Datenmengen zur Verfügung [RLP<sup>+</sup>03] und wurde an der Universität Konstanz am Fachbereich Informatik und Informationswissenschaft in der Arbeitsgruppe Mensch–Computer–Interaktion unter Leitung von Prof. Dr. Harald Reiterer entwickelt.

Mit Hilfe des Metadatenbrowsers VisMeb soll der Benutzer aus der oft sehr großen Treffermenge einer Suchanfrage die für ihn relevanten Daten schnell herausfiltern können [RLP<sup>+</sup>03].

Abbildung 3.2 zeigt, dass der CircleSegmentView dabei nicht nur als Query Preview, sondern auch als visuelles Filterwerkzeug verwendet werden kann. In Abbildung 3.1 ist der CircleSegmentView als Bestandteil von VisMeb zur Veranschaulichung dargestellt.

#### 3.1.2 Der CircleSegmentView als Bestandteil von INVISIP

Wertvolle Vorarbeiten für den Metadatenbrowser VisMeb wurden im Rahmen des EU-Projekts INVISIP <sup>1</sup> geleistet [RLP<sup>+</sup>03]. Der CircleSegmentView wurde im Rahmen von INVISIP als Query Preview entwickelt. INVISIP wird im Rahmen der EU

---

<sup>1</sup>EU-IST-2000-29640 INVISIP

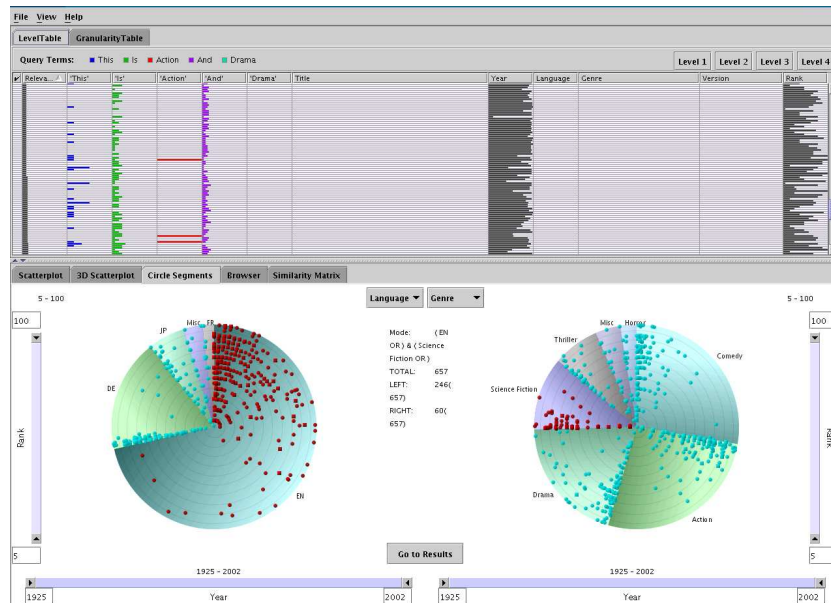


Abbildung 3.1: Benutzeroberfläche von VisMeb

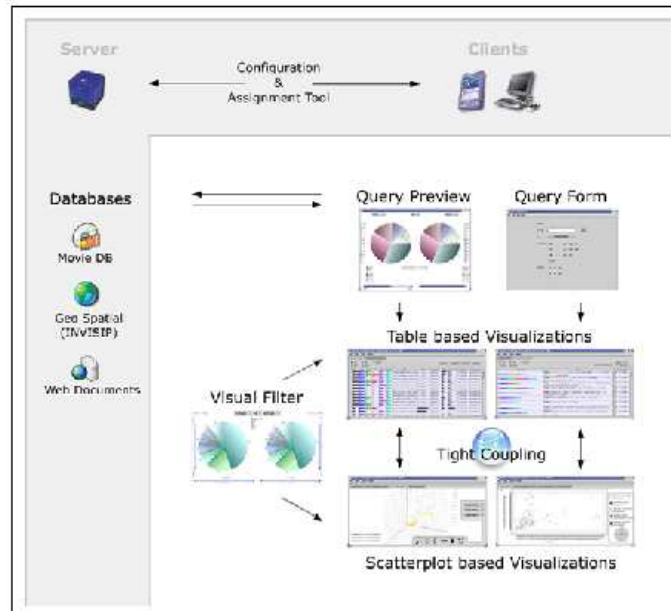
Information Society Technologies Programme gefördert und soll Entscheidungsprozesse bei der Standortwahl erleichtern. Standortentscheidungen und Stadtplanungsprojekte stellen hohe Anforderungen an Planung und Organisation; in den meisten Fällen werden geologische und geographische Daten benötigt, die in großer Anzahl in den entsprechenden Datenbanken vorhanden sind. Dabei hängen die Planungsentscheidungen von den Zugriffsmöglichkeiten auf diese Datenbanken ab. INVISIP erleichtert die Suche auf Geometadaten durch flexibel einsetzbare Visualisierungen.

### 3.1.3 Schwerpunkt

Die Funktion des CircleSegmentView als visueller Filter ist nicht Gegenstand der folgenden Erörterungen. An dieser Stelle soll vielmehr das Entwurfskonzept des CircleSegmentView in seiner Funktion als Query Preview beschrieben und analysiert werden.

Kapitel 3.2 gibt einen Überblick über die Geschichte der kreisdiagrammbasierten Visualisierungen. Es folgt die Beschreibung der Funktionsweise des CircleSegmentView und seiner wichtigsten Interfacekomponenten in Kapitel 3.4. In Kapitel 3.5 wird die Eignung der Oberfläche für die Visualisierung von Daten von verschiedenen



Abbildung 3.2: Die Architektur von VisMeb [RLP<sup>+</sup>03]

Standpunkten aus beschrieben:

Beschrieben wird, wie allgemeine Prinzipien für den Entwurf von Visualisierungen beachtet wurden, in welcher Form prinzipielle Regeln für gutes visuelles Design angewandt wurden und wie die Richtlinien für den Entwurf von Benutzeroberflächen aus dem Bereich der Mensch-Computer Interaktion umgesetzt wurden; denn obwohl der CircleSegmentView als Instrument zur Vorschau auf Datenbankabfragen mit Hilfe von Metadaten entworfen wurde, gelten doch bezüglich der visuellen Gestaltung die gleichen Grundsätze wie bei der visuellen Darstellung von Rohdaten.

### 3.2 Entwicklungsgeschichte

Das Grundkonzept des CircleSegmentView basiert auf der Verwendung von Kreisdiagrammen für die Visualisierung von Daten. Diese Idee wurde zuerst von [AKK96] aufgegriffen: in einem Projekt der technischen Universität München wurde 1996 eine Visualisierung entwickelt, die Börsendaten mit Hilfe von Kreisdiagrammen darstellt. Dabei stellt jedes Segment eine Datendimension dar. Die einzelnen Datensätze werden durch farblich kodierte Pixel auf dem Bildschirm repräsentiert. Die Pixel werden

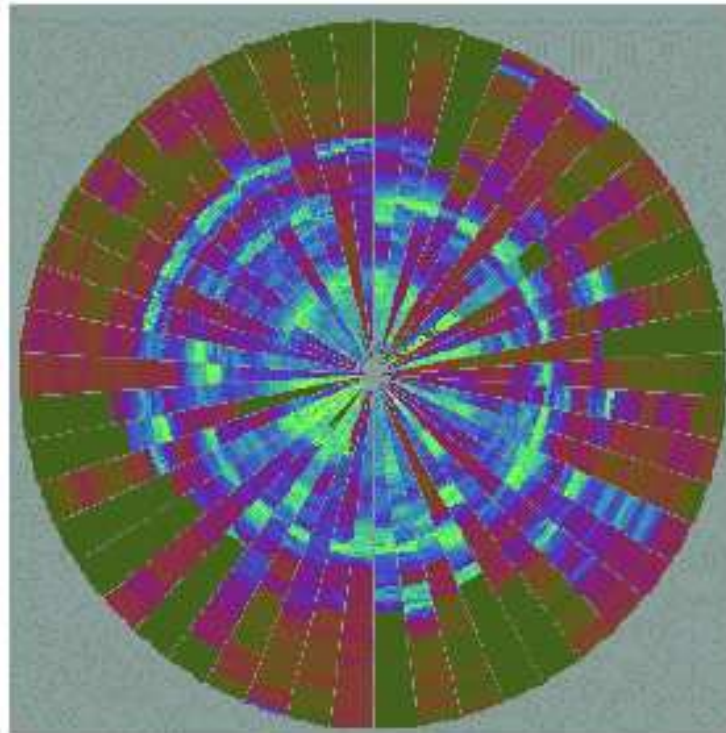


Abbildung 3.3: Ansicht der CircleSegments

innerhalb eines Segments vom Zentrum des Kreisdiagrammes aus bis zu dessen Rand nach folgendem Prinzip angeordnet [AKK96]:

Entlang der 'Zeichenlinie', die orthogonal zu den Segmentbegrenzungen verläuft, wird beginnend im Zentrum abwechselnd vorwärts und rückwärts gezeichnet [AKK96]. Sobald die Segmentbegrenzung erreicht ist, wird die 'Zeichenlinie' Richtung Segmentrand bewegt und die Zeichenrichtung umgedreht [AKK96]. Ältere Datensätze liegen weiter innen, neuere Datensätze ganz außen auf dem Segment. Dabei werden Datensätze mit hohen Werten mit hellen Farben gezeichnet und Datensätze mit niedrigen Werten mit dunklen Farben dargestellt [AKK96]. In Abbildung 3.3 werden mit den CircleSegments die tagesgenauen Kurse von 50 Aktien des FAZ-Index über einen Zeitraum von 20 Jahren visualisiert. Die hellen kreisförmigen Ringe stellen Hochpreisphasen dar [Kei22].

Die Anzahl der darstellbaren Datensätze wird durch die Anzahl der auf dem Bildschirm darstellbaren Pixel beschränkt. Bei heutigen Bildschirmen sind das eine Mil-

lion Pixel [Kei22]. Bei der Arbeit mit sehr großen Datenbanken ist die Beschränkung auf die Bildschirmgröße nicht hinnehmbar. Zudem kann der Benutzer nicht mit der Visualisierung interagieren, er hat keinen Einfluß auf die Darstellung.

Unter Berücksichtigung dieser Punkte wurde mit dem CircleSegmentView eine Visualisierung entworfen, welche die Grundidee der CircleSegments wie folgt erweitert. Die in Kapitel 3.3 genannten Voraussetzungen werden unter Verwendung des in Kapitel 2 vorgestellten Konzeptes der Query Previews auf die bereits bestehende, auf Kreissegmenten basierende Visualisierungstechnik übertragen. Das Ergebnis dieser Vorgehensweise ist der CircleSegmentView, der in Kapitel 3.4 beschrieben und analysiert werden soll.

### 3.3 Grundlegende Einordnung des CircleSegmentView als Visualisierung

Nach [Kei22] können Visualisierungen unter verschiedenen Blickwinkeln betrachtet und kategorisiert werden. Der CircleSegmentView ist gemäß [Kei22] wie folgt einzuordnen:

#### **Art der zu visualisierenden Daten**

Mit dem CircleSegmentView sollen multi-dimensionale Daten aus relationalen Datenbanken visualisiert werden.

#### **Eingesetzte Visualisierungstechnik**

Die dabei zum Einsatz kommende Visualisierungstechnik ist ähnlich wie bei den CircleSegments [AKK96] kreisdiagrammbasiert, kann jedoch nicht wie die CircleSegments zu den pixel-orientierten Visualisierungen gezählt werden, da für die Darstellung der einzelnen Datensätze keine Pixel verwendet werden, sondern kleine Rechtecke oder Kreise. Dadurch weist der CircleSegmentView auch Elemente der icon-basierten Visualisierungstechniken auf.

#### **Verwendete Interaktionstechnik**

Der CircleSegmentView gestattet dem Benutzer mit der Visualisierung zu interagieren. Möglich sind interaktive Selektion, das heißt, der Benutzer kann Datensätze von

Interesse besonders hervorheben, und interaktives Zooming. Interaktives Zooming erlaubt dem Benutzer, Detailinformationen über einzelne Datensätze oder Bereiche abzurufen. Wie diese Techniken realisiert wurden, wird detailliert in Kapitel 3.4 beschrieben.

### 3.4 Beschreibung der Benutzeroberfläche

Grundidee für den Entwurf des CircleSegmentView war die Bereitstellung einer Visualisierung, die dem Benutzer unter Verwendung einer Benutzeroberfläche mit zahlreichen Interaktionsmöglichkeiten einen Überblick über die Verteilung der Daten geben soll. Im Folgenden werden die Interfacekomponenten, die die Anzeige der Daten steuern und die prinzipielle Funktionsweise der Visualisierungstechnik beschrieben:

#### Auswahl der Datenbank

Als oberstes Designziel beim Entwurf des CircleSegmentView wurde die Bereitstellung einer Visualisierung festgelegt, die den Nutzer bei der Analyse und dem Umgang mit multidimensionalen Daten unterstützt und flexible und an die jeweiligen Daten anpassbare benutzerfreundliche Interaktionsmöglichkeiten bereitstellt. Aus diesem Grund wurde der CircleSegmentView so konzipiert, dass dessen Benutzung unabhängig vom Aufbau und Inhalt der zugrundeliegenden Datenbank ist, ein Merkmal, welches vor allem auf den Umstand zurückzuführen ist, dass für die Repräsentation der Datensätze auf der Oberfläche ausschließlich die jeweiligen Metadaten relevant sind. Es ist deshalb unerheblich, ob es sich bei den darzustellenden Daten um Audio-, Text-, Video- oder Bilddaten handelt.

Außerdem kann der CircleSegmentView, wie auch das gesamte System VisMeb mit unterschiedlichen Datenbanken verwendet werden. Bei Systemstart kann der Benutzer die Datenbank auswählen. Abbildung 3.4 zeigt den Einstiegsdialog, welcher beim Start von VisMeb eingeblendet wird. Der Benutzer kann nicht nur die Datenbank auswählen, mit der er arbeiten möchte, sondern mit Hilfe der 'Assignments' genannten Voreinstellungen auch eine Auswahl für die Art der Anzeige der Daten treffen.

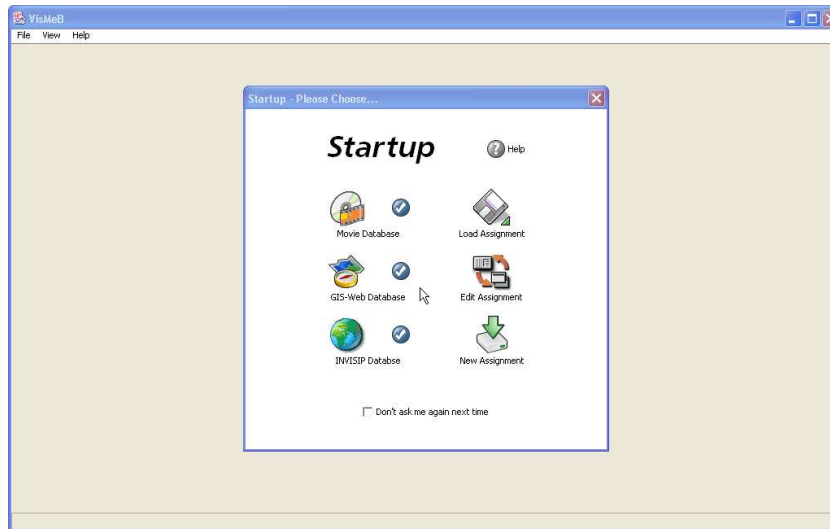


Abbildung 3.4: Ansicht des Einstiegsdialogs von VisMeb

### Basisszenario

Um die Funktionsweise des CircleSegmentView zu veranschaulichen, wird bei der Beschreibung des CircleSegmentView von folgendem Szenario ausgegangen: Man stelle sich eine Filmdatenbank vor. Der Benutzer möchte nun mit dem CircleSegmentView Informationen über die darin enthaltenen Filme bekommen und ist vor allem an Horrorfilmen aus den siebziger Jahren interessiert. In den Erläuterungen wird immer auf das vorliegende Basisszenario Bezug genommen.

### Metadaten

Durch die flexible Schnittstelle zur jeweiligen Datenbank werden die Metadaten in Abhängigkeit von der jeweiligen Datenbank und dem Metadaten typ nach einem bestimmten Grundprinzip auf die vorhandenen Interfacekomponenten verteilt. Die Interfacekomponenten, die die Anzeige der Daten steuern, bieten dem Benutzer jeweils eine andere Möglichkeit, die Darstellung der Datensätze gemäß den Eigenschaften der Metadaten zu beeinflussen. Der Anschaulichkeit wegen soll die genaue Funktionsweise der einzelnen Interfacekomponenten anhand des Basisszenarios erklärt werden. Abbildung 3.5 bietet einen Überblick über die Metadaten dieser Filmdatenbank. Der CircleSegmentView soll Daten nach durch den Benutzer festgelegten

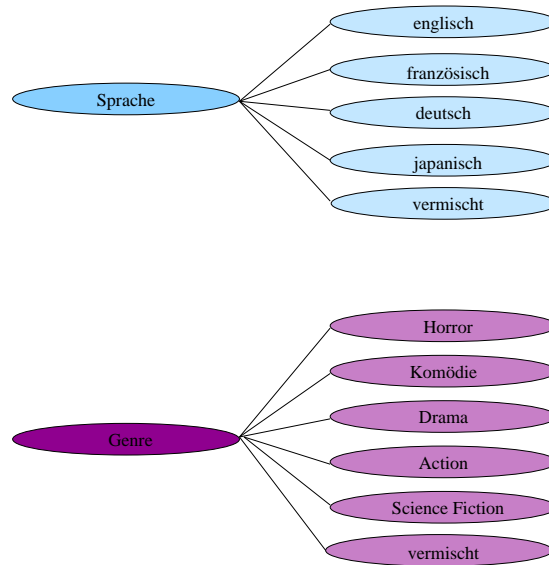


Abbildung 3.5: Die Metadaten der Filmdatenbank vom Typ Kategorie (Ausschnitt)

Auswahlkriterien auf den beiden Kreisdiagrammen darstellen, die auch unabhängig voneinander benutzt werden können. Als Auswahlkriterien stehen dabei die Metadaten der jeweiligen Datenbank zur Verfügung, zwischen denen der Benutzer wie folgt auswählen kann:

### Beschreibung der Funktionen der Interfacekomponenten

Für der Benutzung des CircleSegmentView stehende folgende Interfacekomponenten zur Verfügung: Comboboxen, Auswahlmenüs, Eingabe-/Anzeigefelder, Slider, Buttons, Tooltips.

- Die Comboboxen

Metadaten vom Typ Kategorie werden mit Hilfe der Segmente dargestellt. Schräg oberhalb der Kreisdiagramme befindet sich jeweils eine Combobox, (siehe Abbildung 3.6), anhand derer die Benutzer auswählen können, welche Metadaten durch die Segmente dargestellt werden sollen. Der Wertebereich der Metadaten vom Typ Kategorie umfasst eine endliche Menge von Worten. Jedes Segment stellt dann ein anderes Element des Wertebereichs dar. Hat der Benutzer beispielsweise Sprache ausgewählt, richtet sich die Anzahl der Segmente des Kreisdiagramms nach der Anzahl der verschiedenen Sprachen, in denen

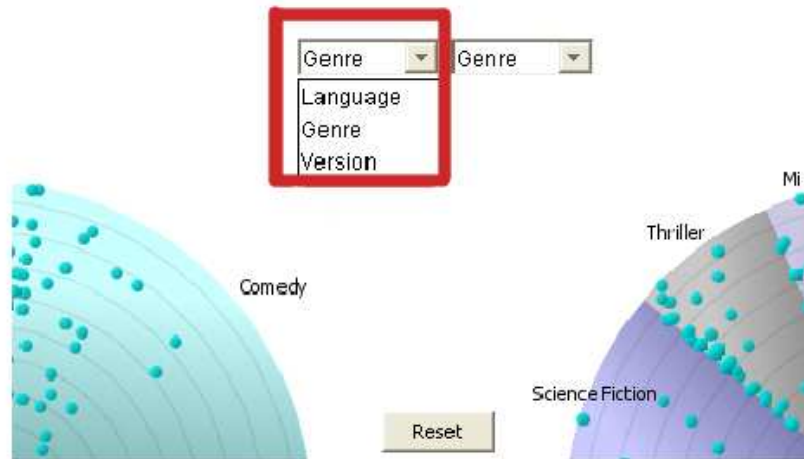


Abbildung 3.6: Ansicht der Combobox des linken Kreisdiagramms

es Filme in der Filmdatenbank gibt. Die aufgrund der Auswahl resultierenden Segmente werden im Uhrzeigersinn nach ihrer Größe sortiert dargestellt. Die Größe der Segmente resultiert aus der Anzahl der Datensätze, die den Wert des jeweiligen Elements aus dem Wertebereich annehmen. Aus der sortierten Darstellung kann der Benutzer sofort Schlüsse über die Anzahl der enthaltenen Datensätze ziehen. Auch ist bei zwei ähnlich großen Segmenten klar erkennbar, dass das im Uhrzeigersinn zuerst gezeichnete Segment das größere ist.

Zur Auswahl stehen prinzipiell alle verfügbaren Metadaten vom Typ Kategorie, ausgeschlossen werden im Vorfeld lediglich solche Metadaten, deren Auswahl zu einer Aufteilung des Kreisdiagramms in mehr als 10 Segmente führen würde. Der rot markierte Bereich in Abbildung 3.6 zeigt die aufgeklappte Combobox des linken Kreisdiagramms.

- Auswahlmenüs

Metadaten numerischen Typs können mit den Menüs ausgewählt werden, die durch Mausklick auf den Bereich neben den horizontalen und vertikalen Slidern erscheinen, (siehe Abbildung 3.7). Der Wertebereich der durch die Menüs

ausgewählten Metadaten wird durch die Slider dargestellt.

- Anzeige-/Eingabefelder

Die minimalen und maximalen Werte des Wertebereichs können von den Anzeigefeldern oberhalb und unterhalb der vertikalen Slider, beziehungsweise links und rechts der horizontalen Slider, abgelesen werden. Die Anzeigefelder können auch als Eingabefelder zur Veränderung der Sliderpositionen verwendet werden.

- Die Slider

Der Wertebereich wird dabei durch die horizontalen und vertikalen Slider repräsentiert. Der kleinste Wert dieses Wertebereichs wird dabei durch die unterste mögliche Sliderposition dargestellt, der größte Wert durch die oberste mögliche Sliderposition. Bei den horizontalen Slidern gilt das gleiche: der kleinste Wert des Wertebereichs wird durch die linke Sliderposition und der größte Wert durch die rechte Sliderposition dargestellt.

- Buttons

Zur Verfügung stehen der Reset Button und der Go-to-Results Button. Das Betätigen des Go-to-Results Button bewirkt die logische UND-Verknüpfung der auf dem rechten und linken Kreisdiagramm selektierten Datensätze. Der Reset Button setzt die Darstellung in den durch die 'Assignments' definierten Ausgangszustand zurück.

- Tooltips

Positioniert der Benutzer die Maus über den Datensätzen, erscheint ein Tooltip mit Detailinformation über den Film.

### **Darstellung der Datensätze**

Die Datensätze selbst werden durch kleine Kreise, die DataButtons, in den Segmenten grafisch dargestellt. Die Position der Datensätze bestimmt sich wie folgt:

Entscheidet sich der Benutzer im Auswahlménü des horizontalen Sliders für z.B. 'Year', wird der Winkel der Segmente für das Mapping der Datensätze benutzt. Im Uhrzeigersinn werden die Datensätze nach dem Entstehungsjahr auf dem Segment angeordnet. Wählt der Benutzer im Auswahlménü neben dem vertikalen Slider z.B.



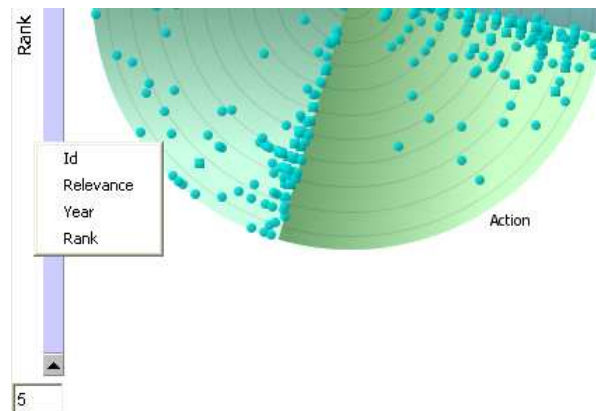


Abbildung 3.7: Metadatenauswahl für den vertikalen Slider

die Bewertung des Films ('Rank') aus, werden die Datensätze auf den Radius gemappt und die Datensätze werden nach ihrer Bewertung aufsteigend sortiert und vom Zentrum des Kreisdiagramms aus zu dessen Rand hin angeordnet.

Die Position der DataButtons auf dem Segmentwinkel wird visuell mit zunehmendem Winkel noch durch eine Aufhellung der Hintergrundfarbe des Segments unterstützt. Hat sich der Benutzer beispielsweise dafür entschieden, das Datum der Datensätze auf den Winkel eines Kreissegments abzubilden, dann liegen die neueren Datensätze im helleren Bereich und die älteren Datensätze werden im dunkleren Bereich des Segments positioniert.

Falls aufgrund ähnlicher oder gleicher Eigenschaften der Datensätze zwei oder mehr DataButtons auf die gleiche Stelle gezeichnet werden, wird der DataButton durch ein kleines Rechteck dargestellt.

Durch die Veränderung der Sliderpositionen kann der Benutzer den Bereich, aus dem Datensätze angezeigt werden, einschränken (Zoom-Funktion). Es werden dann nur die Datensätze auf dem Kreisdiagramm angezeigt, deren Eigenschaften dem ausgewählten Wertebereich entsprechen.

Beispiel:

Der Benutzer interessiert sich nur für englische Horrorfilme aus den siebziger Jahren. Um den Bereich der angezeigten Datensätze entsprechend einzuschränken, muss er an dem für das Jahr verantwortlichen Slider, (hier der horizontale Slider), die linke Sliderposition auf 1970 einstellen und die rechte Sliderposition auf 1979. So werden

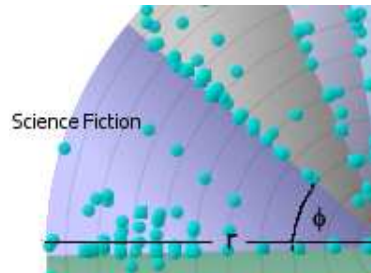


Abbildung 3.8: Radius und Winkel eines Segments

die Flächen aller Segmente für die Anzeige von Filmen aus den siebziger Jahren benutzt.

### Selektion und Deselektion von Datensätzen

Selektierte Datensätze werden in roter Farbe, deselektierte Datensätze cyanfarben dargestellt. Es bestehen folgende Selektionsmöglichkeiten:

- Selektion und Deselektion aller Datensätze eines Kreisdiagramms  
Durch Mausklick mit der rechten Maustaste auf das Kreisdiagramm erscheint ein Auswahlménü mit den Optionen 'Select All' und 'Deselect All'.
- Selektion und Deselektion aller Datensätze eines Segments  
Durch Mausklick auf den Segmenthintergrund können alle Datensätze eines Segments selektiert bzw. deselektiert werden.
- Selektion einzelner Datensätze  
Ein Mausklick auf einen einzelnen DataButton bewirkt Selektion bzw. Deselektion

### UND-Verknüpfung

Zwischen der Menge A der im linken Kreisdiagramm selektierten DataButtons und der Menge B der im rechten Kreisdiagramm selektierten DataButtons kann der Durchschnitt gebildet werden. Im Folgenden wird das als UND-Verknüpfung bezeichnet. Diese Operation kann der Benutzer durch Betätigung des Go-to-Results Buttons ausführen. Die Datensätze, die zur Ergebnismenge gehören, werden in einer

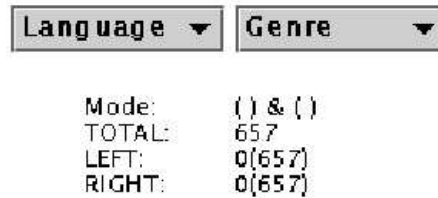


Abbildung 3.9: Anzeige der Anzahl der links und rechts selektierten Datensätze, sowie der Ergebnismenge

einfachen Ergebnistabelle aufgelistet, die in Abbildung 3.10 dargestellt wird.

Hat der Benutzer beispielsweise im linken Kreisdiagramm alle Horrorfilme selektiert und im rechten Kreisdiagramm alle englischsprachigen Filme, kann er sich durch Betätigung des Go-to-Results Button die Ergebnismenge in der Ergebnistabelle anzeigen lassen. In der Ergebnistabelle werden dann alle englischsprachigen Horrorfilme aufgelistet.

Hat sich der Benutzer entschlossen, nur mit einem Kreisdiagramm zu arbeiten, werden bei Betätigung des Go-to-Results Button nur die mit Hilfe der DataButtons ausgewählten Datensätze in der Ergebnistabelle angezeigt.

### Anzeige der Anzahl der Datensätze

Der jeweilige Verknüpfungsmodus wird in dem Textbereich zwischen den beiden Kreisdiagrammen angezeigt, ebenso wie die Anzahl der rechts und links angezeigten Datensätze, die Gesamtanzahl der Datensätze und die Anzahl der momentanen Ergebnismenge nach der Verknüpfungsoperation. (Siehe Abbildung 3.9.)

### Darstellung der Resultate

Hat der Benutzer die Menge der Datensätze zu seiner Zufriedenheit eingeschränkt und den Go-to-Results Button betätigt, werden die Ergebnisdatensätze in einer Tabelle aufgelistet. (Siehe Abbildung 3.10.)

Results (29)

title	year	language	genre	version	rank	plot
Men in Black II	2002	EN	Science Fiction	OF	59	(mond05 /local_tmp/te...
E. T. - The Extra Terrestrial	1982	EN	Science Fiction	OF	65	A group of aliens visit e...
Bicentennial Man	1999	EN	Science Fiction	OF	40	This film follows the life...
Dune	1984	EN	Science Fiction	OF	70	In the distant future, a ...
Vanilla Sky CD 1	2001	EN	Science Fiction	OF	80	A successful publisher fi...
Star Trek: First Contact	1996	EN	Science Fiction	OF	80	The time is the 24th ce...
Star Trek: Generations	1994	EN	Science Fiction	OF	70	Star Trek: Generations i...
Star Trek VI: The Undiscovered Country	1991	EN	Science Fiction	OF	60	An explosion of their ke...
Star Trek V: The Final Frontier	1989	EN	Science Fiction	OF	60	The crew of the Federat...
Star Trek IV: The Voyage Home	1986	EN	Science Fiction	OF	90	A space probe appears...
Star Trek III: The Search for Spock	1984	EN	Science Fiction	OF	75	Admiral Kirk's defeat of...
Lost in Space	1998	EN	Science Fiction	OF	70	In the near future, earth...
Day the Earth stood still, The	1951	EN	Science Fiction	OF	80	An alien (Klaatu) with h...
2001 - A Space Odyssey	1968	EN	Science Fiction	OF	100	Moon explorers encount...
Red Planet	2000	EN	Science Fiction	OF	88	Astronauts search for so...
Final Fantasy	2001	EN	Science Fiction	OF	88	A female scientist make...
Blade Runner	1982	EN	Science Fiction	OF	88	Deckard, a blade runne...
Starship Troopers	1997	EN	Science Fiction	OF	10	In the distant future hig...
Men in Black	1997	EN	Science Fiction	OF	20	Two men who keep an ...
Demolition Man	1993	EN	Science Fiction	OF	80	A cop is brought out of ...
Abyss, The	1989	EN	Science Fiction	OF	50	An American nuclear su...
Gattaca	1997	EN	Science Fiction	OF	99	Futuristic story of a gen...
Matrix	1999	EN	Science Fiction	OF	85	A computer hacker lea...
Star Wars II - The Empire Strikes Back	1980	EN	Science Fiction	OF	85	While Luke takes advan...
Tron	1982	EN	Science Fiction	OF	85	Computer Classic, one o...
Judge Dredd	1995	EN	Science Fiction	OF	20	In a dystopic future, Dre...
Escape from New York	1981	EN	Science Fiction	OF	10	In 1997, when the US Fr...
The Matrix	1999	EN	Science Fiction	OF	100	A computer hacker lea...
Epoch (2000)	2000	EN	Science Fiction	OF	89	Disaster strikes as a sp...

Abbildung 3.10: Für den Test verwendete Ergebnistabelle

### 3.5 Analyse des CircleSegmentView als Visualisierung

Visualisierungen dienen prinzipiell zur Darstellung von Daten. Damit werden im Allgemeinen zwei Ziele verfolgt: Ergebnisse sollen präsentiert und die Analyse der Daten soll erleichtert werden [SM00].

Jedoch stellen Visualisierungen kein neues Themengebiet dar: schon seit Jahrhunderten beschäftigen sich Menschen auf dem Gebiet der Wissenschaft, Technik und Kunst mit der Frage, wie Informationen abgebildet werden können. Frühe Beispiele von Visualisierungen findet man im Bereich der Astronomie und Geographie.

Abbildung 3.11 zeigt eine Visualisierung, die eine Methode zur Berechnung von Längengraden verdeutlichen soll. Dieser Versuch, die Entfernung zwischen Rom und Toledo visuell darzustellen, zeigt die Gefahr, dass bei Nichtbeachtung grundlegender Regeln für visuelles Design Informationen falsch oder verzerrt dargestellt werden können. Deshalb wird beschrieben, welche Regeln beim Entwurf der Oberfläche des CircleSegmentView beachtet wurden, um diese Gefahr zu minimieren.

#### Beachtung von allgemeinen Visualisierungsgrundsätzen

Primär müssen beim Entwurf einer Visualisierung Eigenschaften von Datenmengen auf visuelle Attribute abgebildet werden. Ziel ist es, durch Wahl einer geeigneten Abbildungsform die in den Daten enthaltenen Informationen darzustellen. Da nicht jede Visualisierungsmethode dazu gleich gut geeignet ist, müssen allgemeine

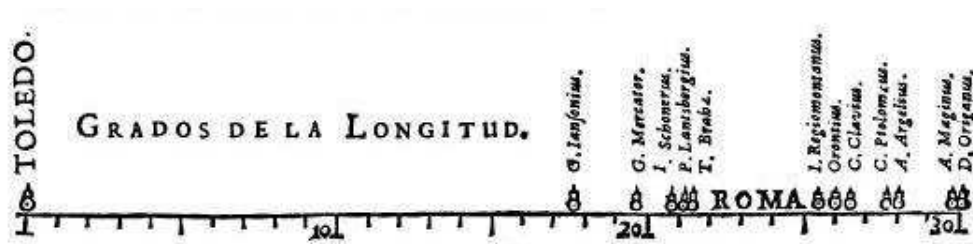


Abbildung 3.11: Beispiel einer frühen Visualisierung

Kriterien für die Beurteilung der Qualität einer Visualisierung formuliert werden. Nach [SM00] definiert sich die Qualität einer Visualisierung durch den Grad, in dem die bildliche Darstellung das kommunikative Ziel der Präsentation erreicht.

### Vorbedingungen

Wichtige Vorüberlegungen sollten die Art und Struktur der Daten, das Bearbeitungsziel und das Vorwissen des Betrachters einbeziehen.

Beim Entwurf des CircleSegmentView wurde folgenden Vorbedingungen als gültig betrachtet:

- Benutzung durch Experten  
Dadurch, dass der CircleSegmentView für die Benutzung durch Experten vorgesehen wurde, konnte der CircleSegmentView so entworfen werden, dass er sich aufgrund des komplexen Grundkonzeptes für eine intuitive Benutzung nur bedingt eignet.
- Überblick über die Verteilung der Daten  
Diese Vorbedingung wurde dadurch erfüllt, dass die visuelle Darstellung auf Kreisdiagrammen basiert [Hol02].
- Nutzung mit verschiedenen Datenbanken  
Der CircleSegmentView ist in den Metadatenbrowser VisMeb eingebettet. VisMeb ermöglicht durch eine flexible Schnittstelle eine Anbindung an gängige Datenbanken.

## Visuelle Codierung der Informationen

Ausgehend von diesen Vorüberlegungen soll nun das Mapping der Daten beschrieben werden. Unter Mapping versteht man die Abbildung der Daten auf visuelle Variablen. Nach [Ber82] stehen dafür folgende Variablen zur Verfügung: die Position in der Ebene, Größe, Helligkeit, Muster, Farbe und nicht zuletzt die Orientierung und die Form des gewählten Elements. Die einzelnen Variablen weisen jeweils eine oder mehrere spezifische Wirkungen auf.

- Anzahl der visuellen Variablen  
In [SM00, S. 138] wird vorgeschlagen, nicht mehr als fünf bis sechs verschiedene Variablen bei Kreisdiagrammen zu verwenden. Diese werden kurz beschrieben
- Beschreibung der visuellen Variablen  
Das Kreisdiagramm wird in verschieden farbig codierte Segmente unterteilt unter Verwendung der visuellen Variablen Größe, Orientierung und Farbe. Hierbei ist nur zu beachten, dass der Kreis eine entsprechende Größe aufweist, da sonst der Vergleich von Flächeninhalten zu schwierig wird. Auch sollte das Kreisdiagramm in nicht mehr als fünf Segmente unterteilt werden, andernfalls können aus der Unterschiedlichkeit der Segmentgrößen keine eindeutigen Schlüsse mehr gezogen werden [Hol02]. Dieses Kriterium wurde hier nicht beachtet, da es im Widerspruch zu der Anforderung an den CircleSegmentView stand, einen möglichst vollständigen Überblick über die zugrunde liegenden Daten zu bieten. Ob die Verletzung dieser Anforderung tatsächlich die Möglichkeit des visuellen Vergleichs der einzelnen Segmente beeinträchtigt, muss durch den in Kapitel 5 umfassend beschriebenen Benutzertest geklärt werden. Die Größe der Segmente dient dazu, die Anzahl der auf ihnen dargestellten Datensätze zu codieren. Farbe wird verwendet, um die einzelnen Segmente besser unterscheiden zu können. Die Orientierung der Segmente um den Mittelpunkt des Kreisdiagramms erfolgt entsprechend ihrer Größe in absteigender Ordnung. Die Datensätze, die auf ein einzelnes Segment entfallen, werden über ihre Position in der Ebene codiert. Die visuellen Variablen können unter Beachtung ihrer besonderen Eignung zur Darstellung bestimmter Dateneigenschaften beliebig verwendet werden. Es muss an dieser Stelle jedoch bemerkt werden, dass die örtliche Auflösung im menschlichen visuellen System im Vergleich zu

allen anderen visuellen Stimuli bezogen auf die wahrgenommene Information am größten ist [SM00, S. 127]. Positions- und Größenvergleiche können besonders genau durchgeführt werden. Die relative Positionierung hinsichtlich eines gemeinsamen Ausgangspunktes stellt hierbei ein besonders geeignetes Mittel zur Visualisierung von Daten dar [SM00, S.127]. Diesem Umstand wurde beim Entwurf des CircleSegmentView dadurch Rechnung getragen, dass die Datensätze nach ihrem Wert sortiert vom Mittelpunkt des Kreisdiagramms aus angeordnet werden.

### **Beachtung von prinzipiellen Wahrnehmungs- und Designgrundsätzen**

Beschrieben wird, wie prinzipielle Wahrnehmungs- und Designgrundsätze beim Entwurf des CircleSegmentView beachtet wurden. Zuerst wird beschrieben, wie Farben verwendet wurden. Das ist vor allem im Hinblick auf die Vermeidung von unerwünschten visuellen Seiteneffekten von entscheidender Bedeutung.

Als Hintergrundfarbe für den CircleSegmentView wurde weiß verwendet. Der Grund hierfür ist der Umstand, dass Farbtöne auf einem helleren Untergrund weniger hell wirken als auf einem dunklen Hintergrund, dafür aber gesättigter erscheinen [Gra86]. Kreisdiagramme fixieren. Bei einem weißen Hintergrund kann teilweise ein sogenannter Sukzessivkontrast als Nebeneffekt entstehen [Gra86]. Dabei entsteht bei längerer Betrachtung einer farbigen Fläche ein Nachbild auf dem hellen Hintergrund als Gegenbild zu der fixierten Fläche. Da die Nutzer während ihrer Arbeit mit dem CircleSegmentView längere Zeit die beiden farbigen Kreisdiagramme intensiv betrachten müssen, erscheint es nötig, das Auftreten dieses Effektes auszuschließen. Dazu wurden die Farben der Segmente grau unterlegt. Prinzipiell sollte die Anzahl der für die Visualisierung verwendeten Farben möglichst klein und konstant gehalten werden, um eine eindeutige Identifizierung der farblich kodierten Information abzusichern [SM00]. Wichtige, besonders zu betonende Informationen sollten in der Regel farblich hervorgehoben werden, beispielsweise durch die Verwendung von roten oder gelben Farbtönen. Grüne und blaue Farbtöne sind dagegen eher zur Visualisierung von Hintergrundinformation geeignet. Da alle Segmente des Kreisdiagrammes das gleiche optische Gewicht haben sollten, fiel die Wahl auf grau unterlegte Grün- und Blautöne. Einige der verwendeten Farben sind in Abbildung 3.12 dargestellt. Der Farbverlauf geht dabei über grün, lila und blau nach braun. Da die selektierten

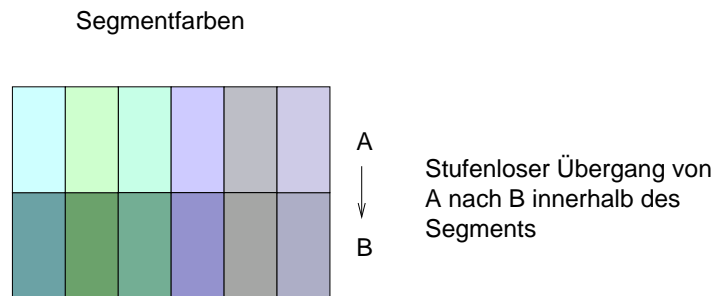


Abbildung 3.12: Segmentfarben des CircleSegmentView

Datensätze farblich besonders hervorgehoben werden sollten, werden sie rot dargestellt. Das hat zudem den Grund, dass der CircleSegmentView für die Verwendung in ingenieurwissenschaftlichen Arbeitsbereichen konzipiert wurde. Dort wird blau als neutrale Farbe und grün als positiv bewertet [SM00]. Von der Verwendung von Warn- und Signalfarben wie rot und gelb, muss beim Entwurf von Visualisierungen für diesen Arbeitsbereich prinzipiell abgeraten werden, da diese Farben mit Gefahren in Verbindung gebracht werden und auch nur für die Vermittlung derartiger Information gebraucht werden sollten.

Was die Bewertung der Anordnung der Interfacekomponenten betrifft, muss auf die Gestaltprinzipien der Wahrnehmungspsychologie zurückgegriffen werden. Grundsätzlich werden das Prinzip der Ähnlichkeit, der Geschlossenheit und Konvexität, sowie der stetigen und guten Fortsetzung unterschieden.

Um diese Grundsätze zu beachten, wurden die Segmente des Kreisdiagramms nach der Größe geordnet und die beiden Kreisdiagramme nebeneinander angeordnet.

Diejenigen Interfacekomponenten, die für die Aktionen, die beide Diagramme betreffen, benutzt werden können, sind mittig dazwischen platziert. Somit kann von der Anordnung der Interfacekomponenten auf deren Funktionalität geschlossen werden.

### **Beachtung von Grundsätzen aus dem Gebiet der Mensch-Computer-Interaktion**

Auf dem Gebiet der Mensch-Computer-Interaktion wurden Kriterien entwickelt, anhand derer sich die Benutzerfreundlichkeit einer Oberfläche feststellen lässt. In [Shn98] werden acht goldene Regeln aufgestellt, die es beim Design von Benutzeroberflächen zu beachten gilt. Diese Regeln beruhen auf Erfahrung und sollten für jede



Benutzeroberfläche gesondert interpretiert und eventuell auch erweitert werden.

- **Konsistenz**

Zuallererst sollte beim Entwurf auf Konsistenz geachtet werden. Konsistenz bedeutet in diesem Zusammenhang Konsistenz bezüglich Farbe, Layout, Eingabefeldern und Menüs. Der CircleSegmentView wird dieser Forderung durch einheitliche Farben, Menügestaltung und Interaktion gerecht.

- **Tastaturkürzel für Expertennutzer**

Expertennutzern sollte die Möglichkeit gegeben werden mit Tastaturkürzeln zu arbeiten, um die Anzahl der Interaktionsschritte zu minimieren. Diese Regel wurde bei der Implementierung des CircleSegmentView eingeschränkt verwirklicht: das Arbeiten mit der Tastatur ist nur bei den Eingabefeldern an den Slidern möglich. Das hat den Grund, dass viele der verwendeten Interfacekomponenten wie zum Beispiel die Slider und die Comboboxen mit Tastaturbefehlen nur eingeschränkt oder schlecht bedient werden können. In Frage käme jedoch eine entsprechende Tastaturbelegung für die Bedienung der Comboboxen, die Selektion bzw. Deselektion einzelner Segmente sowie für die Betätigung des Go-to-Results Button. Eine solche Bedienungsmöglichkeit widerspricht jedoch dem Wesen dieser Interfacekomponenten. Aus diesem Grund wurde auf die Möglichkeit verzichtet, die Komponenten mit Tastaturfunktion auszustatten.

- **Rückmeldung über die ausgeführten Aktionen**

Der Benutzer erhält durch die Textmeldung zwischen den beiden Kreisdiagrammen Informationen über die Auswirkung der gerade durchgeführten Aktion. Dort werden Informationen über die Anzahl der rechts und links sichtbaren und ausgewählten Datensätze und die Anzahl der Resultate einer Und-Verknüpfungsoperation angezeigt und ständig aktualisiert.

- **Geschlossenheit von System-Benutzer Dialogen**

Auf die Geschlossenheit von System-Benutzer Dialogen wurde ebenfalls geachtet. Die Länge einer Interaktionssequenz wird durch den Benutzer festgelegt und umfasst mindestens einen Interaktionsschritt. Abgeschlossen wird jede Sequenz durch die Betätigung des Ergebnisbuttons, der die grafische Auflistung der Ergebnisse in einer Ergebnistabelle bewirkt. (Siehe Abbildung 3.1.)

- Prävention von Benutzerfehlern  
[Shn98] fordert weiterhin, dass gravierende Benutzungsfehler vermieden werden sollen, bzw. bei deren Auftreten der Schaden für den Benutzer minimiert wird. Die Oberfläche trägt dem dadurch Rechnung, dass bei Eingabe von unzulässigen Werten in die Textfelder eines Sliders, dieser sich in seine Ausgangsposition zurückbewegt.
- Umkehrbare Aktionen  
Zudem sollte jede Aktion des Benutzers umkehrbar sein. Jede Einstellung, die der Benutzer an den Interfacekomponenten vornimmt, wird sofort über die Anzahl der auf den Kreisdiagrammen angezeigten Datensätzen sichtbar gemacht. Jede Aktion kann ebenso schnell wieder rückgängig gemacht werden, entweder durch erneute Auswahl des alten Menüpunktes in den Comboboxen und PopupMenus oder bei der Selektion der Datensätze oder Segmente durch erneute Selektion des fälschlicherweise markierten Elements, wodurch dieses in den unselektierten Zustand zurückversetzt wird.
- Der Benutzer im Mittelpunkt  
Es wird darauf geachtet, dass der Benutzer im Mittelpunkt aller Aktionen steht, das heißt, er ist Initiator aller Aktionen. Damit wird dem Benutzer das Gefühl vermittelt, alle Aktionen kontrollieren zu können. Ohne vorherige Interaktion seitens des Benutzers sind beim CircleSegmentView keine Aktionen möglich.
- Entlastung des Kurzzeitgedächtnisses  
Die Zahl der im Kurzzeitgedächtnis speicherbaren Informationseinheiten ist sehr begrenzt. Aus diesem Grund stellt die Bedienung der Oberfläche des CircleSegmentView minimale Anforderungen an das Kurzzeitgedächtnis dadurch, dass auf jede Aktion ein sofortiges Feedback erfolgt: die Anzeige wird den Benutzereinstellungen gemäß aktualisiert. Der Nutzer muss sich seine vorangegangenen Interaktionen nicht merken. Verstärkt wird diese Einfachheit noch durch die Möglichkeit der parallelen Anzeige auf zwei Kreisdiagrammen, wodurch der Nutzer den direkten Vergleich zwischen verschiedenen Darstellungen hat. Gemäß [Tuf97] eignet sich eine parallele Darstellung vor allem zum visuellen Vergleich.

Von [Shn98] wurden in diesem Zusammenhang noch spezielle Richtlinien für den Bereich der Information Visualisation aufgestellt. Diese sind unter dem Begriff Shneidermans Task Taxonomy bekannt und beinhalten folgende Richtlinien, die beim Entwurf des CircleSegmentView besonders beachtet wurden [Shn98].

- Überblick  
Zuerst soll ein Überblick über die Daten geliefert werden. Diese Anforderung wird dadurch erfüllt, dass zu Beginn alle Datensätze auf den Kreisdiagrammen visualisiert werden.
- Zoomfunktion  
Der Nutzer soll durch eine geeignete Zoomfunktion die Möglichkeit haben, interessante Einzelheiten näher und detaillierter betrachten zu können. Das geschieht durch Einschränkung des Bereichs, aus welchem Datensätze dargestellt werden, mit Hilfe der Slider. Der gesamte auf dem Kreisdiagramm zur Verfügung stehende Platz wird dann zur Darstellung des ausgewählten Bereichs verwendet.
- Filterfunktion  
Genaus sollen für den Nutzer uninteressante Informationen ausgeblendet werden können. Beim CircleSegmentView geschieht das auf die gleiche Art wie bei der Zoomfunktion: durch Einschränkung des Bereichs mit Hilfe der Slider.
- Details auf Anfrage  
Besonders wichtig ist, dass auf Anfrage detaillierte Informationen abgerufen werden können. Wenn der Benutzers die Maus über den Datensätzen von Interesse positioniert, erscheinen Tooltips mit näheren Informationen. Durch Selektion von Datensätzen und die anschließende Betätigung des Go-to-Results Button kann der Benutzer sich Details in einer Ergebnistabelle anzeigen lassen.
- Beziehung zwischen den Informationseinheiten  
Nicht zuletzt sollte der Benutzer die Informationseinheiten in Beziehung zueinander setzen können. Beim CircleSegmentView geschieht das durch die Möglichkeit, die rechts und links selektierten Datensätze mit dem logischen Operator 'UND' zu verknüpfen.

### 3.6 Zusammenfassung

Abschließend kann für die Beurteilung der Oberfläche bezüglich allgemeinen und mensch-computer-interaktionsspezifischen Anforderungen an Visualisierungen festgestellt werden, dass die grundlegenden Prinzipien und Regeln beim Entwurf des CircleSegmentView zur Anwendung gekommen sind.

Ob durch den CircleSegmentView dem Nutzer ein flexibles, schnelles und benutzerfreundliches Instrument zur Visualisierung von Datensätzen zur Verfügung gestellt wird, soll durch den Benutzertest festgestellt werden.

In Kapitel 5 werden die Durchführung und die Ergebnisse der Evaluation des CircleSegmentView beschrieben.

# Kapitel 4

## Implementierung

### 4.1 Pflichtenheft

Um einen Einblick in die Implementierung des CircleSegmentView zu vermitteln, wird das der Implementierung zugrunde liegende Pflichtenheft beschrieben. Die Struktur des Pflichtenheftes richtet sich nach den Vorgaben in [Bal96].

#### 4.1.1 Zielbestimmung

Durch die Verwendung des CircleSegmentView im Rahmen des visuellen Metadatenbrowsers VisMeb soll eine flexible und benutzerfreundliche Visualisierung von Ergebnisdatensätzen einer möglichen Datenbankabfrage zur Verfügung gestellt werden.

#### Musskriterien

Ziel ist die Visualisierung von multi-dimensionalen Daten aus einer relationalen Datenbank mit Hilfe von zwei Kreisdiagrammen. Die Art und Weise der Darstellung auf den Kreisdiagrammen richtet sich dabei nach den vom Benutzer ausgewählten Metadaten.

#### Wunschkriterien

Wünschenswert ist dabei eine hohe subjektive Zufriedenheit der Nutzer mit der Oberfläche. Außerdem soll der CircleSegmentView ein schnelleres Arbeiten ermöglichen als mit einem herkömmlichen Suchformular.

## **Abgrenzungskriterien**

Der CircleSegmentView kann auch als visueller Filter innerhalb des Metadatenbrowsers benutzt werden. Die Implementierung dieser Funktion ist aber nicht Bestandteil der Aufgabenstellung.

### **4.1.2 Produkteinsatz**

Der CircleSegmentView dient zur Visualisierung von Ergebnisdatensätzen einer möglichen Anfrage, also zur Vorschau auf ein mögliches Anfrageergebnis.

## **Anwendungsbereiche**

Der Anwendungsbereich für den CircleSegmentView richtet sich nach dem Anwendungsbereich der zugrunde liegenden Datenbank.

## **Zielgruppen**

Der CircleSegmentView ist für die Benutzung durch Experten auf dem jeweiligen Gebiet gedacht.

## **Betriebsbedingungen**

Geplant ist eine Nutzung in normaler Büroumgebung.

### **4.1.3 Produktumgebung**

#### **Software**

Der CircleSegmentView wurde plattformunabhängig entwickelt und ist unter Windows XP und Linux verwendbar.

#### **Hardware**

Ein handelsüblicher PC mit Grafikbildschirm genügt.

#### **Orgware**

Für die Verwendung des CircleSegmentView mit Daten aus Onlinedatenbanken ist eine Internetanbindung erforderlich.

#### 4.1.4 Produktfunktionen

Die Oberfläche besteht aus zwei Kreisdiagrammen. Als wichtigste Interfacekomponenten sind Slider, Comboboxen und einfache Buttons zu nennen. Mit deren Hilfe soll der Benutzer die Metadaten auswählen, die für die Position der darzustellenden Datensätze auf den Kreisdiagrammen zuständig sind.

#### 4.1.5 Benutzeroberfläche

Die üblichen Gestaltungsrichtlinien für grafische Benutzeroberflächen müssen eingehalten werden.

#### 4.1.6 Entwicklungsumgebung

Die Entwicklungsumgebung richtet sich nach der Produktumgebung.

### 4.2 Systemarchitektur

#### 4.2.1 Datenmodell

Die Flexibilität des visuellen Metadatenbrowsers VisMeb basiert vor allem darauf, dass die einzelnen Visualisierungen Daten aus gängigen Datenbanken darstellen können. Um dieses Ziel zu erreichen, wurde eine Schnittstelle eingebaut, die für die gewünschten Datenbanken einen Zugriff mit Hilfe eines JDBC zur Verfügung stellt. Die Visualisierungen lesen die Inhalte der Datenbank nicht direkt aus, sondern verwenden das interne Datenmodell, welches die Daten unabhängig von der zugrunde liegenden Datenbank in einer geeigneten Datenstruktur ablegt. Die einzelnen Visualisierungen greifen dann zur Anzeige der Daten auf den Inhalt dieser Datenstruktur zu.

#### 4.2.2 Abgrenzung

Auf den genauen Aufbau des Datenmodells soll hier nicht näher eingegangen werden, eine detaillierte Beschreibung findet sich in [Mem02].

## 4.3 Codearchitektur

### 4.3.1 UML-Klassendiagramm

Um den Aufbau des `CircleSegmentView` und die Klassenstruktur zu verdeutlichen, findet sich in Abbildung 4.1 ein UML-Klassendiagramm des `CircleSegmentView`. Es wurde mit dem Borland JBuilder 6 automatisch aus dem Source-Code generiert.

Dabei liegt folgender grundlegender Aufbau vor:

Der für den `CircleSegmentView` erforderliche Code findet sich im Paket `View.CircleSegment` die Klasse `CircleSegmentView` beinhaltet die wichtigsten Interfacekomponenten, die Klasse `ViewPanel` ist für die Darstellung der Kreisdiagramme verantwortlich.

An allen Interfacekomponenten, die per Maus bedient werden können, sind Listener registriert, die verschieden auf Mausereignisse reagieren.

In Kapitel 4.3.2 werden die einzelnen Klassen und deren Funktionalität näher beschrieben.



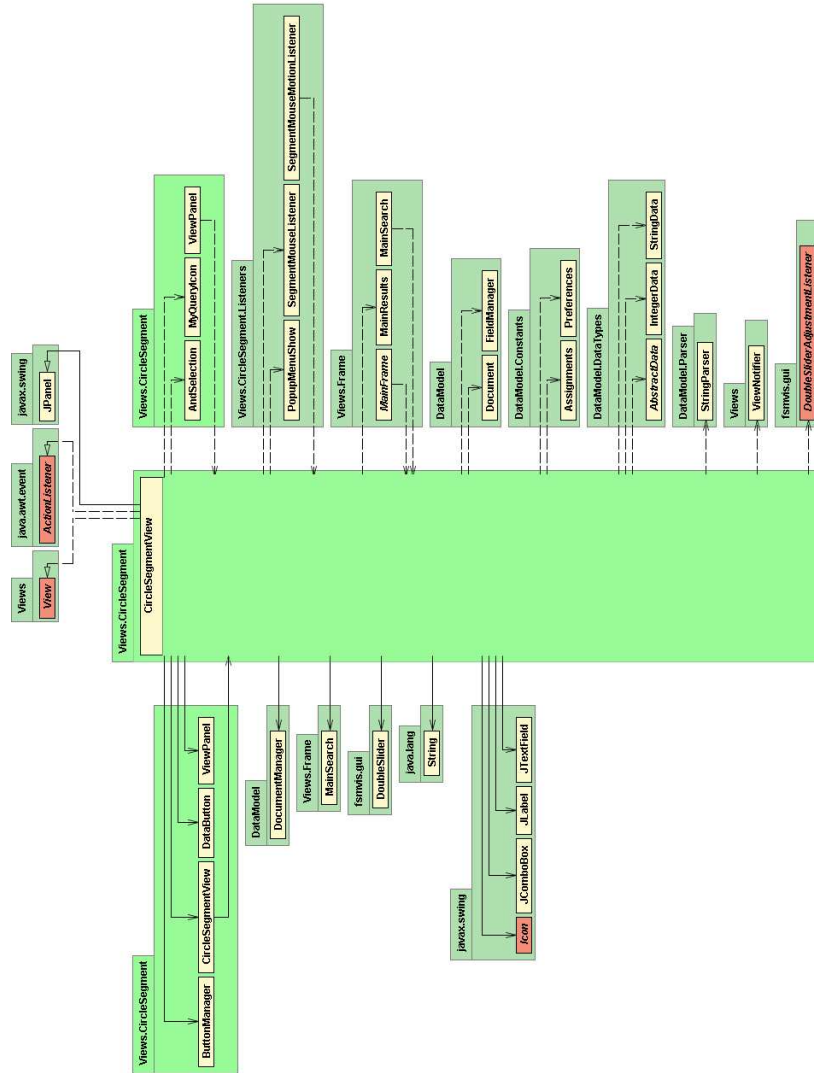


Abbildung 4.1: UML-Klassendiagramm des CircleSegmentView

### 4.3.2 Objektstruktur

Als wichtigste Bestandteile der Objektstruktur sind die Klassen *DataButton*, *Segment* und *MultiDataButton* zu nennen. Durch sie werden die wichtigsten Verhaltensweisen und Zustände der entsprechenden Objekte definiert.

#### Die Klasse *Buttonmanager*

Zu erwähnen ist vor allem die Klasse *Buttonmanager*. In ihr werden wichtige Informationen über die darzustellenden Daten mit Hilfe des Datenmodells gewonnen. Die dafür verantwortlichen Klassen aus dem Paket *DataModel* werden in [Mem02] beschrieben.

Im Konstruktor wird entsprechend der Anzahl der vorhandenen Datensätze ein zweidimensionales Array *buttons* mit den Objekten vom Typ *DataButtons* befüllt. Stellt man sich dieses Array als Matrix vor, so werden in der ersten Spalte die Objekte vom Typ *DataButtons* für das linke Kreisdiagramm und in der zweiten Spalte die Objekte vom Typ *DataButtons* für das rechte Kreisdiagramm gespeichert. Die Information über die Anzahl der Datensätze wird von der Methode *get()* der statischen Klasse *DocumentManager* aus dem Paket *DataModel* geliefert.

Die wichtigsten Methoden und Felder seien hier kurz beschrieben.

#### Methoden

- *createSegments(int index, int panel)*  
Stellt Informationen über die Größe, Reihenfolge und die Segmentbegrenzungen durch Aufruf der statischen Methoden *calcSegments()*, *sortSegments* und *limitSegments()* der Klasse *FieldManager* aus dem Paket *DataModel* bereit. Außerdem wird für alle *DataButtons* die Zugehörigkeit zu einem bestimmten Segment festgelegt.
- *clearSegments(int panelNr)*  
Setzt Informationen über das Segment auf *null* zurück.

#### Felder

- *DataButtons[][] buttons*  
Jedes Element der Datenstruktur repräsentiert einen Datensatz. Beim Aufruf des Konstruktors werden die Elemente in der Datenstruktur gespeichert.

- *String[] segmentKeys*

Dieses Feld enthält die Namen der Segmente.

Sämtliche Informationen über die Segmente des Kreisdiagramms werden mit Hilfe der Klasse *Segment* bereitgestellt.

### Die Klasse *Segment*

Die Segmente des Kreisdiagramms sind Objekte vom Typ *Segment*. Die Eigenschaften der Segmente bestimmen sich vor allem durch die *Arc2D.Double* Repräsentation des Segments, den Namen des Segments, den Selektionsstatus und die Zugehörigkeit zum rechten oder linken Kreisdiagramm. Bereits im Konstruktort findet sich ein wichtiges Merkmal zur Fehlerbehandlung: sollte der Fall eintreten, dass aus welchem Grund auch immer keine Kategorisierung der Daten möglich ist, wird standardmäßig ein segmentloser Kreis gezeichnet.

#### Felder

- *Arc2D.Double segment*

Jedes Segment wird durch ein Objekt vom Typ *Arc2D.Double* aus dem Paket *java.awt.geom* repräsentiert. Dieses Paket stellt standardmäßig Informationen über Startwinkel, Größe des Winkels und die Art der Verbindung zwischen den Schenkeln des Segments bereit.

- *boolean selected*

Hier wird gespeichert, ob das Segment gerade selektiert ist.

- *String segmentName*

Stellt eine Möglichkeit zur Verfügung, den Namen des entsprechenden Segments zu speichern. Der Name wird für die Beschriftung der Segmente verwendet.

### Die Klasse *DataButton*

Eine weitere zentrale Komponente des *CircleSegmentView* ist die Klasse *DataButton*. In ihr werden Methoden und Instanzvariablen verwaltet, die das Verhalten und die Eigenschaften der Objekte vom Typ *DataButton* bestimmen.

### Felder

- *DataButtonList list*

Mit Hilfe der Instanzvariable *list* vom Typ *DataButtonList* wird dem Umstand Rechnung getragen, daß ein *DataButton* auch mehrere Datensätze repräsentieren kann, dann wird er als Quadrat gezeichnet und soll zudem alle Eigenschaften der *DataButtons*, an deren Stelle er tritt, speichern.

Zudem müssen Mausereignisse dergestalt verarbeiten werden können, dass bei Mausklick auf das Objekt *DataButton* dieses die Farbe und den Selektionsstatus ändert. Zu diesem Zweck wird an jedem *DataButton* ein *MouseListener* vom Typ *ButtonMouseListener* registriert.

## 4.4 Leistungsmarkmale

### 4.4.1 Anzeige der Datensätze auf den Kreisdiagrammen

Die Hauptfunktionalität des *CircleSegmentView* liegt in der Anzeige der vorhandenen Daten.

### 4.4.2 Interaktion - benutzerdefinierte Einstellungsmöglichkeiten zur Anzeige der Datensätze

Durch geeignete Bedienung der Interfacekomponenten kann der Benutzer die Menge der zur Anzeige vorgesehenen Datensätze einschränken. (Vgl. Kapitel 3.) Die genaue Berechnung der Position eines *DataButton* auf dem Kreisdiagramm erfolgte dabei mit Hilfe der Berechnung seiner Polarkoordinaten.

Als Bezugspunkt fungiert dabei immer das Objekt *Segment*. Abbildung 4.2 zeigt den Winkel  $\phi$  und den Radius  $r$  eines Segments. Hierbei gilt:

$$x_p = r * \cos\phi \quad (4.1)$$

$$y_p = r * \sin\phi \quad (4.2)$$

Zur eigentlichen Darstellung der Punkte müssen die polaren Koordinaten mit der bekannten Transformationsgleichung wieder in kartesische Koordinaten zurückgerechnet werden, :

$$x = x_p * \cos\phi - y_p * \sin\phi \quad (4.3)$$

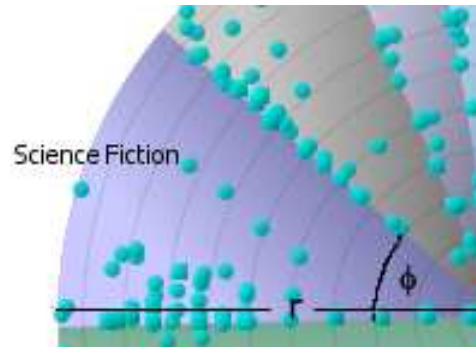


Abbildung 4.2: Ansicht eines Segments

$$y = x_p * \sin\phi - y_p * \cos\phi \quad (4.4)$$

Diese Funktionalität wird in der Methode *setButtonLocations* in der Klasse *View-Panel* wie folgt implementiert:

Der auf einem Segment zur Verfügung stehende Platz wird dabei gleichmäßig auf die Anzahl der anzuzeigenden *DataButtons* verteilt. Listing 4.3 zeigt die Implementierung. Wie diese Verteilung abläuft, soll anhand eines kleinen Beispiels erläutert werden:

Angenommen, der Benutzer hat festgelegt, dass die Datensätze dem Datum nach auf den Radius gemappt werden und der mittels der vertikalen Slider ausgewählte Bereich Datensätze aus den Jahren 1995, 1996 und 1997 beinhaltet. Dann liegen die ältesten Datensätze ganz innen, die Datensätze aus dem Jahr 1996 in der Mitte und die neuesten Daten aus dem Jahr 1997 ganz außen.

#### 4.4.3 Selektion der *DataButtons*

An die Objekte *Segment* und *DataButton* werden Listener vom Typ *SegmentMouseListener* und *ButtonMouseListener* registriert. Die Verarbeitung der registrierten Ereignisse findet dann in den entsprechenden Klassen statt. Sowohl die Klasse *SegmentMouseListener* als auch die Klasse *ButtonMouseListener* erben von der Klasse *MouseListener* aus dem Paket *java.awt.event* und implementieren das Interface *ActionListener*. Dadurch kann bei Empfang eines Mausereignisses auf der jeweiligen Interfacekomponente der Selektionsstatus der Dokumente in der *buttonlist* wie folgt geändert werden: von selektiert nach nicht-selektiert oder umgekehrt.

```
private void setButtonLocations () {
    for (int i=0; i<buttons.length; i++) {
        buttons[i].resetButton ();

        int xData = ((IntegerData) buttons[i].getData(xIndex)).getData ();
        int yData = ((IntegerData) buttons[i].getData(yIndex)).getData ();

        // if button lies within the circle ...
        if (yData >= yMin && yData <= yMax && xData >= xMin
10         && xData <= xMax) {
            int s = buttons[i].getSegmentIndex ();
            double angleStart = segments[s].getAngleStart ();
            double angleExtent = segments[s].getAngleExtent ();
            double angleProp = angleExtent / (xMax - xMin);
            double yProp = (double) radius / (yMax - yMin);
            double yZoom = yMax - yData;
            double xZoom = xData - xMin ;
            double buttonRadius = radius - yZoom * yProp;
            double clockwise = - xZoom * angleProp - angleStart ;
20         double angle = Math.toRadians (315 + clockwise);

            int xp = (int) (Math.cos (angle) * buttonRadius * 0.7 -
                Math.sin (angle) * buttonRadius * 0.7 +
                width / 2);
            int yp = (int) (Math.sin (angle) * buttonRadius * 0.7 +
                Math.cos (angle) * buttonRadius * 0.7 +
                height / 2);

            buttons[i]. setLocation (xp, yp);
        }
    }
}
```

Abbildung 4.3: Listing des Codes für die Platzierung der DataButtons

#### 4.4.4 UND-Verknüpfung

Die UND-Verknüpfungsmöglichkeit zwischen der Menge der links selektierten und rechts selektierten *DataButtons* wird in der Klasse *AndSelection* implementiert. Mit dem Go-to-Results Button zwischen den beiden Kreisdiagrammen kann der Nutzer die UND-Verknüpfung anwenden. Anhand eines kleinen Beispiels aus der Moviedatenbank soll diese Funktionalität näher erklärt werden:

Angenommen, der Nutzer lässt sich auf dem linken Kreisdiagramm die Daten nach den einzelnen Genres kategorisiert anzeigen. Zudem hat er auf dem vertikalen Slider den Rang der Filme ausgewählt und auf dem horizontalen Slider das Entstehungsjahr. Selektiert hat der Nutzer alle Horrorfilme aus den siebziger Jahren. Sichtbar sind somit links alle Filme aus den siebziger Jahren und das Segment mit der Beschriftung Horror ist selektiert. Bezüglich der Bewertung des Films hat der Benutzer von seiner Einschränkungsmöglichkeit keinen Gebrauch gemacht. Auf dem rechten Kreisdiagramm nimmt der Benutzer eine Einteilung nach der Sprache vor. Es sind somit alle Filme zu sehen. Selektiert hat der Benutzer alle englischsprachigen Filme. Sichtbar sind insgesamt immer alle Filme, die den ausgewählten Eigenschaften auf dem linken Kreisdiagramm entsprechen oder den ausgewählten Eigenschaften auf dem rechten Kreisdiagramm. Auch die Selektion ändert daran nichts. Nach Betätigung des Go-to-Results Buttons wird die Ergebnismenge in der Ergebnistabelle angezeigt. Im Beispiel wären das alle englischsprachigen Horrorfilme aus den siebziger Jahren. Bei der Implementierung, (siehe Listing 4.4, werden alle Dokumente überprüft und festgestellt, ob sie sichtbar und selektiert sind. Sichtbar sind alle Dokumente, die den benutzerdefinierten Einstellungen entsprechen. Hier kam die Tatsache zugute, dass im Datenmodell nur eine einzige Dokumentenliste geführt wird. Über ein entsprechende Variable kann dann zur Laufzeit überprüft werden, ob das Dokument links oder rechts sichtbar oder selektiert ist.

#### 4.4.5 Vorschau auf die Ergebnismenge

Mit Hilfe des Codes aus der Klasse *ViewPanel* konnte auch leicht die Anzeige der Anzahl der rechts und links selektierten *DataButtons* und die Anzahl einer potentiellen Und-Verknüpfung realisiert werden: ein Zähler, der erhöht wird, sobald ein *DataButton* gezeichnet wird und ein Zähler, der erhöht wird, wenn der *DataButton* rot gezeichnet wird, liefern die Werte, die dann in dem Anzeigefeld zwischen den

```
for (int i = 0; i < docs; i++) {  
  
    Document doc = DocumentManager.get().getDocument(i);  
    if (selL.isEmpty() && doc.isSelectedRight ()  
        && doc.isVisibleRight ()) {  
        results.add(doc);  
    }  
    if (selR.isEmpty() && doc.isSelectedLeft ()  
        && doc.isVisibleLeft ()) {  
10     results.add(doc);  
    }  
    if ((doc.isSelectedLeft ()) && (doc.isSelectedRight ()) &&  
        doc.isVisibleLeft () && doc.isVisibleRight ()) {  
        results.add(doc);  
    }  
}
```

Abbildung 4.4: Listing des Codes für die Verknüpfung

beiden Kreisdiagrammen dargestellt werden. Die Darstellung dieser Werte wurde in der Klasse *QueryIcon*, die von *JComponent* erbt, durch eine HTML-formatierte einfache Textdarstellung, realisiert.

#### 4.4.6 Anzeige der Resultate

In der Testversion des *CircleSegmentView* werden die Ergebnisse in einer Ergebnistabelle präsentiert, die mit Hilfe der Klasse *JTable* aus dem Paket *java.swing.JTable* implementiert wurde. So kann eine Standardtabelle mit Sortierfunktion in den einzelnen Spalten und verstellbaren Spaltenbreiten erzeugt werden.



## Kapitel 5

# Evaluation des CircleSegmentView

### 5.1 Motivation für die Durchführung des Benutzertests

Ziel beim Entwurf des CircleSegmentView war die Bereitstellung einer Benutzeroberfläche, die dem Benutzer einen Überblick über die Verteilung der vorhandenen Daten bietet, um dem Benutzer das schnellere Auffinden der für ihn relevanten Datensätze zu ermöglichen als mit herkömmlichen Methoden.

Bis heute arbeiten viele gebräuchliche Benutzerschnittstellen zur Abfrage von Datenbanken auf formularbasierten Oberflächen. Deshalb soll durch die Benutzerstudie überprüft werden, ob die Verwendung des CircleSegmentView gegenüber den herkömmlichen formularbasierten Oberflächen Vorteile für den Benutzer bringt.

Zu Testzwecken wurde deshalb ein Formular entworfen, welches die allgemein übliche Formularabfrage ermöglicht. Um eine optimale Vergleichsmöglichkeit zu gewährleisten, wurde darauf geachtet, dass die Formularabfragen die gleiche Mächtigkeit besitzen wie die Abfragen, die mit dem CircleSegmentView durchgeführt werden können. Der Test wird in der Erwartung durchgeführt, dass die Benutzung des CircleSegmentView einen statistisch signifikanten Zeitvorteil gegenüber der Verwendung des Formulars bietet.

Aufgrund einer von [Tan01] durchgeführten Studie, bei der auch ein Query Preview basiertes System mit einem Formular verglichen wurde, wird bezüglich der subjektiven Einschätzung durch den Benutzer davon ausgegangen, dass die Benutzer die

Arbeit mit dem CircleSegmentView der Arbeit mit dem Formular vorziehen.

## 5.2 Testmethoden

Für die Evaluation einer Benutzerberfläche stehen prinzipiell mehrere Methoden zur Verfügung: Zuerst sei hier die heuristische Evaluation genannt. Es handelt sich dabei um eine Methode, bei der die Oberfläche anhand eines normierten Kriterienkataloges von Interfaceexperten überprüft wird [NM94]. Bei Anwendung der heuristischen Evaluation können bei vergleichsweise niedrigen Kosten viele Probleme, auch schwerwiegende gefunden werden [NM94]. Allerdings weist diese Methode den Nachteil auf, dass mehrere Experten nötig sind, die bezahlt werden müssen.

Erwähnt werden muss auch die Methode des Cognitive Walkthrough, die so konzipiert ist, dass von den Entwicklern selbst jedes einzelne Leistungsmerkmal untersucht und mit den spezifizierten Anforderungen verglichen wird. Als vorteilhaft erweist sich dabei die Tatsache, dass die Aufgabe von den Entwicklern selbst durchgeführt werden kann und die Verwendung dieser Methode somit zumindest keine zusätzlichen Kosten für externe Spezialisten zur Folge hat. Im Vergleich mit anderen Methoden [NM94] wurde bei näherer Überprüfung festgestellt, dass prinzipielle und sich wiederholende Probleme mit dem Cognitive Walkthrough oft nicht erkannt werden.

Bewährt hat sich von diesen unterschiedlichen Ansätzen vor allem die Methode des Usability Testing. Testpersonen müssen unter Anleitung bestimmte Aufgaben erfüllen, die im Idealfall so konzipiert sind, dass sie die wichtigsten Funktionsweisen und Benutzungsmöglichkeiten der Oberfläche abdecken. Problematisch an dieser Methode sind einzig und allein die hohen Kosten für Entwicklung und Auswertung des Tests. Hinzu kommen zusätzliche Kosten für ein geeignetes Testlabor, für Testmaterialien und Testpersonen. Ansonsten zeigte sich im Vergleich [NM94], dass vor allem schwerwiegende und wiederkehrende Probleme durch diese Methode aufgedeckt und beseitigt werden können. Bei der Evaluation des CircleSegmentView wurde die Methode des Usability Testing in der Form, wie in [May99] vorgeschlagen, angewandt. Diese Form des Usability Testings beruht auf einem iterativen Konzept und ist daher unter der Bezeichnung Iterative Conceptual Model Evaluation bekannt. Die Benutzerstudie wird dabei in zwei Abschnitte gegliedert: in die Planungs- und Durchführungsphase, deren Anwendung auf den CircleSegmentView im Folgenden

beschrieben werden.

## 5.3 Testbeschreibung

### 5.3.1 Testvoraussetzungen

Der Schwerpunkt der Evaluation besteht darin, festzustellen, ob die entworfene Oberfläche für den geübten Benutzer leicht und schnell zu benutzen ist, (ease of use focus). Voraussetzung dafür ist, dass die Benutzer über eine gewisse Erfahrung im Umgang mit dem zu testenden System verfügen, bevor mit der Durchführung des eigentlichen Tests begonnen wird.

Es mussten Testpersonen gefunden werden, die schon über ein gewisses Grundmaß an Computererfahrung verfügen, damit gleich mit der Schulung für das zu testende System begonnen werden kann, und nicht erst grundlegenden Dinge, wie z.B. der Umgang mit Maus und Tastatur erklärt werden müssen.

Die erste Schritt besteht in der Auswahl geeigneter Testpersonen. In Kapitel 5.3.2 wird beschrieben, wie ausgehend von diesen grundlegenden Anforderungen die Planung der Testdurchführung vorgenommen wurde.

### 5.3.2 Testplanung und Testvorbereitung

#### Testpersonen

Die Testpersonen sollten nicht nur bereits Erfahrung im Umgang mit dem CircleSegmentView haben, sondern auch über eine gewisse Erfahrung im Umgang mit Computern im Allgemeinen verfügen. Um Testpersonen mit den geforderten Computerkenntnissen zu finden, wurde der Kreis von möglichen Testpersonen auf Studenten und ehemalige Studenten beschränkt. Da es sich beim CircleSegmentView um ein neues System handelt, konnten naturgemäß keine Personen gefunden werden, die bereits über Erfahrung im Umgang mit dem CircleSegmentView verfügen. Deshalb soll die Expertennutzersituation durch eine vorangehende Schulung simuliert werden, deren genauer Ablauf in Kapitel 5.3.3 beschrieben wird. Eine Schulung wird auch von [May99] vorgeschlagen.

Erst wenn die Testperson mit den kleinen Übungsaufgaben gut zurechtkommt und keine Fragen mehr hat, wird mit der Durchführung des eigentlichen Benutzer-tests begonnen.

## Testfragen

Die Testfragen an sich wurden so konzipiert, dass folgende Punkte vorrangig überprüft werden können: es sollen Eigenschaften der Oberfläche, die für jeden Nutzer relevant sind, getestet werden. Darüber hinaus sollen Leistungsmerkmale, über die während der Entwicklung Uneinigkeit im Team bestand, überprüft werden [WC97]. Darunter sind vor allem schwer zu bedienende oder versteckte Benutzungsmöglichkeiten zu verstehen.

Anhand dieser Leitlinien wurden die konkreten Testaufgaben entwickelt, die jedoch zusätzlich noch das eigentliche Visualisierungsziel bestätigen sollten: die Bereitstellung einer Oberfläche, die es dem Benutzer erlaubt, Aussagen über die Verteilung der Daten zu machen und extreme Treffermengen bei der Suche (wie z. B. zu lange Trefferlisten oder gar keine Treffer) zu vermeiden.

Demgemäß wurde beim Entwurf der Testaufgaben von folgendem Szenario für den Einsatz des CircleSegmentView ausgegangen: man stelle sich ein Filmarchiv vor, dessen kompletter Inhalt in einer Datenbank gespeichert ist. Nun soll sich der Benutzer anhand des CircleSegmentView einen Überblick über den Inhalt des Archivs machen können und gleichzeitig auch nähere Informationen über die einzelnen Filme bekommen können. Die Fragen wurden nun anhand von typischen Fragen und Problemen, die bei der Suche nach Filmen vorkommen können, entwickelt. Mitverarbeitet wurden die Haupterwartungen, die bei der Entwicklung an die Visualisierung gestellt wurden:

## Erwartungen an den Test und Testaufgaben

Mithilfe des CircleSegmentView sollen die Benutzer schneller einen Eindruck von der Verteilung der Daten bekommen als mithilfe des Formulars, außerdem sollten sich Aufgaben, die darauf abzielen, Gruppen von Daten zu vergleichen, mit dem CircleSegmentView schneller lösen lassen als mit dem Formular. Auch wird davon ausgegangen, dass die Benutzer sich mit der Lösung ganz spezifischer Aufgaben mit dem CircleSegmentView zumindest genauso schwer tun, wie bei der Verwendung des Formulars. Zu jeder dieser Erwartungen wurde eine Aufgabengruppe mit drei Aufgaben entwickelt, deren Lösung die Benutzung der Interfacekomponenten erfordert. Diese werden nachfolgend kurz aufgelistet.

- Aufgabengruppe A

Die erste Aufgabengruppe (Aufgabentyp A) besteht aus unspezifischen Aufgaben, die Antworten über die Verteilung der Daten liefern sollen.

- Aufgabengruppe B

Bei der zweiten Aufgabengruppe (Aufgabentyp B) kommt es darauf an, dass ganz spezifische Fragen über einzelne Datensätze beantwortet werden.

- Aufgabengruppe C

Die letzte Aufgabengruppe (Aufgabentyp C) enthält Fragen, für deren Beantwortung bestimmte Bereiche von Daten verglichen werden sollen. Zu jeder Aufgabe wird die Zeit notiert, die für deren Lösung benötigt wird.

### 5.3.3 Testablauf und Testdurchführung

#### Begrüßung

Die Testpersonen werden zuerst zum Test begrüßt. Ihnen wird eine kurze Projektbeschreibung vorgelesen. (Siehe Anhang B.1.) Dadurch wird die Testperson mit Hintergrundinformation versorgt.

#### Pretestfragebogen

Nach der Begrüßung werden die Testpersonen gebeten, einen Pretestfragebogen mit Angaben zur Computererfahrung auszufüllen. (Siehe Anhang B.2). Dadurch wird gewährleistet, dass alle Testpersonen über die notwendigen grundlegenden Computerkenntnisse verfügen.

#### Schulung der Testpersonen

Da eine Expertennutzersituation simuliert werden sollte, war eine Schulung unerlässlich.

- Schulungsziel

Gemäß [Rub94] ist zuerst ein Schulungsziel festzulegen. Durch die Schulung sollten die Testpersonen mit dem CircleSegmentView so vertraut gemacht werden, dass sie alle Interaktionsmöglichkeiten kennen und bereits unter Anleitung

Testaufgaben bearbeitet haben. Es wurde solange geschult, bis die Testpersonen die Testaufgaben erfolgreich bearbeiten konnten. Auf diese Weise kann die Expertennutzersituation simuliert werden. Ein weiteres Ziel der Schulung bestand darin, alle Testpersonen auf den gleichen Kenntnisstand bezüglich des CircleSegmentView zu bringen.

- Umfang der Schulung

Nach [Rub94] ist der Umfang der Schulung nach dem Schulungsziel auszurichten. Die Dauer bestimmte sich somit danach, wie schnell die zu unterweisende Testperson in der Lage ist, die Übungsaufgaben fehlerfrei zu lösen.

- Schulungsmaterial

Primär sollte sich auch die Auswahl des Schulungsmaterials nach dem Schulungsziel richten. In [Rub94] werden die verschiedensten Schulungsmaterialien aufgezählt, diese gehen von einer Einweisung durch einen Spezialisten über schriftliches Material und Schulungsvideos bis hin zu zweitägigen Seminaren. Um das Schulungsziel zu erreichen, wurden die Testpersonen mündlich unterwiesen. Auf ein Schulungsvideo wurde hier verzichtet, da die Testpersonen dann nicht die Möglichkeit haben, sofort nachzufragen, wenn sie etwas nicht verstanden haben. Außerdem können bei einer mündlichen Schulung die Erklärungen auch dem Niveau der Testperson angepaßt werden. Testpersonen, die Schwierigkeiten haben, eine bestimmte Funktion zu verstehen, kann mit ganz einfachen Worten, wenn nötig auch mehrmals das nicht verstandene erklärt werden. Bei einem Video hingegen läuft eine Standarderklärung ab, die Testperson wagt eventuell nicht, wenn sie etwas nicht verstanden hat, sofort zu unterbrechen oder darum zu bitten, dass das Video noch einmal gezeigt werden soll. Die in [May99] vorgeschlagenen schriftlichen Schulungsmaterialien wurden ebenfalls nicht verwendet, da auch hier die Gefahr besteht, dass die Testperson über wichtige Details hinwegliest oder etwas falsch versteht. Bei einer mündlichen Erklärung hingegen fühlen sich die Testpersonen aufgrund der Kommunikation mit einer anderen Person herausgefordert, besonders aufmerksam zu sein, da sie keinen schlechten Eindruck hinterlassen möchten. Die Schulung wurde erst beendet, wenn die Testpersonen erfolgreich einige Übungsaufgaben lösen konnten.

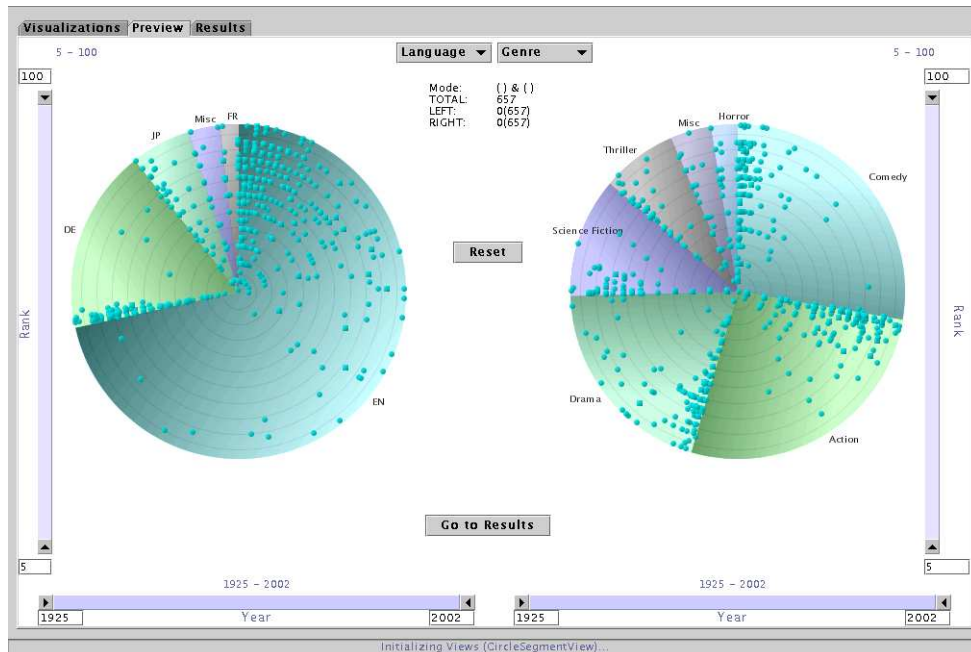


Abbildung 5.1: Testversion des CircleSegmentView

## TestszENARIO

Damit die Testpersonen während der Erledigung der Aufgaben ein Anwendungsbeispiel im Kopf haben, wurden sie vor der Bearbeitung der Testaufgaben gebeten, sich vorzustellen, sie seien in einer Videothek und müssten mit der jeweiligen Oberfläche Informationen über den Filmbestand abrufen.

## Bearbeitung der Testaufgaben

Jede Testperson musste neun Testaufgaben in der gleichen Reihenfolge mit dem Formular und mit dem CircleSegmentView lösen, wobei sich die Aufgaben für die beiden Oberflächen lediglich in kleinen Details unterschieden, um zu verhindern, dass die Testpersonen die Antworten beim Test der zweiten Oberfläche noch im Kopf haben. Wurde beispielsweise beim Formular nach französischen Filmen gefragt, lautet die entsprechende Frage beim CircleSegmentView [...] 'englische Filme' [...]. Die Reihenfolge, in der die Aufgaben gestellt wurden, lautet: A1 - B1 - A3 - B2 - C1 - C2 - A3 - B3 - C3. Um eine eventuelle Verzerrung der Ergebnisse durch bei der

Results (29)						
title	year	language	genre	version	rank	plot
Men in Black II	2002	EN	Science Fiction	OF	59	(mond05 /local_tmp/te...
E.T. - The Extra Terrestrial	1982	EN	Science Fiction	OF	65	A group of aliens visit e...
Bicentennial Man	1999	EN	Science Fiction	OF	40	This film follows the life...
Dune	1984	EN	Science Fiction	OF	70	In the distant future, a ...
Vanilla Sky CD 1	2001	EN	Science Fiction	OF	80	A successful publisher fi...
Star Trek: First Contact	1996	EN	Science Fiction	OF	80	The time is the 24th ce...
Star Trek: Generations	1994	EN	Science Fiction	OF	70	Star Trek: Generations i...
Star Trek VI: The Undiscovered Country	1991	EN	Science Fiction	OF	60	An explosion of their ke...
Star Trek V: The Final Frontier	1989	EN	Science Fiction	OF	60	The crew of the Federat...
Star Trek IV: The Voyage Home	1986	EN	Science Fiction	OF	90	A space probe appears...
Star Trek III: The Search for Spock	1984	EN	Science Fiction	OF	75	Admiral Kirk's defeat of...
Lost in Space	1998	EN	Science Fiction	OF	70	In the near future, earth...
Day the Earth stood still, The	1951	EN	Science Fiction	OF	80	An alien (Klaatu) with h...
2001 - A Space Odyssey	1968	EN	Science Fiction	OF	100	Moon explorers encount...
Red Planet	2000	EN	Science Fiction	OF	88	Astronauts search for so...
Final Fantasy	2001	EN	Science Fiction	OF	88	A female scientist make...
Blade Runner	1982	EN	Science Fiction	OF	88	Deckard, a blade runne...
Starship Troopers	1997	EN	Science Fiction	OF	10	In the distant future hig...
Men in Black	1997	EN	Science Fiction	OF	20	Two men who keep an ...
Demolition Man	1993	EN	Science Fiction	OF	80	A cop is brought out of ...
Abyss, The	1989	EN	Science Fiction	OF	50	An American nuclear su...
Catana	1997	EN	Science Fiction	OF	99	Futuristic story of a gen...
Matrix	1999	EN	Science Fiction	OF	85	A computer hacker lear...
Star Wars II - The Empire Strikes Back	1980	EN	Science Fiction	OF	85	While Luke takes advan...
Tron	1982	EN	Science Fiction	OF	85	Computer Classic, one o...
Judge Dredd	1995	EN	Science Fiction	OF	20	In a dystopic future, Dre...
Escape from New York	1981	EN	Science Fiction	OF	10	In 1997, when the US Fi...
The Matrix	1999	EN	Science Fiction	OF	100	A computer hacker lear...
Epoch (2000)	2000	EN	Science Fiction	OF	89	Disaster strikes as a sp...

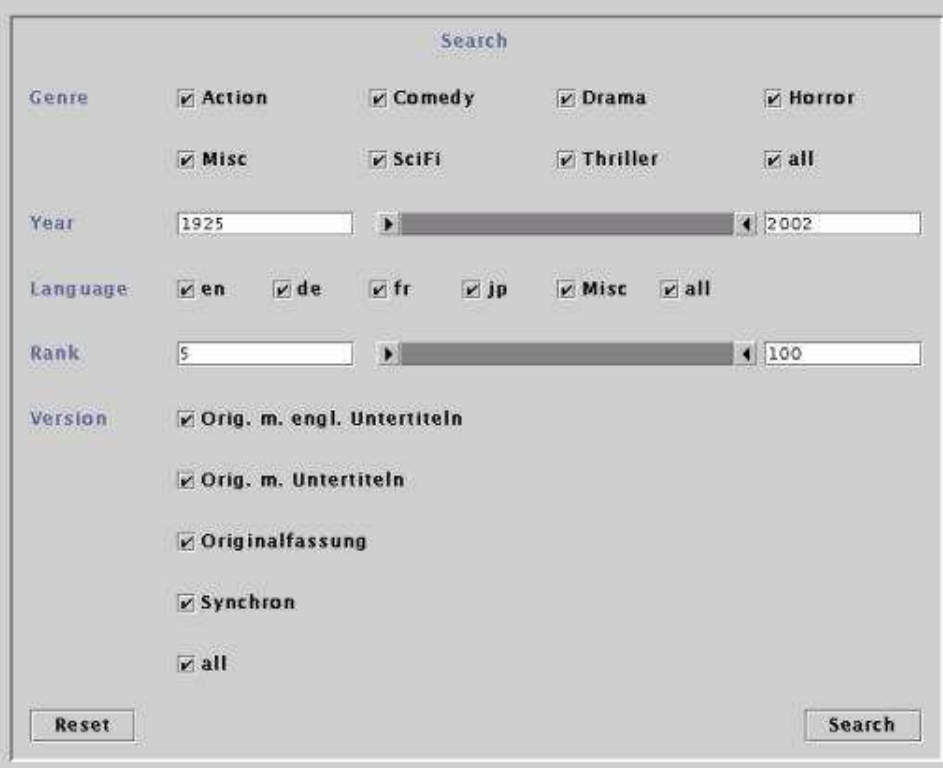
Abbildung 5.2: Für den Test verwendete Ergebnistabelle

Testperson eintretende Müdigkeit beim Test von zwei Oberflächen nacheinander zu verhindern, wurden die Testpersonen in zwei Gruppen eingeteilt. Testpersonen der Gruppe I mussten zuerst die Aufgaben mit dem Formular lösen, (siehe Abbildung 5.3), und dann mit Hilfe des CircleSegmentView, (siehe Abbildung 5.1), während Testpersonen der Gruppe II zuerst das Formular benutzen mussten.

Es wurde entschieden, die Testaufgaben den Testpersonen vorzulesen und nicht vorzulegen. Die Lösung der Aufgaben erfolgte dabei unter Verwendung der Ergebnistabelle in Abbildung 5.2, die sehr intuitiv benutzt werden kann und daher keine zusätzlichen Schwierigkeiten für den Benutzer bedeutet.

Um die Situation während des Tests aufzulockern, und eventuelle Versagensängste seitens des Benutzers zu verhindern, wird der Benutzer mehrfach durch den Testleiter darauf aufmerksam gemacht, dass die Software getestet wird und nicht der Nutzer. Die Zeit, die die einzelnen Benutzer dann für die Erledigung der Aufgaben brauchten, wurde gemessen. Auch wurde ein schriftliches Protokoll geführt, um eventuelle Besonderheiten bei der Aufgabenerledigung festzuhalten. Während der Erledigung der Testaufgaben wurden die Personen von hinten auf Video aufgenommen. Damit auch das Gesicht der Person aufgenommen werden kann, wurde jedoch ein Spiegel links neben dem Monitor platziert, der das Gesicht der Testperson so spiegelt, dass eine Videoaufnahme des Gesichts von hinten mittels des Spiegels möglich war. Das hat den Grund, dass die Vorgänge auf dem Bildschirm und das Gesicht des Nutzers auf den Videos zu sehen sein sollte. Auf den Einsatz einer Screencam





The image shows a search form titled "Search" with the following sections:

- Genre:** Action, Comedy, Drama, Horror, Misc, SciFi, Thriller, all (all checked).
- Year:** Range from 1925 to 2002.
- Language:** en, de, fr, jp, Misc, all (all checked).
- Rank:** Range from 5 to 100.
- Version:** Orig. m. engl. Untertiteln, Orig. m. Untertiteln, Originalfassung, Synchron, all (all checked).

Buttons for "Reset" and "Search" are located at the bottom left and right respectively.

Abbildung 5.3: Das verwendete Formular

musste aus technischen Gründen verzichtet werden. Das Video des Tests wurde mit Ton aufgezeichnet, da der Benutzer am Anfang des Tests dazu angehalten wird, während der Lösung der Testaufgaben laut zu denken. Die Videoaufzeichnung hat den Zweck Kommentare, Mimik und Gestik der Testperson aufzuzeichnen.

### Posttestfragebogen

Direkt nach der Bearbeitung der ersten Oberfläche wurden die Testpersonen gebeten, einen Fragebogen mit Eindrücken über die gerade getestete Oberfläche auszufüllen. Anschließend folgte die Bearbeitung der Aufgaben mit Hilfe der jeweils anderen Oberfläche sowie das Ausfüllen des Fragebogens für die zuletzt getestete Oberfläche. Die Testpersonen mussten ebenfalls einige abschließenden Fragen beantworten. (Siehe Anhang B.3.)



Abbildung 5.4: Ansicht der Versuchsanordnung (nachgestellt).

## 5.4 Ergebnisse

In diesem Kapitel werden die aus den Benutzertests gewonnenen Rohdaten mithilfe statistischer Methoden analysiert. Durch die Auswertung soll überprüft werden, ob ein statistisch signifikanter Unterschied zwischen der Bearbeitung der Testaufgaben mit dem Formular und der Bearbeitung der Testaufgaben mit dem CircleSegmentView besteht.

### 5.4.1 Art und Umfang der Stichprobe

Prinzipiell empfiehlt es sich bei jedem Test, jeder Untersuchung oder Erhebung, möglichst viele Stichproben zu nehmen. Ein großer Stichprobenumfang bildet immer eine sichere Datengrundlage. Hinzu kommt, dass dann bei der Auswertung Verfahren angewendet werden können, die die Normalverteilung der Daten voraussetzen und die im Allgemeinen schärfer testen<sup>1</sup>. In der Praxis bedeutet ein großer Stichprobenumfang jedoch einen hohen Kosten- und Zeitaufwand. Für den Test des CircleSegmentView wurde deshalb beschlossen, ähnlich wie in [Tan01] einen kleineren

<sup>1</sup>Das ergibt sich aus dem zentralen Grenzwertsatz, nach dem Summen beliebig verteilter Zufallsgrößen im Grenzfall normalverteilt sind

Stichprobenumfang zu nehmen. Die Wahl fiel auf 20 Testpersonen. Wie in Kapitel 5.3.3 beschrieben, mussten alle Testpersonen Aufgaben mit dem Formular und dem CircleSegmentView bearbeiten. Diese Art von Stichproben werden als paarige oder verbundene Stichproben bezeichnet. Verbundene Stichproben liegen genau dann vor, wenn eine Beobachtung der einen Stichprobe genau einer Beobachtung der anderen Stichprobe entspricht; das ist insbesondere bei gleichen Untersuchungseinheiten der Fall [Sac97]. Alternativ dazu wäre der Test auch mit vierzig Versuchspersonen durchführbar gewesen, von denen die eine Hälfte die Aufgaben mit dem Formular bearbeitet hätten und die andere Hälfte mit dem CircleSegmentView. Gemäß [Sac97] sind paarige Stichproben vorzuziehen. In Tabelle 5.1 werden die erhaltenen Zeiten nach ihrer Zugehörigkeit zu Aufgabengruppen tabellarisch dargestellt. Die Zeiten geben den Unterschied in Sekunden zwischen der Zeit an, die eine Testperson für die Erfüllung einer bestimmten Testaufgabe mit dem Formular und der Zeit, die eine Testperson zur Erfüllung derselben Aufgabe mit dem CircleSegmentView benötigt hat. Gemessen wurde jeweils die Zeit, die die Benutzer nach der vollständigen Aufgabenstellung bis zum Beginn der richtigen Antwort benötigten. Durch das Messen der reinen Nettozeit werden mögliche Verzerrungen der Ergebnisse durch langsames oder schnelleres Vorlesen der Aufgaben oder weitschweifige Antworten des Benutzers ausgeschaltet.

Benötigte der Benutzer eine Hilfestellung, wurde die Aufgabe als nicht gelöst gewertet. Tabelle 5.1 zeigt einen tabellarischen Überblick über die zeitliche Differenzen  $\Delta t$  in Sekunden, die sich aus den Zeiten  $t_F$ , die die Testpersonen für die Erledigung der Aufgaben mittels der formularbasierten Oberfläche und den Zeiten  $t_{CSV}$ , die die Testpersonen für die Erledigung der Aufgaben mit Hilfe des CircleSegmentView ergeben.

Somit gilt:

$$\Delta t = t_F - t_{CSV} \quad (5.1)$$

Negative Zeiten bedeuten dabei, dass die jeweilige Aufgabe mit dem CircleSegmentView langsamer gelöst wurde als mit dem Formular. Umgekehrt bedeutet eine positive Zeitdifferenz, dass die Aufgabe mit Hilfe des Formulars langsamer gelöst wurde. Ein 'F' bedeutet, dass die Aufgabe gar nicht oder falsch gelöst wurde oder die Testperson länger als fünf Minuten benötigt hat.

Testperson	Aufgabentyp A			Aufgabentyp B			Aufgabentyp C		
	A 1	A 2	A 3	B 1	B 2	B 3	C 1	C 2	C 3
1	+5	F	+21	+41	-47	F	+76	-29	F
2	+1	-8	0	+1	-4	-64	+59	F	-11
3	-1	-2	F	F	+1	F	+53	-15	-35
4	+10	+35	-1	+2	+1	-11	+37	-7	-5
5	+5	0	+11	+1	-13	+14	+173	F	-6
6	+2	-6	-8	+10	-1	F	+37	F	-7
7	+6	-9	-10	+6	+2	F	F	-21	-51
8	0	-4	+2	+11	0	-29	+58	-41	-35
9	-6	F	-12	-35	+2	-36	F	+13	+1
10	F	+4	+1	+2	F	F	+11	F	-18
11	+28	-27	F	+19	+1	F	F	-62	F
12	F	+1	+54	+14	+5	F	+50	+64	F
13	+4	-13	F	+4	-3	-8	+16	F	-7
14	-18	-19	-34	-46	0	+7	F	F	-126
15	-19	+17	+10	+32	-2	-60	+64	-96	F
16	+1	+1	+7	+14	+28	-17	+14	+23	F
17	F	F	F	-13	0	-78	F	F	F
18	-20	0	-6	+16	-13	-35	F	F	+15
19	-2	+16	0	+63	-13	F	+123	+73	-15
20	+22	-5	+5	-104	-6	F	+65	+41	F

Tabelle 5.1: Zeitunterschiede  $\Delta t$  bei der Aufgabenerfüllung mit Formular und Circle-SegmentView. Ein 'F' steht für nicht oder falsch gelöste Aufgaben.

### 5.4.2 Statistische Methoden

Bei der statistischen Auswertung von Rohdaten können zwei Ansätze unterschieden werden:

- Nicht-parametrische Verfahren

Bei dieser Vorgehensweise spielt die den Rohdaten zugrunde liegende Verteilung keine Rolle. Entscheidet man sich eine für eine Auswertung mit den nicht-parametrischen Verfahren, kommen im vorliegenden Fall der Vorzeichenstest und der Wilcoxon-Rang-Test zur Anwendung.

- Parametrische Verfahren

Parametrische Verfahren hingegen setzen Normalverteilung der zugrunde liegenden Daten voraus. Als Beispiel für ein in Frage kommendes parametrisches Verfahren ist hier der T-Test zu nennen.

Bevor mit der eigentlichen Auswertung begonnen wird, sollen in Kapitel 5.4.3 die prinzipielle Vorgehensweise und verwendete Begriffe geklärt werden.

### 5.4.3 Hypothesen und Signifikanzniveau

Gestellt werden muss die Nullhypothese  $H_0$  und die Alternativhypothese  $H_1$ . Wird die Nullhypothese abgelehnt, kann man dieses Ergebnis als statistisch signifikant auf dem 5%-Niveau bezeichnen, wenn das Signifikanzniveau auf 5% festgelegt wird, was bei dem durchgeführten Test der Fall ist. Zu betonen ist in diesem Zusammenhang, dass das Beibehalten der Nullhypothese die Gefahr mit sich bringt, dass das Beibehalten unberechtigt ist, (Fehler zweiter Art) [Sac97]. Die Nichtablehnung der Nullhypothese sagt somit solange nicht über deren Gültigkeit aus, wie die Wahrscheinlichkeit des Auftretens eines Fehlers zweiter Art (Beta-Fehler) unbekannt ist. Da dieser Fehler jedoch im Allgemeinen sehr hoch ist, sollten aus der Nichtablehnung der Nullhypothese keine Schlüsse bezüglich deren Gültigkeit gezogen werden [Sac97]. Die Werte wurden dann als statistisch signifikant auf dem 5%-Niveau bezeichnet, wenn der P-Wert<sup>2</sup> für den jeweiligen Test kleiner als 0.05 war. Dann musste die gestellte Nullhypothese abgelehnt werden. Alle Tests wurden mit der Statistiksoftware SPSS durchgeführt.

---

<sup>2</sup>Der P-Wert bezeichnet das nominelle Signifikanzniveau

Aufgabe	P-Wert
A1	0.205
A2	0.333
A3	0.040
B1	0.010
B2	0.010
B3	0.807
C1	0.026
C2	0.975
C3	0.010

Tabelle 5.2: P-Werte für den Shapiro–Wilk–Test

#### 5.4.4 Test auf Normalverteilung

Zuerst muss geprüft werden, ob die Daten normalverteilt sind. Ist das der Fall, müssen parametrische Verfahren für die Auswertung der Daten angewendet werden.

##### Shapiro–Wilk–Test

Der Shapiro–Wilk–Test gestattet folgende Hypothesen  $H_0$  und  $H_1$ :

$H_0$  : Die Daten sind normalverteilt.

$H_1$  : Die Daten sind nicht normalverteilt.

In Tabelle 5.2 werden die P-Werte für die einzelnen Aufgaben aufgelistet. Da bis auf vier Werte alle Werte kleiner als 0.05 sind, ist in diesen Fällen die Nullhypothese abzulehnen. Das heißt, die Daten sind nicht normalverteilt. Es müssen nicht-parametrischen Verfahren angewendet werden, da eine einheitliche Vorgehensweise für alle Testaufgaben notwendig ist.

##### Normalverteilungsgrafik

Um die mit dem Wilk–Shapiro–Test gewonnenen Ergebnisse zu verdeutlichen, wurde für die Testaufgaben eine Normalverteilungs–Graphik erstellt [DHM<sup>+</sup>95]. Dazu werden die Punkte  $x_i$  bestimmt, für die die Normalverteilungsfunktion  $\Phi(x_i) = i/10$ . In Anhang A sind die Ergebnisse dargestellt. Wären die Daten normalverteilt, wäre

es möglich, eine Gerade so einzuzichnen, dass alle Punkte annähernd auf dieser Gerade liegen. Das ist jedoch bei dieser Verteilung der Daten nicht zweifelsfrei möglich. Die Normalverteilungsgrafik bestätigt die mit dem Wilk–Shapiro–Test erhaltenen Ergebnisse.

#### 5.4.5 Vergleich zweier verbundener Stichproben

Für den nicht-parametrischen Vergleich zweier verbundener oder paariger Stichproben, (siehe Kapitel 5.4.1), eignen sich vor allem der Vorzeichen–Test von Dixon und Mood und der Vorzeichen–Rang–Test von Wilcoxon.

##### Vorzeichen–Test nach Dixon und Mood

Beim Vorzeichentest nach Dixon und Mood handelt es sich um einen Schnelltest zur Prüfung des Unterschiedes der zentralen Tendenz zweier verbundener Stichproben [Sac97]. Wesentliche Voraussetzung für die Anwendbarkeit ist die Unabhängigkeit der Ergebnisse der beiden Paare. Wie bereits erwähnt wurde das durch das Stellen von vergleichbaren aber leicht unterschiedlichen Fragen für das Formular und den CircleSegmentView erreicht. (Siehe Anhang B.) Geprüft wird die Nullhypothese, dass die Verteilung der Differenzen den Median Null hat, und die Differenzen der gepaarten Beobachtungen sich im Durchschnitt nicht von Null unterscheiden.

Abbildung 5.4.5 gibt eine Übersicht, wie viele Aufgaben des jeweiligen Typs überhaupt schneller mit dem CircleSegmentView gelöst wurden. Die Zeiten werden im Folgenden nach Aufgabentyp geordnet als Balkendiagramme präsentiert. Anhand der Abbildung 5.4.5 ist die Tatsache leicht zu erkennen, dass teilweise die Aufgaben von den Testpersonen mit dem CircleSegmentView schneller erledigt wurden, als mit dem Formular. Ob diese Beobachtung auch statistisch signifikant ist, wird nun anhand des Vorzeichentests und des Vorzeichen–Rang–Tests von Wilcoxon geklärt.

Im vorliegenden Fall ist eine Nullhypothese dergestalt gestattet, dass die Benutzung des CircleSegmentView keinen Zeitvorteil bringt, sondern die Aufgaben genauso schnell mit Hilfe des Formulars erledigt werden können.

Es wird somit die Nullhypothese  $H_0$  und die Alternativhypothese  $H_1$  für die Auswertung der Aufgaben wie folgt formuliert:

$H_0$  : Mit Formular und CircleSegmentView lassen sich  
Aufgaben gleich schnell erledigen.

$H_1$  : Eine der beiden Visualisierungen bringt einen Zeitvorteil.

Die Berechnung des Unterschiedes der zentralen Tendenz der Stichproben erfolgt nach folgender Formel:

$H_0 : \tilde{\mu}_d = 0$  gegen  $H_1 : \tilde{\mu}_d \neq 0$  für

$$\hat{z} = (|n - 2h| - 1) / \sqrt{n} > 1.96 = z_{0.05} \quad (5.2)$$

Wobei die Zahl der Vorzeichen  $n$  sei und das seltenere Vorzeichen trete  $h$  mal auf. Dann wird anhand einer geeigneten Tabelle [Sac97] überprüft, ob das erhaltene Ergebnis innerhalb der Schranken des gewählten Vertrauensbereichs (5%-Vertrauensbereich) liegt.

- Ergebnis des Vorzeichen-Tests für Aufgaben des Typs A

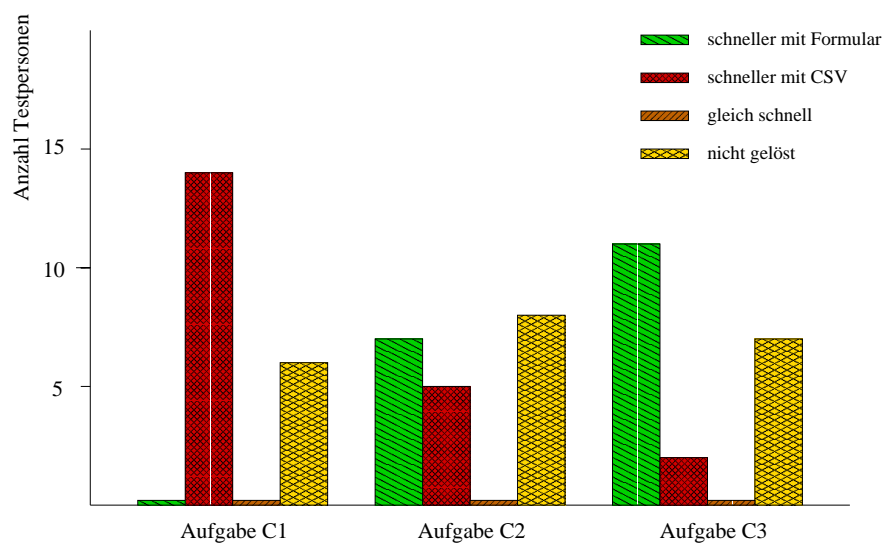
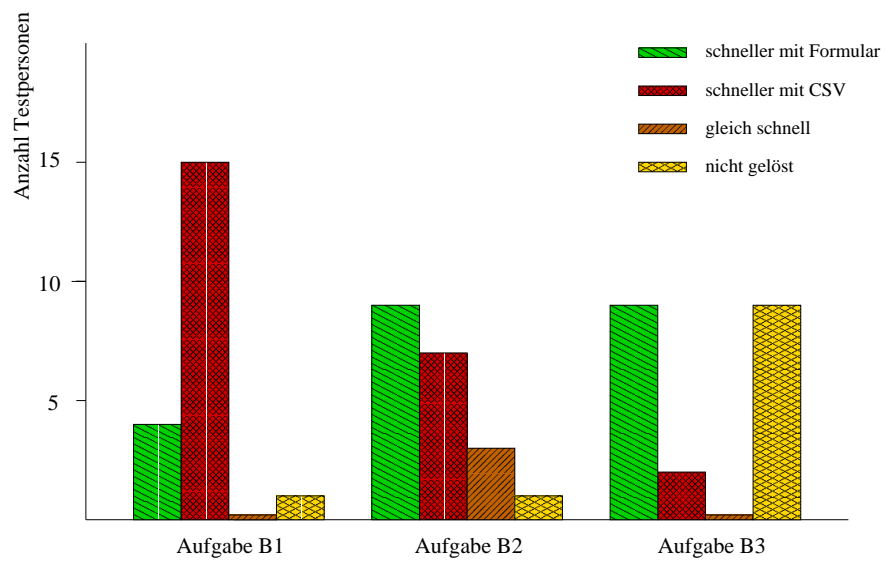
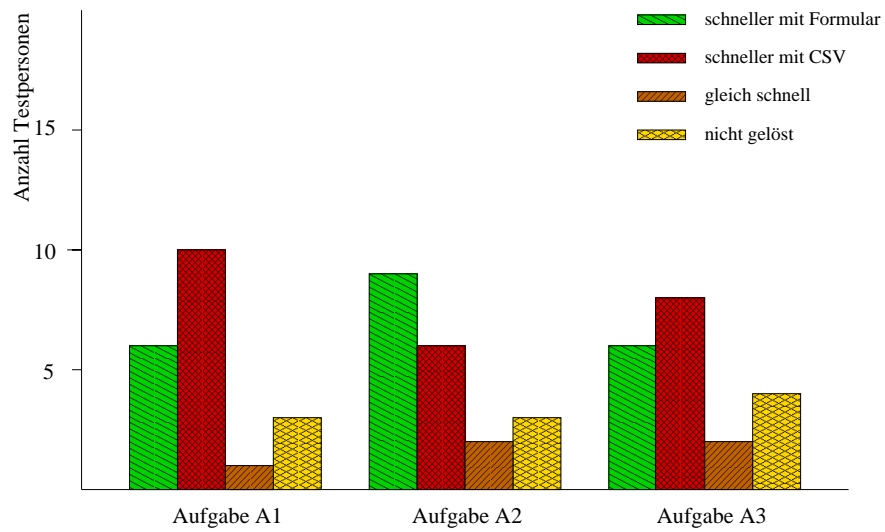
Bei den Aufgaben des Typs A hat dieser Test das Ergebnis, dass bei allen drei Aufgaben die Nullhypothese nicht verworfen werden kann. Aufgaben des Typs A beinhalteten eine Fragestellung nach der groben Verteilung der Daten. Es kann aufgrund der Nichtablehnung der Nullhypothese nicht gezeigt werden, dass bei Aufgaben des Typs A ein statistisch signifikanter Unterschied zwischen der Aufgabenerledigung mit dem Formular und dem CircleSegmentView besteht.

- Ergebnis des Vorzeichen-Tests für Aufgaben des Typs B

Bei Aufgaben des Typs B, also solchen Aufgaben, die eine Fragestellung nach einzelnen Datensätzen zum Gegenstand hatten, kann die Nullhypothese bei den Aufgaben B2 und B3 auf dem 5%-Niveau nicht abgelehnt werden.

Bei Aufgabe B1 allerdings kann die Nullhypothese aufgrund des Testergebnisses verworfen werden. Es besteht somit bei dieser Aufgabe höchstwahrscheinlich ein statistisch signifikanter Unterschied zwischen der Aufgabenerledigung





mit dem Formular und dem CircleSegmentView. Ein Blick auf die Rohdaten sorgt für die richtige Interpretation dieses Ergebnisses: die Anzahl der Testpersonen (fünfzehn waren besser mit dem CircleSegmentView, vier schlechter), die mit dem CircleSegmentView schneller die Aufgabe erledigten, darauf hin, dass der CircleSegmentView bei Aufgabe B1 dem Formular überlegen ist.

- Ergebnis des Vorzeichen-Tests für Aufgaben des Typs C

Bei der dritten Aufgabengruppe, die den Vergleich von Datengruppen zum Gegenstand hatte, waren die Ergebnisse eindeutig. Bei Aufgabe C1 konnte die Nullhypothese mit einem deutlichen Hinweis auf eine schnellere Aufgabenerledigung mit dem CircleSegmentView abgelehnt werden. Bei Aufgabe C2 konnte die Nullhypothese auf dem 5%-Niveau nicht verworfen werden. Bei Aufgabe C3 kann die Nullhypothese auf dem 5%-Niveau abgelehnt werden. Elf Testpersonen erledigten die Aufgabe schneller mit dem Formular, zwei besser mit dem CircleSegmentView. Das deutet auf eine schnellere Erledigung dieser Aufgabe mit dem Formular hin

Nach Anwendung des Vorzeichentests auf die aus dem Test gewonnenen Daten, erhält man lediglich bei der Aufgabengruppe A ein eindeutiges Ergebnis: es besteht kein statistisch signifikanter Unterschied zwischen der Aufgabenerledigung mit Formular und CircleSegmentView. Da es sich beim Vorzeichentest lediglich um einen Schnelltest handelt, erscheint es sinnvoll, die Daten noch mit dem Wilcoxon-Vorzeichen-Rang-Test für Paardifferenzen zu überprüfen. Das kann vor allem bei den Aufgabentypen B und C zu eindeutigeren Ergebnissen führen.

### **Auswertung der Ergebnisse mit dem Vorzeichen-Rang-Test von Wilcoxon**

Vorteil des Wilcoxon-Tests ist, dass er auch auf nicht normalverteilte Daten angewendet werden kann, und darüber hinaus normalverteilte Daten fast ebenso scharf testet wie der T-Test [Sac97]. Wirksamkeit und Effizienz liegen bei großen und kleinen Stichproben bei 95 Prozent.

Mit dem Wilcoxon-Test wird geprüft, ob die Differenzen paarig angeordneter Beobachtungen symmetrisch mit dem Median gleich Null verteilt sind [Sac97]. Hierzu werden die Differenzen zwischen den gepaarten Beobachtungen nach ihren absolu-

Aufgabe	Rangsumme	kritischer Wert
A1	55	34
A2	45.5	34
A3	46	34
B1	60	29
B2	44.5	40
B3	5	46
C1	0	21
C2	35.5	13
C3	8.5	17

Tabelle 5.3: Kritische Werte und Rangsummen für den Wilcoxon-Test

ten Beträgen aufsteigend geordnet, mit Ausnahme der Paare mit gleichen Einzelwerten. Die positiven und negativen Werte werden mit Rangzahlen entsprechend ihrem Vorzeichen belegt, gleichen Werten werden gemittelte Rangzahlen zugeordnet. Die Summen der positiven und negativen Rangzahlen werden addiert, wobei die jeweils kleinere Summe als Testgröße benutzt wird. Die Nullhypothese wird verworfen, wenn die Testgröße kleiner oder gleich dem Schwellwert einer geeigneten statistischen Tabelle ist.

Tabelle 5.3 enthält die Rangsummen und kritischen Werte. Tabelle 5.4 zeigt die P-Werte der Auswertung mit SPSS. Somit ergibt sich auch nach diesem Test auf dem 5%-Niveau kein statistisch signifikanter Unterschied bei Aufgaben des Typs A. Bei den Aufgaben des Typs B zeigt sich hier, dass bei Aufgabe B1 die Nullhypothese nicht verworfen werden kann. Es handelt sich um einen Grenzfall, da immerhin 15 Personen schneller mit dem CircleSegmentView waren und lediglich 4 Testpersonen langsamer. Bei Aufgabe B3 besteht ein statistisch signifikanter Unterschied zwischen dem Formular und dem CircleSegmentView. Die Rohdaten deuten auf eine schnellere Aufgabenerledigung mit Formular hin.

Bei Aufgaben des Typs C wird die Nullhypothese bei den Aufgaben C1 und C3 verworfen. Bei Aufgabe C1 erledigten alle Testpersonen die Aufgabe schneller mit dem CircleSegmentView, während bei Aufgabe C3 der CircleSegmentView schlechter abschneidet.

Aufgabe	P-Wert
A1	0.501
A2	0.410
A3	0.683
B1	0.159
B2	0.223
B3	0.013
C1	0.001
C2	0.784
C3	0.010

Tabelle 5.4: P-Werte beim Wilcoxon-Test für die Aufgaben

#### 5.4.6 Vergleich der Gruppen

Interessant ist außerdem, ob es einen Unterschied macht, ob die Testpersonen die Aufgaben zuerst mit dem Formular oder CircleSegmentView erledigen. Die Testpersonen, die die Aufgaben zuerst mit dem Formular lösen und dann mit dem CircleSegmentView gehören Gruppe I an. Die Testpersonen, die die Aufgaben zuerst mit dem CircleSegmentView lösen mussten, gehören der Gruppe II an.

#### U-Test

Um zu überprüfen, ob die Reihenfolge der getesteten Benutzeroberfläche einen Einfluß hat, wird der U-Test nach Wilcoxon, Mann und Whitney benutzt [Sac97]. Er gestattet folgenden Hypothesen:

$H_0$  : Es besteht kein Unterschied zwischen  
den Gruppen I und II

$H_1$  : Es besteht ein Unterschied

In Tabelle 5.5 werden die P-Werte für den Test dargestellt.

Die Nullhypothese kann für keine der Aufgaben abgelehnt werden. Ein signifikanter Unterschied zwischen den Zeiten für die Aufgabenerledigung in Gruppe I und in Gruppe II kann nicht festgestellt werden.

---

Aufgabe	P-Wert
A1	0.413
A2	0.885
A3	0.525
B1	0.414
B2	0.935
B3	0.855
C1	1.000
C2	0.337
C3	0.938

Tabelle 5.5: P-Werte für den Mann-Whitney U-Test

#### 5.4.7 Subjektive Zufriedenheit

Die Benutzer wurden sowohl nach Erledigung der Aufgaben mit dem Formular als auch nach Erledigung der Aufgaben mit dem CircleSegmentView gebeten, einen Fragebogen auszufüllen, der Fragen nach der subjektiven Zufriedenheit mit dem jeweiligen Interface enthielt.

Um auszuschließen, dass die Benutzer aufgrund von Schwierigkeiten, die Testaufgaben zu verstehen, diese nicht richtig oder nur sehr langsam lösen konnten, wurden die Testpersonen zuerst gefragt, ob die Aufgaben sprachlich und inhaltlich zu verstehen waren. Auf einer Skala von 1 bis 7, wobei 1 unverständlich und 7 besonders verständlich bedeutete, gaben 70 Prozent an, dass sie die Aufgaben als sehr verständlich formuliert empfanden (6 und 7), 25 Prozent waren der Ansicht, dass die Aufgaben verständlich formuliert waren (5 und 4), während 5 Prozent Schwierigkeit mit dem Verständnis der Aufgaben haben.

Ähnlich deutlich äußerten sich die Testpersonen über die Frage, ob die Aufgaben klar voneinander abgegrenzt gewesen seien. Hier waren 80 Prozent der Befragten der Meinung, dass die Aufgaben klar abgegrenzt waren (6 und 7), 15 Prozent äußerten sich auf der Skala mit einer 5 oder einer 4, während 5 Prozent die Aufgaben als nicht klar abgegrenzt empfanden.

Nach Erledigung der Testaufgaben für die jeweilige Benutzeroberfläche wurden die Testpersonen gebeten, Fragen zu ihrer subjektiven Zufriedenheit mit der gerade ge-

testeten Oberfläche zu beantworten, (siehe Anhang B.3). Auch hier waren Antworten auf einer Skala von eins bis sieben möglich.

Auf Fragen bezüglich der Verständlichkeit und Klarheit des Layouts waren deutlich mehr Testpersonen dazu bereit, dem Formular eine deutlich positiverer Bewertung im oberen Bereich der Bewertungsskala zu geben. Im mittleren Notenbereich (Bewertung vier und fünf) schneidet der CircleSegmentView deutlich besser ab.

Was die Bewertung der Schnelligkeit der Informationsbeschaffung und der allgemeinen Nützlichkeit betrifft, wurden beide Oberflächen relativ ähnlich bewertet.

Bei der Frage nach der intuitiven Navigation und ob sich die Navigation eventuell verwirrend ausgewirkt habe, ergibt sich wieder dasselbe Bild wie bei der Beurteilung der Verständlichkeit:

mehr Testpersonen waren im oberen Bewertungsbereich dazu bereit, das Formular besser zu bewerten als den CircleSegmentView.

Deutlich positiver fielen die Angaben zu der Frage aus, ob die Arbeit mit dem jeweiligen Interface der Testperson Spaß bereitet hatte und ob die verwendeten Farben von der Testperson als angenehm empfunden wurden: hier wurde der CircleSegmentView sehr gut bewertet im Vergleich zum Formular.

Bis auf eine Testperson gaben alle an, dass sie gerne mit dem CircleSegmentView arbeiten würden; demgegenüber waren drei Testpersonen nicht dazu bereit, dass Formular bei ihrer täglichen Arbeit zu verwenden.

Insgesamt wurde eine hohe subjektive Zufriedenheit mit beiden Oberflächen beobachtet.

#### 5.4.8 Kommentare

Die Testpersonen hatten beim Ausfüllen des Posttestfragebogens, (siehe Anhang B.3), die Möglichkeit, spontan Begriffe mit der zuvor getesteten Benutzeroberfläche frei zu assoziieren. Das bietet bei der Auswertung die Möglichkeit herauszufinden, ob der Benutzer entgegen seiner eigenen Einschätzung eventuell mit der Oberfläche nicht so gut zurecht gekommen ist. Tendenziell neigen Versuchspersonen dazu bei abschließenden Fragebögen sich aus Höflichkeit prinzipiell positiv zu äußern. Beim freien Assoziieren wird die Selbstkontrolle durch den Wunsch spontan zu sein, überdeckt. Beim vorliegenden Test wurden die assoziierten Begriffe in positive, negative und neutrale Begriffe eingeteilt. Auffällig ist dabei, dass mit dem Formular weniger

negative Begriffe assoziiert wurden, als mit dem CircleSegmentView. Bei Betrachtung der positiven Assoziationen fällt genau das Gegenteil auf: mit dem Formular wurden mehr positive Begriffe in Verbindung gebracht, als mit dem CircleSegmentView. Das mag zum einen daran liegen, dass der Mehrheit der Benutzer der Umgang mit formularbasierten Oberflächen vertraut ist, und das völlig neue Konzept des CircleSegmentView wohl auch verwirrend auf den Benutzer gewirkt hat. Das dadurch gewonnene Bild rundet die bisher erhaltenen Ergebnisse ab, darf jedoch nicht zur alleinigen Interpretation genutzt werden.

## 5.5 Diskussion der Testergebnisse

Das Ergebnis der durchgeführten Benutzerstudie entspricht teilweise den an den CircleSegmentView gestellten Erwartungen. Überraschenderweise schnitt der CircleSegmentView bei Fragen nach einzelnen Datensätzen bei einer Aufgabe gut ab. Bei Aufgaben, die den Vergleich von Datengruppen zum Gegenstand hatten, schnitt der CircleSegmentView bei einer Aufgabe sehr gut ab. Bei Aufgaben nach der groben Verteilung der Daten konnte kein statistisch signifikanter Unterschied zwischen CircleSegmentView und Formular festgestellt werden.

An dieser Stelle soll nun diskutiert werden, wie die Ergebnisse zu werten sind und warum die Ergebnisse nicht in allen Punkten den Erwartungen entsprechen:

- Die Schulung

Der CircleSegmentView ist für die Benutzung durch Experten vorgesehen. Die Testpersonen wurden zwar geschult, es war jedoch deutlich zu beobachten, dass die Dauer der Schulung nicht ausreichend war. Die Testpersonen waren zwar nach Abschluß der Schulung in der Lage die Übungsaufgaben zu beantworten, während des Tests wurde jedoch deutlich, dass die Testpersonen das Gelernte noch nicht verinnerlicht hatten.

Ein gutes Beispiel dafür ist die Aufgabe C2. Bei dieser Frage mussten die Testpersonen herausfinden, aus welchem Jahrzehnt es die meisten, bzw. die wenigsten Filme gibt. Ein kurzer Blick auf das Kreisdiagramm genügt hier um herauszufinden, dass die meisten Filme sehr neu sind. Trotzdem suchten die Testpersonen alle Jahrzehnte durch. Des weiteren konnte aufgrund der Dauer der Schulung nicht sichergestellt werden, dass wirklich alle Testpersonen über das exakt gleiche Wissen über den CircleSegmentView verfügen. Das

liegt zum einen daran, dass 15 Minuten nicht ausreichen, um selbst bei einer ausführlichen Erläuterung alle Einzelheiten aufmerksam aufzunehmen und zu verinnerlichen, zum anderen daran, dass viele Testpersonen offensichtlich nicht alles verstanden hatten, aber trotzdem behaupteten, dass keine Fragen mehr offen seien. Dieses Problem hängt eng mit der Tatsache zusammen, dass viele Testpersonen zwar bestrebt waren, die Aufgaben korrekt und schnell zu lösen, aber nicht vor Augen hatten, dass es bei der Aufgabenerledigung auf Sekunden ankommt.

Problematisch erscheint auch, dass die Testpersonen nicht davon abgehalten werden konnten, während des Tests Fragen zu stellen, diese wurden zwar meisten mit dem Hinweis, dass der Test jetzt laufe und die Aufgaben selbständig zu bearbeiten seien, auf die Zeit nach dem Test zurückgestellt.

- Der Lösungsweg

Bei der Aufgabenerledigung gab es bezüglich des Lösungswegs keine Vorgaben. Problematisch erscheint daran, dass eine Testperson, die nicht den richtigen Weg zur Aufgabenerledigung einschlägt und deshalb sehr viel Zeit braucht, nicht mit einer Person verglichen werden kann, die die Aufgabe korrekt löst. Viele Testpersonen benutzten einfach die Ergebnistabelle, um daraus die Antwort abzulesen. Bei nur 657 Datensätze führte diese Vorgehensweise auch oft zum Ergebnis, war aber zeitraubender als der richtige Lösungsweg. Bedingt durch die Anzahl der Datensätze konnte auch nicht überprüft werden, inwieweit der CircleSegmentView für die Verwendung mit großen und sehr großen Datenmengen geeignet ist.

Im Folgenden werden Lösungsansätze für die genannten Probleme vorgestellt, die in späteren Arbeiten benutzt werden können, um eine sehr genaue Prüfung der Benutzerfreundlichkeit des CircleSegmentView vornehmen zu können.



## Kapitel 6

# Zusammenfassung und Ausblick

Der CircleSegmentView wurde unter Beachtung der Richtlinien für die Gestaltung von Oberflächen und Visualisierungen entworfen. Es liegt auf der Hand, dass der CircleSegmentView bei der Suche dem Benutzer einen Zeitvorteil bringt. Diese Erwartung wurde durch den durchgeführten Test teilweise bestätigt. Um die Ergebnisse des Tests zu bestätigen und zu erweitern, sollte ein zweiter Test vorgenommen werden, der gegenüber dem durchgeführten Test folgende Verbesserungen aufweist: Die Testpersonen sollten für die Schulung und Durchführung des Tests bei entsprechender Bezahlung einen Tag zur Verfügung stehen. Um ein tiefgehendes Verständnis und eine entsprechenden Übung im Umgang mit dem CircleSegmentView seitens der Testpersonen garantieren zu können, sollte die Schulung zwei Doppelstunden in Anspruch nehmen. Dabei werden alle Testpersonen gleichzeitig in einem geeigneten Seminarraum geschult. Während dieser Zeit sollen die Testpersonen über alle wesentlichen Hintergründe für den Entwurf des CircleSegmentView aufgeklärt werden. Des weiteren werden Funktionsweise und Anwendungsbereiche detailliert anhand von geeigneten Beispielen erläutert. Nach einer Pause bekommen die Testpersonen die Aufgabe, an einem kleinen Anwendungsszenario die Funktionalität selbst unter Anleitung auszuprobieren. Das dürfte ungefähr den Zeitraum von einer Doppelstunde beanspruchen. Nach dieser Phase beginnt dann der eigentliche Test.

Alternativ dazu wäre es auch möglich, dass die Testpersonen sich zu Hause selbständig anhand von geeignetem Unterrichtsmaterial in die Benutzung des CircleSegmentView einarbeiten.

Was die Durchführung des eigentlichen Tests betrifft, werden folgende Verbesserun-

gen vorgeschlagen:

Damit die Zeit für die Aufgabenerledigung exakt und zweifelsfrei gemessen werden kann, wird ähnlich wie in [CD00] vorgeschlagen, dass die Testfragen auf dem Bildschirm eingeblendet werden. Hat die Testperson die Frage verstanden, muss eine Taste gedrückt werden. Das Tastaturkommando bewirkt, dass die Zeitmessung läuft und das Fenster mit der Frage an den Rand der Arbeitsfläche verschoben wird. Hat die Testperson die richtige Antwort gefunden, ist wieder ein Tastaturbefehl erforderlich, der bewirkt, dass die Zeitmessung gestoppt wird, und ein Eingabefeld für die Antwort eingeblendet wird. Schafft die Testperson es nicht, eine bestimmte Aufgabe zu erledigen, kann sie aufgeben. Aufgaben, die die Testpersonen nicht über den definierten richtigen Lösungsweg erledigen, werden nicht gewertet. Mit dieser Durchführung wird eine einheitliche und zweifelsfreie Zeitmessung gewährleistet. Eine korrekte Zeitmessung ist besonders wichtig, da es sich oft um geringe Zeitunterschiede handelt.

Weiterhin wird empfohlen, den CircleSegmentView auch mit einer sehr großen Datenbank zu testen.

Auch wenn die Anwendung des CircleSegmentView auf Datenbanken mit sehr vielen Datensätzen noch nicht getestet und evaluiert wurde, steht die prinzipielle Eignung des CircleSegmentView als Query Preview für die Arbeit mit Datenbanken außer Frage. Das ergibt sich aus den Erfahrungen im Umgang mit dem CircleSegmentView und den Ergebnissen der durchgeführten Benutzerstudie:

Die Testpersonen erledigten bei zwei von drei Aufgabengruppen jeweils eine Aufgabe statistisch signifikant schneller mit dem CircleSegmentView als mit dem Formular. Der CircleSegmentView eignet sich auch für den Vergleich von Datengruppen. Ein Zeitvorteil bei der Beantwortung von Fragen nach der Verteilung der Daten konnte noch nicht nachgewiesen werden.

Vor allem im Kontext von VisMeb und im Zusammenspiel mit den anderen Visualisierungen bietet die Benutzung des CircleSegmentView eine nicht unerhebliche Erleichterung bei der Suche in großen Datenmengen.

Somit bedeutet die Entwicklung des CircleSegmentView einen weiteren Schritt in Richtung Query Preview und dynamische Datenexploration.

# Literaturverzeichnis

- [AKK96] M. Ankrest, D. Keim, and H.-P. Kriegel. Circle segments': A Technique for Visually Exploring Large Multidimensional Data Sets. In *Proceedings of Visualization '96, Hot Topic Session, San Francisco, CA, 1996*, 1996.
- [AS94] C. Ahlberg and B. Shneiderman. Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays. In *CHI Conference Proceedings*, 1994.
- [AWS92] C. Ahlberg, C. Williamson, and B. Shneiderman. Dynamic Queries for Information Exploration: an Implementation and Evaluation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, volume 11, pages 619–626, 1992.
- [Bal96] H. Balzert. *Lehrbuch der Software-Technik*. Spektrum Lehrbücher der Informatik, Heidelberg, 1996.
- [Ber82] J. Bertin. *Graphische Darstellungen und die graphische Weiterverarbeitung der Information*. de Gruyter, Berlin, New York, 1982.
- [BYRN99] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, Reading, Massachusetts, 1999.
- [CD00] H. Chen and S. Dumais. Bringing Order to the Web: Automatically Categorizing Search Results. Technical report, Microsoft Research, Redmond WA, 2000.
- [Cle85] W.S. Cleveland. *The Elements of Graphing Data*. Wadsworth Advanced Book Program, Pacific Groove, 1985.

- [DHM<sup>+</sup>95] F. Daly, D.J. Hand, Jones M.C., A.D. Lunn, and K.J. McConway. *Elements of Statistics*. Prentice-Hall, London, UK, 1995.
- [DPSB97] K. Doan, C. Plaisant, B. Shneiderman, and B. Bruns. Query Previews for Networked Information Systems: A Case Study with NASA Environmental Data. *ACM SIGMOD Record*, 26:75–81, 1997.
- [Gra86] L.D. Grandis. *Theory and Use of Color*. Prentice-Hall, Englewood Cliffs, New Jersey, 1986.
- [Hol02] A. Holzinger. *Basiswissen Multimedia: technologische Grundlagen multimedialer Informationssysteme*. Vogel–Fachbuch, Würzburg, 2002.
- [Kei22] D. Keim. Information visualization and visual data mining. *Datenbank Spektrum*, 1, 20022.
- [KMRE02] P. Klein, F. Müller, H. Reiterer, and M. Eibl. Visual Information Retrieval with the SuperTable+Scatterplot. In *Proceedings of the 6th International Conference on Information Visualization (IV 02)*, pages 59–66, 2002.
- [LV03] P. Lyman and H. Varian. How much information? Technical report, University of California at Berkeley, 2003.
- [May99] D. Mayhew. *The Usability Engineering Lifecycle*. Morgan Kaufmann Publishers, Inc., San Francisco, 1999.
- [Mem02] T. Memmel. Implementation einer tabellenbasierten Visualisierung für geo-räumliche Metadaten, 2002.
- [NM94] J. Nielsen and R. Mack. *Usability Inspection Methods*. John Wiley, New York, 1994.
- [PVN<sup>+</sup>99] C. Plaisant, M. Venkatraman, K. Ngamkajornwiwat, R. Barth, B. Harberts, and W. Feng. Refining Query Preview Techniques for Data with Multivalued Attributes. In *Proceedings of the IEEE Forum on Research and Technology Advances in Digital Libraries*, pages 50–59, Los Alamitos, CA, 1999. IEEE Computer Society Press.

- 
- [RLP<sup>+</sup>03] H. Reiterer, T. Limbach, K. Peter, F. Müller, and C. Jetter. Ein visueller Metadatenbrowser für die Erkundung großer Datenmengen. *Mensch-Computer 2003*, 2003.
- [RRH00] A. Ralston, E. Reilly, and D. Hemmendinger. *Encyclopedia of Computer Science*. Nature Publishing Group, London, 2000.
- [Rub94] J. Rubin. *Handbook of Usability Testing*. Wiley, Chichester, New York, 1994.
- [Sac97] L. Sachs. *Angewandte Statistik*. Springer Verlag, Berlin, 8 edition, 1997.
- [SB90] H.S. Smallman and R.M. Boynton. Segregation of Basic Colors in an Information Display. *Journal of the Optical Society of America*, 7:1985–1994, 1990.
- [Shn94] B. Shneiderman. Dynamic Queries for Visual Information Seeking. *IEEE Software*, 11:70–77, 1994.
- [Shn98] B. Shneiderman. *Designing the User Interface*. Addison-Wesley, Reading, Massachusetts, 1998.
- [SM00] H. Schumann and W. Müller. *Visualisierung*. Springer-Verlag, Berlin, Heidelberg, 2000.
- [Tan01] E. Tanin. *Browsing Large Online Data Using Generalized Query Previews*. PhD thesis, University of Maryland, 2001.
- [TLH<sup>+</sup>00] E. Tanin, A. Lotem, I. Haddadin, B. Shneiderman, L. Slaughter, and C. Plaisant. Facilitating Data Exploration with Query Previews: A Study of User preference and performance. *Behavior and Information Technology*, 19:393–403, 2000.
- [TPS00] E. Tanin, C. Plaisant, and B. Shneiderman. Browsing Large Online Data with Query Previews. In *Proceedings of the Symposium on New Paradigms in Information Visualization and Manipulation (NPIVM) 2000*. ACM Press, 2000.

- 
- [TS01] E. Tanin and B. Shneiderman. Exploration of Large Online Data Tables Using Generalized Query Previews. Technical report, University of Maryland, 2001.
- [Tuf97] E.R. Tufte. *Visual Explanations*. Graphics Press, Cheshire, Connecticut, 1997.
- [WC97] D. Wixon and Wilson C. *The Usability Engineering Framework for Product Design and Evaluation*. In *Handbook of Human-Computer-Interaction*. Elsevier Science, Englewood Cliffs, 1997.
- [WS92] C. Williamson and B. Shneiderman. The dynamic Home Finder: Evaluating dynamic queries in a real-estate information exploration system. In *Proceedings of the ACM SIGIR 1992*, pages 295–307, 1992.

## Anhang A

# Test auf Normalverteilung

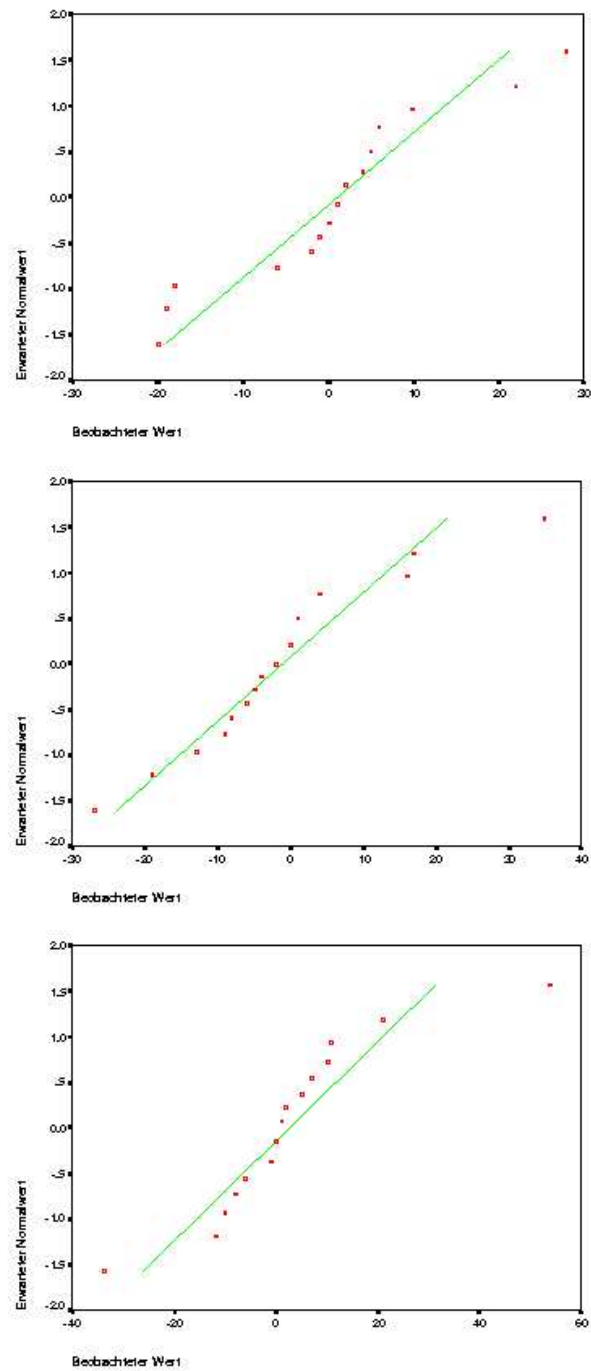


Abbildung A.1: Test auf Normalverteilung - Aufgabentyp A



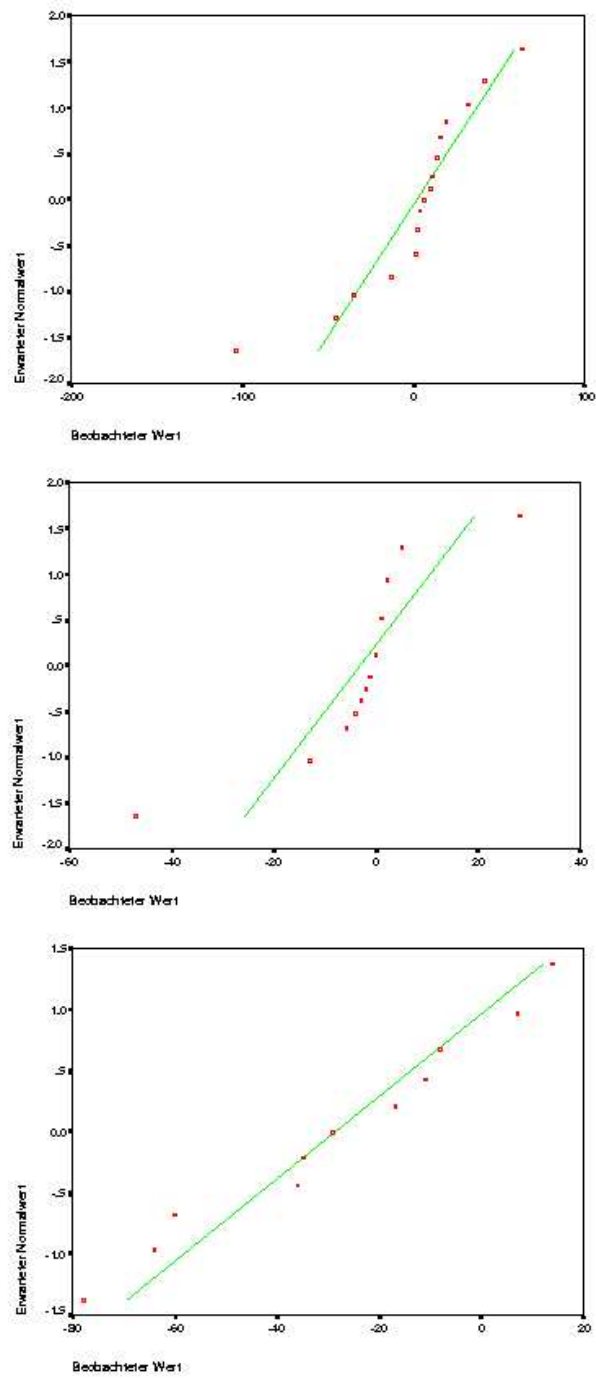


Abbildung A.2: Test auf Normalverteilung - Aufgabentyp B

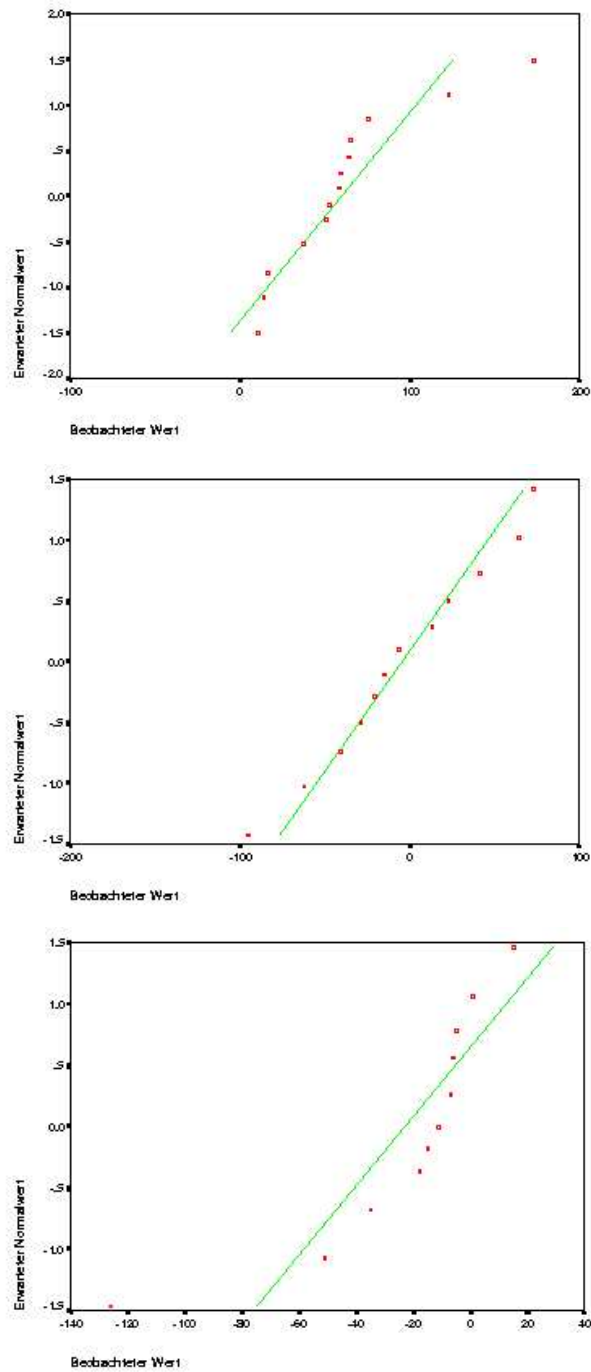


Abbildung A.3: Test auf Normalverteilung - Aufgabentyp C

## Anhang B

# Testmaterialien

### B.1 Begrüßung

#### Willkommen

Herzlich Willkommen zum Usability Test von VisMeB. VisMeB ist ein Projekt der Arbeitsgruppe Human Computer Interaction des Fachbereichs Informatik und Informationswissenschaft der Universität Konstanz. VisMeB wurde mit dem Ziel entwickelt, einen visuellen Metadatenbrowser bereitzustellen, der die Vorteile von verschiedenen Visualisierungstechniken vereinigt.

Mit diesem Test sollen verschiedene Aspekte des Systems im Hinblick auf die Leichtigkeit der Benutzung getestet werden. Hierzu werden wir Ihnen verschiedene Aufgaben stellen und Sie bitten, diese am System zu lösen.

Dazu ist es notwendig, dass wir die Zeit, die Sie für die Erfüllung der Aufgaben benötigen, messen. Es ist wichtig, dass sie sich davon nicht unter Druck gesetzt fühlen, sondern ganz normal arbeiten. Es wird nur die Software getestet und nicht Ihr Können! Falls Sie mit einer Aufgabe nicht zurecht kommen, können Sie natürlich jederzeit mit der nächsten Aufgabe fortfahren. Ebenfalls steht es Ihnen frei den Test an einer beliebigen Stelle komplett abzubrechen.

Nach dem Test werden Sie die Möglichkeit haben, Fragen zu bestimmten Aufgaben oder dem System zu stellen. Beachten Sie jedoch, dass wir Ihnen während dem Test keine Hilfestellung geben können. Falls Sie noch Fragen haben, bitten wir Sie, diese vor Beginn des eigentlichen Tests zu stellen.

Wir bedanken uns im Voraus für Ihre Teilnahme.

## B.2 Pretest Fragebogen

(wird vom Testleiter ausgefüllt)

TESTPERSON

TESTLEITER

Protokollführer

---

Herzlichen Dank, dass Sie sich bereit erklärt haben, an diesem Test teilzunehmen. Bevor wir anfangen, benötigen wir von Ihnen noch einige Angaben, insbesondere auch zu Ihrer generellen Computererfahrung und Ihrer Erfahrung mit Suchsystemen. Dies wird uns bei der Auswertung Ihres Umgangs mit dem System helfen.

### Angaben zur Person

1. Alter ...
2. Geschlecht
  - männlich
  - weiblich
3. Schulbildung
  - Gymnasium
  - Realschule
  - Hauptschule
  - Sonstiges: ...
4. Studium/ Ausbildung/ Lehre:
5. Beruf/ berufliche Stellung:

### Computererfahrung

1. Wie lange arbeiten Sie bereits mit Computern?
    - weniger als sechs Monate
    - zwischen sechs Monaten und einem Jahr
    - länger als ein Jahr
  2. Wie viele Stunden am Tag benutzen Sie einen Computer?
  3. Fällt es Ihnen schwer, sich mit neuer Software vertraut zu machen?  
(1 bedeutet: 'ja, fällt mir eher schwer', 7 bedeutet: 'nein, bereitet mir keine Probleme')
- 1     2     3     4     5     6     7

- 
4. Wie schätzen Sie Ihre Erfahrung mit Computern allgemein auf einer Skala von 1-7 ein?  
(1 bedeutet: 'keine Erfahrung', 7 bedeutet: 'sehr viel Erfahrung')
- 1     2     3     4     5     6     7
5. Welches Betriebssystem benützen Sie an Ihrem PC?
- Microsoft Windows 9x (95, 98, ME)  
 MS Windows NT/2000/XP  
 Apple Mac OS  
 Linux
- Bitte nennen Sie die verwendete Oberfläche:
6. Haben Sie Freunde mit besonderer Computererfahrung?
- Ja  
 Nein
7. Benutzen Sie im Internet Suchmaschinen ?
- Ja  
 Nein
8. Wenn ja, können Sie den Namen der von Ihnen am häufigsten genutzten Suchmaschine nennen?
9. Wie würden Sie ihre Erfahrung mit Internet Suchmaschinen auf einer Skala von 1-7 einschätzen ?
10. Wie würden Sie Ihre generelle Einstellung zu Computern auf einer Skala von 1-7 einschätzen ?  
(1 bedeutet: 'arbeite sehr ungern mit Computern' und 7 bedeutet: 'arbeite sehr gerne mit Computern')

Vielen Dank und Viel Spaß nun beim Test!

## B.3 Posttest Fragebogen

(wird vom Testleiter ausgefüllt)

TESTPERSON

TESTLEITER

Protokollführer

---

Wir möchten uns bei Ihnen noch einmal für Ihre Mithilfe bedanken. Dieser letzte Fragebogen dient dazu, uns mitzuteilen, was sie über das getestete System denken und was sie eventuell für Anmerkungen dazu haben.

Die Fragen sind in 2 Bereiche unterteilt. Der erste Teil besteht aus allgemeinen Fragen zu beiden Tests. Der zweite Teil beschäftigt sich mit dem CircleSegmentView. Bitte geben Sie Bewertungen auf einer Skala von 1 bis 7 ab. (1 bedeutet: 'nicht mit einverstanden' und 7 bedeutet 'Ja, das stimmt!')

### Allgemeine Fragen

1. War die Sprache bei allen Testfragen einfach zu verstehen?

1  2  3  4  5  6  7

2. Waren die Aufgaben klar abgegrenzt?

1  2  3  4  5  6  7

**CircleSegmentView**

1. War das generelle Layout der Oberfläche klar?  
1  2  3  4  5  6  7
2. War die verwendete Terminologie verständlich?  
1  2  3  4  5  6  7
3. War der Gebrauch von Farben angenehm?  
1  2  3  4  5  6  7
4. War die Navigation intuitiv?  
1  2  3  4  5  6  7
5. Haben Sie sich manchmal 'verloren' gefühlt?  
1  2  3  4  5  6  7
6. Konnten Sie an die benötigten Informationen Ihrer Meinung nach schnell und problemlos gelangen?  
1  2  3  4  5  6  7
7. Empfanden Sie den CircleSegmentView als nützlich?  
1  2  3  4  5  6  7
8. Könnten Sie sich vorstellen, damit zu arbeiten?  
Ja  Nein
9. Hat es Ihnen Spaß gemacht, damit zu arbeiten?  
1  2  3  4  5  6  7
10. Welche drei Begriffe würden Sie spontan damit assoziieren?  
...  
...  
...



**Formular**

1. War das generelle Layout der Oberfläche klar?  
1  2  3  4  5  6  7
2. War die verwendete Terminologie verständlich?  
1  2  3  4  5  6  7
3. War der Gebrauch von Farben angenehm?  
1  2  3  4  5  6  7
4. War die Navigation intuitiv?  
1  2  3  4  5  6  7
5. Haben Sie sich manchmal 'verloren' gefühlt?  
1  2  3  4  5  6  7
6. Konnten Sie an die benötigten Informationen Ihrer Meinung nach schnell und problemlos gelangen?  
1  2  3  4  5  6  7
7. Empfanden Sie das Formular als nützlich?  
1  2  3  4  5  6  7
8. Könnten Sie sich vorstellen, damit zu arbeiten?  
Ja  Nein
9. Hat es Ihnen Spaß gemacht, damit zu arbeiten?  
1  2  3  4  5  6  7
10. Welche drei Begriffe würden Sie spontan damit assoziieren?  
...  
...  
...

## B.4 Testfragen

### Formular

- A1. Finden Sie heraus, wieviele Komödien die Movie-Datenbank enthält!
- A2. Suchen Sie bitte zwei beliebige französische Komödien aus der Movie-Datenbank heraus und lesen Sie die Titel laut vor!
- A3. Wieviele französische Komödien gibt es in der Movie-Datenbank?
- B1. Schauen Sie in der Movie-Datenbank nach, aus welchem Jahr der Film Road Trip stammt!
- B2. Wieviele verschiedene Genres gibt es in der Movie-Datenbank?
- B3. Gibt es Action-Filme aus den Jahren 1987 und 1988?
- C1. Finden Sie heraus von welchem Genre die meisten Filme sind?
- C2. Aus welchem Jahrzehnt gibt es die wenigsten Filme?
- C3. Vergleichen Sie die Genres Komödie und Thriller.  
In welchem der beiden Genres wurden mehr Filme mit Rank 20 bewertet?

### CircleSegmentView

- A1. Finden Sie heraus, wieviele Science Fiction Filme die Movie-Datenbank enthält!
- A2. Suchen Sie bitte zwei beliebige englische Dramen aus der Movie-Datenbank heraus und lesen Sie die Titel laut vor!
- A3. Wieviele deutsche Komödien gibt es in der Movie-Datenbank?
- B1. Schauen Sie in der Movie-Datenbank nach, aus welchem Jahr der Film American Pie stammt!
- B2. In wievielen verschiedenen Sprachen gibt es in der Movie-Datenbank?
- B3. Gibt es Thriller aus den Jahren 1987 und 1988?
- C1. Finden Sie heraus in welcher Sprache die meisten Filme sind?
- C2. Aus welchem Jahrzehnt gibt es die meisten Filme?
- C3. Vergleichen Sie die Genres Science Fiction und Drama.  
In welchem der beiden Genres wurden mehr Filme mit Rank 20 bewertet?

**Anhang C**

**CD-Rom**