

Universität Konstanz
FB Informatik und Informationswissenschaft
Bachelor-Studiengang Information Engineering

Bachelorarbeit

Konzeption und Realisierung einer zoom- und fassettenbasierten Visualisierung

Zur Erlangung des akademischen Grades eines
Bachelor of Science (B.Sc.)

Studienfach: Information Engineering

Schwerpunkt: Computer Science

Themengebiet: Angewandte Informatik

von

Oliver Runge

Mat.-Nr: 01/623924

Erstgutachter: Prof. Dr. Harald Reiterer

Zweitgutachter: Prof. Dr. Marc H. Scholl

Betreuer: Mathias Heilig

Einreichung: 30.04.2010

Zusammenfassung

Der immense Informationsanstieg der heutigen Zeit führt schnell zu einer Überforderung der Anwender bei der Bewältigung der stetig anwachsenden Masse an Daten. Die allgemeine Herausforderung besteht darin, diese Datenansammlungen effizient zu verarbeiten und dem Benutzer so das Arbeiten mit den Daten zu erleichtern.

Die vorliegende Bachelorarbeit setzt sich mit dieser Thematik auseinander und führt die zoom- und facettenbasierte Visualisierung *FacetBrowsing* ein. Nach einer Einführung in die aktuellen Problematiken werden anhand bestimmter Orientierungs- und Übersichtskonzepte Ideen aufgezeigt, die den Anwender dabei unterstützen, sich in großen Datenmengen zurechtzufinden. Ebenso werden verschiedene Navigations- und Interaktionstechniken vorgestellt, die die Bewegung innerhalb des Informationsraums erleichtern. Die entwickelten Konzepte werden anschließend daraufhin untersucht, ob die im Vorfeld definierten Forschungsfragen beantwortet werden und sie somit zu einer Lösung der Problematiken beitragen.

Abstract

The enormous increase of information leads users to be quickly overwhelmed by the ever-growing mass of Data. The overall challenge is to facilitate the workflow by processing these data collections efficiently.

The present Bachelor thesis argues with this topic and implements the zoom and facet based visualization *FacetBrowsing*. After an introduction of the current problems, a range of orientation and overview concepts help to present ideas for a better user bearing within huge data volume. Furthermore several navigation and interaction techniques are presented to facilitate the movement within the information space. The developed concepts are then examined to determine whether the pre-defined research questions are answered and they thus contribute to a solution of the problem.

Inhaltsverzeichnis

Zusammenfassung	2
Inhaltsverzeichnis	3
Abbildungsverzeichnis.....	5
Tabellenverzeichnis	6
Quellcodelistings	6
Abkürzungsverzeichnis.....	7
1 Einführung	8
1.1 Problemstellung und Forschungsfragen	9
1.2 Vorgehensweise.....	10
2 Grundlagendefinition.....	12
2.1 Informationsvisualisierung.....	12
2.2 ZOIL Framework	12
2.3 MedioVis 2.0	14
2.4 Facettenklassifikation.....	15
3 Related Work	17
3.1 Orientierung in großen Datenmengen.....	17
3.1.1 Breadcrumb Navigation.....	18
3.1.1.1 Location Breadcrumbs	18
3.1.1.2 Look-Ahead Breadcrumbs	19
3.1.1.3 Path Breadcrumbs	19
3.1.1.4 Attribute Breadcrumbs	20
3.1.2 Cluster.....	21
3.2 Navigation in großen Datenmengen	23
3.2.1 Breadcrumb Navigation.....	23
3.2.2 Navigation über Kontextmenüs	24
3.2.2.1 Pie Menu	24
3.2.2.2 Marking Menu	24
3.2.2.3 Flower Menu	25
3.3 Visualisierung großer Datenmengen.....	26
3.3.1 Facettenklassifikation.....	26
4 FacetBrowsing: Konzept und Design	31
4.1 Problemstellung	31
4.2 Vorstellung des Prototypen FacetBrowsing	31
4.3 Vorstellung der Konzepte.....	34

4.3.1	Visualisierung durch Facettenklassifikation	35
4.3.1.1	Facettenklassifikation	35
4.3.1.2	Facetten	36
4.3.2	Semantische Anordnung	38
4.3.3	Navigation und „austauschbare Zoomstufen“	40
4.3.4	Orientierung innerhalb der Visualisierung	42
5	FacetBrowsing: Implementation und Umsetzung	47
5.1	Navigation	47
5.2	Austauschbare Zoomstufen	50
5.3	Clusterbildung & semantische Anordnung	54
5.3.1	LSA	54
5.3.2	SpringLayout	55
6	Fazit und Ausblick	58
6.1	Rückblick	58
6.2	Ergebnisse	58
6.3	Ausblick	61
	Glossar	64
	Literaturverzeichnis	65
	Erklärung	70

Abbildungsverzeichnis

Abbildung 1:	Aufbau der Arbeit und wesentliche Inhalte	11
Abbildung 2:	Veranschaulichung des semantischen Zoomes anhand von ZOIL.....	13
Abbildung 3:	Startzustand von MedioVis 2.0 mit nach Genre gruppierten Filmobjekten	14
Abbildung 4:	Zoom auf einen Film im Genre Action in unterschiedlichen Detailstufen.....	15
Abbildung 5:	Aufbau des Kapitels Related Work.....	17
Abbildung 6:	Unterschiedliche Wege zu einem Artikel resultieren im gleichen Bread-crumb Pfad	19
Abbildung 7:	Beispiel eines „Look-Ahead-Breadcrumb“ Pfades mit Drop-Down-Box	19
Abbildung 8:	Vertikaler <i>Path Breadcrumb Pfad</i> des Systems FacetLens mit genauer Reihenfolge der Filterschritte.	20
Abbildung 9:	Attribute Breadcrumb auf Amazon.com mit Metainformationen	21
Abbildung 10:	Google Directory Suchergebnis mit Attribute Breadcrumbs zum schnelleren Ausschluss von irrelevanten Ergebnissen	21
Abbildung 11:	Einfache Darstellung eines Pie Menus mit 8 Menüeinträgen	24
Abbildung 12:	Beispielhafte Darstellung eines Marking Menus.....	25
Abbildung 13:	Beispielhafte Abbildung eines Flower Menu.....	25
Abbildung 14:	Flamenco Interface mit Datensatz „UC Berkeley Architecture Library“	28
Abbildung 15:	Relation Browser++ Interface	28
Abbildung 16:	FacetLens Interface im Startzustand.....	30
Abbildung 17:	Beispielhafter Filtervorgang (a-d) bei FacetBrowsing bis hin zu den gefilterten Filmobjekten	32
Abbildung 18:	FacetBrowsing Startbildschirm mit facetierter Visualisierung (oben) und Head-Up-Display (unten)	33
Abbildung 19:	FacetBrowsing: Navigation in tiefere Navigationsebene (oben) und Head-Up-Display mit einer Breadcrumb Navigation (unten)	33
Abbildung 20:	Aufbau einer Facette und deren Attributwerte	37
Abbildung 21:	Formel für die lineare Skalierung der Facetten.	37
Abbildung 22:	Ablauf des semantischen Zooms bei FacetBrowsing	38
Abbildung 23:	Darstellung ähnlicher Genres auf der Informationslandschaft.....	39
Abbildung 24:	Flower Menu innerhalb von FacetBrowsing	41
Abbildung 25:	Beispiel eines „Path Breadcrumb“ Pfades bei FacetBrowsing	43
Abbildung 26:	Mittige Darstellung der aktuellen Facette („Facet: Rating“) in textueller Form.....	44
Abbildung 27:	Dendrogramm mit Abbildung der gebildeten Hauptcluster	45

Abbildung 28: Breadcrumb Pfad zur Navigation durch die Historie.....	47
Abbildung 29: Flower Menu: Einschränkung der Auswahl	49
Abbildung 30: Beispielhafter Weg einer Filterung vom ersten Schritt bis hin zum eigentlichen Filmobjekt	51
Abbildung 31: Beispielhafte Auswahl der ersten Zoomstufe des Attributwerts „Mystery“, basierend auf Listing 5	52
Abbildung 32: (oben) Ablauf einer Filterauswahl mit Berechnung der nächsten Attributwerte	53
Abbildung 33: Darstellung eines mittels SLA erstellten Netzwerks	55
Abbildung 34: Ausschnitt der Ergebnismatrix (nXn) mit allen Genres und ihren Ähnlichkeitswerten.....	56

Tabellenverzeichnis

Tabelle 1: Beispiel einer Facettenklassifikation	16
Tabelle 2: Farbliche Hervorhebung der Hauptcluster	45
Tabelle 3: Farbliche Hervorhebung der Nebengendre.....	46

Quellcodelistings

Listing 1: Methode ZoomOnTo()	48
Listing 2: Implementation des Flower Menus mittels ElementMenu	48
Listing 3: Methode TranslatePossibleSpliCriterias()	49
Listing 4: Beispiel semantische Zoomstufen ZOIL Framework	50
Listing 5: Aufbau der ersten Zoomstufe eines Attributs bei FacetBrowsing.....	53
Listing 6: Beispielhafte LinQ-Anfrage.....	53
Listing 7: Input String – Spring Layout Algorithmus	56
Listing 8: Methode 'addEdge()' zur Suche von Knoten	57

Abkürzungsverzeichnis

CC	Colon Classification
HUD	Head-up Display
IMDb	Internet Movie Database
ISO	International Organization for Standardization
LAB	Look-Ahead Breadcrumb
LSA	Latent Semantic Analysis
SLA	Spring Layout Algorithm
WPF	Windows Presentation Foundation
XAML	Extensible Application Markup Language
XML	Extensible Markup Language
ZOIL	Zoomable Object-oriented Information Landscape

1 Einführung

Die heutige Technik und der aktuelle Stand in der Informatik ermöglichen das Speichern immer größerer Datenmengen. Jeder Mensch trifft im beruflichen sowie im privaten Umfeld auf eine große und vielfältige Ansammlung von Informationen, die es zu bewältigen gilt. Zahlreiche Kollektionen von Musikdateien, Bildern oder Dokumenten aller Art sind nur wenige Beispiele.

Die allgemeine Herausforderung ist es, diese stetig anwachsenden Informationen für den Menschen so zu verarbeiten und darzustellen, dass der Einstieg und das damit verbundene Arbeiten mit den Daten erleichtert wird. Die Unterstützung des Benutzers bei seinem Such- und Browse-Verhalten spielt eine wichtige Rolle bei der Visualisierung von Informationen. Zusätzlich sind Interaktions- sowie Navigationskonzepte als Hilfestellung von großer Bedeutung. Verschiedene Ansätze in der Forschung der Mensch-Computer Interaktion und anderer Bereiche haben sich in den letzten Jahren damit auseinandergesetzt und versucht, diese Herausforderungen mit Hilfe vielfältiger Visualisierungen zu bewältigen.

Ein Ansatz, der in den letzten Jahren, vor allem auch aufgrund der stetigen Verbesserung der Computertechnologie immer häufiger zum Einsatz kommt, ist die Gruppierung von Informationsobjekten in so genannte multiple orthogonale Facetten, die eine Zuordnung von mehreren Sichten auf dasselbe Objekt ermöglichen. Dieser Ansatz wird auch als „facettierte Browsing“ bezeichnet.

Facettierte Browsen ist eine moderne Weiterentwicklung der Facettenklassifikation, die bereits in den 1930er Jahren ihren Ursprung in der Bibliothekswissenschaft hatte [Taulbee 1965]. Bis in die heutige Zeit haben mehrere Studien gezeigt, dass facettierte Browsen ein erfolgreicher Weg ist, viele Daten einer Person zugänglich zu machen [Yee et al. 2003] [Zhang et al. 2005] [Gildea et al. 2008] (vgl. Kapitel 3).

Im Rahmen dieser Arbeit wird der Prototyp *FacetBrowsing* vorgestellt, der im Verlauf des Bachelor Projektes konzipiert und umgesetzt, sowie in Verbindung mit dieser Arbeit weiterentwickelt wurde. *FacetBrowsing* ist eine zoombare und facettierte sowie browsing-orientierte Visualisierung und wird innerhalb des Projektes MedioVis 2.0 verwirklicht (detaillierter Einblick in MedioVis 2.0 vgl. Kapitel 2.3).

Der Ansatz ist hierbei eine Visualisierung zu schaffen, die es dem Benutzer ermöglicht, den aktuellen Datenraum schnell und effizient zu erkunden. Objekte liegen dabei nicht mehr direkt sichtbar auf der Informationslandschaft sondern sind in so genannten Facetten kategorisiert. Zusätzlich bietet sich dem Benutzer die Möglichkeit, über die Facetten einzelne Informationen aus mehreren Sichtweisen zu betrachten und Ziele im Datenraum auf verschiedenen Wegen zu erreichen. Dabei soll das konzipierte User Interface nicht den Eindruck einer komplizierten Anwendung erwecken, sondern dem Benutzer den Einstieg in den komplexen Datenraum erleichtern. Ebenso sollen die Übersicht und die Orientierung zu jeder Zeit gewährleistet sein.

Eine weitere Zielsetzung innerhalb dieses Projektes ist es, den Prototypen als eine ergänzende Visualisierung für MedioVis 2.0 anzubieten und so dem Benutzer eine andere, alternative Sicht auf die Daten zu gewähren.

Die Motivation der Realisierung dieser Anwendung lag vor allem in der bestehenden Problematik und den existierenden technischen sowie konzeptionellen Grenzen des zu Grunde liegenden ZOIL Frameworks und MedioVis 2.0.

Im Folgenden werden die Motivation und die dazugehörigen Problemstellungen detaillierter aufgezeigt, um einen Einblick in die Zielsetzung dieser Arbeit zu erlangen.

1.1 Problemstellung und Forschungsfragen

Das in der Arbeitsgruppe Mensch-Computer-Interaktion der Universität Konstanz entwickelte ZOIL Framework (**Z**oomable **O**bject-oriented **I**nformation **L**andscape) wird intern bereits für viele Anwendungen und Visualisierungen verwendet. Aufgrund der Entwicklung und Umsetzung von ZOIL in C# und WPF unter dem .NET 3.5 Framework, treten teilweise technische Grenzen auf, die ein flüssiges Arbeiten erschweren können (Stand Dezember 2009). Beobachtet werden kann dies beispielsweise bei dem Forschungsprototypen MedioVis 2.0, der auf dem ZOIL Framework aufbaut. Das Hauptproblem hierbei ist das .NET 3.5 Framework von Microsoft, welches sich als sehr ressourcenlastig herausstellt. Bei jeder Interaktion mit der Anwendung (zoomen und panen) muss vom .NET Framework aus neu gerendert werden. Je mehr Objekte in der Datengrundlage enthalten sind, desto mehr fällt dieses Problem ins Gewicht. Ab einer bestimmten Anzahl an Objekten veranlasst diese Problematik die Anwendung dazu, sehr langsam zu reagieren bzw. stark zu ruckeln. Ein flüssiges Interagieren ist oftmals nicht mehr möglich.

Die Hauptproblematik, sowie die Motivation für die in Kapitel 4 aufgezeigten Konzepte, liegt auf den aktuell vorhandenen konzeptionellen Grenzen, die die allgemeine Interaktion mit Anwendungen beeinträchtigen können. Eines der konzeptionellen Probleme stellt die aktuelle Anordnung bzw. Darstellung der einzelnen Filmobjekte innerhalb von MedioVis 2.0 dar. Filme werden je nach Genre auf der Informationslandschaft verortet, d.h. Objekte, die das gleiche Genre enthalten, werden in unmittelbarer Nähe zueinander platziert. Je größer die Datengrundlage ist, desto wahrscheinlicher wird es, dass beispielsweise sehr viele Action oder Drama Filme existieren. Ab einer gewissen Anzahl an Objekten lässt es sich somit, auch aufgrund der begrenzten Displaygröße, nicht mehr verhindern, dass sich einzelne Filme überlappen oder sogar vollkommen verdecken. Aus diesem Aspekt ergibt sich die Problematik, dass das Gefühl für die wirkliche Menge an Daten verloren gehen kann und somit auch die Übersichtlichkeit für den Benutzer schwindet.

Des Weiteren können bei einer zoombaren Visualisierung in bestimmten Situationen Orientierungsschwierigkeiten auftreten. Zoomt ein Benutzer beispielsweise innerhalb von MedioVis 2.0 auf ein Filmobjekt, so verschwinden andere Informationsobjekte aus seinem Sichtfeld. Dieser Vorgang führt dazu, dass sich die Anzahl an Orientierungs-

hinweisen stetig verringert. In diesem Fall muss sich der Benutzer auf seine mentale Repräsentation des Informationsraumes verlassen, um hierbei noch gezielt navigieren zu können [Gerken 2006]. Ist dies nicht möglich, kann sich der Benutzer aufgrund des Orientierungsverlustes im Datenraum verloren fühlen und kommt um ein ständiges „Zoom Out“ und „Zoom In“ nicht herum.

In diesem Zusammenhang kam die Überlegung auf, ob zusätzlich zu den im ZOIL Framework vorhandenen Navigationsmöglichkeiten, wie der Klickzoom bzw. die herkömmlichen Zoom und Pan Gesten, weitere alternative Navigationskonzepte existieren, die die Interaktion mit großen Datenmengen erleichtern und den Benutzer bei seinem Suchen und Browsen innerhalb der Filmdaten unterstützen.

Die genannten Problematiken veranlassten die Bildung folgender Forschungsfragen:

- (1) *Ist es möglich durch eine zoom- und facettenbasierende Visualisierung die vorhandenen technischen Grenzen, sowie die aktuelle Anordnung von Objekten bei großen Datenmengen zu verbessern bzw. zu lösen?*
- (2) *Welche Möglichkeiten gibt es, die Orientierung des Benutzers bei einer zoom- und facettenbasierenden Visualisierung zu gewährleisten?*
- (3) *Existieren alternative Navigationsmöglichkeiten zu der herkömmlichen Zoom und Pan Navigation?*

1.2 Vorgehensweise

Diese Arbeit ist in weitere fünf Kapitel unterteilt. Zunächst wird in Kapitel 2 ein gemeinsames Verständnis für wichtige Begriffe und Erläuterungen geschaffen, die für den Verlauf der Arbeit wichtig sind.

Um einen Einblick in die Thematik zu erhalten, werden anschließend in Kapitel 3 einige verwandte Arbeiten sowie bestehende Systeme vorgestellt, die sich speziell mit den Thematiken auseinandergesetzt haben, die als Konzepte innerhalb von *FacetBrowsing* realisiert wurden. Hierbei werden auch teilweise Unterschiede und verschiedene Forschungsansätze beschrieben sowie Entwicklungen aufgezeigt.

Kapitel 4 stellt im Anschluss den Prototypen *FacetBrowsing* vor und beschreibt den allgemeinen Aufbau sowie das User Interface. Darauf folgend werden die umgesetzten Konzepte detailliert dargestellt und anhand von Studien und weiteren wissenschaftlichen Arbeiten ergründet.

In Kapitel 5 wird der Fokus auf die technische Umsetzung der Konzepte gelegt, wobei unter anderem verschiedene Algorithmen vorgestellt werden, die bei der Realisierung der Anwendung zum Einsatz kamen.

Zum Abschluss wird ein Fazit, speziell in Bezug auf die gestellten Forschungsfragen gezogen und erläutert, in wie fern diese Fragen beantwortet wurden. Schließlich endet die Arbeit mit einem Ausblick, in dem mögliche Erweiterungen des Prototypen dargestellt werden.

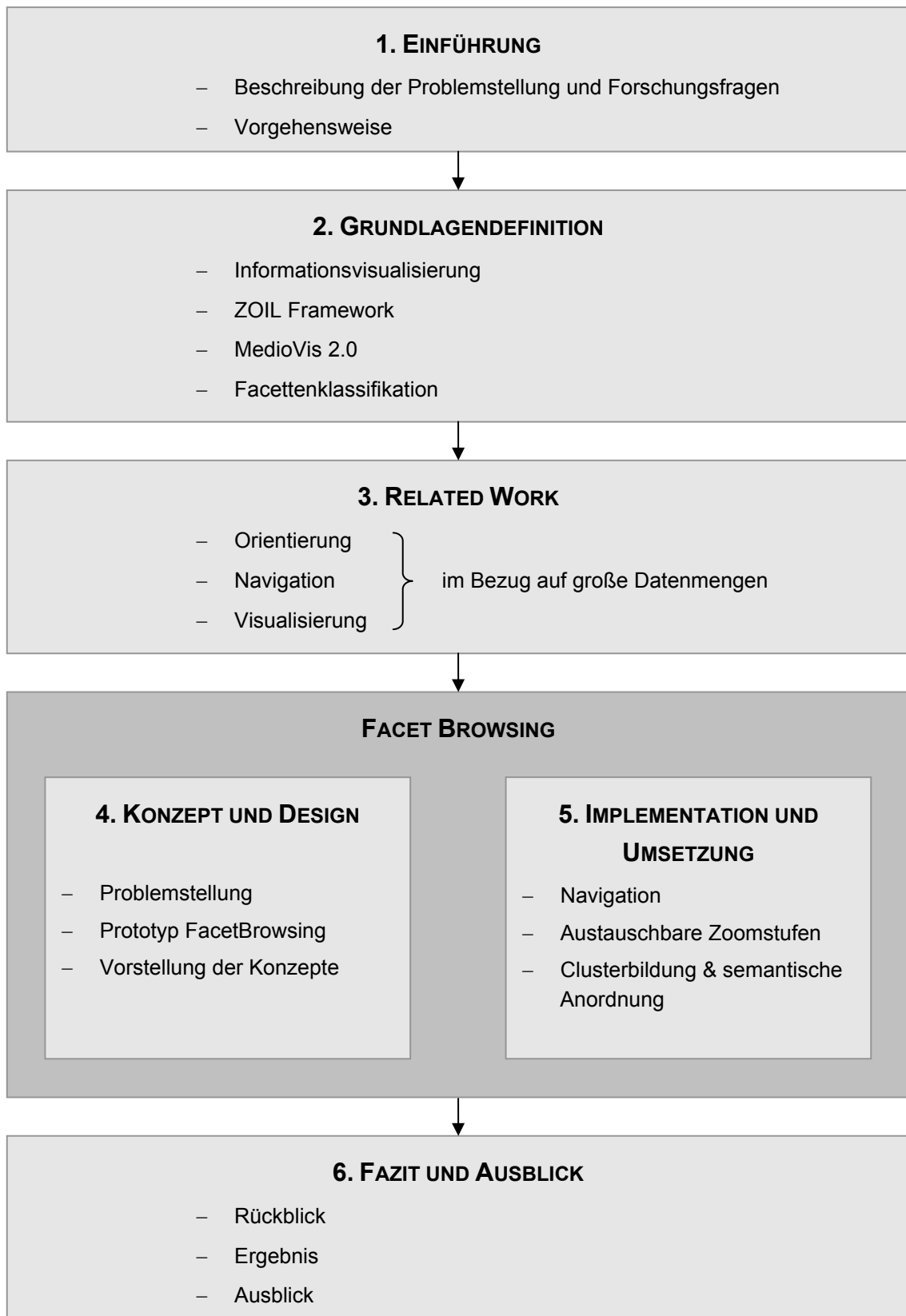


Abbildung 1: Aufbau der Arbeit und wesentliche Inhalte

2 Grundlagendefinition

Die Grundlagendefinition befasst sich mit wichtigen Begriffen bzw. Themen, die für die vorliegende Arbeit von großer Relevanz sind. Im Folgenden werden nun beschrieben:

- Informationsvisualisierung
- ZOIL Framework
- MedioVis 2.0
- Facettenklassifikation

2.1 Informationsvisualisierung

“Information visualization is the visual presentation of abstract information spaces and structures to facilitate their rapid assimilation and understanding.”

[Andrews 2009]

Durch die ständig wachsende Anzahl von Daten ist es notwendig diese auf eine effektive und anschauliche Weise graphisch zu repräsentieren und die darin enthaltenen Informationen zugänglich zu machen.

Stuart Card betont in diesem Zusammenhang, dass die Informationsvisualisierung darauf zielt, die menschliche Erkenntnis zu erweitern, um abstrakte Informationen mit Hilfe der menschlichen visuellen Fähigkeiten verständlich zu machen. Gleichzeitig sollen Hilfsmittel bereitgestellt werden, die Personen mit konstanten Wahrnehmungsfähigkeiten dabei unterstützen sollen, mit großen Mengen an Daten zurechtzukommen [Card et al. 2005].

Gegenstand der Informationsvisualisierung ist es also, die Ansammlung von Informationen graphisch so darzustellen, dass es Benutzern möglich ist, die Daten auszuwerten und aus ihnen neue Erkenntnisse zu gewinnen.

2.2 ZOIL Framework

Das ZOIL Framework ist eine Entwicklung der Arbeitsgruppe Mensch-Computer-Interaktion der Universität Konstanz und wurde im Rahmen des Projektes permaedia¹ realisiert. Das Ziel des Frameworks ist es, eine konsistente und intuitive Interaktion bzw. Navigation im Umgang mit digitalen Medien zu erreichen [Jetter 2007].

Das ZOIL Interface integriert dabei alle Arten von Informationsräumen in ein einziges verständliches Model – die Informationslandschaft. Diese ist eine zoombare, visuelle

¹ Permaedia Projekt der Universität Konstanz: <http://hci.uni-konstanz.de/permaedia>

Komponente, die ihren Inhalt in der Theorie unendlich vergrößern bzw. verkleinern (zooming) und in X- sowie Y- Richtung verschieben (panning) kann [Zöllner 2009]. Der Benutzer hat hier die Möglichkeit, auf verschiedene Funktionen und Inhalte zuzugreifen, indem animiert auf den Teil der Informationslandschaft gezoomt wird, der gerade von Belangen ist. Der Hauptaugenschein dieser Navigationstechnik fällt vorwiegend auf die menschliche Fähigkeit der visuellen und räumlichen Orientierung. Um diese Empfindung zu intensivieren, kommen sinusartig beschleunigte Animationen zum Einsatz, die den Bewegungen in der realen Welt nachempfunden sind [Reiterer et al. 2010].

Auf der Informationslandschaft werden alle Objekte und Inhalte der aktuellen Datenmenge verortet. Die Besonderheit hier ist die direkte Manipulation am visuellen Objekt, ohne auf andere Anwendungen oder Geräte zurückgreifen zu müssen.

Eine weitere Besonderheit des ZOIL Framework ist das Konzept des semantischen Zooms. Je mehr der Benutzer auf ein für ihn interessantes Objekt zoomt, desto mehr Platz steht auf diesem zur Verfügung. Dieser Platzgewinn wird nicht zwingend für eine größere Darstellung des Objektes genutzt, sondern um mehr Funktionalitäten sowie Informationen preiszugeben [Zöllner 2009].

König 2006 charakterisiert den Nutzen eines semantischen Zoomings daher wie folgt: "Im Optimalfall steht dem Anwender mittels semantischem Zooming bei jeder Skalierung eine gut wahrnehmbare und sinntragende Objektdarstellung zur Verfügung" (vgl. Abbildung 2) [König 2006].

Das ZOIL Framework wurde in C# und WPF unter dem .NET 3.5 Framework² entwickelt und umgesetzt. Die Navigation innerhalb des Frameworks ist auf eine „zooming“ und „panning“ Interaktion begrenzt, welche die menschliche Fähigkeit der visuellen räumlichen Orientierung und die Erinnerung an visuelle Landmarken ausschöpft [Gerken 2006].

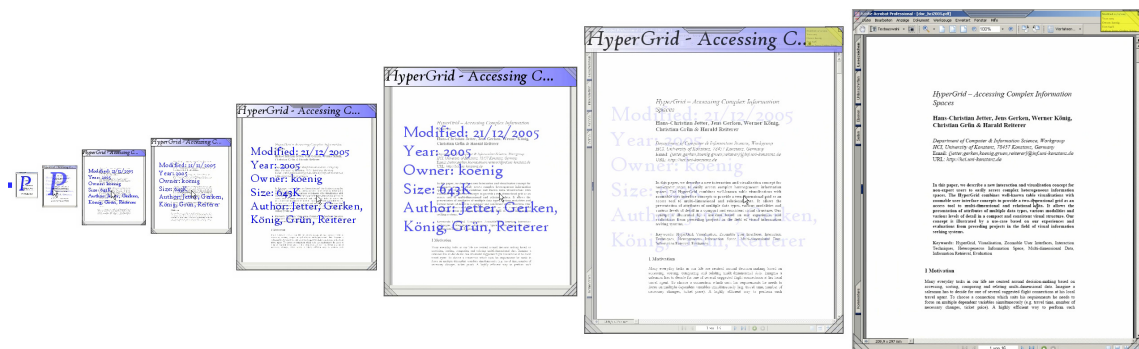


Abbildung 2: Veranschaulichung des semantischen Zoomes anhand von ZOIL [Gerken 2006]

² .NET Framework: Softwareplattform mit umfangreicher Sammlung an Klassenbibliotheken und Programmierschnittstellen der Firma Microsoft

2.3 MedioVis 2.0

MedioVis ist ein visuelles Informationssuchsystem mit dem Ziel, den Benutzer bei der natürlichen Suche, vor allem in komplexen Informationsräumen im Kontext der digitalen Bibliothek, zu unterstützen. Um dieses Ziel zu verwirklichen, werden verschiedene flexibel zusammenstellbare Visualisierungen angeboten, um Einblicke aus verschiedenen Perspektiven auf den Informationsraum zu erhalten. Gleichzeitig baut MedioVis auf ein einfaches und konsistentes Interaktionsprinzip [Heilig et al. 2008].

Der weiterentwickelte Forschungsprototyp MedioVis 2.0 versucht, eine umfassende Unterstützung aller Aktivitäten im Arbeitsbereich der digitalen Bibliothek anzubieten. Primär basiert MedioVis 2.0 auf dem ZOIL Paradigma (vgl. Kapitel 2.2) und setzt dementsprechend auf das fundamentale Interaktionsmodell einer zoombaren und objektorientierten Informationslandschaft [Heilig et al. 2009]. Die dabei verwendete Datengrundlage besteht hauptsächlich aus Filmdaten der Internet Movie Database (IMDb), welche anhand ihres Genres auf der Informationslandschaft angeordnet sind (vgl. Abbildung 3).

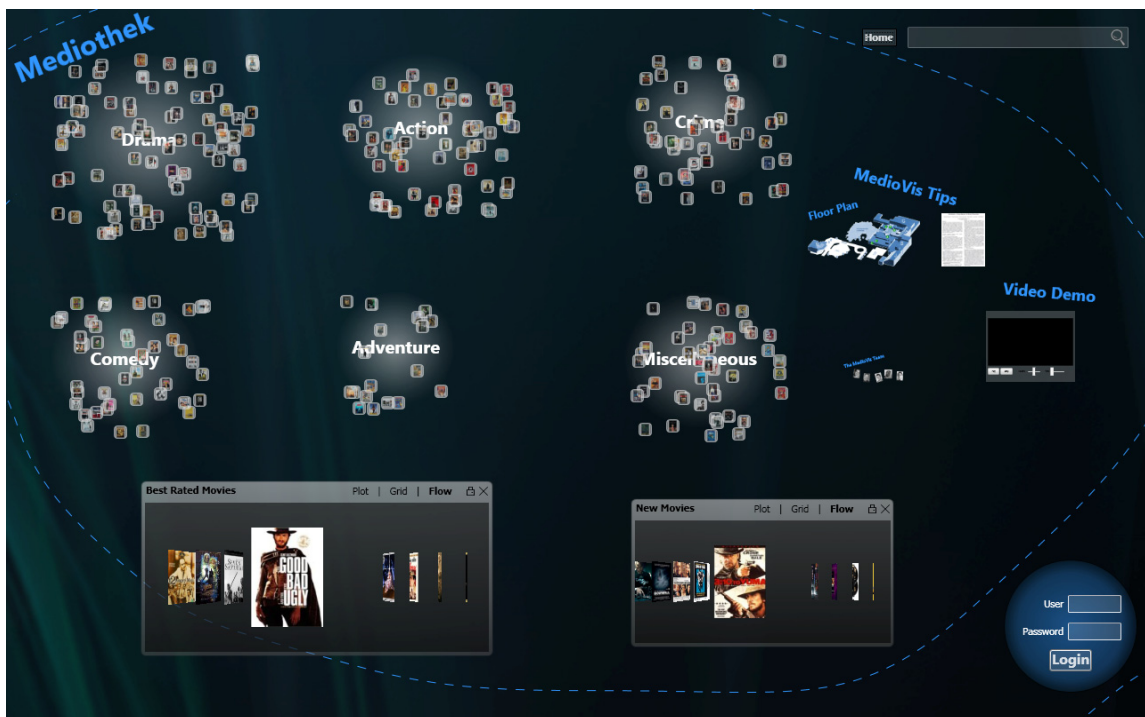


Abbildung 3: Startzustand von MedioVis 2.0 mit nach Genre gruppierten Filmobjekten [MedioVis 2010]

Dem Benutzer eröffnet sich so die Möglichkeit, durch animierte Zoom- und Pan-Operationen zu den verschiedenen Filmen bzw. Genres zu navigieren. Anhand des Prinzips eines semantischen Zooms ergibt sich die Möglichkeit, zu jedem einzelnen Filmobjekt, abhängig von der eigener Größe, mehrere Informationen und Funktionen abzurufen (vgl. Abbildung 4). Solche ergänzenden Informationen können beispielsweise die Schauspieler, die Handlung oder sogar der Trailer des Filmes sein.

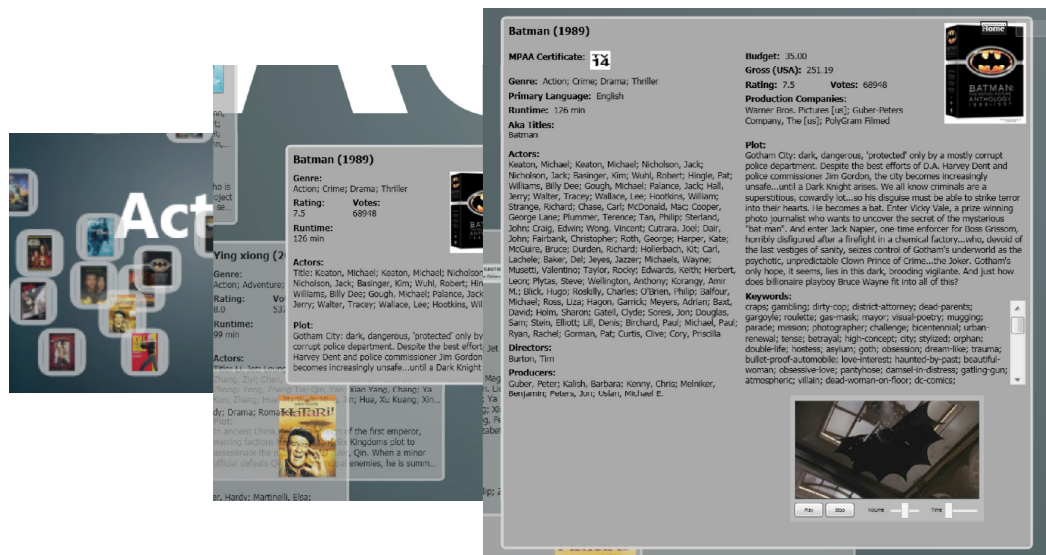


Abbildung 4: Zoom auf einen Film im Genre Action in unterschiedlichen Detailstufen [MedioVis 2010]

2.4 Facettenklassifikation

Ein facettenklassifiziertes System erlaubt die Zuordnung von mehreren Klassifizierungen bzw. Sichten auf ein Objekt. Dorloff beschreibt den Grundgedanken einer Facettenklassifikation als die Möglichkeit, einen Sachverhalt aus mehreren Perspektiven untersuchen zu können [Dorloff]. Diese Form der Klassifizierung zeichnet sich dadurch aus, dass einzelne Attribute eines Objektes in multiple orthogonale Kategorien gruppiert werden [Lee et al. 2009]. Diese Kategorien werden auch als Facetten bezeichnet. Der allgemeine Unterschied zu anderen Klassifizierungssystemen liegt darin, dass eine Facettenklassifikation versucht, aus ausgewählten Dokumenten Einfachklassen zu extrahieren und zu Facetten zusammen zu fassen. Eine Facette selber definierte Shiyali Ranganathan bereits 1967 als Oberbegriff, der die Bestandteile (Inhalte) sowie deren gewichteten Formen, Begriffe und Zahlen eines zusammengesetzten Fachgebiets kennzeichnet [La Barre 2006].

Tabelle 1 veranschaulicht die Facettenklassifikation an einem Beispiel einer Klassifikation von verschiedenen Film-Metadaten. Dabei sind in der ersten Spalte die Facetten angegeben. Eine Facette wäre hier beispielsweise der „Drehort“, der die verschiedenen Merkmalsausprägungen bzw. Attributwerte „USA“, „Deutschland“ und „Frankreich“ annehmen kann. Die möglichen Kombinationen dieser Merkmalsausprägungen ergeben die Klassen einer Facettenklassifikation. Eine mögliche Klasse wäre zum Beispiel alle „Actionfilme“, die in „Frankreich“ gedreht wurden und das Rating „8“ besitzen. Die Einordnung eines bestimmten Filmes in eine Klasse erfolgt anhand der bei diesem konkreten Film vorliegenden Kombinationen von Attributwerten. Dies bedeutet, dass der Film über verschiedene Facetten und Kombinationen gefunden werden kann und somit ein Objekt immer mehreren Facetten und somit auch Merkmalsausprägungen zugeordnet ist. Der Benutzer hat auf diese Weise die Möglichkeit, die Informationsmenge aus verschiedenen Perspektiven zu erkunden und somit neue Erkenntnisse

über die Daten zu generieren. Die Facettenklassifikation erleichtert es daher, sich bei einer großen Anzahl von Daten einen Überblick über die vielen Informationen zu verschaffen [Schaier 2007].

Facettenklassifikation einzelner Metainformationen von Filmen

Genre	Action	Drama	Thriller
Drehort(e)	USA	Deutschland	Frankreich
Rating	7	8	9

Tabelle 1: Beispiel einer Facettenklassifikation

3 Related Work

Dieses Kapitel stellt bereits bestehende Systeme und verwandte Arbeiten vor und beschreibt Entwicklungen innerhalb von verschiedenen Forschungsansätzen. In der Hinsicht auf die in Kapitel 1.1 gestellten Forschungsfragen ist die Unterteilung dieses Kapitels festgelegt auf die für diese Arbeit wichtigen Begriffe der *Orientierung* und *Navigati-on* in großen Datenmengen, sowie die *Visualisierung* vieler Objekte. Innerhalb dieser Unterkapitel wird der Fokus auf die Systeme und Arbeiten gelegt, die sich speziell mit den Thematiken auseinandergesetzt haben, die als Konzepte innerhalb des Prototypen *FacetBrowsing* realisiert wurden.

RELATED WORK	ORIENTIERUNG IN GROßEN DATENMENGEN Breadcrumb Navigation Location Look-Ahead Path Attribute Breadcrumbs Cluster
	NAVIGATION IN GROßEN DATENMENGEN Breadcrumb Navigation Kontextmenüs Pie Marking Flower Menu
	VISUALISIERUNG GROßER DATENMENGEN Facettenklassifikation

Abbildung 5: Aufbau des Kapitels Related Work

3.1 Orientierung in großen Datenmengen

Die Orientierung spielt im Zusammenhang mit Visualisierungen eine entscheidende Rolle. Es darf nicht vernachlässigt werden, dem Benutzer auf eine gewisse Weise zu veranschaulichen, wo er sich gerade im Datenraum befindet bzw. welchen Weg er eingeschlagen hat, um die aktuelle Informationsmenge nach seinen Wünschen einzuschränken. Fehlen Orientierungspunkte, kann sich ein Benutzer schnell verloren fühlen, da die eigene Position nicht mehr bestimmt werden kann [Gerken 2006].

Im Folgenden werden verwandte Arbeiten sowie Entwicklungen und Unterschiede, innerhalb eines Forschungsansatzes, aufgezeigt.

3.1.1 Breadcrumb Navigation

Eine Breadcrumb Navigation, zu Deutsch Brotkrümelnavigation, ist im Allgemeinen eine Auflistung von Elementen (meistens eine textuelle Repräsentation), die dem Benutzer einer Webseite oder einer Anwendung den bereits zurückgelegten Pfad innerhalb der Informationsmenge anzeigt. Die Brotkrümel Metapher geht auf die Geschichte von Hänsel und Gretel zurück, die auf ihrem Weg durch den Wald Brotkrumen zur Orientierung hinterließen, um bei der späteren Rückkehr den Weg aus dem Wald wiederzufinden [Hudson 2004]. In der heutigen Zeit wird die Breadcrumb Navigation vor allem im World Wide Web verwendet, um Benutzern die Orientierung innerhalb der teilweise verzweigten Webseitenstrukturen zu erleichtern. Auch außerhalb des Internets kann diese Art der Orientierungshilfe immer häufiger bei Anwendungen mit großer Datengrundlage beobachtet werden.

Bei der Darstellung und Umsetzung solcher Breadcrumb Pfade existieren allerdings Unterschiede, die auf dem Aufbau und der allgemeinen Zusammensetzung bzw. Funktionalität des Pfades beruhen. Es existieren vier verschiedene Arten einer Breadcrumb Navigation – „Location“, „Look-Ahead“, „Path“ und „Attribute Breadcrumbs“. Im Folgenden werden diese Unterschiede erläutert und anhand von Webseiten bzw. existierenden Systemen verdeutlicht.

3.1.1.1 Location Breadcrumbs

Location Breadcrumbs, oder auch ortsbezogene Breadcrumbs, sind heutzutage die am weitesten verbreitete Darstellungsvariante von Breadcrumbs innerhalb einer Webseite. Sie vermitteln dem Benutzer die aktuelle Position innerhalb der Seiten-Hierarchie, d.h. es können mehrere Wege zum gewünschten Ziel existieren, jedoch bleibt der Breadcrumb Pfad immer derselbe (vgl. Abbildung 6). Der Pfad ist demnach vergleichbar mit einem hierarchischen Baum, der von jedem Knoten, egal wie dieser erreicht wurde, immer den direkten Weg zur Wurzel aufzeigt. Bei den meist statischen Auftritten im Internet scheint dies die sinnvollste Lösung zu sein und wird deshalb immer wieder verwendet [Instone 2002]. Typische Beispiele für Location Breadcrumbs sind aktuelle Webseiten, wie „Spiegel Online“³.

³ Spiegel Online: <http://www.spiegel.de>

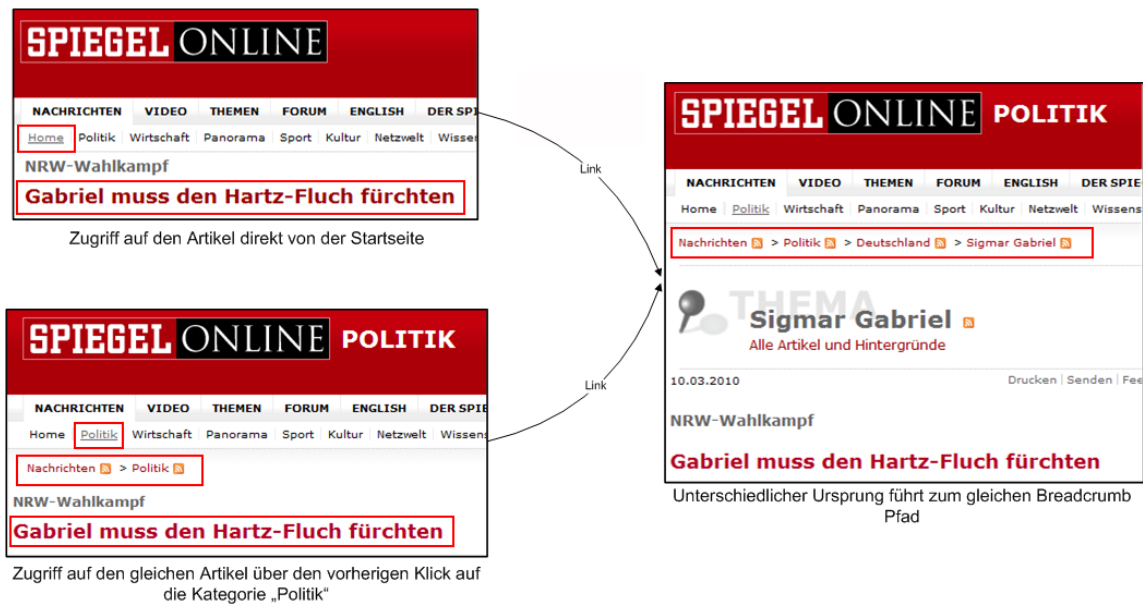


Abbildung 6: Unterschiedliche Wege zu einem Artikel resultieren im gleichen Breadcrumb Pfad
[Spiegel 2010]

3.1.1.2 Look-Ahead Breadcrumbs

Eine Weiterentwicklung des bisherigen Breadcrumb Pfades, speziell der Location Breadcrumbs, sind die von Teng vorgestellten „Look-Ahead-Breadcrumbs“ (LAB) [Teng 2003]. Der Unterschied besteht darin, dass für den Benutzer zusätzliche Interaktionsmöglichkeiten mit dem Breadcrumb Pfad bestehen. Für jeden Abschnitt des Pfades, außer dem aktuellen Standpunkt, bieten die LAB eine Art „Drop-Down-Liste“ an, die von dem jeweiligen Punkt weiterführende Links anzeigt und so dem Benutzer ein Zurückspringen in der Hierarchie ersparen [Vora 2009]. (vgl. Abbildung 7)

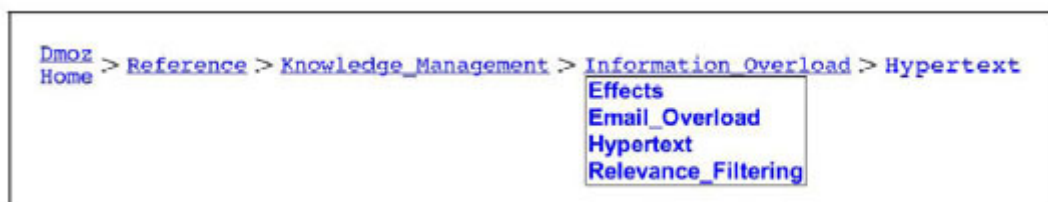


Abbildung 7: Beispiel eines „Look-Ahead-Breadcrumb“ Pfades mit Drop-Down-Box
[Blustein et al. 2005]

3.1.1.3 Path Breadcrumbs

Path Breadcrumbs schlagen einen anderen Weg der Darstellung ein. Der Breadcrumb Pfad ist hier nicht mehr statisch sondern dynamisch [Vora 2009] und repräsentiert die

eigentliche Metapher des Terms⁴. Innerhalb der Informationsmenge ist es, wie bei den Location Breadcrumbs (siehe Kapitel 3.1.1.1), ebenfalls möglich, das gewünschte Ziel auf mehrere Arten zu erreichen. Der Unterschied liegt allerdings darin, dass sich der Breadcrumb Pfad an den individuell zurückgelegten Weg des Benutzers, den er durch die Datenmenge geht, anpasst und sich somit verschiedene Wege zu dem gleichen Ziel unterscheiden. Path Breadcrumbs zeigen dem Benutzer demnach zusätzlich zu der aktuellen Position genau an, „wie er hierhergekommen ist“ [Instone 2002].

Abbildung 8 zeigt ein Beispiel eines Path Breadcrumb Pfades anhand des facettierten Systems FacetLens [Lee et al. 2009]. Die zurückgelegte Historie wird über der Hauptvisualisierung präsentiert und erleichtert so dem Benutzer das Zurechtfinden innerhalb der Anwendung. Es kann genau nachvollzogen werden, welcher Weg durch die Informationsmenge genommen wurde und welche Filterschritte bereits ausgeführt wurden.

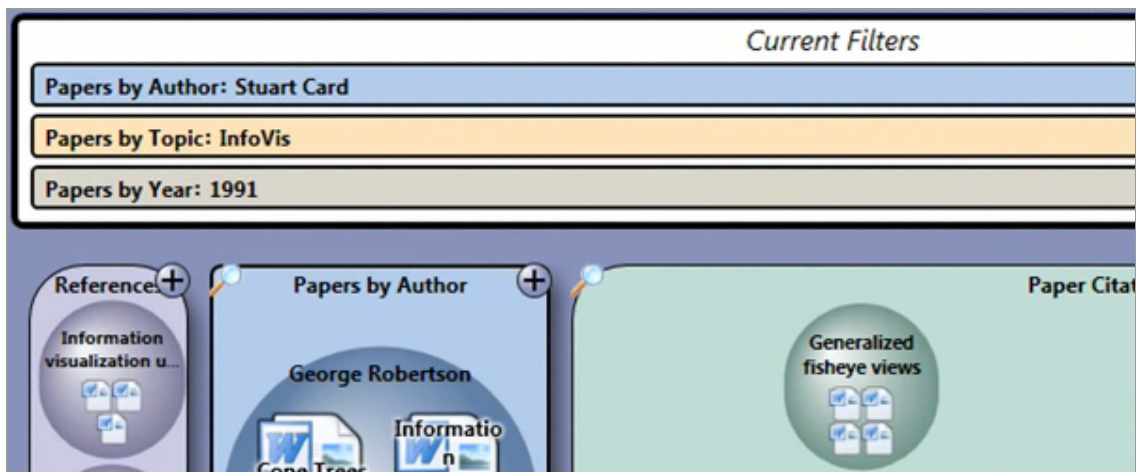


Abbildung 8: Vertikaler *Path Breadcrumb Pfad* des Systems FacetLens mit genauer Reihenfolge der Filterschritte.

3.1.1.4 Attribute Breadcrumbs

Diese Art des Breadcrumb Pfades wird vor allem von großen Webseiten, wie beispielsweise Amazon⁵ verwendet, bei denen die gleiche Seite über viele unterschiedliche Wege erreicht werden kann. Hier werden Breadcrumb Verknüpfungen angezeigt, die unter anderem Metainformationen über das aktuelle Objekt auf der Seite beschreiben. Ebenfalls wird dabei verdeutlicht, in welchen anderen Kategorien sich das Objekt befindet [Vora 2009]. Dabei kann ein Attribute Breadcrumb sowohl ein „Path-“ als auch ein „Location Breadcrumb“ sein.

⁴ Metapher des Terms entspricht der Hänsel und Gretel Geschichte (vgl. [Hudson 2004])

⁵ Amazon: <http://www.amazon.com>



Abbildung 9: Attribute Breadcrumb auf Amazon.com mit Metainformationen [Amazon 2010]

Ein Anwendungsbeispiel für Attribute Breadcrumbs liefert Amazon.com. Unterhalb von gefundenen Artikeln bietet Amazon eine Liste weiterer Breadcrumb Pfade an, aus denen sich erkennen lässt, in welchen zusätzlichen Kategorien der Artikel verlinkt ist.

Des Weiteren werden Teile der Metadaten preisgegeben (vgl. Abbildung 9). Ein weiteres Anwendungsbeispiel liefert „Google Directory“, bei dem der Breadcrumb Pfad bereits in den Suchergebnissen vermerkt ist. Hier können die ergänzenden Metainformationen in den Attribute Breadcrumbs helfen, im Vorfeld bereits die richtige und die vom Benutzer erwünschte Auswahl zu treffen. So können, wie Abbildung 10 erkennen lässt, schon frühzeitig falsche Ergebnisse ausgeschlossen werden. [Google Directory 2010]

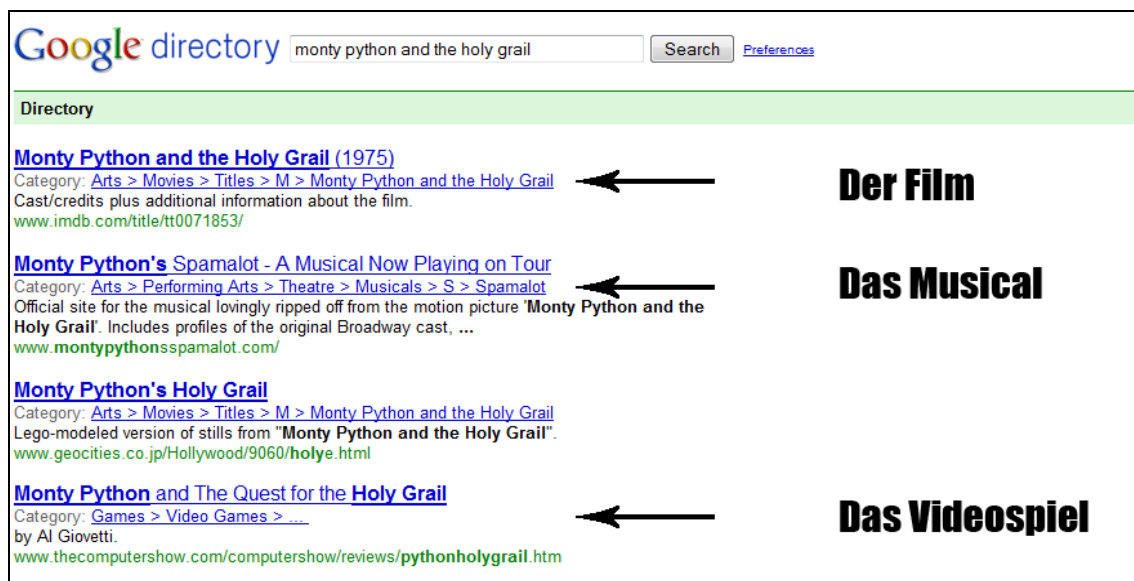


Abbildung 10: Google Directory Suchergebnis mit Attribute Breadcrumbs zum schnellen Ausschluss von irrelevanten Ergebnissen [Google Directory 2010]

3.1.2 Cluster

Im Allgemeinen kann „Clustering“ als eine Methode der Kategorisierung bezeichnet werden, die ähnliche Dokumente und Objekte zusammenfasst. Dabei werden diese Objekte in Gruppen aufgeteilt, so dass die Ähnlichkeit innerhalb der Gruppen sehr hoch ist und dementsprechend niedrig zu anderen Gruppierungen [Käki 2005] [Yiu et al. 2004]. Die Verdeutlichung von dominierenden Thematiken in Clustern, verhilft dem

Benutzer zu einer besseren Übersicht und damit besseren Orientierung innerhalb großer Datenmengen [Hearst 2006].

Dumais et al. (1988) zählen in der Mensch-Computer Interaktion zu den Ersten, die die Technik der Kategorisierung, speziell das Clustering, nahegelegt haben, um den Zugriff auf Informationen zu beschleunigen [Käki 2005].

Eines der ersten Systeme, die den Ansatz des Clusterings umgesetzt und getestet haben, ist Scatter/Gather von Cutting et al. im Jahre 1992 [Cutting et al. 1992]. Die dabei zugrunde liegende Informationsmenge wird vom System in kleinere Gruppen bzw. Cluster aufgeteilt („Scatter“) und zeigt dem Benutzer lediglich eine kleine Zusammenfassung der Daten innerhalb des Clusters an. Der Benutzer kann in weiteren Schritten ein oder auch mehrere gewünschte Cluster auswählen, in Folge dessen diese Informationen vom System gesammelt („Gather“) werden und eine neue Sub-Kollektion bilden. Das System offenbarte allerdings noch einige Probleme in der Benennung bzw. der Zusammenfassung der automatisch erzeugten Cluster. Diese Zusammenfassungen stellten sich häufig als zu lang und unlesbar heraus, da zu einer Analyse der gesamte Text eines Dokumentes verwendet wurde. Aus diesem Grund fiel es Benutzern schwer, relevante Cluster zu identifizieren. Außerdem erweckte Scatter/Gather den Eindruck, langsam und wenig präzise in den Aufgabengebieten der Informationsbeschaffung zu sein [Käki 2005]. Andererseits demonstrierte Scatter/Gather aber das hohe Potenzial, welches der ursprüngliche Ansatz der Kategorisierung von Dumais et al. aufzeigt [Käki 2005].

Im späteren Verlauf der Forschung entwarfen Zamir und Etzioni 1998 ebenfalls eine Clustering Technik. Diese diente speziell der Verwendung in Webumgebungen. Basierend auf dieser Technik entwickelten Zamir und Etzioni 1999 das System Grouper, welches als Interface für Suchmaschinen wie beispielsweise die Meta Suchmaschine HuskySearch diente und dynamisch die gelieferten Suchergebnisse in Clustern gruppierte [Zamir et al. 1999].

Der Vorteil zu Scatter/Gather lag darin, dass hier anstelle des gesamten Dokumentes nur kleine Ausschnitte als Input genommen wurden. So musste zur Analyse nicht das gesamte Dokument heruntergeladen werden. Somit war der Algorithmus schnell genug für die Online Analyse innerhalb eines Web Interfaces [Zeng et al. 2004]. Gleichzeitig sollten die generierten Cluster anhand der kleinen Textschnipsel nun lesbarer sein, um schnelles Browsen zu ermöglichen.

Zamir und Etzioni testeten ihre Entwicklung anhand einer Zusammenstellung von Dokumentenclustern im Web mit Hilfe von Benutzern, die über das System Suchanfragen starteten. Grouper löste die Problematik, die Scatter/Gather aufzeigte, besser, allerdings zeigte die Testphase weitere Probleme auf. Grouper erstellt die Cluster, indem es Basiscluster zusammenführt. Dies kann, laut Zamir und Etzioni, oft nützlich sein, allerdings auch den Benutzer verwirren, besonders wenn die entstandenen Cluster nicht die semantischen Unterscheidungen aufzeigen, die der Benutzer eigentlich erwartet [Zamir et al. 1999].

Eine Weiterentwicklung der von Zamir und Etzioni entwickelten Clustering Technik präsentierte Zeng et al. [2004]. Sie identifizierten das Hauptproblem, welches bei dem Clustern von Suchergebnissen auftritt, als ein hervorstechendes Problem des Rankings einzelner Phrasen. Zeng et al. integrierten zusätzlich eine Technik, die ständig von Trainingsdaten lernt und so die aufgezeigte Problematik lösen sollte. Experimentelle Ergebnisse bestätigten diese Vermutung [Zeng et al. 2004].

Zeng zeigte, dass korrekte Cluster generiert werden konnten, die folglich das Browsen durch Suchergebnisse effektiver und schneller gestalteten.

Der größte Vorteil des Clusterings heutzutage ist die Tatsache, dass es vollständig automatisiert abläuft und sehr einfach auf jegliche Kollektionen angewandt werden kann [Hearst 2006]. Des Weiteren besteht beim Clustering die Möglichkeit, potenziell Unerwartetes bzw. neue Tendenzen in großen Datenmengen zu entdecken. Allerdings besteht laut Hearst (2006) nach wie vor der Nachteil des Mangels an Vorhersehbarkeit bzw. Berechenbarkeit und die Benennung der einzelnen Gruppen.

3.2 Navigation in großen Datenmengen

Gerade in der heutigen Zeit, in der immer mehr Daten und Informationen vorhanden sind, ist es unumgänglich eine zuverlässige Navigation innerhalb einer Visualisierung anzubieten und somit den Benutzer optimal bei seinem Arbeiten zu unterstützen.

Um in diesen Datenmengen navigieren zu können, bieten sich die folgenden Methoden an:

3.2.1 Breadcrumb Navigation

Die bereits weiter oben in diesem Kapitel angesprochenen Breadcrumbs (vgl. Kapitel 3.1.1) dienen nicht nur der Orientierung, sondern unterstützen ebenso die Navigation innerhalb des Datenraumes. Breadcrumbs bieten in der Regel Links als Verknüpfung an, um sehr schnell und einfach in der Historie zu vorherigen Kategorien zurück springen zu können, ohne dabei beispielsweise eine neue Suchanfrage zu starten oder sich erneut vom Startpunkt durch alle Kategorien klicken zu müssen [Hudson 2004]. Studien [Bowler 2001] [Maldonado, Resnick 2002] zeigen, dass eine Breadcrumb Navigation bei Benutzern eine effizientere und befriedigendere Auswirkung im Bezug auf die Navigation innerhalb einer Datenmenge hat. Einen Schritt weiter gehen die bereits vorgestellten „Look-Ahead-Breadcrumbs“ (LAB). Der Benutzer hat durch die zusätzliche Drop-Down-Liste den Vorteil, noch schneller in der Breadcrumb Historie an einen gewünschten Punkt zu gelangen. Die LAB bieten an dem jeweiligen Breadcrumb Element Links an, die von diesem Punkt erreichbar sind (vgl. Abbildung 7). Ein Zurückspringen ist somit überflüssig.

3.2.2 Navigation über Kontextmenüs

3.2.2.1 Pie Menu

Bei einem Pie Menu handelt es sich um ein Menü, welches kreisförmig um einen bestimmten Referenzpunkt angeordnet wird und dem Benutzer verschiedene Optionen anbietet, die ausgeführt werden können (vgl. Abbildung 11).

Bereits 1988 stellten Hopkins et. al. ihren Ansatz eines *Pie Menus* vor und belegten anhand einer Studie mit 16 Probanden, dass Benutzer mit Hilfe eines *Pie Menus* bis zu 15% schneller arbeiten können und die Fehlerrate wesentlich geringer ausfällt als mit einem herkömmlichen linearen Menü [Hopkins et al. 1988].

Laut den Autoren bringt die Benutzung von *Pie Menus* auch Nachteile mit sich. Zum einen verbraucht ein solches Menü, je nachdem welche Länge die Beschriftungen der einzelnen „Kuchenstücke“ aufweisen, sehr viel Platz auf dem Bildschirm. Zum anderen besteht ein Pie Menu immer aus einer limitierten Anzahl an Einträgen [Hopkins 1991]. Eine zu große Anzahl an Optionen würde das Arbeiten mit Hilfe eines *Pie Menus* eher erschweren, da die einzelnen „Kuchenstücke“ zu klein ausfallen würden.

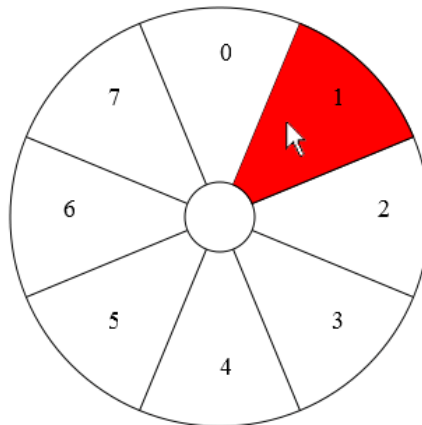


Abbildung 11: Einfache Darstellung eines Pie Menüs mit 8 Menüeinträgen
[Mozilla 2010]

3.2.2.2 Marking Menu

Eine Erweiterung des *Pie Menus* stellen die sogenannten *Marking Menus* [Kurtenbach 1993] dar. Sie basieren auf dem gleichen Prinzip, weisen allerdings zusätzliche Funktionalitäten auf. Das charakteristischste Feature eines *Marking Menus* ist die ergänzende Funktion der skaleninvarianten gestischen Auswahl eines Menüpunktes, in dem der Benutzer einen Pfad in Richtung des Ziels zeichnet, ohne dabei das Menü öffnen zu müssen. Anhand solcher Gesten ist es ebenso möglich, „zick-zack“ Bewegungen auszuführen, um Ziele in tieferen Ebenen zu erreichen (vgl. Abbildung 12). Mehrere Studien [Kurtenbach 1993] [Kurtenbach 1994] haben gezeigt, dass *Marking Menus* signifikante Vorteile gegenüber regulären *Pie Menus* und linearen Menüs aufweisen. Kurtenbach et al. erläutern in diesem Zusammenhang, dass *Marking Menus* die Zeit zur Auswahl eines Menüpunktes dramatisch für erfahrene Benutzer verkürzen und

gleichzeitig Anfänger dabei unterstützen, sich schneller in die Materie einzufügen [Kurtenbach et al. 1995]. Laut Zhao et al. liegt dieser Zuwachs an Geschwindigkeit bei bis zu 3,5x [Zhao et al. 2004].

Allerdings zeigten Kurtenbach et al. auch, dass je mehr Level (Menü Tiefe) und je mehr Elemente pro Level (Menü Breite) vorhanden sind, die Fehlerrate umso höher ausfällt – auch bei Experten [Zhao et al. 2004].



Abbildung 12: Beispielhafte Darstellung eines Marking Menus

(a) Menüeintrag über mehrere Stationen erreicht. (b) Gleicher Menüeintrag anhand einer Geste ohne Menüansicht [Betriebsraum 2009]

3.2.2.3 Flower Menu

Ein neuartiger Ansatz eines *Marking Menus* stellt das von Bailly et al. [2008] vorgestellte *Flower Menu* dar. Den Autoren zufolge wurde dieser Ansatz verfolgt, um eine hohe Anzahl (>1000) an Elementen innerhalb dieses Menüs anzeigen zu können und somit die Nachteile eines herkömmlichen *Marking Menus* oder auch *Pie Menus* zu umgehen. Um dies zu gewährleisten, bietet das *Flower Menu* sieben verschiedene kurvenförmige Gesten für jede der vorhandenen acht Richtungen an. So ist es theoretisch möglich, auf einer Ebene 56 verschiedene Elemente darzustellen. Bailly et al. weisen jedoch darauf hin, dass für die meisten Anwendungen deutlich weniger Elemente ausreichend sind (vgl. Abbildung 13).

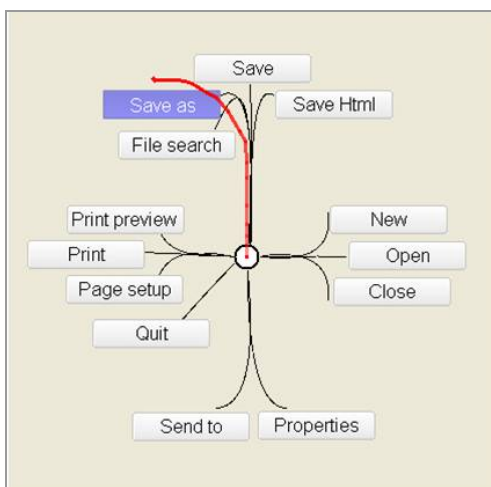


Abbildung 13: Beispielhafte Abbildung eines Flower Menu
[Baily et al. 2008]

Das besondere an *Flower Menus* ist die Möglichkeit, durch gezielte Gesten Menüpunkte bereits vor dem Erscheinen des Menüs schneller ausführen zu können. In einer Studie bestätigt sich dieser zeitliche Vorsprung gegenüber linearen Menüs, ebenso wie das richtigere Verhalten der Benutzer beim Erlernen dieser Möglichkeit der Steuerung.

3.3 Visualisierung großer Datenmengen

Die Visualisierung von großen Datenmengen hat in den letzten Jahren immer weiter an Bedeutung gewonnen. Durch den aktuellen Stand der Informationstechnologie ist es heute möglich, eine nicht geringe Anzahl an Informationen zu sammeln bzw. diese abzuspeichern [Schweizer 2009]. Diese ständig anwachsenden Datenansammlungen gilt es zu verarbeiten und so dem Benutzer einen komfortablen Einstieg bzw. einen einfachen Zugang zu den Informationen zu ermöglichen.

Forschungen in vielen Sektionen, unter anderem im Bereich der Mensch-Computer-Interaktion, haben in den vergangenen Jahren verschiedene Möglichkeiten hervorgebracht, diese Ansammlungen von Informationen und den Zugang für den Benutzer mit Hilfe unterschiedlichster Visualisierungsarten zu handhaben.

Im Folgenden wird auf die Entstehung und Entwicklung der für diese Arbeit wichtigen Visualisierungsart eingegangen – die Facettenklassifikation. Zusätzlich werden bestehende Systeme vorgestellt, die einen gewissen Einfluss auf die Konzipierung und Umsetzung des Prototypen *FacetBrowsing*, der im Rahmen des Bachelor Projektes und dieser Arbeit entstand, hatten.

3.3.1 Facettenklassifikation

Ihren Ursprung hatte die Facettenklassifikation in den Forschungsgebieten der Bibliothekswissenschaft. Bereits 1924 arbeitete Shiyali Ranganathan an dem Ansatz, ob es nicht möglich sei ein Klassifikationssystem zu entwickeln, welches multiple Blickwinkel erlaubt [Taulbee 1965]. 1933 veröffentlichte er sein Ergebnis unter dem Namen „Colon Classification“ (CC). Ausgelegt ist dieses besonders auf die Klassifikation von kleinen Artikeln aus beispielsweise Zeitungen oder Zeitschriften. Die Colon Classification teilt dabei Ausdrücke oder Bezeichnungen in Kategorien ein, welche sich anhand charakteristischer Beschriftungen unterscheiden. Ranganathan bezeichnete diesen Vorgang bereits als „Facettenanalyse“ [Taulbee 1965]. Demnach existierte bereits in den frühen 1930er Jahren die Idee, multiple Blickwinkel auf Daten durch verschiedene Facetten Darstellungen zu repräsentieren. Die Colon Classification wurde im Laufe der Jahre immer weiterentwickelt und erschien 1972 in ihrer siebten und bisher letzten Version. Die CC ist ein aufgrund ihrer sehr frühen Entstehung ein gutes Beispiel einer facettierten Klassifikation in der realen Welt [Lee et al. 2009].

In den 1970er Jahren wurde speziell der Fokus auf die Verwendung der Facetten Analyse in Verbindung mit Thesauri gelegt. Thesaurofacet war 1970 das erste explizit genutzte facettierte Klassifikationssystem in dieser Zusammensetzung. Die zu damaliger Zeit neuartige Integration von Begriffen in Klassifizierungslisten und in Thesauren führ-

te zu der Schaffung eines Systems, das sowohl in computergestützten Indexierungs- und Informationsbeschaffungssystemen sowie in traditionellen Bibliotheks- und Dokumentationszentren Verwendung fand [La Barre 2006].

Diesem Ansatz folgend wurden einige Systeme entwickelt, die heute noch zur Anwendung kommen. Zu diesen Systemen gehören beispielsweise die „London Education Classification“ [Fosket 1974] und die „London Classification of Business Studies“ [Vernon 1979].

Bei der Verwendung der Facetten Analyse lag in den 1980er Jahren der Fokus nicht mehr auf dem Einsatz von Thesauri. Bestehende Systeme wurden kritisiert [La Barre 2006] und neue Systeme unter einem überarbeiteten Forschungsfokus realisiert. Die „Dickens House Classification“ verfolgte 1987 beispielsweise den Ansatz, die Facetten Klassifikation für den Zugriff auf Bilder zu verwenden. Ebenso kam es 1989 zur Diskussion über den Einsatz der FC im Forschungsfeld des Webs. [Duncan 1989]

Über die Jahre hinweg haben Forscher besonders im Hinblick auf die historische Bezeichnung der Facetten Analyse und dem Potential in einer digitalen Umgebung geforscht. In diesem digitalen Zusammenhang wurde der Begriff der Facetten Klassifikation immer mehr durch den des „facettierten Browsens“ ersetzt.

Flamenco Search Framework Das „Flamenco Search Interface Framework“⁶ (2001) war eine recht frühe Demonstration, dass facettiertes Browsen in Verbindung mit einer umfangreichen Fähigkeit der „Dynamic Queries“ sehr effektiv war und eine erfreuliche User Experience im Bereich des Suchens und Browsens darlegte [Lee et al. 2009]. Das Flamenco Framework ist ein Interface mit der primären Zielsetzung, den Benutzern eine flexible Bewegung durch große Informationsräume zu ermöglichen, ohne sich dabei verloren zu fühlen. Das Framework benutzt dazu hierarchisch facettierte Metadaten, die es dem Benutzer erlauben, die aktuelle Anfrage zu verfeinern oder zu erweitern, während es die konsistente Repräsentation der Datenstruktur beibehält. [Hearst 2003] [Flamenco 2010]. Eine der Neuerungen zu den bisher bestehenden facettierten Systemen war der vereinfachte Zugriff auf eine große digitale Kollektion von Bildern (vgl. Abbildung 14). Dabei legten die Systementwickler besonderen Wert darauf, unter den Gesichtspunkten des Benutzerzentrierten Designs zu entwickeln. Der große Vorteil liegt darin, dass das Framework sehr leicht auf verschiedene Datengrundlagen angewandt werden kann. [Hearst et al. 2002]. Einsatz findet das Framework beispielsweise innerhalb von Forschungsreihen mit Bilderkollektionen des „*Fine Arts Museums of San Francisco*“⁷ und der „*UC Berkeley Architecture Visual Resources Library*“⁸.

⁶ Flamenco Framework: **FL**exible information **A**ccess using **ME**tadata in **N**ovel **C**ombinations
<http://flamenco.berkeley.edu/>

⁷ Fine Arts Museums of San Francisco (letzter Aufruf Januar 2010):
<http://orange.sims.berkeley.edu/cgi-bin/flamenco.cgi/famuseum/Flamenco>

⁸ UC Berkeley Architecture Visual Resources Library (letzter Aufruf Januar 2010):
<http://orange.sims.berkeley.edu/cgi-bin/flamenco.cgi/spiro/Flamenco>

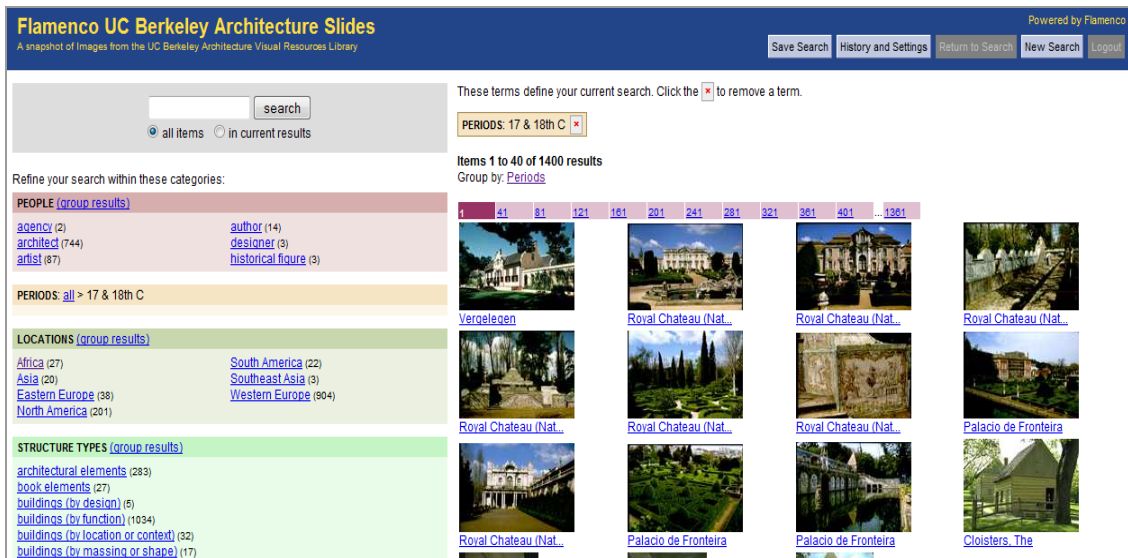


Abbildung 14: Flamenco Interface mit Datensatz „UC Berkeley Architecture Library“
 Links: Facetten mit Merkmalsausprägungen zum filtern
 Rechts: die Ergebnismenge in Form einer Bilderkollektion
 [UC Berkeley]

Bemerkenswert ist ebenfalls, dass das *Flamenco Search Interface Framework* das am meisten zitierte Beispiel im Bezug auf Facettierte Klassifikationen, ist [La Barre 2006].

RelationBrowser++ Einen ähnlichen Weg schlägt der RelationBrowser++ (2003) ein. Als eigenständige Anwendung zeigt der RelationBrowser++ (RB++) multiple Facetten an, die es ermöglichen sollen, Beziehungen zwischen den einzelnen Daten besser erkennen zu können. Hierzu wird unter anderem eine Mouseover Funktion verwendet, die dem Benutzer, sobald er sich mit der Maus über einer Facette befindet, dynamisch anzeigt, welche weiteren Attributwerte in den nächsten Schritten – in den unterschiedlichen Facetten – zur Verfügung stehen. Gleichzeitig wird dargestellt, mit welchen weiteren Facetten dieser Begriff in Verbindung steht (vgl. Abbildung 15). Auch der RB++ ermöglicht eine Verwendung verschiedener Datengrundlagen, allerdings existiert hierbei eine Limitierung in der Menge der erlaubten Metadaten, da die Navigation innerhalb des Systems schnell beschwerlich werden kann [Lee et al. 2009].

Chapter	Field of Science	Location	Format	Year
190 K-12	571 Computer	2422 United States	2010 HTML	312 1993
343 Higher Ed	259 Environmental	0 Asia	421 Excel	188 1996
207 Labor force	500 Life	0 Europe	0 PDF	210 1998
375 R&D Funds	842 Mathematics	0 Africa	5 Image	352 2000
339 Academic R&D	316 Physical	0 South America	0 Text	368 2002
421 Marketplace	112 Psychology	0 North America	0 Powerpoint	484 2004
170 Public attitudes	338 Social			508 2006
66 State indicators	131 Engineering			

Abbildung 15: Relation Browser++ Interface
 Der Mouseover Effekt (Attributwert „United States“ in Facette „Location“) zeigt, welche weiteren Attributwerte nach der Auswahl zur Verfügung stehen und wie viele Elemente darin vertreten sind. [Lee et al. 2009]

FacetMap / FacetLens FacetMap (2006) ermöglicht ein grafisches und dynamisch facettiertes Browsen in einer großen Auswahl an verschiedenen Datenbeständen, ohne dabei jedes Mal das Interface neu zusammenstellen zu müssen [Lee et al. 2009]. Laut den Autoren offenbart FacetMap allerdings Einschränkungen in der Fähigkeit, Beziehungen zwischen Daten darzustellen.

Als Weiterentwicklung der FacetMap und der PaperLens⁹ [Lee et al. 2004] entstand das FacetLens System (2009). FacetLens [Lee et al. 2009] ist ein interaktives System, das Beziehungen und Zusammenhänge in einem facettierten Datenraum aufdeckt und visualisiert (vgl. Abbildung 16). FacetLens weist im Vergleich zu den bisher bestehenden Systemen zwei wesentliche Veränderungen bzw. Erweiterungen auf. Traditionelle Facettentypen können eine (bspw. das Jahr, in dem die Arbeit veröffentlicht wurde) bzw. mehrere (z.B. die Autoren) Ausprägungen pro Attribut aufweisen. Des Weiteren können sie hierarchische Attribute, wie beispielsweise verwiesene Quellen, die wiederum andere Quellen oder Arbeiten referenzieren, abbilden. FacetLens erweitert diese traditionellen Typen um die linearen Facetten. Während Attribute an sich kategorisch sind, ist es bei linearen Facetten möglich, die innere visuelle Repräsentation der Daten, nach z.B. der zeitlichen Beziehung, zu bewahren und darzustellen. Eine weitere Neuerung gegenüber anderen Systemen ist die Möglichkeit des Users, jederzeit während der Exploration zwischen zusammenhängenden Facetten zu wechseln („pivoting“). Dies hat den Vorteil, dass der User das Gefühl für den Kontext behält, während er schnell und effizient den Datenraum erkundet. [Lee et al. 2009].

Dank dem heutigen ansteigenden Bedarf an der effektiven Handhabung des enormen Informationswachstums, gewinnt die facettierte Darstellung von Daten immer weiter an Bedeutung. Vor allem das facettierte Browsen erfreut sich in den letzten Jahren immer größerer Beliebtheit [Lee et al. 2009]. Die Anzahl der Applikationen, die diese Technik der Visualisierung in den verschiedensten Bereichen (u.a. Websuche, Online Shopping, Bibliotheken) nutzen, steigt stetig an [Kules et al. 2009].

⁹ PaperLens ist eine Visualisierung die Trends und Verbindungen offenbart. Es verbindet dabei Sichten über wissenschaftliche Arbeiten, Autoren und Referenzen [Lee et al. 2004].

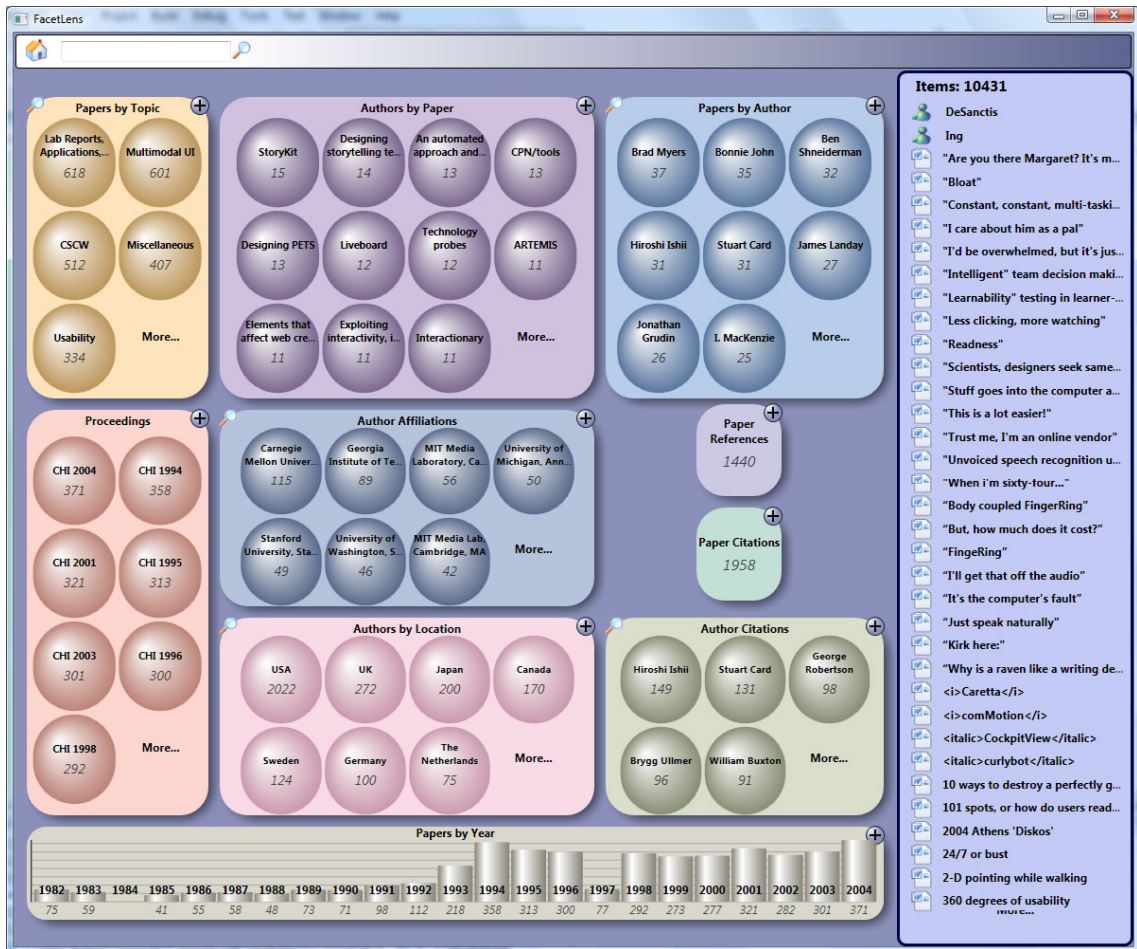


Abbildung 16: FacetLens Interface im Startzustand.

Facetten und ihre Attributwerte sind farblich hervorgehoben
 [FacetLens 2010]

4 FacetBrowsing: Konzept und Design

Die Problemstellung sowie die Grundfunktionen des *FacetBrowsing* Prototypen werden zu Beginn dieses Kapitels in einer kurzen Zusammenfassung beschrieben. Die einzelnen Konzeptideen werden im Anschluss detailliert aufgezeigt und ihre Verwendung und Funktionsweise innerhalb der Visualisierung erläutert.

4.1 Problemstellung

Viele der bisher umgesetzten Visualisierungen, die auf dem ZOIL Framework basieren und innerhalb der Arbeitsgruppe Mensch-Computer Interaktion der Universität Konstanz entstanden sind, haben die Gemeinsamkeit, Objekte direkt auf der Informationslandschaft zu platzieren. Je mehr Objekte in der Datengrundlage vorhanden sind, desto größer fällt in den meisten Fällen die Anzahl der direkt auf der Landschaft verorteten Objekte aus. Die Folge können die vorgestellten technischen sowie konzeptionellen Grenzen sein, die ein übersichtliches und problemloses Arbeiten bzw. ein problemloses Suchen und Browsen, erschweren können. Ebenso kann es ohne spezielle Hilfsmittel zu Orientierungsproblemen kommen, bei denen sich der Benutzer verloren fühlt und ein gezieltes Navigieren beeinträchtigt wird.

Die Motivation der Konzeption und der letztendlichen Umsetzung des Prototypen *FacetBrowsing* lag darin, diese technischen und konzeptionellen Grenzen zu verbessern oder gar zu lösen.

Eine detailliertere Darstellung der technischen und konzeptionellen Grenzen des ZOIL Frameworks sowie MedioVis 2.0 kann in Kapitel 2 nachgelesen werden.

4.2 Vorstellung des Prototypen FacetBrowsing

Um einen Überblick über den Prototypen *FacetBrowsing* zu erhalten, wird dieser in dem nun folgenden Unterkapitel vorgestellt und seine Funktionsweise sowie die realisierten Konzepte kurz erläutert.

FacetBrowsing ist eine zoom- und facettenbasierte Visualisierung, die den Benutzer bei seinem browsing-orientierten Suchverhalten unterstützen soll. Der Prototyp wurde in der Arbeitsgruppe Mensch-Computer Interaktion (Universität Konstanz) innerhalb des Projektes MedioVis auf Basis des ZOIL Frameworks realisiert. Wie auch bei MedioVis 2.0 (vgl. Kapitel 2.3), besteht die momentan verwendete Datengrundlage dementsprechend aus Film-Metadaten der *Internet Movie Database* (IMDb). Durch die Zusammenführung dieser Metadaten, wie beispielsweise das Rating oder das Land in dem der Film gedreht wurde, in semantisch zoombare Facetten, wird eine sehr schnelle Filterung bzw. Eingrenzung der Datenmenge ermöglicht.

Seinen Vorstellungen entsprechende Filme kann der Benutzer auf recht einfachem Wege ausfindig machen, indem er durch verschiedene Facetten navigiert und somit die Datenmenge anhand der einzelnen Attributwerte filtert bzw. einschränkt. Abbildung 17 zeigt einen beispielhaften Filtervorgang nach Filmen mit dem Genre „Crime“, die in den „USA“ gedreht wurden und das „Rating 8“ aufweisen. Dabei ist es belanglos, ob gezielt nach einem Film gesucht wird oder ob der Benutzer lediglich solange durch die Datenmenge browsst, bis ein interessantes Objekt gefunden ist. Aufgrund der ständigen Anzeige kommender Auswahlmöglichkeiten, erlaubt die Facettenklassifikation einen flüssigen Wechsel zwischen Suchen und Browsen [Hearst 2006].

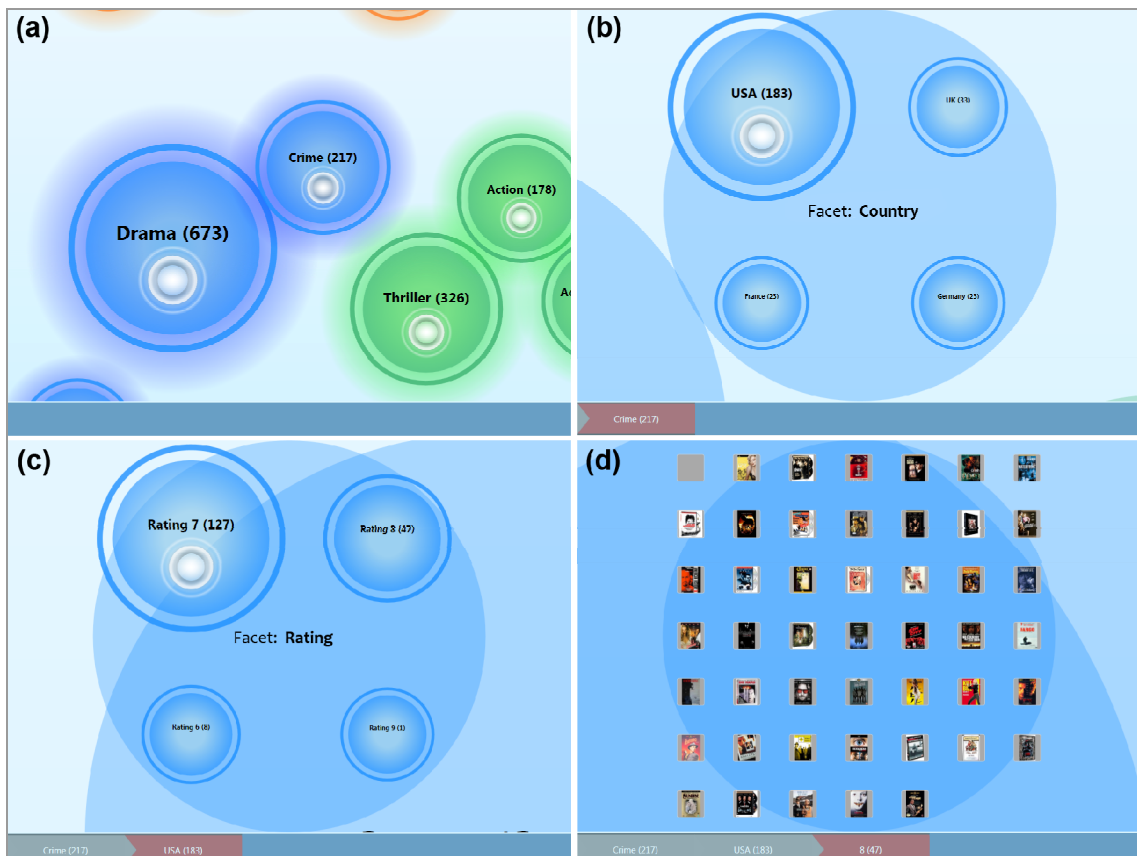


Abbildung 17: Beispielhafter Filtervorgang (a-d) bei FacetBrowsing bis hin zu den gefilterten Filmobjekten

User Interface *FacetBrowsing* besteht aus zwei wesentlichen Komponenten, anhand derer der Benutzer interagieren kann. Der im oberen Abschnitt gelegene Teil der Anwendung ist der Hauptteil der Anwendung. Hier befindet sich die zoombare Informationslandschaft (vgl. Kapitel 2.2) und die facettierte Visualisierung mit mehreren verorteten Facetten. Die Daten sind im Startzustand nach dem Genre gruppiert. Im unteren Bereich des Prototypen liegt das Head-Up-Display (HUD), welches zu jeder Zeit Informationen über den aktuellen Aufenthaltsort innerhalb der Datenmenge enthält. Diese Informationen werden mit Hilfe einer Breadcrumb Navigation ermöglicht, die direkt in das HUD eingebunden ist.

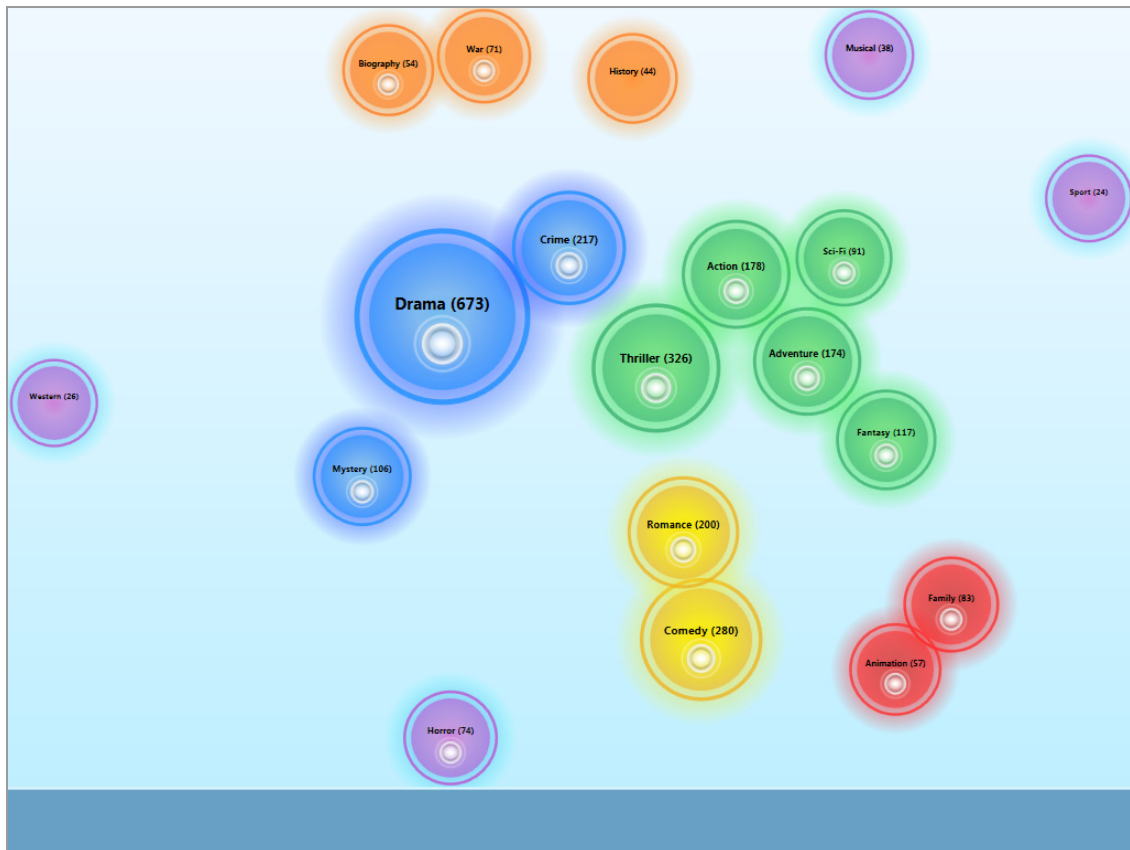


Abbildung 18: FacetBrowsing Startbildschirm mit facettierter Visualisierung (oben) und Head-Up-Display (unten)

In Abbildung 18 ist im Head-Up-Display noch nichts zu sehen, da der Startbildschirm angezeigt wird und vom Benutzer bisher nicht navigiert wurde. Befindet sich der Anwender bereits in einer tieferen Navigationsebene, so wird dies wie in Abbildung 19 zu sehen dargestellt.

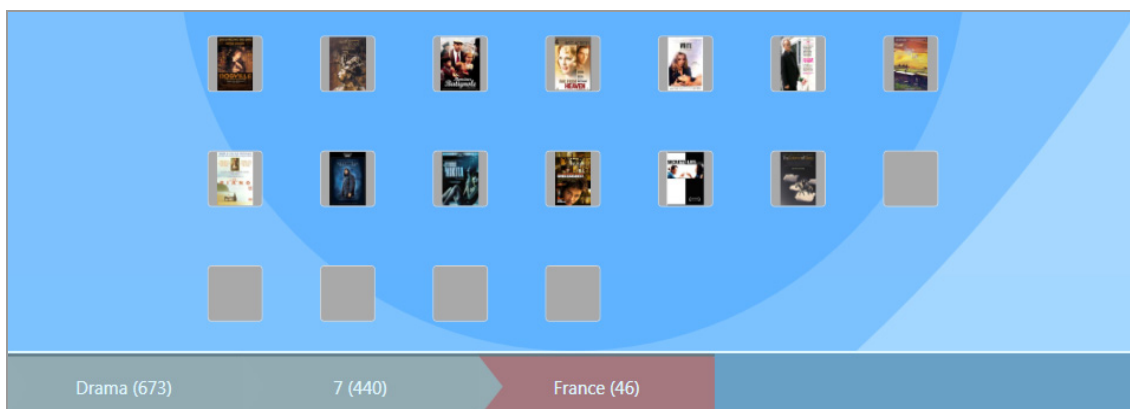


Abbildung 19: FacetBrowsing: Navigation in tiefere Navigationsebene (oben) und Head-Up-Display mit einer Breadcrumb Navigation (unten)

Navigation und Interaktion Für die Navigation und Interaktion mit dem Prototypen stehen dem Benutzer verschiedene Möglichkeiten zur Verfügung, um sich innerhalb der zoombaren Informationslandschaft zu bewegen. Zum Einen kann innerhalb der

Anwendung mit Hilfe der im ZOIL Framework vorhandenen Zoom- und Pan-Gesten sowie mit dem einfachen „Klickzoom“ navigiert werden. Der Klickzoom ermöglicht die Auswahl bzw. den „Klick“ auf eine Facette und bewirkt somit das automatische heranzoomen des Objektes, ohne manuell mit den entsprechenden Gesten bzw. mit der Maus zoomen zu müssen. Zum Anderen steht dem Benutzer zur Interaktion mit den Facetten zusätzlich ein *Flower Menu* zur benutzerdefinierten Filterung der Daten zur Verfügung. Das *Flower Menu* fungiert dabei mit dem gleichen Navigationsprinzip wie der Klickzoom, allerdings behält der Benutzer hiermit die Kontrolle über die Einschränkung der Datenmenge.

Orientierungshilfen Ein weiteres Ziel bei der Konzeption und Umsetzung des Prototypen ist die Gewährleistung der Orientierung während der Interaktion. Der Benutzer soll bei seiner Suche bzw. seinem Browsen durch den Datenraum nicht die Übersicht verlieren und zu jeder Zeit erkennen können, wo er sich innerhalb der Daten gerade befindet. Dazu werden verschiedene Hilfsmöglichkeiten angeboten, wie beispielsweise eine Breadcrumb bzw. Brotkrümelnavigation. Anhand des Breadcrumb Pfades kann jederzeit nachvollzogen werden, welcher Weg zu einem bestimmten Ziel zurückgelegt wurde und welcher Filterschritt welche Auswirkung auf die Anzahl der verbleibenden Daten hatte (vgl. Abbildung 19). Weitere angebotene Hilfsmittel sind die semantische Anordnung der Facetten nach ihrer Ähnlichkeit zueinander. Durch die semantische Anordnung bilden sich verschiedene Cluster, die zusätzlich farblich hervorgehoben sind und somit leicht unterschieden werden können. Die Clusterbildung und die farbliche Betonung sollen ebenfalls als Orientierungshilfe und zur besseren Übersicht beitragen.

Die aktuelle Datengrundlage besteht derzeit (Stand Februar 2010) aus ca. 3000 Filmobjekten, die anhand der Facettenklassifikation nach den jeweiligen Genres in der Informationslandschaft kategorisiert sind. Um einen schnellen Zugriff auf diese vielen Informationen zu gewährleisten, wurde BaseX¹⁰ als XML-Server angebunden. BaseX ist eine schnelle und kompakte Open Source XML Datenbank, die am Lehrstuhl „Datenbanken und Informationssysteme“ von Prof. Dr. Marc H. Scholl an der Universität Konstanz entwickelt wurde. Gleichzeitig besticht BaseX durch eine sehr effiziente Verarbeitung von XPath bzw. XQuery Anfragen und unterstützt vor allem sehr große XML Instanzen [BaseX 2010].

4.3 Vorstellung der Konzepte

Im Folgenden wird der im Rahmen des Bachelor Projektes entstandene Prototyp *FacetBrowsing* anhand der bereits kurz angesprochenen Konzeptideen detailliert beschrieben, sowie anhand von Studien und wissenschaftlichen Arbeiten ergründet. Die Konzepte entstanden während der Anforderungsanalyse an den Prototypen und wurden im Rahmen der Implementation umgesetzt (vgl. Kapitel 5).

¹⁰ BaseX: <http://www.inf.uni-konstanz.de/dbis/basex/>

Es werden nun die folgenden Konzepte detailliert beschrieben:

- Visualisierung durch Facettenklassifikation
- Semantische Anordnung
- Navigation und austauschbare Zoomstufen
- Orientierung innerhalb der Visualisierung

4.3.1 Visualisierung durch Facettenklassifikation

4.3.1.1 Facettenklassifikation

Der Grundgedanke, um die aufgezeigten Grenzen zu verbessern ist der Ansatz, die Informationsobjekte innerhalb der Datenmenge erst dann anzeigen zu lassen, wenn die Auswahl an Filmen, die der Benutzer sehen möchte, ersichtlich ist. Der Benutzer soll demnach die Möglichkeit haben, innerhalb der Informationsmenge mit Hilfe von gruppierten Daten nach seinen Bedürfnissen und Zielen zu filtern. Diese Vorgehensweise hat den Vorteil, dass die einzelnen Objekte, bedingt durch die Filterung des Benutzers, erst ab einer gewissen Restanzahl angezeigt und somit zu keiner Zeit zu viele Informationsobjekte gleichzeitig dargestellt werden.

Als Visualisierung bietet sich hierbei das Konzept der Facettenklassifikation an, bei der verschiedene Attribute eines Objektes, wie beispielsweise das Rating oder die Schauspieler eines Filmes, in sogenannten Facetten kategorisiert werden. Facettiertes Browsing ist dabei eine neuartige Erweiterung der Facettenklassifikation, welcher in mehreren Studien [Yee et al. 2003] [Zhang et al. 2005] [Gildea et al. 2008] bereits großes Potenzial bei der Darstellung großer Datenmengen zugesprochen wurde. Die effektive und effiziente Art Daten abzubilden, unterstützt den Benutzer auf unterschiedlichste Art und Weise bei seinem browsing-orientierten Suchverhalten, welche im Folgenden näher erläutert werden.

Zhang et al. belegen in ihrer Studie, dass facettierte Kategorien eine Möglichkeit darstellen, Benutzer dabei zu unterstützen, den Aufbau einer Ansammlung von Informationen besser zu verstehen [Zhang et al. 2005]. Zhang erläutert weiter, dass es ein facettierter Ansatz ermöglicht, den Informationsraum durch verschiedene Sichtweisen und Dimensionen zu erkunden und es dem Benutzer somit erlaubt wird, die Daten aus verschiedenen Perspektiven zu betrachten. Desweiteren kann das Explorieren der Beziehungen verschiedener Facetten das Verständnis des Benutzers für die Daten vertiefen und ihm dabei helfen, neue Erkenntnisse zu gewinnen [Zhang et al. 2005].

Yee et al. [2003] zeigen ebenfalls anhand einer Studie mit 32 Studenten, dass es eine starke Präferenz zur Nutzung der facettierten Kategorien gibt. Probanden sollten hier mit Hilfe eines facettierten Interfaces – dem Flamenco System (vgl. Kapitel 3.3.1) – eine große Anzahl von Bildern durchsuchen und verschiedene Aufgaben lösen.

Um Vergleiche anstellen zu können, wurden alle Aufgaben ebenfalls mit Hilfe einer herkömmlichen Bilder Suche bewältigt [Yee et al. 2003].

Die Mehrheit der Probanden äußerte sich positiv im Bezug auf das facettierte Interface. Es wurden größere Such-Erfolge erzielt und gleichzeitig stieg die Zufriedenheit der Probanden im Bezug auf die erzielten Ergebnisse. Des Weiteren zeigte sich in Verbindung mit dem facettierten Suchen eine erhöhte Vertrautheit mit den Inhalten der Daten. Hearst et al. kommen zu dem Ergebnis, dass aufgrund dieser Resultate der facettierte Ansatz einer Visualisierung einen erfolgreichen Weg darstellt, Benutzern viele Daten zugänglich zu machen [Yee et al. 2003].

Kules et al. [Kules et al. 2009] zeigen ebenso das Potenzial der facettierten Technik auf und verweisen auf die hohe Flexibilität des Ansatzes, der einen flüssigen Wechsel zwischen der Strategie des Suchens und der des Browsens ermöglicht. Dieser Wechsel wird durch ein facettiertes Interface unterstützt, da stets die nächsten ausführbaren Filtermöglichkeiten angezeigt werden [Hearst 2006].

Zusammenfassend ist die Facettenklassifikation eine gute Möglichkeit, viele Daten effizient und effektiv zugänglich zu machen, ohne dabei den Überblick über die gesamte Datenmenge zu verlieren.

4.3.1.2 Facetten

Facetten sind innerhalb der Anwendung der Hauptbestandteil der facettierten Visualisierung. Sie bieten die Möglichkeit, die zugrunde liegenden gruppierten Daten zu durchsuchen und zu filtern. Eine detaillierte Beschreibung einer Facettenklassifikation kann in Kapitel 2.4 nachgelesen werden.

Facetten werden bei *FacetBrowsing* kreisförmig dargestellt und enthalten den jeweiligen Namen der Metainformation, nach der anhand der verschiedenen Attributwerte bzw. Merkmalsausprägungen gefiltert werden kann (vgl. Abbildung 20b). Im Startzustand des Prototypen repräsentiert die Facette *Genre*, ähnlich wie in MedioVis 2.0 beobachtet werden kann, alle in der aktuell verwendeten Datenmenge vorkommenden Filmgenres. Als einzige Ausnahme wird die Startfacette *Genre* nicht wie die übrigen Facetten kreisförmig dargestellt, sondern wird dem Benutzer in Gestalt der Informationslandschaft veranschaulicht. Wie bereits in Abbildung 18 zu sehen, werden die Attributwerte in Form der verschiedenen Genres dargestellt. Die Informationslandschaft fungiert dabei als „unsichtbare“ Genre-Facette. Jeder Attributwert repräsentiert im Startzustand demnach ein Genre, welches mit dem entsprechenden Namen, beispielsweise „Action“ oder „Comedy“, gekennzeichnet ist und die betreffenden Filmobjekte enthält. Diese textliche Repräsentation des Namens des Attributwertes wird durch die genaue Anzahl der dahinter liegenden Daten ergänzt. (vgl. Abbildung 20a).

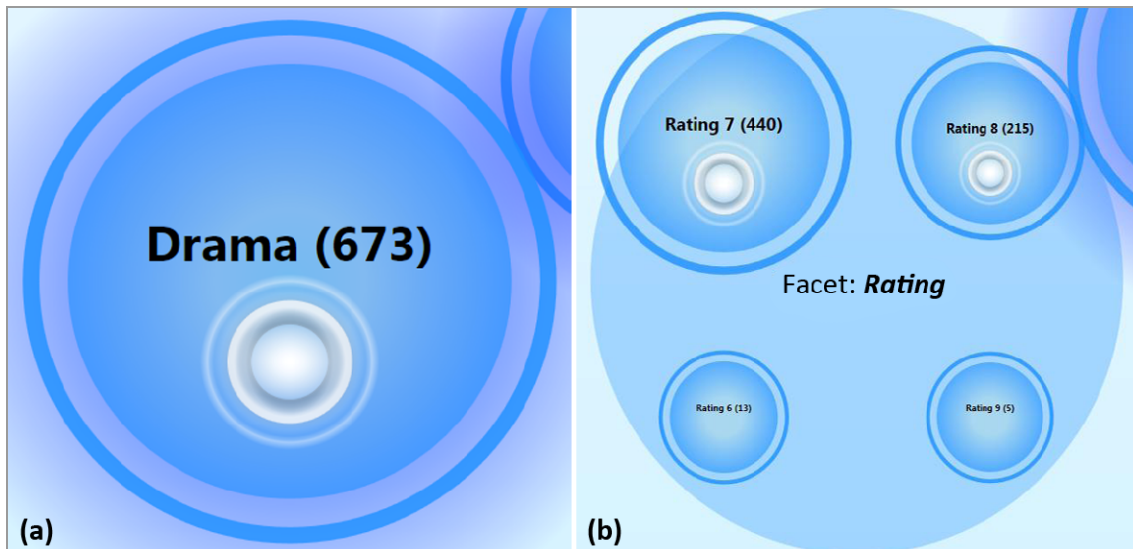


Abbildung 20: Aufbau einer Facette und deren Attributwerte

(a) Ansicht des Attributwertes „Drama“ mit Anzahl der enthaltenen Filmdaten, (b) Zoom ermöglicht weitere Filtermöglichkeiten anhand der Facette Rating und ihren Merkmalsausprägungen „Rating 6-9“

Um dem Benutzer die Menge der Daten zusätzlich zu veranschaulichen, wird die Größe der einzelnen Attributwerte linear im Bezug auf die Menge der enthaltenen Informationen skaliert (vgl. Abbildung 21). Demnach ist eine Merkmalsausprägung die sehr viele Objekte enthält größer dargestellt als eine mit wenigen Objekten. Diese Darstellung soll dem Benutzer ein schnelleres Identifizieren von informationsreichen Attributen ermöglichen. Mit Hilfe der Facette und ihren Attributwerten bzw. Merkmalsausprägung hat der Benutzer die Möglichkeit, den aktuellen Datensatz nach seinen Bedürfnissen einzuschränken und in der Folge die unerwünschten Filme heraus zu filtern.

$$x * \frac{\text{current_count}(\text{Movies}) - \min(\text{count_allMovies})}{\max(\text{count_allMovies}) - \min(\text{count_allMovies})} + y$$

Abbildung 21: Formel für die lineare Skalierung der Facetten.

„current_count(Movies)“ steht dabei die Anzahl der Filme in der aktuellen Kategorie. „min(count_allMovies)“ bzw. „max(count_allMovies)“ steht für die kleinste/ größte Anzahl an Filmen aus allen Kategorien der aktuellen Facette.

Das Besondere an einem Filtervorgang ist die Eigenschaft des semantischen Zooms, das jede einzelne Kategorie aufweisen kann. Zoomt ein Benutzer auf ein Attribut innerhalb der Facette, wird der entsprechende Filter aktiviert und weitere Einschränkungsmöglichkeiten der Daten werden sichtbar. Der Benutzer hat nun beispielsweise die Möglichkeit, anhand der neuen Facette mit ihren Attributwerten die Daten nach den verschiedenen „Ratings“ zu filtern (vgl. Abbildung 20b).

Einzelne Filmobjekte werden allerdings erst dann für den Benutzer sichtbar, sobald nach mehreren Filterschritten eine gewisse Restanzahl an Daten übrig ist und es sinnvoll erscheint, diese anzeigen zu lassen (vgl. Abbildung 22). Die Grenze für die Anzeige der gefilterten Objekte liegt in der aktuellen Version von *FacetBrowsing* bei 50 Objekten, welche sich allerdings manuell im Quellcode verändern lässt. Objekte werden demnach erst für den Benutzer sichtbar gemacht, nachdem die aktuelle Datenmenge auf 50 Filme reduziert wurde. Diese Objektgrenze wurde nach mehrmaligem „Try & Error“ festgelegt, da mit dieser Anzahl die Darstellung und die Übersicht noch gewährleistet sind.

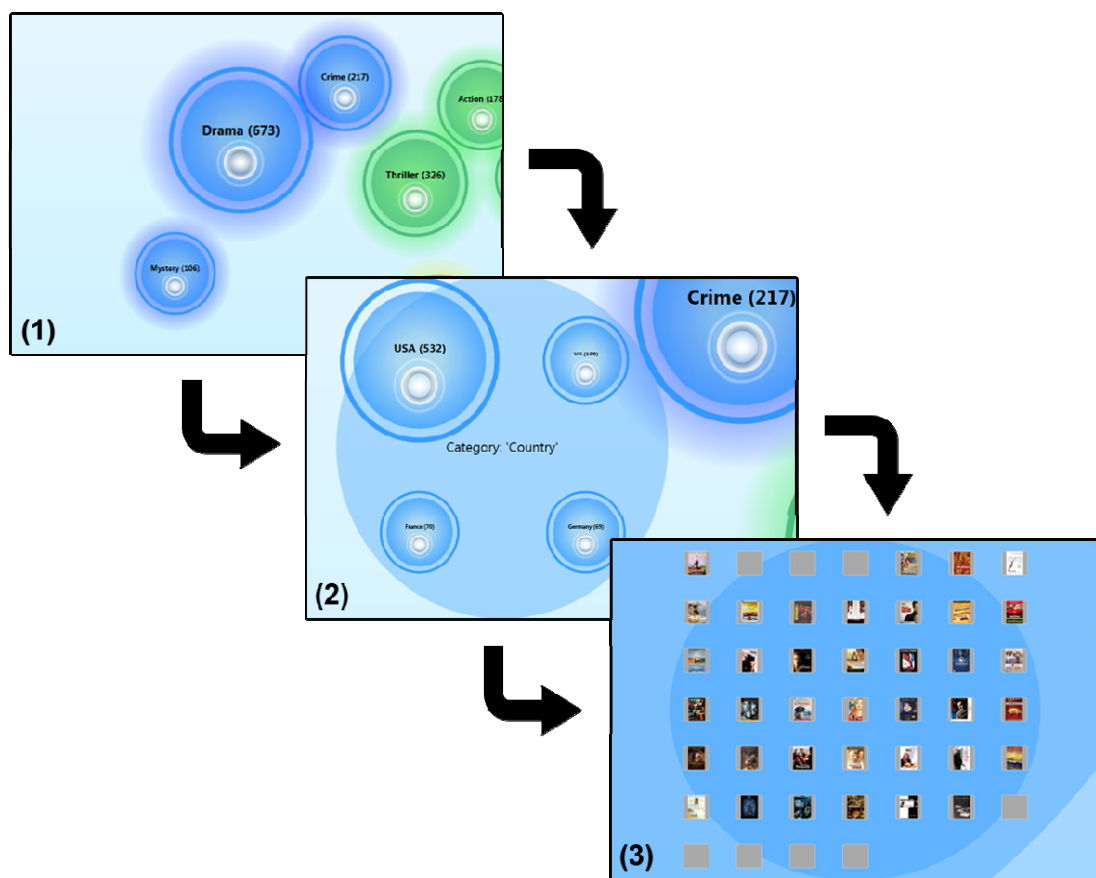


Abbildung 22: Ablauf des semantischen Zooms bei FacetBrowsing

- (1) Zoom auf den Attributwert „Drama“.
- (2) Weitere Merkmalsausprägungen werden sichtbar und können zur Filterung genutzt werden - hier Länder in dem ein Film gedreht wurde.
- (3) Vorgang ist solange wiederholbar, bis einzelne Filmobjekte letztendlich dargestellt werden

4.3.2 Semantische Anordnung

Alle vorhandenen Genres der zugrunde liegenden Filmobjekte sind als Einstiegspunkte in den Datenraum auf der Informationslandschaft verortet. Der Unterschied zu dem bisherigen Aufbau (Stand: Dezember 2009) von MedioVis 2.0 besteht darin, dass Genre-Attribute, wie beispielsweise „Action“ oder „Thriller“, nicht willkürlich und ohne Hin-

tergedanken angeordnet sind. Genres, die sich ähnlich sind, befinden sich nahe beieinander und Genres die sich eher fremd sind, liegen weiter auseinander (vgl. Abbildung 23). Diese Idee der semantischen Anordnung der Genres soll den Benutzer unter anderem dabei unterstützen, eine bessere Wahrnehmung der einzelnen Genres zu erlangen und diese genauer unterscheiden zu können. Ebenso dient sie der besseren Übersicht und Orientierung innerhalb der Informationslandschaft.

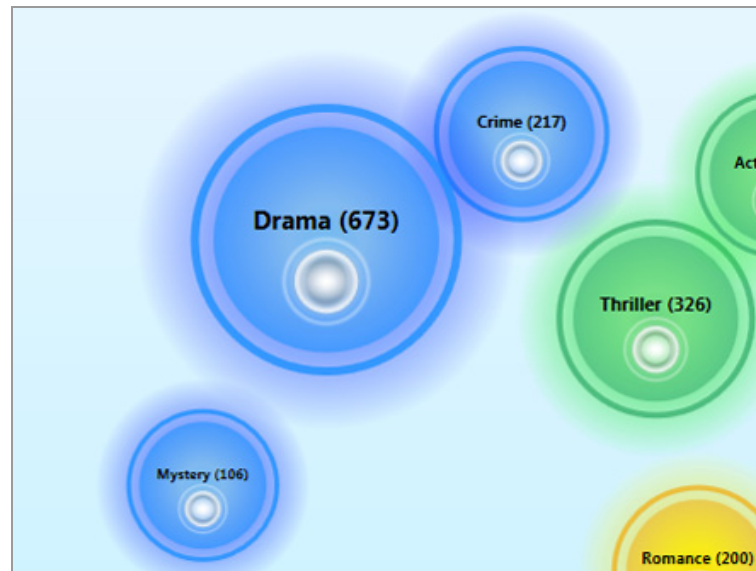


Abbildung 23: Darstellung ähnlicher Genres auf der Informationslandschaft

Dieser Ansatz entstand aus der Tatsache heraus, dass es im Allgemeinen sehr schwer ist Genres zu unterscheiden bzw. klare Abgrenzungen festzustellen. Daniel Chandler bestätigt diese Aussage in einem Paper über Genre Theorien. Er hebt an diesem Punkt hervor, dass Genres sich oft überlappen und es ebenfalls „gemischte Genres“ (bspw. Action-Komödien) gibt, die eine eindeutige Abgrenzung schwer machen. [Chandler 2000]. Nicholas Abercrombie vertritt ebenfalls die Auffassung, dass sich die Grenzen der Unterscheidung mittlerweile zwischen den Genres bewegen und allgemein durchlässiger werden [Abercrombie 1996].

FacetBrowsing versucht mit Hilfe der Anordnung der Attributwerte dem Benutzer zu ermöglichen, die schwammigen Grenzen zwischen den einzelnen Genres zu identifizieren, indem einzelne Genres klar getrennt voneinander auf der Landschaft liegen. Selbst ähnliche Genres, die sehr nahe zueinander abgebildet sind, haben aufgrund ihrer Darstellung klare Grenzen. Ebenfalls erlaubt die semantische Anordnung alternative Genres, die dem Lieblingsgenre ähnlich sind und für den Benutzer ebenfalls interessant sein könnten, schneller zu erkennen. Das Erkennen alternativer Genres ist allerdings nicht die Regel, betont Nicholas Abercrombie in diesem Zusammenhang. Er erläutert, dass Genres die Hervorbringung und Pflege eines treuen Publikums erlauben, welches gewohnheitsbedingt lediglich Filme innerhalb eines bestimmten Genres anschaut [Abercrombie 1996]. Chandler fügt hinzu, dass Genres wichtige Bezugsrahmen liefern, die es den Zuschauern ermöglichen bestimmte Filme zu identifizieren und auszuwählen [Chandler 2000]. Schaut eine Person gerne Actionfilme, würde diese

anhand der aufgeführten Gründe, vielleicht nicht auf die Idee kommen, einen Film aus dem Genre „*Drama*“ zu suchen bzw. anzuschauen, obwohl sich beide, wie die semantische Anordnung in *FacetBrowsing* zeigt, anhand der Schlüsselwörter sehr ähnlich sind. Mit Hilfe der semantischen Anordnung kann demnach sofort erkannt werden, dass sich beide Genres sehr ähnlich sind. Das potenzielle Interesse des Benutzers, der ansonsten auf ein bestimmtes Genre fokussiert ist, wird geweckt.

Zusammenfassend lässt sich folglich sagen, dass *FacetBrowsing* anhand der semantischen Nähe der einzelnen Genres zueinander nicht nur die Abgrenzung aufzeigt, sondern dem Benutzer ebenfalls erlaubt zu erkennen, welche alternativen Genres interessant sein könnten. Auf diese Weise wird eine gezielte und breiter gefächerte Suche nach relevanten Objekten ermöglicht.

4.3.3 Navigation und „austauschbare Zoomstufen“

Die Interaktion mit der facettierten Visualisierung und somit auch die Navigation durch den Informationsraum können auf mehreren Wegen erfolgen. Zum Einen besteht die Möglichkeit mit Hilfe der, aus dem ZOIL Framework bekannten, Touch-Gesten oder der Maus innerhalb der Visualisierung zu Zoomen und zu Panen, um somit zu einem bestimmten Objekt zu gelangen oder sich auf der Informationslandschaft fortzubewegen. Zum Anderen erlaubt der „Klickzoom“ ein direktes Interagieren mit einem gewünschten Objekt. Durch das Berühren bzw. das Anklicken eines Attributwertes innerhalb einer Facette wird der „Klickzoom“ ausgelöst und ermöglicht das automatische und zentrierte Heranzoomen des Objektes.

Sobald ein Attributwert durch die entsprechenden Interaktion vergrößert wurde, wird diese, bedingt durch den semantischen Zoom, semantisch umgeschaltet. In der Folge wird dem Benutzer eine Facette mit mehreren Attributwerten als Filtermöglichkeiten angeboten, mit denen die aktuelle Datenmenge erneut eingeschränkt werden kann. Das System erstellt dabei die „neue“ Facette automatisch und bietet diese dem Benutzer an. Allerdings sollte darauf geachtet werden, trotz der automatisierten Bereitstellung von Filtermöglichkeiten, die Usability Prinzipien der Steuerbarkeit sowie der Individualisierbarkeit durch den Benutzer zu garantieren. Benutzer können sich unsicher fühlen, insbesondere wenn es zu Veränderungen kommt, die sie nicht initiiert haben und nicht nachvollziehen können [Mauder 2006].

Diese Richtlinien sind in der ISO 9421¹¹ für „Grundsätze der Dialoggestaltung“ festgelegt:

Steuerbarkeit: Die Steuerbarkeit besagt, dass ein Dialog dann steuerbar ist, „wenn der Benutzer in der Lage ist, den Dialogablauf zu starten sowie seine Richtung und Geschwindigkeit zu beeinflussen, bis das Ziel erreicht ist“ [ISO 2005].

¹¹ DIN EN ISO 9421-110: „Grundsätze der Dialoggestaltung“

Individualisierbarkeit: Die Individualisierbarkeit besagt, dass ein Dialog dann individualisierbar ist, „wenn das Dialogsystem Anpassungen an die Erfordernisse der Arbeitsaufgabe, individuelle Vorlieben des Benutzers und Benutzerfähigkeiten zulässt“. [ISO 2005]

Dementsprechend sollte der Benutzer weiterhin die Freiheit haben, selber Entscheidungen zu treffen, bzw. einen Einfluss auf die Anwendung haben. Das alleinige Anbieten der automatisierten Filtermöglichkeiten bei *FacetBrowsing* unterstützt diese Prinzipien der Steuerbarkeit und Individualisierbarkeit nicht zwingend, da der Benutzer keinen direkten Einfluss auf die kommenden Facetten und Attributwerte hat.

Aus diesem Grund wurde das Konzept einer zusätzlichen Interaktions- und Navigationstechnik verfolgt, die dem Benutzer die nötige Freiheit beim Filtern der Datenmenge einräumt. Die Wahl fiel hier auf das von Bailly vorgestellte *Flower Menu* [Bailly et al. 2008]. Benutzertests der Autoren haben gezeigt, dass die Interaktion mit einem *Flower Menu* sehr effizient im Bezug auf das Erlernen dieser Technik ist und, im Vergleich mit linearen Menüs, ein wesentlich schnelleres auswählen von Menüeinträgen ermöglicht.

Jede Kategorie innerhalb einer Facette besitzt ein *Flower Menu*, das durch längeres Drücken aktiviert wird und auf diese Weise dem Benutzer alle noch möglichen Einschränkungen als Auswahl zur Verfügung stellt (vgl. Abbildung 24). Die Person kann in der Folge den nächsten Schritt der Filterung selbst bestimmen, indem er den Pfad bis zur Auswahl innerhalb des Menüs „entlangfährt“. Es bietet sich somit die Möglichkeit auf der einen Seite den vorgegebenen Weg der Filterung zu gehen und auf diese Weise den Datenraum zu durchstöbern oder auf der anderen Seite gezielt nach bestimmten Kriterien mit Hilfe des *Flower Menus* zu filtern. Infolgedessen steht es dem Benutzer zu jeder Zeit frei für welche Art des Suchens er sich entscheidet. Ein Wechsel der Strategie ist während der Filterung zu jeder Zeit möglich, z.B.:

- **Strategie 1:** Benutzung des *Flower Menus* zur individuellen Filterung
- **Strategie 2:** Benutzung des Klickzooms zur vom System vorgegebenen Filterung



Abbildung 24: Flower Menu innerhalb von FacetBrowsing

Nach der Auswahl der Kategorie mit Hilfe des *Flower Menus*, wird das Objekt, wie auch bei der direkten Klickzoom-Selektion, automatisch heran gezoomt und stellt neue Filtermöglichkeiten, anhand des Prinzips des semantischen Zooms, zur Verfügung.

Der allgemeine Unterschied des semantischen Zooms bei *FacetBrowsing* zu dem des ZOIL Frameworks liegt darin, dass im Vorfeld keine festen Zoomstufen definiert werden können, da je nach Datenmenge bzw. je nach Auswahl des Benutzers per *Flower Menu* vorher nicht bekannt ist, welche Kategorien als nächstes angeboten werden sollen. Aus diesem Grund ist eine im Vorfeld fest definierte Zoomstufenreihenfolge, wie es beispielsweise bei den Filmobjekten in MedioVis der Fall ist, nicht möglich.

Das Konzept welches dieses Problems löst, ist die Idee der dynamisch austauschbaren Zoomstufen. Der Grundgedanke dabei ist es, keine fest implementierten Zoomstufen anzubieten, sondern auf die jeweilige Interaktion des Benutzers zu interagieren und so dynamisch zur Laufzeit der Anwendung entsprechende Facetten sowie Attributwerte zu erstellen. *FacetBrowsing* berechnet daher für jede Auswahl bzw. jeden Filterschritt in der aktuellen Datenmenge, je nach vorheriger Auswahl des Benutzers, die nächsten Filtermöglichkeiten und erstellt dynamisch die entsprechende Zoomstufe. Auf diese Weise wird die fehlende Möglichkeit, fest definierte Zoomstufen anzugeben, umgangen. Ein weiterer Unterschied der bisherigen Anwendung des semantischen Zooms ist es, dass die bei *FacetBrowsing* dynamisch generierten Zoomstufen jederzeit ersetzt werden können. Dies ist notwendig, falls der Benutzer in ein Objekt hinein zoomt und sich entscheidet, wieder einen Schritt zurück zu gehen, um einen alternativen Filterweg einzuschlagen. Die zuvor generierte Zoomstufe wird durch die neue Auswahl des Benutzers ersetzt

Eine weitere Möglichkeit innerhalb des Datenraumes zu navigieren ist die Interaktion mit dem Breadcrumb Pfad. Zur Umsetzung innerhalb des Prototypen *FacetBrowsing* wurden die aus Kapitel 3.1.1 bekannten unterschiedlichen Varianten eines Breadcrumb Pfades diskutiert und entschieden, dass der „Path Breadcrumb“ Pfad die geeignetste Variante darstellt, da der Benutzer auf diese Weise genau nachvollziehen kann, welchen exakten Weg er bereits zurückgelegt hat. Mit Hilfe des Breadcrumb Pfades bietet sich ebenso die Möglichkeit, an jeden Punkt der Hierarchie zurück zu navigieren, indem der entsprechende Breadcrumb Eintrag ausgewählt wird. Die Visualisierung reagiert auf diese Interaktion und zoomt zu der jeweiligen Stelle im Datenraum zurück. Der Benutzer kann in folge dessen schneller mehrere Schritte auf einmal zurückspringen und einen neuen Filterweg einschlagen, ohne durch viele einzelne Klicks in der Hierarchie zurück zu gehen oder erneut vom Startpunkt aus zum gewünschten Ziel zu gelangen.

4.3.4 Orientierung innerhalb der Visualisierung

Ein weiterer wichtiger Bestandteil bei Visualisierungen von großen Datenmengen ist die Unterstützung des Benutzers bei der Orientierung. Stehen keine Hilfsmittel für die Orientierung zur Verfügung, kann der Benutzer nach kürzester Zeit den Überblick über die Datenmenge verlieren. Es kann nicht mehr nachvollzogen werden, welcher Weg

bereits zurückgelegt wurde, um an das gewünschte Ziel zu gelangen bzw. in welchem Teil der Visualisierung sich der Benutzer gerade befindet.

Zur Verbesserung der Orientierung innerhalb des Prototypen wurden mehrere Konzeptideen entwickelt und ausgebaut. Die gerade erwähnte Breadcrumb Navigation stellt, neben der Navigations- und Interaktionsmöglichkeit (vgl. Kapitel 4.3.3), ebenfalls eine sehr gute Orientierungshilfe für einen Benutzer dar. Auch im Hinblick auf die Orientierung, sollte der im Punkt 3.1.1.3 erwähnte Path Breadcrumb Pfad im Prototyp umgesetzt werden. Wie beschrieben, reflektiert ein Path Breadcrumb Pfad dabei die gesamte Tiefe der Hierarchie bzw. den exakt zurückgelegten Weg durch die Datenmenge. Der Zweck, den ein solcher Pfad demnach erfüllen soll, ist dem Benutzer ständig die Information bereitzustellen, wo genau er sich innerhalb des Informationsraumes befindet [Lida et al. 2003]. Ergebnisse einer Studie der „Wichita State University“ aus dem Jahr 2003 heben an diesem Punkt hervor, dass das Anbieten eines Breadcrumb Pfades eine signifikante Verbesserung der Benutzbarkeit aufzeigt und von Benutzern dementsprechend hoch honoriert wird [Hudson 2004].

Zoomt eine Person auf ein für ihn interessantes Filterkriterium, wird der jeweilige Filter aktiviert und dem Breadcrumb Pfad hinzugefügt. Der aktuell zuletzt angewandte Filterschritt wird dabei farblich hervorgehoben. Anhand der zusätzlich vermerkten Zahl neben einem Breadcrumb-Eintrag kann erkannt werden, auf wie viele Objekte die aktuelle Datenmenge eingegrenzt wurde. Diese Hervorhebung dient als eine ergänzende Orientierungs- und Übersichtshilfe.

Abbildung 25 zeigt ein typischen Path Breadcrumb Pfad in der Anwendung und veranschaulicht dem Benutzer die bereits zurückgelegte Hierarchie innerhalb der Daten. In diesem Beispiel wären dies:

- alle Filme aus dem Genre „Drama“,
- mit einem Rating von 7,
- die im Vereinigten Königreich (UK) gedreht wurden,
- sowie Ethan Coen als Regisseur aufweisen.

Im Zuge der Filterung, wurde die verbleibende Datenmenge von 673 auf 3 Filmobjekte reduziert.



Abbildung 25: Beispiel eines „Path Breadcrumb“ Pfades bei FacetBrowsing

Angezeigt wird der genaue Weg durch den Datenraum und die Anzahl der gefilterten Objekte. Der aktuelle Standpunkt ist farblich hervorgehoben (rot)

Zusätzlich zum Breadcrumb Pfad wird, wie in Abbildung 26 zu sehen ist, der Name der aktuell im Fokus stehenden Facette in textueller Form angezeigt. Dieser Hinweis prä-

sentiert sich dem Benutzer immer in der Mitte der Facette und unterstützt damit ergänzend die Orientierung durch die Informationslandschaft, indem er den Überbegriff der aktuell möglichen Filterschritte angibt.

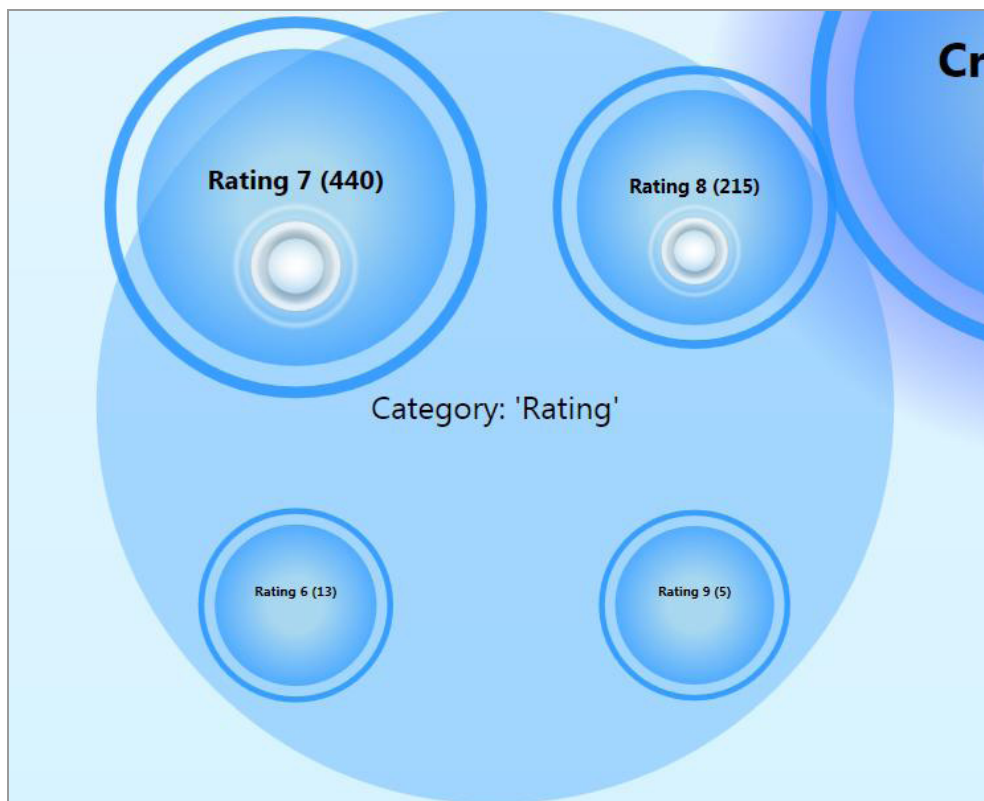


Abbildung 26: Mittige Darstellung der aktuellen Facette („Facet: Rating“) in textueller Form

Eine weitere Orientierungshilfe stellt die semantische Anordnung (vgl. Kapitel 4.3.2) der Genres und die daraus entstandene Bildung der Cluster dar. Ähnliche Genres liegen nahe beieinander und bilden Gruppierungen. Mit Hilfe dieser Zusammenführung der Genres ist es möglich, ähnliche oder verwandte Filme schneller finden zu können, da nicht erst die gesamte Informationslandschaft nach einem entsprechenden alternativen Genre abgesucht werden muss. Animationsfilme (Genre *Animation*) beispielsweise sind im Normalfall Filme für die gesamte Familie (wie z.B. von Disney oder Pixar). Demnach ist das Genre *Family* auch im gleichen Cluster vertreten und stellt eine schnell zu findende Alternative dar, in der ebenfalls interessante Filme enthalten sein könnten.

Durch eine farbliche Hervorhebung der einzelnen Cluster, ergibt sich für den Anwender ein zusätzlicher Mehrwert bezüglich der Orientierung. Die Hauptcluster konnten mit Hilfe des Dendrogramms, das durch Christian Rohrdantz' Berechnungen der semantischen Anordnung entstanden, nachvollzogen werden (vgl. Abbildung 27).

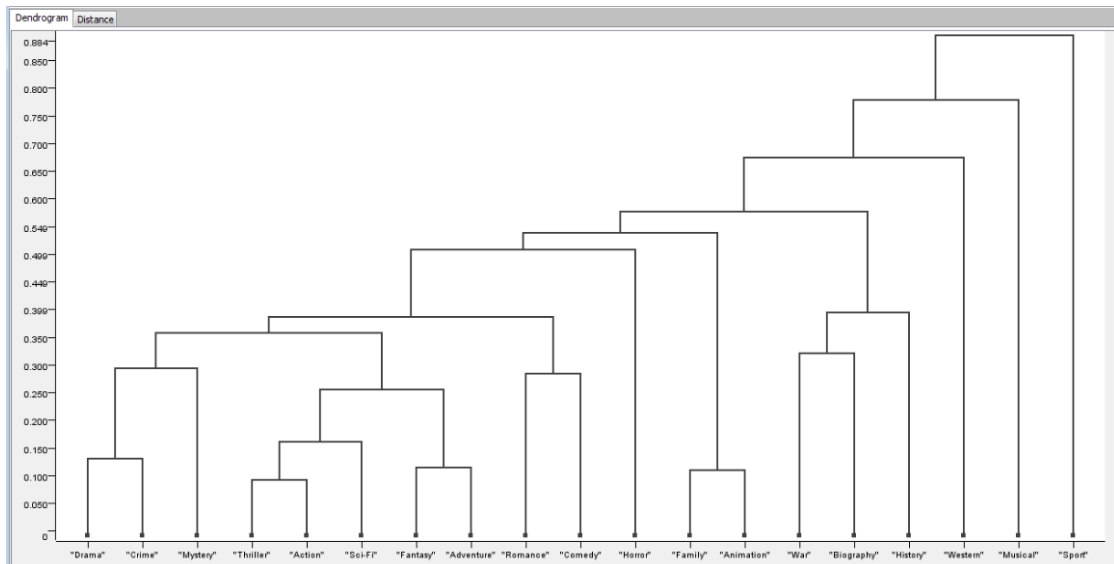


Abbildung 27: Dendrogramm mit Abbildung der gebildeten Hauptcluster

Um eine schnellere Identifikation im Falle des Prototypen *FacetBrowsing* zu gewährleisten, sind die Hauptcluster wie folgt gruppiert und farblich hervorgehoben:

Drama Crime Mystery
Thriller Action Sci-Fi Fantasy Adventure
Romance Comedy
Family Animation
War Biography History

Tabelle 2: Farbliche Hervorhebung der Hauptcluster

Die Genres Western, Horror, Musical und Sport gehören in keines der oben genannten Hauptcluster. Ebenso bilden sie aufgrund ihrer Entfernung zueinander keine eigene Gruppierung. Aus diesem Grund sind diese vier Genres in einem neutralen Farbton (lila) markiert und somit von den übrigen Clustern zu unterscheiden.



Western Horror Musical Sport

Tabelle 3: Farbliche Hervorhebung der Nebengener

Die farbliche Codierung der einzelnen Facetten innerhalb der Cluster ist durchweg konstant gehalten, das bedeutet, das bei jedem Schritt der Filterung und somit bei allen neu entstehenden Facetten die farbliche Markierung beibehalten wird. Diese konstante Einhaltung der Farbe dient ebenfalls der Orientierung bei tiefem Zoom. Abbildung 22 (Kapitel 4.3.1) zeigt dieses Verhalten grafisch auf.

5 FacetBrowsing: Implementation und Umsetzung

Im nun folgenden Umsetzungsteil, wird die Implementation der gerade beschriebenen Konzepte in den Prototypen *FacetBrowsing* aufgezeigt. Im Hinblick auf die definierten Forschungsfragen, ist auch dieses Kapitel untergliedert in die Punkte Navigation, austauschbare Zoomstufen, Clusterbildung und semantische Anordnung der Genres.

5.1 Navigation

Breadcrumb Navigation *FacetBrowsing* ermöglicht dem Benutzer, anhand des Breadcrumb Pfades, ein Navigieren innerhalb der bereits zurückgelegten Hierarchie. Diese Funktionalität erleichtert das Springen zu den Punkten innerhalb der bereits passierten Informationsmenge, um die Daten zu einem gewünschten Ziel einzuschränken. Um dieses Konzept der Navigation zu ermöglichen war es wichtig, nach jedem Filterschritt mit einem Attributwert die dazugehörige Facette zu speichern und dementsprechend ein Zurückspringen an diesen Punkt zu einem späteren Zeitpunkt zu ermöglichen.

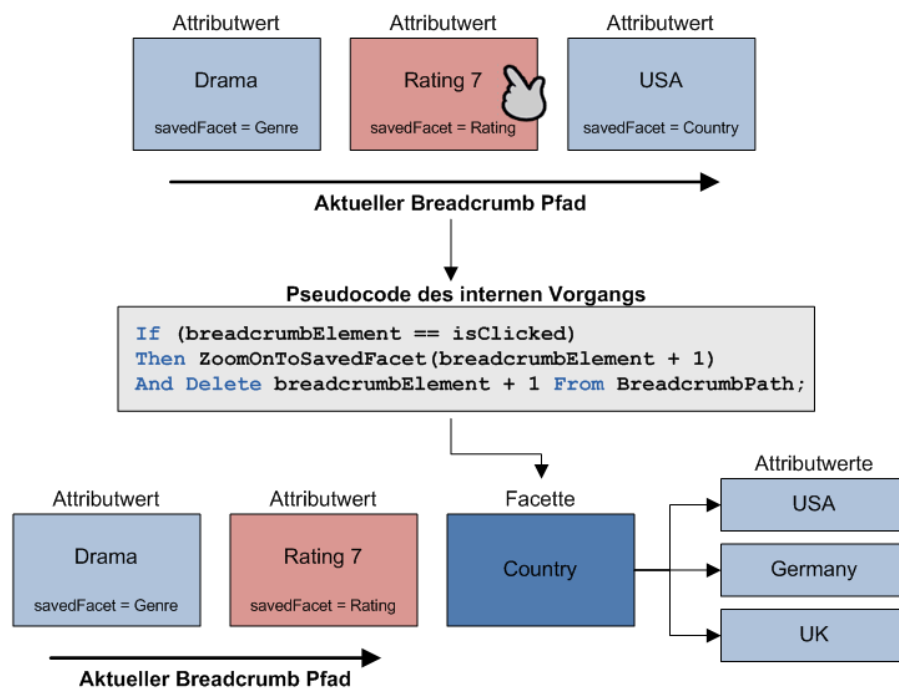


Abbildung 28: Breadcrumb Pfad zur Navigation durch die Historie

Abbildung 28 veranschaulicht beispielhaft den implementierten Ablauf einer Navigation mit Hilfe des Breadcrumb Pfades. Der Benutzer filtert zunächst die aktuellen Daten nach seinen Wünschen (Drama – Rating 7 – USA), während das System die jeweiligen Facetten dazu in einer Liste zwischenspeichert. Im genannten Beispiel entscheidet sich der Benutzer schließlich, die Informationsmenge nach einem anderen Land einzuschränken und wählt den Attributwert aus, der unmittelbar vor dem auszutauschenden

Eintrag „USA“ steht (in diesem Fall: Rating 7). Das System reagiert darauf, indem es zu der Facette zurückzoomt, in der die Filterung nach dem Land „USA“ zuvor stattgefunden hat. In der Folge wird der Eintrag aus dem Breadcrumb Pfad gelöscht und die entsprechenden Filtermöglichkeiten der Facette „Country“ erneut angeboten. Der Benutzer hat die Möglichkeit einen alternativen Weg durch die Daten zu wählen.

Der mittels Breadcrumb Pfad ausgelöste Zoom auf eine Facette unterscheidet sich nicht grundlegend von dem aus dem ZOIL Framework bekannten direkten Zoom auf ein Objekt. Die Objekte, die unmittelbar auf der Informationslandschaft verortet sind, reagieren unverzüglich auf eine direkte Interaktion, indem ein Zoom auf das jeweilige Objekt ausgelöst wird. Im Falle des Breadcrumb Pfades interagiert der Benutzer allerdings nicht direkt mit dem Objekt welches gezoomt werden soll, sondern wählt den indirekten Weg über einen Breadcrumb Eintrag. Um an dieser Stelle einen gezielten Zoom auslösen zu können ist die interne Methode - `ZoomOnto()` - aufzurufen, die das gewünschte Objekt übergeben bekommt und anschließend den Zoom auf der Landschaft auslöst (vgl Listing 1).

```
// „crumbFacet“ repräsentiert die Ziel-Facette
root.ZoomOnto(positionDic[crumbFacet.ToLower().Trim()], false, true);
```

Listing 1: Aufruf der Methode `ZoomOnto()` mit dem Objekt auf das gezoomt werden soll

Flower Menu Die zusätzliche Navigationstechnik, die eine individuelle Filterung der Daten ermöglicht und damit den Usability Prinzipien der Steuerbarkeit und Individualisierbarkeit folgt, stellt das *Flower Menu* dar. Jeder Attributwert innerhalb einer Facette besitzt ein solches Menü, um es dem Benutzer zu ermöglichen selbst zu entscheiden, nach welchen Kriterien als nächstes gefiltert werden soll. Zur Umsetzung des *Flower Menus* wurde das bereits im Surface SDK¹² bereitgestellte *ElementMenu*¹³ verwendet. Listing 2 zeigt den einfachen Einbau eines *ElementMenu* bzw. *Flower Menu* in *FacetBrowsing* mittels XAML-Datei.

```
<Viewbox x:Name="elementMenuViewBox" HorizontalAlignment="Center">
  <s:ElementMenu x:Name="elementmenu" ActivationMode="AlwaysActive"
    ContactDown="elementmenu_ContactTapGesture"
    SubmenuOpened="elementmenu_SubmenuOpened"/>
</Viewbox>
```

Listing 2: Implementation des Flower Menus mittels `ElementMenu`

Eine wichtige Funktion stellt der Befehl `ActivationMode="AlwaysActive"` dar. Dieser bedeutet, dass der Punkt des *Flower Menus*, der bei einer Berührung das Menü

¹² Surface SDK wird benötigt, um touchfähige Applikationen auf dem Microsoft Surface Tisch zu kreieren und zu testen

¹³ `ElementMenu` vorhanden seit der Veröffentlichung des Surface SDK Service Pack 1 (SP1) (11.5.2009)

aktiviert, immer sichtbar ist. Diese Einstellung wurde gewählt, damit der Benutzer die Anwesenheit des *Flower Menu* nicht übersieht. Einzelne Menüeinträge (`ElementMenuItems`) sind, wie Listing 2 erkennen lässt, nicht fest implementiert, sondern werden stetig berechnet. Diese Menüeinträge sollen dem Benutzer die Möglichkeit geben die Datenmenge im nächsten Schritt einzuschränken. Alle möglichen Filterkriterien sind zu Beginn der Anwendung innerhalb einer Liste gespeichert. Hat der Benutzer ein Element (z.B. *Country*) im *Flower Menu* gewählt und wurde in die entsprechende Facette gewechselt, soll dieses Element in weiteren Schritten nicht mehr zur Filterung angeboten werden, da dieser Filter bereits genutzt wurde (vgl. Abbildung 29).

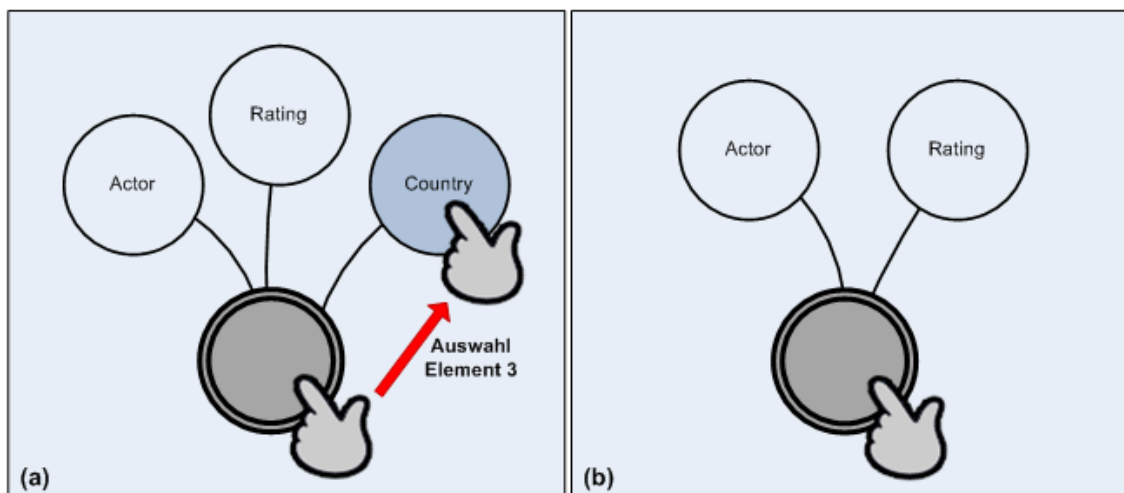


Abbildung 29: Flower Menu: Einschränkung der Auswahl

(a) Auswahl des Menüeintrages *Country* über ein *Flower Menu*. (b) Da dieser Eintrag bereits zur Filterung genutzt wurde, wird er in weiteren Schritten nicht mehr zur Verfügung stehen

Über eine ständige LinQ-Anfrage wird das aktuell ausgewählte Element aus der Liste aller noch vorhandenen Kriterien gelöscht (vgl. Listing 3). Anschließend wird die neu errechnete Liste der noch möglichen Filterkriterien in den folgenden Attributwerten gespeichert.

```
static public List<string> TranslatePossibleSpliCriterias
(List<string> AllSplitCriterias, string children_split_crit)
{
    IEnumerable<string> temp =    from splits in AllSplitCriterias
                                where splits != children_split_crit
                                select splits;
                                return temp.ToList();
}
```

Listing 3: Methode `TranslatePossibleSpliCriterias()`. Bereits verwendete Elemente werden aus der Liste gelöscht und im *Flower Menu* nicht mehr angeboten

5.2 Austauschbare Zoomstufen

Jeder Attributwert, beispielsweise „Rating 7“ oder „Rating 8“, innerhalb einer Facette besitzt die Möglichkeit, je nach eigener Größe semantisch umzuschalten. Dieses semantische Umschalten wird auch als „Wechsel in eine andere Zoomstufe oder Detailstufe“ bezeichnet und bietet die Möglichkeit weitere Informationen zum Objekt darzustellen oder, wie in *FacetBrowsing*, neue Filtermöglichkeiten bereitzustellen, um die aktuellen Daten weiter einzuschränken. Die Implementation des semantischen Zooms bei Anwendungen die auf dem ZOIL Framework basieren, verläuft anhand der vorherigen festen Einbindung des Codes direkt am Objekt in der .XAML-Datei¹⁴. Listing 4 zeigt eine typische beispielhafte Implementation eines Rechtecks (`Rectangle`) mit zwei vorher definierten Detailstufen (`ZComponentFrame`). Die erste Detailstufe (`WidthNeeded="0"`) repräsentiert das zu Beginn sichtbare Rechteck, welches zur Anzeige keine Mindestgröße benötigt. In die zweite Zoomstufe bzw. Detailstufe wird semantisch umgeschaltet, sobald, bedingt durch das Zoomen, die definierte Mindestgröße von 200 Pixeln (`WidthNeeded="200"`) erreicht ist. Aufgrund der zusätzlichen Textbox wird das Rechteck in dieser Anzeige mit zusätzlichen Informationen dargestellt.

```
// Erste Zoomstufe
<p:ZComponentFrame WidthNeeded="0" x:Name="ersteZoomstufe">
  <Grid>
    <Rectangle x:Name="Rechteck" Height="50" Width="100"/>
    <Label x:Name="lbl" Content="Beschriftung Rechteck"/>
  </Grid>
</p:ZComponentFrame>

// Zweite Zoomstufe
<p:ZComponentFrame WidthNeeded="200" x:Name="zweiteZoomstufe">
  <Grid>
    <Rectangle x:Name="Rechteck" Height="100" Width="200"/>
    <Label x:Name="lbl" Content="Beschriftung Rechteck"/>
    // Ergänzenden Textbox mit mehreren Informationen
    <TextBox x:Name="tbl" Content="Weitere Informationen"/>
  </Grid>
</p:ZComponentFrame>
```

Listing 4: Beispiel eines Rechteckes mit zwei fest implementierten semantischen Zoomstufen (ZOIL Framework)

Ein markanter Unterschied bei der Implementation des semantischen Zooms bei *FacetBrowsing* zu dem des ZOIL Frameworks besteht in der Unwissenheit, welche Zoomstufen bzw. Filtermöglichkeiten als nächstes dargestellt werden müssen. Der

¹⁴ XAML: **EX**tensible **A**pplication **M**arkup **L**anguage ist eine auf XML basierende Sprache, die dazu verwendet wird, grafische Elemente oder Benutzeroberflächen innerhalb einer WPF .NET Anwendung zu definieren.

Benutzer kann bei seiner Suche entweder durch das direkte Heranzoomen eines Facettenattributes den vom System vorgeschriebenen Weg der Filterung wählen, oder sich für den individuellen Pfad durch die Informationsmenge, mit Hilfe des *Flower Menu*, entscheiden. Bei der Umsetzung innerhalb des Prototypen *FacetBrowsing* ist eine vorherige feste Implementation des gesamten Inhaltes einer Zoomstufe folglich nicht möglich. Abbildung 30 veranschaulicht beispielhaft verschiedene Wege der Filterung durch unterschiedliche Attribute. Der Benutzer hat auf jeder Ebene (Facette) die Möglichkeit mehrere Richtungen der Datenfilterung einzuschlagen. Welchen Weg er letztendlich wählt ist vorher nicht bekannt.

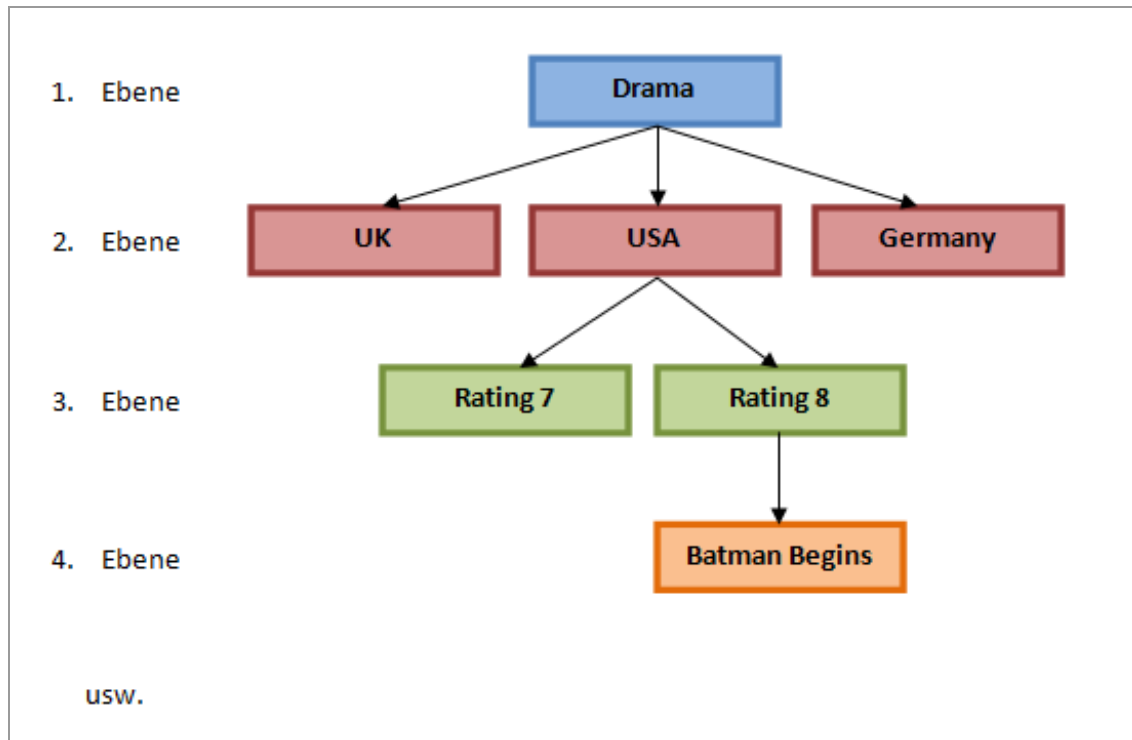


Abbildung 30: Beispielhafter Weg einer Filterung vom ersten Schritt bis hin zum eigentlichen Filmobjekt

Der Grundgedanke dieses Problem zu lösen, war die Idee die Detailansichten dynamisch und erst zur Laufzeit zu generieren, um auf die Bedürfnisse des Benutzers bei dem Filtern der Daten einzugehen. Der Hauptbestandteil dabei sind die Attributwert bzw. Merkmalsausprägungen, die sich innerhalb einer Facette befinden. Diese Attributwerte besitzen die Möglichkeit semantisch Umschalten und darauffolgend weitere Funktionalitäten anzuzeigen. Das Besondere an einem Filtervorgang bei *FacetBrowsing* ist die Verschachtelung mehrerer Attributwerte nacheinander, so dass der Benutzer die Möglichkeit hat, mehrmals hintereinander die Daten zu filtern. Dabei ist jeder Attributwert im Prinzip das gleiche Objekt – d.h. es wurde nur einmal implementiert –, wird allerdings rekursiv und mit den entsprechend unterschiedlichen Datengrundlagen erstellt und hinzugefügt.

Listing 5 zeigt die Implementation eines Attributwertes, welcher aus nur zwei unvollständig implementierten Zoomstufen besteht. In der ersten Detailstufe befinden sich

ein leeres Label und das *Flower Menu*, welches während des Zoomens mit den übrigen Filtermöglichkeiten gefüllt wird (vgl. Kapitel 5.1). Das Label beinhaltet Informationen über das aktuelle Objekt, wie z.B. der jeweilige Titel und die Anzahl der enthaltenen Daten (vgl. Abbildung 31). Die zweite Zoomstufe enthält anfangs keinerlei Informationen bezüglich weiterer Attributwerte bzw. Filtermöglichkeiten. Lediglich eine im Hintergrund liegende Facette, die die späteren Kategorien in Form einer Ellipse als zusätzliche Orientierungshilfe umrahmen soll, ist vorhanden. Ferner ist ein weiteres Label definiert, welches den späteren Namen der Facette darstellt. Als Container für die noch nicht vorhandenen Filterkategorien dient ein leeres UniformGrid.

```
<p:ZComponentFrame WidthNeeded="0" x:Name="firstZoomLevel">
  <Grid>
    <Ellipse x:Name="rootAttribute" .../>
    <Label x:Name="lblAttribute" Content="{Binding Path=Title,
      Elementname=myCloud}" .../>

    // Viewbox mit Flower Menu
    <Viewbox x:Name="vb_flowerMenu">
      <s:ElementMenu x:Name="flowerMenu" .../>
    </Viewbox>
  </Grid>
</p:ZComponentFrame>
```

Listing 5: Aufbau der ersten Zoomstufe eines Attributs bei FacetBrowsing

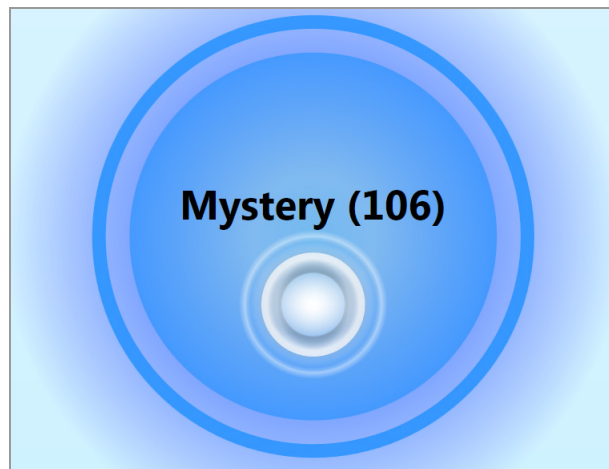


Abbildung 31: Beispielhafte Auswahl der ersten Zoomstufe des Attributwerts „Mystery“, basierend auf Listing 5

Während des Zoomens auf einen Attributwert wird dynamisch zur Laufzeit die neue Facette gekennzeichnet, sowie die entsprechenden Attribute erstellt und in den UniformGrid geladen. Auf diese Weise wird die zweite Zoomstufe eines Attributes zur Laufzeit befüllt. Da, wie bereits beschrieben, ein neuer Attributwert lediglich ein rekursiver Aufruf des gleichen Objektes darstellt, kann dieser Vorgang theoretisch unendlich weit angewandt werden. Daraus ergeben sich bei großen Datenmengen viele potenzielle Facetten und dementsprechend viele Filterschritte, um die Informationsmenge einzuschränken. Dieser rekursive Vorgang wird solange wiederholt, bis die Daten so weit eingeschränkt sind, bis die „50-Filme-Grenze“ (vgl. Kapitel 4.3.1) erreicht ist und es somit möglich ist die einzelnen Filmobjekte anzuzeigen.

Der Prozess der „austauschbaren Zoomstufen“ findet im Quellcode von *FacetBrowsing* anhand der Methode `BuildTree()` statt. Für den aktuellen Knoten im hierarchischen Baum, an dem sich der Benutzer befindet, werden zunächst die möglichen Kinderknoten während des Zoomens auf Basis des gewählten Filterkriteriums berechnet. Über eine LinQ-Anfrage werden aus der aktuellen Datenmenge die Top 3 Attribute bestimmt und zusammen den jeweilig enthaltenen Filmen in einer Liste zwischengespeichert. Attribute, welche nicht zu den Top 3 Werten gehören, werden zu einem „Misc“-Attribut¹⁵ zusammengefasst.

Abbildung 32 sowie Listing 6 sollen diesen Ablauf innerhalb des Systems anhand eines Beispiels verdeutlichen: Der Benutzer wählt in dem für ihn interessanten Genre „Action“ mit Hilfe des *Flower Menus* das „Country“ als nächsten Filterschritt aus. Während des Heranzoomens werden über eine LinQ-Anfrage die Top 3 Drehländer der aktuellen Filmliste im Genre „Action“ berechnet. Die restlichen Drehorte werden in einem extra Attribut „Misc“ zusammengefasst. Für diese vier neuen Attribute (die Top 3 und das Misc-Attribut) werden in der Folge neue Knoten im hierarchischen Baum erstellt. Diese Knoten speichern den Namen, die Größe, die Datenliste sowie noch mögliche Filterschritte. Im späteren Verlauf wird für jeden im Baum neu erstellten Knoten mittels der Methode `CreateZAttribute()` ein neuer Attributwert der zweiten Zoomstufe, genauer dem darin enthaltenen UniformGrid, hinzugefügt.

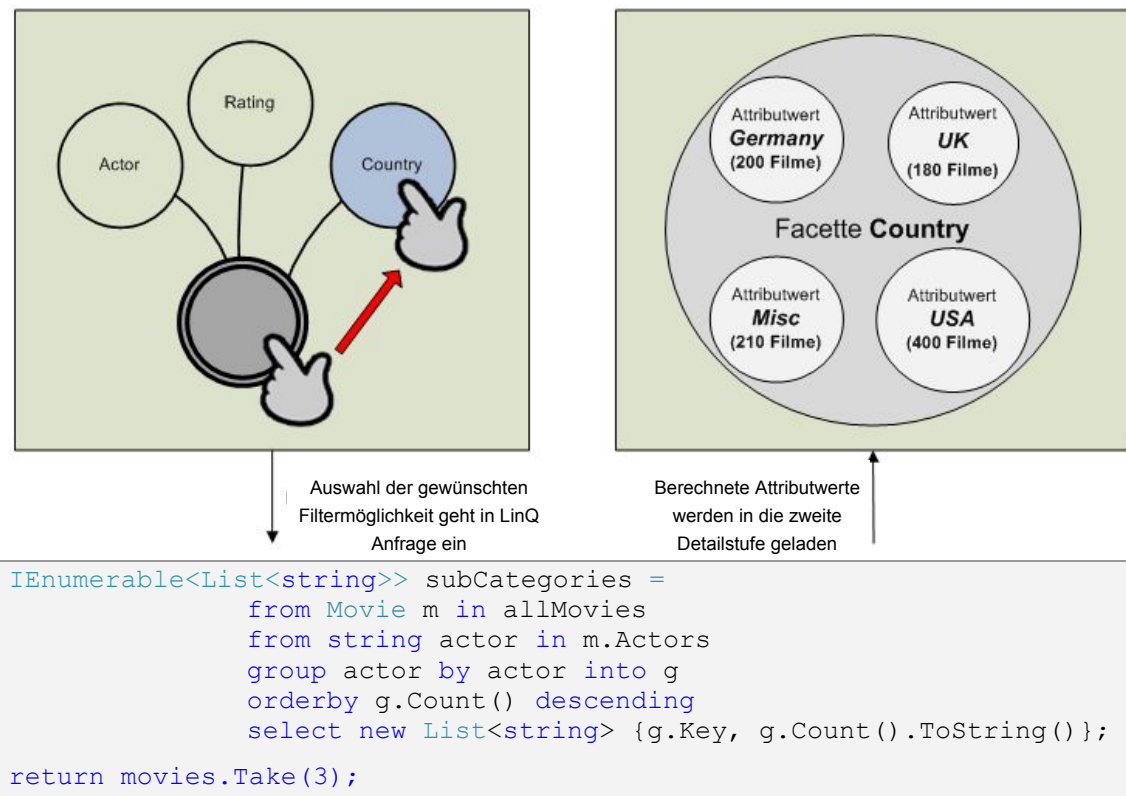


Abbildung 32: (oben) Ablauf einer Filterauswahl mit Berechnung der nächsten Attributwerte

Listing 6: (unten) Beispielhafte LinQ-Anfrage im Bezug auf Abbildung 32

¹⁵ „Misc“ entspricht dem engl. Wort „miscellaneous“ und bedeutet im dt. „Sonstiges“ / „Diverses“

5.3 Clusterbildung & semantische Anordnung

Die semantische Nähe und die Anordnung der Genre-Attribute nach ihrer Ähnlichkeit wurden, in einer Zusammenarbeit mit Christian Rohrdantz von der Arbeitsgruppe „Datenbanken, Datenanalyse und Visualisierung“ von Prof. Dr. Daniel A. Keim der Universität Konstanz, verwirklicht.

Die der Visualisierung zugrunde liegenden Informationen bestehen hauptsächlich aus Metadaten, die die einzelnen Filme auf verschiedene Art beschreiben, wie beispielsweise das Rating oder die mitwirkenden Schauspieler. Die für die Berechnung der semantischen Nähe wichtigen Metadaten, sind die vielen einzelnen Keywords, die jeden Film beschreiben. Diese Keywords werden herausgefiltert und dem jeweiligen Genre des Filmes zugewiesen. Die Zusammenhänge wurden anschließend von Christian Rohrdantz in einer „Genre-Keyword-Matrix“ gespeichert, wobei jede Zeile einem Genre und jede Spalte einem Keyword entspricht. Jeder Eintrag in dieser Matrix besagt, wie oft ein bestimmtes Keyword in einem bestimmten Genre vorkommt.

Die grundsätzliche Idee hierbei ist, dass verschiedene Genres sich inhaltlich ähnlich sind, wenn die jeweils gleichen Keywords häufig bzw. selten vorkommen. Kommt demnach beispielsweise das Keyword „Crash“ sehr häufig bei Actionfilmen sowie Dramafilmen vor, sind diese beiden Genres sich dementsprechend sehr ähnlich. Dahingegen ähneln sich Genres geringfügiger, je weniger Keywords übereinstimmen.

In der Folge wäre es möglich, die Ähnlichkeit zweier Genres anhand ihrer Feature-Vektoren in jeder Zeile der „Genre-Keyword-Matrix“ durch die Cosinus-Ähnlichkeit zu errechnen. Das Ergebnis wäre ein Wert, der anzeigt wie ähnlich sich zwei Genres sind. Allerdings besteht hierbei das Problem der hohen Dimensionalität der Vektoren und der Häufigkeit der 0-Werte. Aufgrund dessen wird versucht, eine Dimensionsreduktion durchzuführen, bei der die möglichst aussagekräftigen Dimensionen bewahrt und die restlichen verworfen werden. Um dies erreichen zu können, wird die „Latent Semantic Analysis“ (LSA) Methode verwendet, die generell die Ähnlichkeit zweier Dokumente aufgrund ihrer Terme bestimmen kann.

Die Grundlegende Funktionsweise der LSA Methode wird im Folgenden beschrieben:

5.3.1 LSA

Die Latent Semantic Analysis (LSA) ist ein Verfahren des Information Retrieval und wird generell angewandt, um durch den Einsatz statistischer Berechnungen aus einem großen Textkorpus die kontextbezogene Bedeutung von Wörtern herauszufiltern. Bedeutend ist dabei nicht das einzelne Wort, sondern die thematischen Wortbeziehungen, da Sachverhalte von Personen durch die unterschiedlichsten Wörter beschrieben und dargestellt werden. Daher beruht die Funktionsweise der „Latent Semantic Analysis“ auf der Idee, nach sprachlichen Mustern und Konzepten zu filtern, welche genau diese unterschiedlich genutzten Wörter und Begriffe repräsentieren [Landauer et al. 1998]. Demzufolge wird beispielsweise die Ähnlichkeit zweier Dokumente anhand ihrer Terme berechnet.

Die aus der Analyse resultierende Termfrequenz wird in einer Matrix dargestellt, welche sich nicht nur auf einen einzigen Text, sondern auch auf viele unterschiedliche Kapitel oder sogar ganze Teile des Internets beziehen kann. [Landauer et al. 1998]. Die in der Matrix aufgezeigte Häufigkeit der enthaltenen Thematiken innerhalb eines Textkorpus, dient als Grundlage für die Darstellung der Termfrequenzen in unterschiedlichen Systemen durch eine algorithmische Anordnung.

So liegen auch für das hier beschriebene *FacetBrowsing* die durch die Latent Semantic Analysis gewonnenen Daten zu Grunde. Im Falle des *FacetBrowsing* Prototypen wird dementsprechend die Ähnlichkeit zweier Genres anhand ihrer Keywords berechnet.

Anhand der Ergebnisse der LSA Methode wurde innerhalb des Prototypen der Spring Layout Algorithmus (SLA) zur Anordnung der Genres verwendet. Die Beschreibung und Funktionalität in *FacetBrowsing* wird im folgenden Unterkapitel aufgezeigt.

5.3.2 SpringLayout

Der Spring Layout Algorithmus (SLA) ist eine Form der „kräfte-basierenden-Algorithmen“. Er ist eine effiziente Methode ein ungerichtetes Netzwerk zu visualisieren. Hierbei werden die einzelnen Kanten der Knoten als Spiralfedern interpretiert.

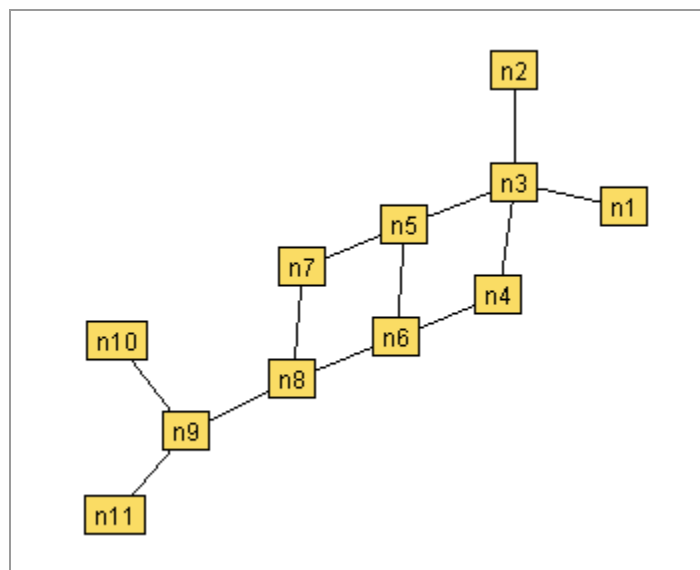


Abbildung 33: Darstellung eines mittels SLA erstellten Netzwerks

Eine Feder bzw. Kante zwischen zwei Knoten u und v hat eine natürliche Länge l_{uv} und versucht diese ständig einzuhalten. Wenn die Kante kürzer bzw. größer ist als die natürliche Länge, versuchen die Federn sich zusammen bzw. auseinander zu ziehen, um so den richtigen Abstand wieder herzustellen [Pesch 2009]. Darüber hinaus ist der Spring Layout Algorithmus ein erfolgreicher Weg, Überlappungen einzelner Knoten zu verhindern [Li et al. 2006]. Abbildung 33 zeigt ein beispielhaftes Netzwerk, welches durch einen Spring Layout Algorithmus visualisiert wurde.

Längere Recherchen, diesen Algorithmus in einem verwendbarem Format für den Prototypen *FacetBrowsing* in WPF C# zu finden, schlugen fehl. In der Folge wurde er in

Eigeninitiative aus einem Java Beispiel in WPF C# .NET 3.5 portiert und lauffähig gemacht. Dabei wurden die Hauptfunktionen und Berechnungsmethoden übernommen und entsprechend angepasst.

Der Algorithmus verlangt zu Anfang einen Input String, der sämtliche zu verbindenden Knoten (Genres) sowie deren Abstand zueinander enthält (vgl. Listing 7). Der Abstand der einzelnen Knoten ergibt sich aus der Ergebnismatrix von Christian Rohrdantz, die die Ähnlichkeitswerte aller Genres zueinander beinhaltet (vgl. Abbildung 34). Je kleiner der Wert bzw. je näher der Wert Richtung Null geht, desto ähnlicher sind sich die Genres. Dementsprechend ungleicher sind sich Genres mit einem höheren Wert. Für den Input String wird dieses Ergebnis mit 1000 multipliziert und als Entfernungsmaß in Pixeln verwendet. Beispielsweise entspricht der Ähnlichkeitswert von 0,10656 zwischen dem Genre „Action“ und „Adventure“ auf der Informationslandschaft einem Abstand von ca. 106 Pixeln.

```
String splitString = "Action-Adventure/M106, Action-Animation/M516,..."
```

Listing 7: Input String mit zu verbindenden Knoten und Abstand (hier u.a. Action - Adventure mit einem Abstand von 106px)

	Action	Adventure	Animation	Biography	Comedy	Crime	Drama	Family	Fantasy	History
Action	0	0,10656	0,51686	0,50794	0,37403	0,26680	0,27115	0,51322	0,24451	0,45268
Adventure	0,10656	0	0,38298	0,58687	0,36035	0,46759	0,40168	0,35187	0,12146	0,51609
Animation	0,51686	0,38298	0	0,71356	0,39016	0,67298	0,59275	0,11694	0,45678	0,69957
Biography	0,50794	0,58687	0,71356	0	0,58815	0,46817	0,33875	0,72363	0,63972	0,46872
Comedy	0,37403	0,36035	0,39016	0,58815	0	0,37258	0,31631	0,44435	0,46101	0,62389
Crime	0,26680	0,46759	0,67298	0,46817	0,37258	0	0,13774	0,69468	0,60383	0,60328
Drama	0,27115	0,40168	0,59275	0,33875	0,31631	0,13774	0	0,61266	0,51985	0,49392
Family	0,51322	0,35187	0,11694	0,72363	0,44435	0,69468	0,61266	0	0,36853	0,69658
Fantasy	0,24451	0,12146	0,45678	0,63972	0,46101	0,60383	0,51985	0,36853	0	0,57334
History	0,45268	0,51609	0,69957	0,46872	0,62389	0,60328	0,49392	0,69658	0,57334	0

Abbildung 34: Ausschnitt der Ergebnismatrix (nXn) mit allen Genres und ihren Ähnlichkeitswerten

Aus den übergebenden Daten des Input Strings wird errechnet, wie viele Kanten im Gesamten erstellt werden müssen. Die Kanten sind dabei die Abstandshalter zwischen zwei Genre Attributen. Im Anschluss suchen sich die Kanten selbstständig die Objekte die sie verbinden sollen, indem sie die einzelnen Genres suchen (vgl. Listing 8).

Anschließend werden die Knoten sowie ihre Kanten in einer Liste abgespeichert, auf die der Algorithmus beim Anordnen der Genres zugreift.


```
void addEdge(String from, String to, int len)
{
    Edge e = new Edge();
    e.from = findNode(from);
    e.to = findNode(to);
    e.lengthEdge = len / 1.5;
    _edges.Add(e);
}
```

Listing 8: Methode „addEdge()“, die Kanten erzeugt und die zwei zu verbindenden Knoten für diese Kante selber über den Aufruf „findNode()“ sucht.

Der Spring Layout Algorithmus ist ein zufällig arbeitender Algorithmus, d.h. das entstandene Netzwerk sieht bei jedem Durchgang, trotz der Berücksichtigung der jeweiligen Abstände der Knoten, unterschiedlich aus.

Im Falle von *FacetBrowsing* ist allerdings eine einzige Darstellung mit korrekten Abständen ausreichend, da unterschiedliche Ausführungen verwirrend erscheinen können. Der Benutzer müsste sich jedes Mal erneut auf der Landschaft zurechtfinden und die Positionen der Lieblingsgenres ausfindig machen. Damit beim Start der Anwendung dieses Problem nicht auftritt und der Algorithmus nicht jedes Mal neu durchlaufen werden muss, wurden die X- und Y-Positionen der Facetten nach dem Beenden des SLA abgespeichert. Beim Laden von *FacetBrowsing* werden diese X- und Y-Positionen für jeden Genre-Attributwert ausgelesen und so an die richtige Position auf der Informationslandschaft gesetzt.

6 Fazit und Ausblick

6.1 Rückblick

Zu Beginn dieser Arbeit wurden verschiedene Forschungsfragen definiert, die aus den aktuellen konzeptionellen und technischen Grenzen des ZOIL Frameworks und Medio-Vis 2.0 entstanden sind. Es sollte untersucht werden, ob es eine Möglichkeit gibt, diese Problematiken anhand einer zoom- und facettenbasierten Visualisierung zu verbessern oder sogar zu lösen. Dabei wurden verschiedene Konzeptideen anhand der Forschungsfragen entwickelt und mit Hilfe von wissenschaftlichen Studien sowie Forschungsergebnissen untersucht.

Zur besseren Veranschaulichung werden die in der Einleitung definierten Forschungsfragen erneut aufgezeigt:

- (1) Ist es möglich durch eine zoom- und facettenbasierende Visualisierung die vorhandenen technischen Grenzen, sowie die aktuelle Anordnung von Objekten bei großen Datenmengen zu verbessern bzw. zu lösen?*
- (2) Welche Möglichkeiten gibt es, die Orientierung des Benutzers bei einer zoom- und facettenbasierenden Visualisierung zu gewährleisten?*
- (3) Existieren alternative Navigationsmöglichkeiten zu der herkömmlichen Zoom und Pan Navigation?*

6.2 Ergebnisse

Die Forschungsfragen werden nun im Hinblick auf die Resultate der vorliegenden Arbeit beantwortet.

- (1) Ist es möglich durch eine zoom- und facettenbasierende Visualisierung die vorhandenen technischen Grenzen, sowie die aktuelle Anordnung von Objekten bei großen Datenmengen zu verbessern bzw. zu lösen?*

Die Facettenklassifikation liefert im Allgemeinen sehr gute Ergebnisse bei der Handhabung vieler Informationen und bietet gleichzeitig ein enormes Potenzial für Visualisierungen die geeignet sind, Daten für eine Person anschaulich und leicht verständlich zu präsentieren. Benutzer haben ebenso die Möglichkeit, den Informationsraum aus verschiedenen Perspektiven zu betrachten. Des Weiteren hilft das Explorieren der Beziehungen verschiedener Facetten, um beim Benutzer das Verständnis für die verwendeten Daten zu vertiefen und dabei eventuell sogar neue Erkenntnisse zu gewinnen. Forschungsergebnisse und Studien haben ebenso gezeigt, dass Personen bei der Ver-

wendung einer facettierten Visualisierung eine erhöhte Vertrautheit mit den Dateninhalten aufweisen und dementsprechend größere Sucherfolge sowie zufriedenstellendere Ergebnisse erzielen.

Der facettierte Visualisierungsansatz bei *FacetBrowsing* gruppiert die Attributwerte verschiedener Metainformationen aus den zugrundeliegenden Filmdateien in Facetten, und ermöglicht so dem Benutzer der Anwendung ein übersichtliches Navigieren durch den Informationsraum. Einzelne Filmobjekte liegen nicht mehr im Gesamten auf der Informationslandschaft (vgl. MedioVis 2.0), sondern werden dem Benutzer erst präsentiert, nachdem er – mit Hilfe der Facetten und ihren Attributwerten – genügend Filme aus der Datenmenge herausgefiltert hat. Somit wird die konzeptionelle Grenze der Überlappung und Verdeckung von einzelnen Objekten verhindert und gewährleistet eine bessere Übersicht bei einer großen Anzahl an Daten. Gleichzeitig wird aufgrund der späten Darstellung der Filmobjekte verhindert, dass seitens des .NET Frameworks bei jeder Interaktion (Zoomen bzw. Pannen) ressourcenlastig gerendert werden muss, da dem Benutzer zu keinem Zeitpunkt mehr als 50 Objekte angezeigt werden. Die aktuelle technische Grenze, wie beispielsweise die träge Reaktion bei vielen Objekten aufgrund des ständigen Renderings von vielen Textpassagen, wird somit umgangen und erlaubt ein flüssigeres Arbeiten.

(2) Welche Möglichkeiten gibt es, die Orientierung des Benutzers bei einer zoom- und facettenbasierenden Visualisierung zu gewährleisten?

Orientierungspunkte sowie verschiedene Hilfsmittel beim Zurechtfinden in großen Datenmengen spielen im Zusammenhang mit Visualisierungen eine nicht zu vernachlässigende Rolle. Bedingt durch das Zoomen kann es vor allem in einem zoombaren User Interface (ZUI) aufgrund des Verschwindens so genannter Landmarken aus dem eigenen Sichtfeld zu Orientierungsschwierigkeiten kommen. Die Folge ist, dass die eigene Position innerhalb der Visualisierung oftmals aufgrund einer fehlerhaften mentalen Repräsentation nicht mehr bestimmt werden kann. Verschiedene Forschungsansätze, wie beispielsweise eine Breadcrumb Navigation, versuchen hier Abhilfe zu schaffen. Studien haben gezeigt, dass ein angebotener Breadcrumb Pfad eine signifikante Verbesserung der Benutzerfreundlichkeit aufzeigt und dementsprechend von Benutzern honoriert wird. Der Breadcrumb Pfad repräsentiert den vom Benutzer zurückgelegten Weg durch die Datenmenge und hilft auf diese Weise, die Orientierung auch bei tieferem Zoom zu erhalten. Andere Ansätze der Forschung, wie die Bildung von Clustern, haben ebenfalls einen Einfluss auf die Orientierung und die Übersicht des Benutzers. Mit Hilfe des Clustering besteht potenziell die Möglichkeit, neue Tendenzen sowie Un-erwartetes innerhalb großer Datenmengen zu entdecken. Weiterhin besagen Studien, dass Cluster in Bezug auf die Verdeutlichung und Hervorhebung dominanter Motive innerhalb eines Informationsraumes sehr nützlich sind.

FacetBrowsing veranschaulicht dem Benutzer mit Hilfe eines Path Breadcrumb Pfades den exakten Weg, welchen er eingeschlagen hat, um die aktuelle Datenmenge zu filtern bzw. zu reduzieren. Dabei wurde darauf Wert gelegt, dass diese Art der Orientierungshilfe zu jedem Zeitpunkt sichtbar und dementsprechend auffallend auf dem Head-Up Display (HUD) hinterlegt ist. Ebenso die Größe des HUD und demnach auch die Größe des Breadcrumb Pfades tragen dazu bei, dass der Benutzer jederzeit ohne Probleme seinen zurückgelegten Weg sowie den aktuell ausgeführten Filterschritt nachvollziehen kann. Selbst bei einer fehlerhaften mentalen Repräsentation des Informationsraumes kann sich der Benutzer während des Zoomens anhand dieser Informationen orientieren.

Zusätzliche Orientierungs- und Übersichtshilfen innerhalb des Prototypen sind die semantische Anordnung der Genre-Attributwerte und die daraus entstandenen Cluster. Ähnliche Genres liegen bei diesem Konzeptansatz nah beieinander und bilden auf diese Weise gemeinsame Cluster, welche farblich hervorgehoben sind. Somit können auf der einen Seite schneller interessante Alternativen zu den bisherigen Lieblingsgenres gefunden werden. Auf der anderen Seite fördert die farbliche Unterscheidung der Cluster ebenso die Übersicht und Orientierung, da die farbliche Hervorhebung der einzelnen Cluster bis auf die letzte Zoomstufe, in der letztendlich die einzelnen Filmobjekte angezeigt werden, gleich ist. Der Benutzer hat auf diese Weise selbst bei einem tiefen Zoom noch einen Anhaltspunkt, in welchem Cluster er sich gerade befindet.

Die lineare Skalierung der Größe einzelner Attributwerte innerhalb einer Facette unterstützt des Weiteren das bessere und schnellere Erkennen von Objekten mit vielen Daten.

(3) *Existieren alternative Navigations- bzw. Interaktionsmöglichkeiten zu der herkömmlichen Zoom und Pan Navigation?*

Alternative Navigationsmethoden bzw. Interaktionsmöglichkeiten, die ebenfalls in einem zoombaren User Interface (ZUI) funktionieren und für den jeweiligen Benutzer verständlich sind, existieren in mehreren Forschungsansätzen. Beispielsweise werden in der heutigen Zeit zunehmend so genannte *Marking Menus* eingesetzt, um den Benutzer ein anschauliches und übersichtliches Funktionsangebot zu präsentieren. Ein noch recht neuer Ansatz in diesem Forschungsgebiet sind *Flower Menus*, die gegenüber herkömmlichen *Marking Menus* (wie z.B. dem *Pie Menu*) von der Eigenschaft profitieren, über mehrere Ebenen hinweg eine sehr hohe Anzahl von Elementen innerhalb des Menüs anzeigen zu können. So ist eine theoretische Anzahl von 56 Elementen pro Ebene darstellbar. Eine weitere Besonderheit des *Flower Menus* ist die mögliche Unterstützung von gezielten Gesten, um bestimmte Menüpunkte schneller auswählen zu können. Neben dem Vorteil der schnelleren Menüauswahl wurde ferner das bessere und intuitivere Verhalten der Benutzer beim Erlernen dieser Technik in einer Studie bestätigt.

Aufgrund der vorteilhaften höheren Anzahl an möglichen Menüpunkten und der intuitiveren Steuerung wurde das *Flower Menu* als eine alternative Navigations- sowie Interaktionstechnik für *FacetBrowsing* gewählt. Die große Anzahl an Metainformationen, die einen Film beschreiben, und die Möglichkeit diese vielen Informationen als Filter über ein solches Menü dem Benutzer zu präsentieren, unterstützen diese Wahl. Das *Flower Menu* ist bei *FacetBrowsing* direkt am Objekt verortet und bietet dem Benutzer eine alternative Interaktionsmöglichkeit. Neben den herkömmlichen Zoom und Pan Gesten kann der Benutzer mit Hilfe des *Flower Menus* selbst entscheiden, nach welcher Metainformation er als nächstes filtern möchte. Sobald ein Menüpunkt ausgewählt wird, fungiert das Menü ebenfalls als alternative Navigationsmöglichkeit, in dem der jeweilige Attributwert innerhalb der Facette heran gezoomt wird und die Auswahl des *Flower Menus* als neue Filtermöglichkeit zur Verfügung steht.

6.3 Ausblick

Die in dieser Arbeit vorgestellten Konzepte stellen eine funktionale Basis dar, um den Einsatz von *FacetBrowsing* als alternative Visualisierung innerhalb von MedioVis 2.0 zu ermöglichen. Dennoch bedarf es einer tiefer gehenden Analyse, um einzelne Aspekte der Konzepte zu optimieren.

Attributwerte bzw. Merkmalsausprägungen Um die *Auswahlmöglichkeit an Attributwerten* zur Datenfilterung innerhalb einer Facette flexibler zu gestalten, sollte eine platzoptimierte Darstellung bedacht werden. Auf diese Weise könnten dem Benutzer mehr als die bisherigen vier möglichen Merkmalsausprägungen zur Verfügung stehen, um die Datenmenge einzuschränken. Ebenso könnte eine intensivere Benutzerbefragung zeigen, nach welchen Kriterien ein Film gesucht bzw. gefunden wird. Anhand der Ergebnisse könnten speziell im *Flower Menu* Einträge ergänzt sowie ausgetauscht werden, um eine höhere Zufriedenheit bei der natürlichen Suche des Benutzers zu ermöglichen.

Semantische Anordnung einzelner Filmobjekte Neben der semantischen Anordnung der Genre-Attribute wäre das Konzept der semantischen Abbildung der einzelnen Film Objekte auf der untersten Zoomstufe bzw. nach dem letzten Filtervorgang, eine denkbare Erweiterung und zusätzliche Unterstützung für den Benutzer. Das Wesentliche hierbei wäre die Darstellung dieser Objekte nach ihrer Relevanz innerhalb des entsprechenden Genres. Mit Hilfe der Schlüsselwörter eines Filmes könnte errechnet werden, wie repräsentativ dieser für ein bestimmtes Genre ist. Befindet sich also ein Film innerhalb eines Genres, das nicht seinem Hauptgenre entspricht, wäre dieser Film für die gewählte Kategorie nicht maßgebend. Zur Veranschaulichung wird dieser Fall im Folgenden anhand eines kurzen Beispiels erläutert:

- Der Film „*Lethal Weapon 2*“ aus dem Jahre 1989 ist mit den Genres „Action“, „Thriller“, „Crime“ und „Comedy“ getagt¹⁶.
- Obwohl der Film einige lustige Passagen enthält, ist er in erster Linie nicht als reine Komödie zu sehen. „Action“ und „Crime“ sind hier die ausschlaggebenden Genres.
- Anhand der Keywords innerhalb des Filmes kann nun ein Wert errechnet werden, der veranschaulicht, ob der Film repräsentativ für das jeweilige Genre ist.
- In der untersten Zoomstufe bzw. nach dem Erreichen der „50-Filme-Grenze“ (vgl. Kapitel 4.3.1) wird der Film „*Lethal Weapon 2*“ innerhalb des Genres „Comedy“ so angeordnet, dass sofort ersichtlich ist, dass der Film nicht als reine Komödie zu sehen ist und somit nicht als charakterisierend gilt.

Dem Beispiel zufolge wäre das Genre „Action“ bzw. das Genre „Crime“ als repräsentatives Hauptgenre und „Comedy“ eher als „Untergenre“ zu betrachten. Das Problem der eindeutigen Genredefinition liegt oft darin, dass Film-Thematiken, wie beispielsweise die Handlung, als Grundlage zur Genre-Bestimmung dienen. Bereits 1989 vertritt David Bordwell die Auffassung, dass dieses Vorgehen unzureichend ist. Er schlussfolgert, dass Handlungen deswegen unzureichende Grundlagen sind, da jedes Thema in jedem Genre vorkommen kann [Bordwell 1989]. Desweiteren fügt er hinzu, dass „keine Anordnung zwingender und hinreichender Bedingungen ein Genre derart klar von Gruppierungen anderer Art so abgrenzen kann, dass alle Experten oder das gewöhnliche Filmpublikum dem zustimmen können“.

Filme zeigen demnach oft Konventionen von mehr als einem Genre. Hier unterscheiden zu können, welches Genre am wesentlichsten für einen Film ist, stellt sich als schwierig heraus. Ebenso zu erkennen, ob ein Film im Genre „Comedy“ eine reine Komödie ist, oder eher als Actionfilm anzusehen wäre, ist im Normalfall ohne das Anschauen des Filmes oder des dazugehörenden Film-Trailers schwierig. Selbst das Lesen der Zusammenfassung bzw. der Handlung des Filmes hilft nicht immer, da, wie John Hartley anmerkt, „der gleiche Text zu verschiedenen Genres gehören kann“ [Hartley 2002].

Hinsichtlich dieses Problems könnte innerhalb von *FacetBrowsing* versucht werden, dem Benutzer aufgrund der semantischen Anordnung das Identifizieren eines repräsentativen Filmes zu erleichtern und somit das Suchen und Finden des richtigen Filmes zu beschleunigen.

Flower Menu Das vom Surface SDK angebotene ElementMenu, welches innerhalb des Prototypen *FacetBrowsing* als *Flower Menu* dient, bietet zwar eine gute und alternative Navigations- und Interaktionsmöglichkeit, weist allerdings auch Probleme auf. Ein Problem besteht darin, dass sich die Position und die Richtung der Pfade zu einem

¹⁶ Der Film *Lethal Weapon 2* in der IMDb: <http://www.IMDb.com/title/tt0097733/>

Menüeintrag nicht beeinflussen lassen. Je nach Position des Attributwertes kann es vorkommen, dass Menüeinträge über den Rand der Visualisierung hinausragen und damit für den Benutzer nicht komplett sichtbar sind. Ein weiteres Manko stellt die schwierige Skalierbarkeit des ElementMenus dar. Je nachdem, wie viele Daten innerhalb eines Attributwertes vorhanden sind, desto größer werden diese anhand der logarithmischen Skalierung dargestellt. Das ElementMenu passt sich automatisch an diese Größe an. Die sich daraus ergebende Problematik stellt die resultierende Größe des ElementMenus dar. Je nach Größe des Attributwertes kann das ElementMenu unverhältnismäßig groß respektive klein dargestellt werden.

Hier könnten weitere Nachforschungen alternative Ansätze eines *Flower Menus* in C# WPF aufzeigen, die zum Zeitpunkt der Implementation noch nicht vorhanden waren und eine bessere Individualisierbarkeit aufweisen.

Vergleichen verschiedener Interessengebiete Gewisse Situationen verlangen die Möglichkeit, unterschiedliche Bereiche der Informationslandschaft zu vergleichen. Das Betrachten zweier Genres in Bezug auf die Verteilung des Ratings (Welches Genre hat prozentual mehr sehr gute Filme hervorgebracht?) könnte eine solche Situation darstellen. Hier wäre es denkbar, einen multifokalen Ansatz anzubieten, wie er in der Bachelor Arbeit von Jonas Schweizer¹⁷ vorgestellt wurde. Ein derartiger Split Screen könnte es ermöglichen, die Landschaft in mehrere Abschnitte zu teilen und separat die gewünschten Vergleichsobjekte heran zu zoomen.

¹⁷ Nachzulesen unter: http://hci.uni-konstanz.de/downloads/BA_JonasSchweizer.pdf

Glossar

.NET Frameworks: Das .NET Framework bezeichnet eine von Microsoft entwickelte Software-Plattform, bestehend aus einer Laufzeitumgebung, einer Sammlung von Klassenbibliotheken, Programmierschnittstellen und Dienstprogrammen. Version 3.5 des Frameworks ist die, zum Zeitpunkt der Umsetzung des Prototypen *FacetBrowsing*, verwendete.

Attributwert / Merkmalsausprägung: Bezeichnet innerhalb des Prototypen *FacetBrowsing* eine Filtermöglichkeit, die innerhalb einer Facette angeordnet ist. Mit Hilfe eines Attributwerts oder einer Merkmalsausprägung kann der Benutzer die Datenmenge eingrenzen.

Dendrogramm: Ein Dendrogramm wird oft dazu verwendet, Beziehungen zwischen Clustern darzustellen. Ein Dendrogramm zeigt dabei die multidimensionalen Abstände zwischen Objekten in einer baumähnlichen Struktur an.

Facette: Eine Facette ist definiert als Oberbegriff, der die Bestandteile (Inhalte) sowie deren gewichteten Formen, Begriffe und Zahlen eines zusammengesetzten Fachgebiets kennzeichnet.

Landmarken: Landmarken sind auffälligen Objekte oder Merkmale der eigenen Umgebung, welche diese strukturieren und der Orientierung dienen können.

LinQ-Anfrage: LinQ (**L**anguage **I**ntegrated **Q**uery) ist eine Komponente des Microsofts .NET-Frameworks zur Abfrage von Datenquellen wie Datenbanken oder XML-Dateien

Panen: Das Panning ist eine Funktion, um einen Bildausschnitt zu verschieben (X- und Y- Richtung)

Skaleninvarianz: Skaleninvarianz beschreibt die Eigenschaft eines Zustands oder Vorgangs, bei dem auch bei der Veränderung der Betrachtungsgrößen (Skalierung) die Eigenart oder Charakteristik inklusive seiner Werte weitestgehend exakt gleich bleibt.

User Interface: Benutzerschnittstelle, die es Menschen erlaubt Systeme zu bedienen

Visualisierung: Veranschaulichung bzw. Sichtbarmachen von Daten

XAML: XAML (**E**xtensible **A**pplication **M**arkup **L**anguage) ist eine auf XML basierende Sprache, und ermöglicht es innerhalb von WPF deklarativ Benutzerschnittstellen zu programmieren, ohne dabei über Kenntnisse einer üblichen Programmiersprache zu verfügen.

Zoomen: Das Zooming ist eine Funktion, die es ermöglicht, einen Bildausschnitt oder ein Objekt zu vergrößern bzw. zu verkleinern.

Windows Presentation Foundation (WPF): WPF ist ein Softwareframework innerhalb des .NET Frameworks der Firma Microsoft. WPF integriert dabei XAML und übliche Programmiersprachen des .NET Frameworks.

Literaturverzeichnis

- [Abercrombie 1996]: Abercrombie, N.: Television and Society. Cambridge: Polity Press, Seiten 43-45, 1996
- [Adkissen 2002]: Adkisson, H.: Identifying De-Facto Standards for E-Commerce Web Sites. University of Washington, Master Thesis, 2002
- [Andrews 2009]: Andrews, K.: Information Visualisation. Graz University of Technology, Course Notes, 2009
- [Amazon 2010] Online-Shopping. Zugriff am 10.03.2010 unter <http://www.amazon.com>
- [Bailey et al. 2008]: Bailey G., Lecolinet E., Nigay L.: Flower Menus: A New Type of Marking Menu with Large Menu Breadth, Within groups and Efficient Expert Mode Memorization. AVI'08, 28-30 May , 2008, Napoli, Italy, 2008
- [BaseX 2010]: Base X: Processing and Visualizing XML with a native XML Database. Zugriff am 31.03.2010 unter <http://www.inf.uni-konstanz.de/dbis/basex/index>
- [Betriebsraum 2009]: Extremely Efficient Menu Selection: Marking Menus for the Flash Platform, Zugriff 26.4.2010 unter: <http://www.betriebsraum.de/blog/2009/12/11/extremely-efficient-menu-selection-marking-menus-for-the-flash-platform/>
- [Blustein et al. 2005]: Blustein, J., Ahmed, I., Instone, K.: An Evaluation of Look-ahead Breadcrumbs for the WWW. In: HT'05, September 6–9, 2005, Salzburg, Austria, 2005
- [Bonino et al. 2009]: Bonino D., Corno F., Farinetti L.: FaSet: A Set Theory Model for Faceted Search. IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, 2009
- [Bordwell 1989]: Bordwell, D.: Making Meaning: Inference and Rhetoric in the Interpretation of Cinema, Seite 147, Cambridge, MA: Harvard University Press, 1989
- [Bowler 2001]: Bowler, D., Ng, W., and Schwartz, P.: Navigation bars for hierarchical websites. Retrieved 01/20/03 from University of Maryland, Student HCI Online Research, 2001
- [Card et al. 2005]: Card, S., Heer, J., Landay, J.: prefuse: a toolkit for interactive information visualization. CHI 2005, April 2–7, 2005, Portland, Oregon, USA, 2005
- [Chandler 2000]: Chandler, D.: An Introduction to Genre Theory, 2000. Zugriff am 02.04.2010 unter http://www.aber.ac.uk/media/Documents/intgenre/chandler_genre_theory.pdf
- [Cutting et al. 1992] Cutting D., Kargerl D., Jan O. Pedersen J., Tukey J.: Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. 15th Ann Int'l SIGIR '92 / Denmark-6 /92. Seiten 318-329, 1992

- [Dorloff 2004]: Dorloff F., Leukel J., Schmitz V.: E-Business- Standardisierung und Integration. Tagungsband zur Multikonferenz Wirtschaftsinformatik 2004. Cuvillier Verlag Göttingen, 2004
- [Duncan 1989]: Duncan, E.: A faceted approach to hypertext?. In R. McAleese. (Ed.), HYPERTEXT: theory into practice (pp. 157-163).Oxford: Intellect Press, 1989
- [FacetLens 2010]: Faceted Browsing for Scalable Information Exploration: Zugriff am 23.02.2010 unter <http://research.microsoft.com/en-us/um/redmond/groups/cue/facetlen/>
- [Flamenco 2010]: Flamenco Framework: FLEXible information Access using METadata in Novel Combinations, Zugriff am 10.02.2010 unter <http://flamenco.berkeley.edu/>
- [Fosket 1974]: Foskett, D. J., Foskett, J.: The London Education Classification: A thesaurus/classification of British educational terms. University of London Institute of Education Library, 1974
- [Gerken 2006] Gerken, J.: Orientierung und Navigation in zoombaren Benutzerschnittstellen unter besonderer Berücksichtigung kognitions-psychologischer Erkenntnisse. Universität Konstanz, Master Thesis, 2006
- [Gildea et al. 2008]: Gildea, N., Villa, R., Jose, J.: A Faceted Interface for Multimedia Search, In: SIGIR'08 Conference, Singapore, Singapore, Seiten: 775-776, 2008
- [Google Directory 2010] Google Directory. Zugriff am 11.03.2010 unter <http://directory.google.com/>
- [Hartley 2002]: Hartley J.: Communication, Cultural and Media Studies: The Key Concepts. Taylor & Francis Verlag, Auflage: 0003, 2002
- [Hearst 2006]: Hearst, M.: Clustering versus Faceted Categories for information exploration. In COMMUNICATIONS OF THE ACM April 2006/Vol. 49, No. 4, Seiten 59-61, 2006
- [Hearst et al. 2002]: Hearst, M., Elliott, A., English, J., Sinha, R., Swearingen, K., Yee, K.-P.: *Finding the flow in web site search*. In: COMMUNICATIONS OF THE ACM September 2002/Vol. 45, No. 9, Seiten 42-49, 2002
- [Heilig et al. 2008]: Heilig, M., Demarmels, M., König, W., Gerken, J., Rexhausen, S., Jetter, H.-C., Reiterer, H.: MedioVis – Visual Information Seeking in Digital Libraries. Universität Konstanz, HCI Group, 2008
- [Heilig et al. 2009]: Heilig, M., Demarmels, M., Rexhausen, S., Huber, S. and Runge, O.: Search, Explore and Navigate – Designing a Next Generation Knowledge Media Workbench. Universität Konstanz, HCI Group, 2009
- [Hopkins et al. 1988]: Hopkins, D., Callahan, J., Weiser, M., Shneiderman, B.: An empirical comparison of pie vs. linear menus. CHI 1988, Seiten 95-100, 1998

- [Hopkins 1991]: Hopkins, D.: The Design and Implementation of Pie Menus. There're Fast, Easy, and Self-Revealing. Published in Dr. Dobb's Journal, 1991
- [Hudson 2004]: Hudson, W.: Breadcrumb Navigation: There's More to Hansel and Gretel than Meets the Eye, 2004. Zugriff am 02.02.2010 unter <http://www.syntagm.co.uk/design/articles/breadcrumbs.htm>
- [Instone 2002]: Instone K.: Location, Path & Attribute Breadcrumbs. Information Architect, ibm.com. Zugriff am 27.03.2010 unter <http://keith.instone.org/breadcrumbs/>
- [ISO 2005]: Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten Teil 10: Grundsätze der Dialoggestaltung (ISO 9241-10 : 1995), Zugriff am 12.04.2010 unter: http://www.interactive-quality.de/site/DE/int/pdf/ISO_9241-10.pdf
- [Jetter 2007] Jetter, H.-C.: Informationsarchitektur und Informationsvisualisierung für die Post-WIMP Ära. Universität Konstanz, Master Thesis, 2007
- [Käki 2005]: Käki, M.: Findex: Search Result Categories Help Users when Document Ranking Fails. University of Tampere, Finland, Department of Computer Sciences, Papers: Document Interaction, CHI, 2005
- [König 2006] König, W.: Referenzmodell und Machbarkeitsstudie für ein neues Zoomable User Interface Paradigma. Universität Konstanz, Master Thesis, 2006
- [Kules et al. 2009]: Kules B., Capra R., Banta M., Sierra T. What Do Exploratory Searchers Look at in a Faceted Search Interface? *JCDL'09*, June 15–19, , Austin, Texas, USA, 2009
- [Kurtenbach 1993]: Kurtenbach G.: The Design and Evaluation of Marking Menus, University of Toronto, Doktorarbeit, 1993
- [Kurtenbach et al. 1994] Kurtenbach G., Buxton W.: User Learning and Performance with Marking Menus. CH194-4/94 Boston, Massachusetts USA, 1994
- [Kurtenbach et al. 1995]: Kurtenbach G., Tapia M.: Some Design Refinements and Principles on the Appearance and Behavior of Marking Menus, UIST '95 Pittsburgh, PA, USA, 1995
- [La Barre 2006]: La Barre, K.: The use of faceted analytico-synthetic theory as revealed in the practice of website construction and design. Indiana University, Doktorarbeit, 2006
- [Landauer et al. 1998]: Landauer, T. K., Foltz, P. W., & Laham, D. (1998): Introduction to Latent Semantic Analysis. *Discourse Processes*, 25, Seiten 259-284, 1998
- [Lee et al. 2004]: Lee B., Czerwinski M., Robertson, G., Bederson B.: Understanding Eight Years of InfoVis Conferences using PaperLens. *Posters Compendium of InfoVis*, pp. 53-54, 2004

- [Lee et al. 2009] Lee, B., Smith, G., Robertson, G., Czerwinski, M., Tan, D.: FacetLens: Exposing Trends and Relationships to Support Sensemaking within Faceted Datasets. Papers: Visualization 2, CHI, 2009
- [Li et al. 2006] Li W., Eades P., Nikolov N. Using Spring Algorithms to Remove Node Overlapping Paper. APVIS 2005, Conferences in Research and Practice in Information Technology, Vol. 45, , Sydney, Australia, Seiten: 131 – 140, 2006
- [Lida et al. 2003]: Lida B., Hul S., Pilcher K.: Breadcrumb Navigation: An Exploratory Study of Usage, Usability News Vol.5 Issue 1, Software Usability Research Laboratory, Wichita State University, Zugriff am 12.03.2010 unter: <http://www.surl.org/usabilitynews/51/breadcrumb.asp>
- [Maldonado, Resnick 2002]: Maldonado, C. A. & Resnick, M.L.: Do common user interface design patterns improve navigation? Proceedings of the Human Factors and Ergonomics Society 46th Annual Meeting 1315-1319, 2002
- [Mauder 2006]: Mauder M. : Context-Aware Applications, Computing in Context, Lehr und Forschungseinheit Medieninformatik LMU München. Zugriff 12.04.2010 unter <http://www.medien.ifi.lmu.de/lehre/ws0607/mmi1/essays/Markus-Mauder.xhtml>
- [MedioVis 2010]: MedioVis 2.0: Universität Konstanz, HCI Group, 2010. Zugriff am 15.03.2010 unter <http://hci.uni-konstanz.de/MedioVis2>
- [Mozilla 2010]: Pie Menu vs. Rectangular Menu Demo Zugriff am 10.03.2010 unter <http://people.mozilla.com/~jdicarlo/piemenus.html>
- [Pesch 2009]: Graph-Layouts für die Visualisierung von biologischen Datenbanken. Hochschule Bonn-Rhein-Sieg, Master Arbeit, 2009
- [Reiterer et al. 2010] *Reiterer H., Heilig M., Rexhausen S., Demarmels M.*: Idee der Blended Library – neue Formen der Wissensvermittlung durch Vermischung der realen und digitalen Welt. In: Ein neuer Blick auf Bibliotheken, in: Bibliothek & Information Deutschland e.V. (BID), Dinges & Frick, Wiesbaden, Erfurt, Deutschland, Seiten 108-116, 98. Deutscher Bibliothekartag, 2010
- [Schaier 2007]: Schaier S.: Konvergenz von internem und externem Rechnungswesen: Bedarf für eine Neustrukturierung des Rechnungswesens?. Gabler Verlag, Auflage 1, 2007
- [Schweizer 2009]: Schweizer, J.: Gestaltung und Implementierung eines Multi-Fokus und Multi-Display Management-Systems. Universität Konstanz, Bachelorarbeit, 2009
- [Spiegel 2010] Gabriel muss den Hartz-Fluch fürchten. Zugriff am 10.03.2010 unter <http://www.spiegel.de/politik/deutschland/0,1518,682707,00.html>
- [Taulbee 1965]: Taulbee, O.: Classification in Information Storage and Retrieval, Goodyear Aerospace Corporation, Akron, Ohio, 1965

- [Teng 2003]: Teng, H.: Location breadcrumbs for navigation: An exploratory study. Master's thesis, Dalhousie University, NS, Canada, Faculty of Computer Science, 2003
- [UC Berkeley] UC Berkeley Architecture Visual Resources Library. Zugriff am 14.01.2010 unter <http://orange.sims.berkeley.edu/cgi-bin/flamenco.cgi/spiro/Flamenco>
- [Vernon 1979]: Vernon, K., & Lang, V.: The London Classification of Business Studies: A Classification and Thesaurus for Business Libraries. London Graduate School of Business Studies, 1979
- [Vora 2009]: Vora, P.: Web Application Design Patterns. Kaufmann, 2009
- [Yee et al. 2003] Yee, K., Swearingen, K., Li, K., Hearst, M.: Faceted Metadata for Image Search and Browsing, Paper: Searching and Organizing, Chi Letters Volume No 5 Issue 1, 2003
- [Yiu et al. 2004]: Yiu, M., Mamoulis, N.: Clustering on a Spatial Network. University of Hong Kong. In ACM SIGMOD, Seiten: 443 – 454, 2004
- [Zamir et al. 1999]: Zamir, O. and Etzioni, O.: Grouper: Grouper: A Dynamic Clustering Interface to Web Search Results, University of Washington, 1999
- [Zeng et al. 2004]: Zeng, H.; He, Q. , Chen, Z., Ma, W., Ma, J.: Learning to Cluster Web Search Results. In: SIGIR'04, July 25–29, 2004, Sheffield, South Yorkshire, UK, S. 210 – 217, 2004
- [Zhang et al. 2005]: Zhang, J., Marchionini, G.: Evaluation and Evolution of a Browse and Search Interface: Relation Browser, University of North Carolina. In: dg.o; Vol. 89, Seiten: 179 – 188, Atlanta, Georgia, 2005
- [Zhao et al. 2004]: Zhao S., Balakrishnan R.: Simple vs. Compound Mark Hierarchical Marking Menus, *UIST '04*, October 24–27, Santa Fe, New Mexico, USA, 2004
- [Zöllner 2009]: Zöllner, M.: Ein persistentes objektorientiertes Datenmodell für das Personal Information Management in ZOIL, Universität Konstanz, Bachelorarbeit, 2009

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Bachelorarbeit selbständig angefertigt habe. Es wurden nur die in der Arbeit ausdrücklich benannten Quellen und Hilfsmittel benutzt. Wörtlich oder sinngemäß übernommenes Gedankengut habe ich als solches kenntlich gemacht.

Ort, Datum

Unterschrift