

**Interaktionsdesign für Information Retrieval Mehrwertdienste am  
Beispiel einer Search Term Recommender Anwendung**

# Diplomarbeit

zur Erlangung des Grades eines Diplom-Informatikers  
im Studiengang Computervisualistik

vorgelegt von

**Dominik Hering**

Erstgutachter: Prof. Dr. Jürgen Krause  
Institut für Computervisualistik

Zweitgutachter: Dipl. Inform. Philipp Schaer  
GESIS, Informationelle Prozesse für die Sozialwissenschaften

Koblenz, im Januar 2010

# Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ja    Nein

Mit der Einstellung der Arbeit in die Bibliothek bin ich einverstanden.

Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.

.....  
(Ort, Datum)

.....  
(Unterschrift)

# Inhaltsverzeichnis

<b>1</b>	<b>Danksagungen</b>	<b>1</b>
<b>2</b>	<b>Zusammenfassung</b>	<b>2</b>
<b>3</b>	<b>Vorwort</b>	<b>3</b>
<b>4</b>	<b>State of the Art</b>	<b>11</b>
4.1	Das klassische Suchmaschinenmodell . . . . .	11
4.2	Interfacedesign von Search-Term-Recommenderdiensten . . . . .	17
4.3	Pre-Search-Recommendations . . . . .	19
4.3.1	Anfragespezifikation . . . . .	19
4.4	Post-Search-Recommendations . . . . .	26
4.4.1	Trefferanzeige . . . . .	26
4.4.2	Reformulierung der Anfragespezifikation . . . . .	33
4.5	Die betrachteten Systeme . . . . .	38
4.5.1	oSkope . . . . .	38
4.5.2	Kartoo . . . . .	39
4.5.3	Grokker („ambient findability“) . . . . .	40
4.5.4	Searchme . . . . .	40
4.5.5	MyTag . . . . .	42
4.5.6	Google . . . . .	43
4.5.7	Bing . . . . .	44
4.5.8	Wolfram Alpha . . . . .	45
4.5.9	Picollator („image driven search“) . . . . .	45
4.6	Fazit State of the Art . . . . .	48
<b>5</b>	<b>Designentwurf für den Search Term Recommender</b>	<b>50</b>
5.1	User - Interface - Design . . . . .	50
5.2	(Software) Prototyping . . . . .	55
5.3	Usability Testing von Userinterfaces . . . . .	58
5.3.1	Standard Information Retrieval Evaluation . . . . .	59

---

5.3.2	Informelle Usability Evaluation . . . . .	59
5.3.3	Formale Usability Evaluation und Kontrollierte Experimente . . . . .	60
5.3.4	Longitudinal Studien . . . . .	61
5.3.5	Large-Scale Log-Based Usability Evaluation (Bucket-Testing) . . . . .	61
5.4	Erster Interfaceentwurf für den STR und Benutzertest . . . . .	63
5.4.1	Die DropDown-Funktion . . . . .	63
5.4.2	Der Benutzertest . . . . .	65
5.4.3	Die Aufgaben . . . . .	67
5.4.4	Das Ergebnis . . . . .	67
<b>6</b>	<b>Implementierung STR-Prototyp als DropDown-Variante</b>	<b>68</b>
6.1	Grails, Groovy, Mindserver . . . . .	68
6.1.1	Grails . . . . .	68
6.1.2	Groovy . . . . .	70
6.1.3	MindServer . . . . .	71
6.2	Implementierung . . . . .	72
6.2.1	Anforderungen an den Prototyp . . . . .	72
6.2.2	Funktionen des Prototyps . . . . .	73
6.2.3	Implementierte Methoden und Algorithmen . . . . .	75
6.3	Evaluation . . . . .	81
6.3.1	Testaufbau - Durchführung - Aufgaben . . . . .	81
6.3.2	Ergebnisse . . . . .	85
6.3.3	Fazit Evaluation . . . . .	91
<b>7</b>	<b>Fazit und Ausblick</b>	<b>93</b>
<b>A</b>	<b>Anhang - Fragebogen</b>	<b>101</b>

# Abbildungsverzeichnis

3.1	Darstellung der polyzentrischen Informationslandschaft . . . . .	3
3.2	Vagheitsproblem (Krause 2003) . . . . .	7
4.1	Aufbau der Googlearchitektur (Brin und Page, 1998) . . . . .	12
4.2	Was macht eine Suchmaschine gut? - Eigene Darstellung - siehe vergleichsweise Barker, 2003 . . . . .	14
4.3	Informationssuchprozess nach Salton und McGill . . . . .	16
4.4	"Pre-Search-Recommendations"; Google . . . . .	17
4.5	"Post-Search-Recommendations"; Google . . . . .	18
4.6	Deutsche Seite von Google . . . . .	24
4.7	US-Amerikanische Seite von Google . . . . .	25
4.8	Newsmap.jp . . . . .	29
4.9	Kartoo - Karten-Anzeige . . . . .	29
4.10	Grokker - „Blasen“1 . . . . .	30
4.11	Grokker - „Blasen“2 . . . . .	31
4.12	MyTag . . . . .	32
4.13	Google Wonderwheel . . . . .	32
4.14	abstrakte Papierskizze - Eigene Darstellung . . . . .	33
4.15	oSkope.com . . . . .	38
4.16	Kartoo.com . . . . .	39
4.17	Kartoo.com . . . . .	40
4.18	Grokker.com . . . . .	41
4.19	searchme.com . . . . .	41
4.20	searchme.com . . . . .	42
4.21	mytag.uni-koblenz.de . . . . .	43
4.22	Google.com (Google timeline) . . . . .	43
4.23	Google.com (Google Wonderwheel) . . . . .	44
4.24	Bing.com . . . . .	45
4.25	Wolframalpha.com . . . . .	46
4.26	Picollator.com . . . . .	46

---

5.1	Brainstorming1 . . . . .	63
5.2	Brainstorming2 . . . . .	64
5.3	Paperprototyp; Screenauswahl . . . . .	65
5.4	Paperprototyp1 . . . . .	66
5.5	Paperprototyp2 . . . . .	66
6.1	Model-View-Controller . . . . .	69
6.2	STR-Protoyp als DropDown-Variante . . . . .	76
6.3	STR-Protoyp als DropDown-Variante . . . . .	76
6.4	STR-Protoyp als DropDown-Variante . . . . .	76
6.5	Beoblab Uni Koblenz . . . . .	81
6.6	Usabilitytest1 . . . . .	83
6.7	Usabilitytest2 . . . . .	84
6.8	Benutzerbefragung; siehe Anhang . . . . .	85
6.9	Bewertungen durch die VP . . . . .	87

# Listings

- 6.1 Thesaurus der Sozialwissenschaften - Auszug . . . . . 74
- 6.2 searchAJAX . . . . . 77
- 6.3 askMindserver . . . . . 78
- 6.4 askTheSoz . . . . . 80

# 1 Danksagungen

- *Vielen Dank an Herrn Prof. Dr. Jürgen Krause, der auch nach seiner Pensionierung als Betreuer und Gutachter dieser Arbeit zur Verfügung stand.*
- *Vielen Dank an Herrn Dipl.-Inform. Philipp Schaer, der stets ein offenes Ohr, Zeit, Verständnis und eine Antwort auf meine vielen Fragen hatte.*
- *Vielen Dank an die Mitarbeiter/-innen des GESIS-IZ Leibnizinstitut für Sozialwissenschaften in Bonn.*
- *Ein besonderer Dank gilt meinen Großeltern.*



## 2 Zusammenfassung

Das Hauptaugenmerk dieser Arbeit, liegt auf einer softwareergonomisch empfehlenswerten Integration eines *Search Term Recommender Moduls (STR)* mithilfe von Usability-Tests und dem gezielten Blick auf den State of the Art des Interaktionsdesigns bei Retrieval-Mehrwertdiensten.

Daniela Holl (Holl, 2009) hat in ihrer Diplomarbeit ein Search Term Recommender Modulprototyp unter der Verwendung der Software *MindServer* entwickelt und die Vorzüge eines Search Term Recommenders in Bezug auf die Behandlung der verbleibenden Vagheit zwischen Benutzer und kontrolliertem Vokabular erläutert. Nach Entwicklung eines lauffähigen Prototyps führte sie eine empirische Studie zu den gelieferten Ergebnissen bzw. der Qualität der zurückgegebenen Ergebnisse durch. Somit wurde zu einem Großteil eine korrekte Funktionalität dieses Prototyps gewährleistet.

Sinn und Zweck des Search Term Recommenders ist es, dem Benutzer nur die Terme vorzuschlagen, welche nicht bereits durch Termentransformationen des Heterogenitätsservices behandelt wurden. Vornehmlich steht die Unterstützung der gezielten Suche nach bestimmten Daten zur Befriedigung des Informationsbedürfnisses im Mittelpunkt. Dabei gilt es den Benutzer sowohl in seiner Suchanfrageformulierung visuell, als auch beim Herausfiltern, der für ihn relevanten Ergebnisse in der Trefferanzeige zu unterstützen.

Da bislang lediglich maschinelle und interne Daten zum Testabgleich verwendet wurden, liegt die Priorität dieser Arbeit auf der Untersuchung von Dialoggestaltung bzw. der Benutzerinteraktion mit dem Search Term Recommender. Schwerpunkt war eine umfassende Evaluation von Designprototypen und (Papier)-Mockups, mittels Methoden des Usability-Engineering direkt am Benutzer selbst. Dies hinsichtlich Machbarkeit und Gebrauchstauglichkeit des Search Term Recommenders.

### 3 Vorwort

Seit mehreren Jahren ist eine zunehmende Entwicklung zu einer zentralistischen Struktur im WWW zu beobachten. Veränderungen aus technologischer, sowie marktwirtschaftlicher Sicht tragen maßgeblich dazu bei (Krause, 2008), (Cigan, 2002). Selbst für den Otto-Normalverbraucher sind leistungsfähige Rechner und kostengünstige Webzugänge (Flatrates) zu erwerben. Damit besteht für jeden Benutzer permanent die Möglichkeit auf alle verfügbaren Informationen im WWW zugreifen zu können, aber auch selbst jederzeit Informationen im WWW zugänglich zu machen und zu verbreiten. Dadurch hat das WWW erheblich an Größe gewonnen und wächst zunehmend weiter.<sup>1</sup>

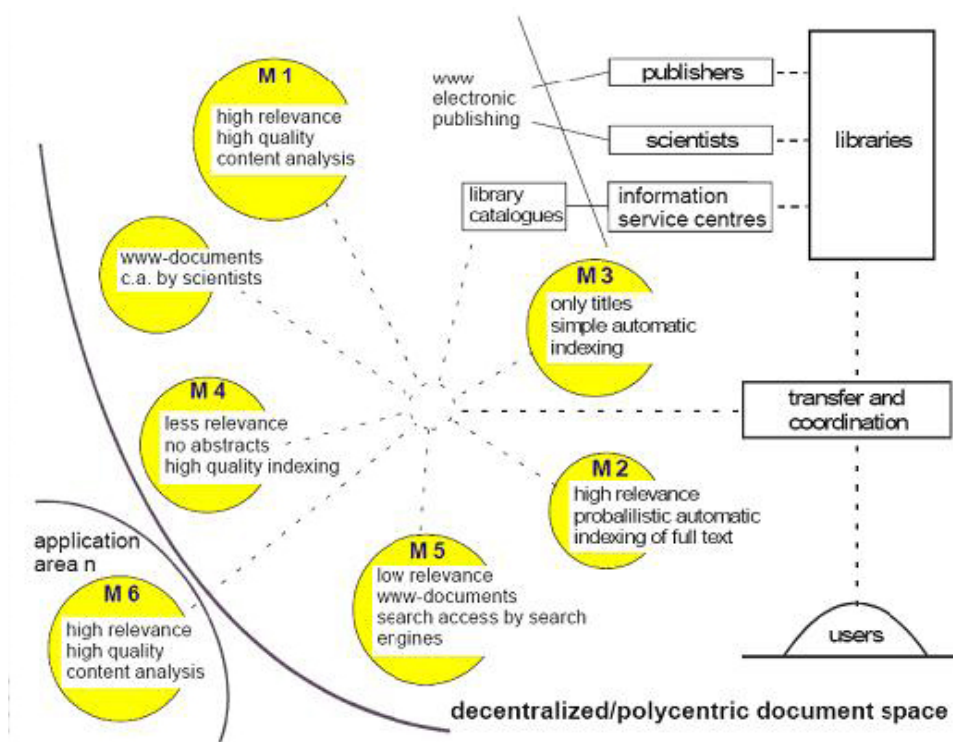


Abbildung 3.1: Darstellung der polyzentrischen Informationslandschaft

Diese zunächst positiv zu bewertende Entwicklung birgt aber auch ihre Probleme. Mit der wachsenden Fülle an Informationen, wächst die Heterogenität im WWW. Eine Folge sind unterschiedlichste

<sup>1</sup>Gemeint ist dabei die Anzahl der im WWW verfügbaren Informationen, nicht unbedingt die technologische Vernetzung

Konsistenzbrüche (Krause, 2008). Neben relevanten und qualitätskontrollierten Daten finden sich irrelevante oder sogar falsche.

War es früher traditionellen Anbietern wie z.B. Bibliotheken, Verlegern und Betreibern von Datenbanken mit Fachinformationen vorbehalten zu bestimmten Themen, Informationen anzubieten, ist es heute jedem möglich. Dementsprechend entfällt eine Qualitätssicherung der Daten. Dies spräche für die Verwendung von Anbietern von Fachinformationen, um inhaltlich korrekte Informationen zu erhalten. Allerdings handelt es sich dabei um sehr fachspezifische Datenbestände. Um sich fachübergreifend orientieren zu können, würde das die Nutzung von mehreren Fachinformationsportalen bedeuten. Da sich unterschiedliche Portale nicht standardisierten Erfassungstechniken bedienen, muss sich der Anwender jeweils auf eine andere Art von Inhaltserschließung und der Bedienung der Suchfunktion einstellen. Desweiteren würde das so entstehende Konglomerat an gefundenen Informationen dem Informationsbedürfnis des Benutzer nicht sachdienlich sein. Um dies zu begegnen, muss ein Fachportal nicht nur in der Lage sein, mehrere Fachinformationsgebiete zu vereinen, sondern darüber hinaus auch gewährleisten, dass der Benutzer seine Suche in dem von ihm vertrauten Vokabular formulieren kann.

Letztlich behält aber der Benutzer trotz der auftretenden Probleme den Anspruch bei, auf die Datenbestände zugreifen zu können und die für ihn relevanten und qualitativ hochwertigen Informationen angeboten zu bekommen. Insgesamt gesehen, zeigt sich, dass eine neue Form des Information Retrieval entstanden ist. Stand im klassischen Information Retrieval die Vollständigkeit der Treffermenge im Vordergrund, so geht es nun vornehmlich darum, möglichst relevante und qualitativ gesicherte Informationen liefern zu können, um dem Allgemeinziel, Befriedigung des Informationsbedürfnis, nachkommen zu können.

Im Laufe der Zeit wurden zur Problematik der Inhaltserschließung verschiedene Lösungsstrategien wie z.B. *Ontologie-* (siehe *Semantic Web*) oder *Standardisierungsansätze, Transfermodule* (z.B. *Crosskordanzen*) mehr oder weniger erfolgversprechend formuliert. Aufgrund unterschiedlicher Inhaltserschließungsverfahren und Standards ist eine Lösung der Problematik Heterogenität im WWW nicht absehbar (Krause, 2008, 2003). Unabhängig von der Inhaltserschließung und Güte der Informationen stellen noch andere Probleme bei der Informationssuche bzw. der Informationsfindung eine Herausforderung dar. Allein die schier unüberschaubare Menge an, der im WWW kursierenden unterschiedlichen Dokumente und Dateien bedürfen zur Inhaltserschließung verschiedenster Technologien. Die Suche im WWW auf rein intellektuelle Weise gestaltet sich so äußerst schwierig. Um diese Fülle von Informationen übersichtlicher zu gestalten, bedienen sich Information-Retrieval-Systeme diversen Methoden. Beispielsweise wäre, die Optimierung der Suchmechanismen hinsichtlich einer exakteren und relevanten Ergebnisfindung als ein Ansatz zu nennen. Eine solche, weit verbreitete Optimierung der Suche, ist die *Stichwortbasierenden Suche*:

„Bei der Stichwortbasierenden Suche werden die eingegebenen Schlagworte mit den in der Dokumentensammlung abgeglichen, indem die Dokumente identifiziert werden, die das eingegebene Schlagwort beinhalten. Passende Dokumente stellen die Ergebnismenge.“ (Puzicha, 2008)

Siehe andere: *Vektor-Raum-Suchmodell*, *Konzeptbasierende Suche* etc. Ein anderer Ansatz wären die Rankingverfahren, die noch vor der eigentlichen Trefferanzeige zum Tragen kommen. Mittels diversen Rankingalgorithmen wird die erhalten Ergebnismenge entsprechend der Suche oder der Benutzerpräferenzen sortiert. Für gewöhnlich werden die Ergebnisse als eine relevanzhierarchisch-absteigende Liste dargestellt. Die schon erwähnte unüberschaubare Menge an Informationen im WWW, macht das Ranking unerlässlich. Bei den Rankingverfahren, gibt es eine ähnlich große Vielfalt, wie auch bei den Suchmechanismen und hängen von sehr unterschiedlichen Faktoren ab (Lewandowski, 2005). Zum einen wären da die *dokument-* und *abfrageabhängigen Faktoren*. Dabei werden beispielsweise Dokumente, mit hohen relativen Vorkommenshäufigkeiten eines Suchbegriffs höher gerankt, als Dokumente mit niedrigeren Vorkommen. Weiter spielt die Reihenfolge der eingegebenen Suchbegriffe und/oder die Aktualität der Dokumente eine entscheidende Rolle. Dem zuerst eingegebenen Begriff und den aktuellsten Dokumenten werden mehr Relevanz zugewiesen. Zum anderen spricht man von *dokument-spezifischen Faktoren*. Darunter versteht man den Vorzug von Dokumenten, die in der selben Sprache wie die Suchanfrage verfasst sind. Weiter finden Groß-/Kleinschreibung und Betonung Berücksichtigung. Beim *Geo-Targeting*, erhalten die Dokumente von Servern, welche sich geografisch näher dem „Anfrageserver“ befinden, den Vorzug in der Rankingreihenfolge. Faktoren, wie Dokumentenlänge, Größe der Seite oder Linkpopularität sind von der Anfrage unabhängig. Bezüglich ausführlichen Erläuterungen zu diesen und anderen Faktoren, sei auf Lewandowski (2005) verwiesen.

Ein weiteres Problem des Benutzers im Information Retrieval, ist die Schwierigkeit, sein Informationsbedürfnis exakt formulieren zu können. Die Ergebnisse von Logfile-Analysen und Tests haben ergeben, dass hauptsächlich die einfache Suche verwendet wird und weitere Suchmodi in der Regel nur dann Verwendung finden, wenn diese direkt auf der Einstiegsseite zur Verfügung stehen. Unverständliche oder zu kurze Beschreibungen der selbigen veranlassen den Benutzer, eine Suchfunktionalität nicht weiter zu verfolgen. Nur in 0,5 Prozent aller Fälle werden die Hilfetexte und in den wenigsten Fällen die Ergebnisse über dem sechsten Ergebnis hinaus konsultiert. Durchschnittlich werden bei einer Suche zwei Suchbegriffe pro Anfrage verwendet. Dabei werden in der Regel Suchanfragen ohne Verwendung von Fachvokabular spezifiziert. Dadurch bleibt das Potential des bibliothekarischen Wissens oft ungenutzt (Schneider, 2009).

In vielen Fällen ist der teils unbefriedigende Umgang mit Suchmaschinen auch auf mangelndes Wissen über die Arbeitsweise und die Optionen der Suchoptimierung zurückzuführen. Beispielsweise sind viele Benutzer von der Syntax der Adresszeile und des Sucheingabefeldes irritiert, und vertauschen die

beiden Eingabefelder. Weiter zeigten Untersuchungen nach Muramatsu u. Pratt (2001) die Problematik einiger Nutzer im Umgang mit einfachen *Booleschen Operatoren* (*AND*, *OR*). Die Variation der Ergebnismenge bei unterschiedlicher Anwendung der Operatoren war für viele Probanden zu undurchsichtig. Das Phänomen einer leeren Ergebnismenge bei Eingabe der Phrase „to be or not to be“, oder der unterschiedlichen Ergebnismenge bei Eingabe von „boat fire“ und „fire boat“ waren von viele Testpersonen nicht zu erklären.

Aus der Sicht des Suchprozesses zusammenfassend formuliert, beschreibt man dieses Problem als *Transferproblem* oder auch als *Vagheitsproblem*. Genauer gesagt, ist es die Diskrepanz zwischen einer beliebigen fachlichen Fragestellung, bei der der Benutzer natürlichsprachige Terme verwendet, und der Terme, die die Inhaltserschließung für entsprechende Dokumente vorgenommen hat (Petras, 2006). Die Vagheit zwischen Anfrage- und Dokumentebene wird bei Krause als V1 3.2 bezeichnet und kann zum Beispiel durch Verfahren der Termerweiterung behandelt werden. Dies kann „manuell“ durch den Benutzer unter Verwendung eines Thesaurus mit Synonymbeziehungen geschehen, oder in ähnlicher Weise durch das Informationssystem (teil-)automatisch. Dementsprechend stellt sich die Frage, inwieweit die Initiative beim Benutzer bleibt oder inwieweit man dem Benutzer diese zusätzlichen Aufgaben zumuten sollte.

„Jedem Bibliothekar und jedem, der sich mit Information Retrieval befasst, war schon immer klar, dass zwischen den semantischen Termen, die in der Datenbank ein Dokument charakterisieren, und dem Term, den der Benutzer anwendet, nicht immer eine 1:1-Relation besteht.“ (Krause, 2003)

Im Information Retrieval wurden diverse Strategien zur Lösung dieser Problematik vorgeschlagen. Zu nennen wären da *Crosskonkordanzen* zu Klassifikationen und Thesauri, *qualitativ-deduktive*, oder auch *quantitativ-statistische* Ansätze. Bei den Crosskonkordanzen geht es nicht um eine neue Standardisierung bestehender Begriffswelten. Vielmehr unternimmt man den Versuch verschiedene Begriffssysteme im Anwendungskontext zu analysieren und ihre Begrifflichkeiten intellektuell aufeinander zu beziehen. Qualitativ-deduktive Verfahren finden sich beim Intelligenten Information Retrieval, bei intelligenten Recherchesystemen und im Bereich der Expertensysteme. Quantitativ - statistische Verfahren hingegen, eignen sich besonders für die Zusammenführung von Bibliotheksbeständen mit Fachdatenbanken der Informationszentren (Krause, 2008). Ein weiterer Lösungsvorschlag für die Vagheitsbehandlung der ersten Ebene V1 stellt ein *Search Term Recommender System* dar (Petras, 2006).

Recommender Systeme an sich, haben seit den frühen 1990er Jahren zunehmend in der Informationswissenschaft und besonders bei Entwicklern von Dokumentretrievalsystemen und digitalen Bibliotheken an Bedeutung gewonnen. Im Allgemeinen, ist der Kern eines jeden Recommender Systems, ein spezifizierter Informationsfilterungsmechanismus, welcher aus einer großen Menge an existierenden

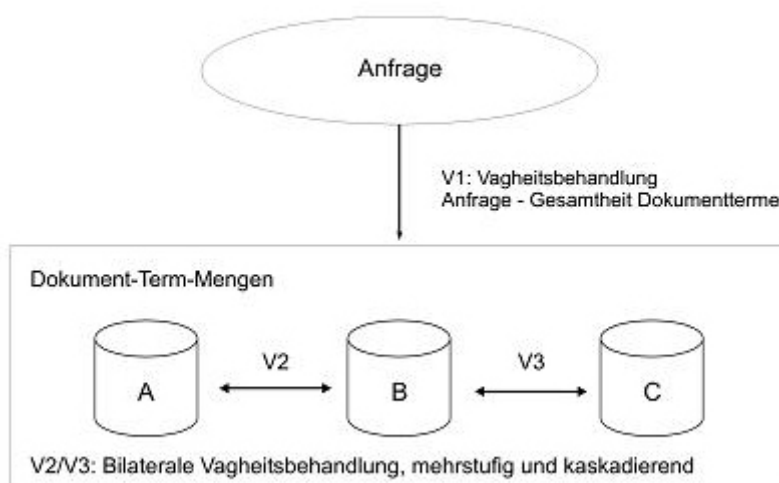


Abbildung 3.2: Vagheitsproblem (Krause 2003)

Objekten, Informationen analysiert und extrahiert. Allerdings gilt es dabei verschiedene Formen solcher Systeme zu unterscheiden. Zwar ist die Grundidee aller Varianten, den Benutzer zu unterstützen, aber dennoch unterscheiden sie sich in der technischen Realisierung. Dabei sind z.B. die Komplettierung des eingegebenen Suchbegriffs, Suchempfehlungen wie „Meinten Sie vielleicht...?“ , oder auch Formen anderer grafischer Art (siehe: *Google Wonderwheel*) als Recommending Funktionen anzusehen. Betrachten wir als konkretes Beispiel *Amazon.com*<sup>2</sup>. *Amazon* ist ein sogenannter Online Retailer. *Amazon* bietet diverse neue und gebrauchte Produkte zum Kauf oder Verkauf an. Aber auch den Benutzern ist die Möglichkeit gegeben Produkte zum Verkauf anzubieten. *Amazon* bedient sich eines Produkt Recommendationsystems. Ein solches System kann bzw. soll zum einen den Benutzer in seiner Produktsuche auf *Amazon* unterstützen, schlägt aber auch zum anderen Produkte vor, die eventuell für den Benutzer von Interesse sein könnten. Das Interesse des Benutzers auf Produkte zu lenken, nach denen er ursprünglich nicht primär gesucht hat, dürfte zumindest für derartige Shopping-Portale mindestens einen wirtschaftlichen Mehrwert bringen.<sup>3</sup> Für jeden angemeldeten Benutzer lässt sich ein (Präferenz-)Profil erstellen. Dies kann aufgrund expliziter oder impliziter Datenerfassung erfolgen. Bei der expliziten Erfassung wirkt der Benutzer „aktiv“ mit. Er wird beispielsweise gebeten, bestimmte Objekte nach Güte zu bewerten oder eine Liste von beliebten Objekten zu erstellen. Die impliziten Datenerfassung laufen hintergründig, während der Benutzung der Webseite ab. Beispielsweise werden die vom Benutzer online gekauften Artikel, oder die angeschauten Objekte mitprotokolliert. Das

<sup>2</sup><http://www.amazon.org>

<sup>3</sup>siehe: „Benutzer die das Produkt X, für das Sie sich interessieren, gekauft haben, haben auch folgende Produkte Y, Z gekauft.“

Recommender-System vergleicht anschließend die gesammelten Daten mit ähnlich erstellten Datenprofilen und berechnet so eine Liste von empfohlenen Objekten für den jeweiligen Benutzer. Solche Empfehlungen können daher als eine Art von Vorhersage der Benutzerpräferenzen betrachtet werden (Varian 1996).

Solche Recommender Systeme basieren auf den Ansätzen des *Content-based Filtering* (inhaltsbasiertes Filtern) und dem *Collaborative Filtering* (gemeinschaftsbasiertes Filtern). Hinzu kommen hybride Ansätze, die beide Techniken vereinen.

Bei der inhaltsbasierten Filterung werden Ähnlichkeiten von Objekten über deren Eigenschaften ermittelt. Ein Objekt wird unabhängig seiner Eigenschaften beschrieben und bekommt daraufhin bestimmte Attribute zugewiesen. Sehr komplexe Objekte stellen durch ihre schlechte Beschreibbarkeit, und damit verbundene schlechte Erfassbarkeit, ein Problem dar. Beispielsweise lässt sich der Inhalt einer Internetseite problemlos beschreiben, jedoch Auskünfte über die Eigenschaften der Bilder, Ladezeiten oder Darstellungsqualität fehlen meistens (Balabanovic u. Shoham, 1997). Ein weiteres Problem ist die Überspezialisierung, welche sich negativ auf die Qualität der Empfehlung auswirken kann. Die inhaltsbasierte Suche neigt dazu, dem Benutzer ähnlich den bereits bewerteten Objekten, Objekte zu empfehlen. Dabei werden Objekte, die den Benutzer potentiell interessieren könnten, aber mit dem Benutzerprofil divergieren, nicht angezeigt. Ein inhaltsbasiertes System lässt sich über *Latent Semantic Indexing (LSI)* realisieren. Weitere existente Methode, sollen an dieser Stelle nicht weiter erläutert werden, da insbesondere die LSI für die Entwicklung des Search Term Recommender von Interesse ist. Die Grundannahme beim LSI, ist eine latent semantische Struktur hinter allen Worten einer Sprache. Die Objekte werden nicht durch Schlagwörter beschrieben, sondern einer semantischen Struktur zugeordnet (Foltz u. Dumais, 1992). Das bedeutet, dass Begriffe wie „Kfz“, „PKW“, „Auto“ als ähnlich erfasst würden.

Die gemeinschaftsbasierte Filterung beschränkt sich auf die Präferenzprofile anderer Nutzer, die ähnlich den aktiven Benutzern sind, und vernachlässigt die Informationen über die zugrunde liegenden Objekte. Hingegen der inhaltsbasierten Filterung, stellen komplexe Objekte kein Problem dar, da sie nicht explizit beschrieben werden. Nachteile ergeben sich aus der Tatsache, dass nur bereits bewertete Dokumente empfohlen werden können und gegebenenfalls eine verhältnismäßig kleine Anzahl von Nutzern, einer großen Anzahl an zu bewertenden Objekten gegenübersteht. Weiter werden „exotischen“ Benutzerprofilen eher unbrauchbare Empfehlungen vorgeschlagen, da das System dazu neigt populäre Objekte zu empfehlen.

Hybride Systeme verfolgen das Ziel, die individuellen Schwächen zu minimieren und die Stärken zu nutzen. Hybride Systeme sind eine Kombination aus inhaltsbasierter Filterung und gemeinschaftsbasierter Filterung. Die beschriebenen Recommender Systeme finden heutzutage selbst auf dem Gebiet der *Human Resources* (Färber u. a., 2003) , also z.B. bei der Rekrutierung von Arbeitskräften in einem Betrieb bereits Verwendung.

Eine andere Form eines Recommending Systems stellt der Search Term Recommender dar. Ein Search Term Recommender bietet die Möglichkeit die Anfrage des Benutzers zu erweitern. Dadurch wird der Benutzer in der Formulierung seiner Suche unterstützt. Dies geschieht, indem man dem Benutzer bezüglich seiner Anfragespezifikation, vermeintlich semantisch verwandte Suchbegriffe vorschlägt. Durch eine solche Empfehlung versucht man dem Vagheitsproblem V1 entgegen zu wirken. Dadurch wird dem Benutzer nicht nur eine genauere Suchspezifikation ermöglicht, sondern es können sich auch die Chancen erhöhen, die Differenzen zwischen Eingabevokabular und der vom System über die Inhaltserschließung vergebenen Terme zu vermindern. Dabei werden diejenigen Begriffe eines Dokumentenbestandes zur Empfehlung herangezogen, welche im kontrollierten Vokabular des jeweiligen Systems enthalten sind (Petras, 2006). Es kann sich um Thesauri, Listen von Subject Headings oder auch Klassifikationen handeln. Diese Vokabularien unterscheiden sich in Kontrollfähigkeit und Aufbau. Bei der Klassifikation werden die Dokumente eindeutig und sich gegenseitig ausschließend vollständig kategorisiert. Bei einem Thesaurus werden hingegen die Beziehungen der Deskriptoren untereinander dargestellt. Die Beziehungen stehen für die unter den Deskriptoren herrschende Hierarchie und der Assoziativität untereinander. Bei kontrollierten Listen der Subject Headings werden lediglich die Terme dieser Liste zur Darstellung eines Dokuments verwendet und in der Reihenfolge der Darstellung berücksichtigt (Holl, 2009). Wie die Auswertungen von Logfiles bei Schneider (2009) ergeben haben, empfiehlt es sich die Suchanfrage des Benutzers, durch die adäquaten Terme des kontrollierten Vokabulars zu erweitern, beziehungsweise dem Benutzer als Empfehlung zugänglich zu machen. Würde beispielsweise ein Benutzer nach dem Begriff „Flugzeug“ suchen, so liefern gängige Suchmaschinen wie *Google*<sup>4</sup>, *LiveSearch*<sup>5</sup> u.a. Ergebnisse, in denen der Begriff „Flugzeug“ vorkommt und Ergebnisse in denen das Wort „Flugzeug“ oder zumindest Teile des Suchterms enthalten sind (siehe: „Flugkapitän“, „Flugzeugturbine“,...). Mit dem Search Term Recommender im Verbund mit dem *Mindserver* wurde allerdings der Ansatz realisiert, dem Benutzer neben den genannten Ergebnissen auch thematisch bzw. semantisch verwandte Ergebnisse zur Suchanfrage liefern zu können. Empfehlungen zum Suchbegriff „Flugzeug“ könnten dann z.B. „Airbus“, „Flieger“, „Jet“ etc. sein.

Nach Petras (2006), Mayr u. a. (2008) lässt sich ein Search Term Recommender folgendermaßen realisieren. Zunächst wird ein Wörterbuch von Assoziationen zwischen zwei Vokabularien aufgebaut. Das eine Vokabular besteht aus natürlichsprachlichen Termen und Phrasen eines Dokumentes, das andere ist das kontrollierte Vokabular, das für die Darstellung der Dokumente Verwendung findet. Treten natürlichsprachliche Terme des einen Vokabulars gemeinsam mit Termen des kontrollierten Vokabulars mit einer bestimmten Mindesthäufigkeit auf, so werden diese assoziiert. Die Assoziationsstärke ergibt sich aus der Häufigkeit des gemeinsamen Auftretens. Je häufiger ein solches Auftreten vorkommt,

---

<sup>4</sup><http://www.google.com>

<sup>5</sup><http://www.livesearch.com>



---

desto größer die Assoziationsstärke. Zusätzlich übernimmt der Search Term Recommender die Unterteilung der Dokumentensammlung in verschiedene Fachgebiete, um den Suchraum zu unterteilen. Da die Assoziationsstärke zwischen einem natürlichsprachlichen Term und den kontrollierten Vokabularen der Fachgebiete variieren kann, müssen die Assoziationsgewichte für jedes Fachgebiet berechnet werden. Bei Verwendung von mehreren natürlichsprachlichen Suchformulierungen werden, die aus allen Fachgebieten assoziierten Terme, ausgegeben. Da die Assoziationsberechnungen auf einer statistischen Analysis basieren, ist es darüber hinaus mit einem Search Term Recommender möglich, auch Abbildungen zwischen unterschiedlichen Arten von kontrollierten Vokabularen, zwischen Dokumenten und kontrolliertem Vokabular oder zwischen natürlich sprachlichen Termen einer Sprache und einem kontrolliertem Vokabular einer anderen Sprache zu berechnen. Genauere Ausführungen zur Funktionalität und Eigenschaften eines Search Term Recommender sind bei Petras (2006) zu finden. Am *GESIS Leibniz - Institut - für Sozialwissenschaften* in Bonn wird im Rahmen von *Sowiport* ein <sup>6</sup> solches Search Term Recommender (STR) - Modul entwickelt. *Sowiport* ist ein Portal zur Suche von sozialwissenschaftlicher Literatur. *Sowiport* bringt wissenschaftliche Informationen aus verschiedenen Quellen an einem Punkt zusammen. Von Literatur über Forschungsprojekte bis hin zu Neuigkeiten aus den Sozialwissenschaften stehen diese vernetzt zur Verfügung.

Im folgenden Kapitel „State of the Art“, sind an konkreten Beispielen gängige Recommendingdienste erläutert. Inhaltserschließungsverfahren, Ranking- und Suchverfahren sollten an dieser Stelle zur Einordnung der Thematik genannt werden, auf eine ausführliche Ausführung wurde jedoch dementsprechend verzichtet.

---

<sup>6</sup><http://www.sowiport.de>

# 4 State of the Art

In diesem Kapitel geht es darum, einen Überblick über die aktuellen Visualisierungs- und Interaktionsansätze generell bei Informationsanbietern, insbesondere aber bzgl. der Entwicklungen von Recommendingdiensten zu vermitteln. Einleitend wird das klassische Suchmaschinenmodell betrachtet. Zu erwähnen ist jedoch, dass hier nur Suchmaschinen, Suchanwendungen und Suchinterfaces an aktuellen Anwendungsbeispielen behandelt werden sollen, da ein umfassender Blick auf alle möglichen Interfaces, dem Kontext Search Term Recommender über die Maße hinaus entfliehen würde.

## 4.1 Das klassische Suchmaschinenmodell

Betrachtet man die Typen von Suchanwendungen im WWW, so lassen sich prinzipiell zwei Formen unterscheiden: einerseits die *manuell erstellten Verzeichnisse*, andererseits die *algorithmischen Suchmaschinen* in unterschiedlicher Ausprägungen. Die algorithmischen Suchmaschinen lassen sich je nach Ausprägung weiter in *Universalsuchmaschinen*, *Spezialsuchmaschinen* und *Archivsuchmaschinen* unterteilen (Lewandowski, 2005).

Universalsuchmaschinen sind thematisch, geografisch oder sprachlich unbegrenzt. Ziel hingegen ist es, das gesamte WWW zu erfassen. Spezialsuchmaschinen beschränken sich bewusst auf ein Themengebiet, wie nur das wissenschaftliche Web und/oder bestimmte Inhalte. Eine Unterscheidung von algorithmischen Suchmaschinen auf technologischer Ebene, unterteilt die algorithmischen Suchmaschinen in *klassische algorithmische Suchmaschinen*, *Meta-Suchmaschinen* und *Suchagenten*.

Klassische algorithmische Suchmaschinen (auch nur Suchmaschinen genannt) durchsuchen automatisch das WWW und erfassen die gefundenen Dokumente in einer eigenen Datenbank. Erfolgt eine Suchanfrage, wird die Datenbank durchsucht und mittels Rankingalgorithmen eine Ergebnisliste ausgegeben.

Meta-Suchmaschinen (*MyTag*)<sup>1</sup> besitzen keine eigene Datenbank. Erfolgt eine Suchanfrage, so wird diese an andere Suchdienste weitergeleitet. Nach Rückgabe der Ergebnisse, werden diese dann gelistet. Bei Suchagenten handelt es sich um Programme, die vom Benutzer auf dem Rechner installiert werden müssen. Zum einen besteht die Möglichkeit einen Suchagenten ähnlich einer Suchmaschine mit einer direkten Anfrage, oder zum anderen mit regelmäßigen Suchanfragen zu beauftragen. Der Benutzer

---

<sup>1</sup><http://mytag.uni-koblenz.de>

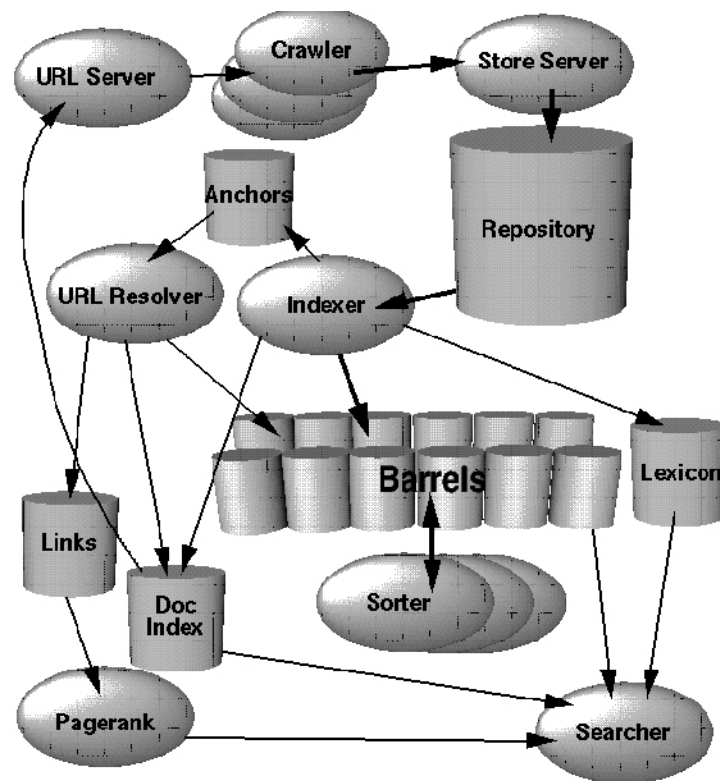


Abbildung 4.1: Aufbau der Googlearchitektur (Brin und Page, 1998)

bekommt dann stets die aktuellsten Ergebnisse angezeigt (Lewandowski, 2005).

Die Abbildung 4.1 zeigt den Aufbau der Architektur von *Google*. In *Google* ist der Web Crawler, der für das Hernunterladen der gefundenen Webseiten verantwortlich ist, durch mehrere Crawler realisiert. Der URL-Server sendet die Liste mit URLs, welche erfasst werden sollen, an die Crawler. Die erfassten Webseiten werden im Anschluss daran an den *Storeserver* geschickt, der die Seiten komprimiert und in einem Repository speichert. Jede Webseite bekommt eine ID Nummer, die sogenannte *docID* zugewiesen, sobald eine neue URL auf einer Webseite gefunden wird. Die Indexierungsfunktion übernimmt bei *Google* der *Indexer* und der *Sorter*. Der Indexer hat mehrere Aufgaben.

Er liest das Repository, dekomprimiert die Dokumente und parst sie schließlich. Jedes Dokument wird in eine Menge von Wörtern (*Hits*) konvertiert. Der Indexer unterteilt die Hits in eine Menge von *Barrels*, um einen teilsortierten Index zu erstellen. Weiter filtert der Indexer alle auf einer Webseite vorkommenden Links in einer *Anchor file*. Diese Datei enthält genug Informationen, um berechnen zu können, von welchen Linkpunkten der Link kommt und zu welchen Linkpunkten der Link geht (siehe: *PageRankAlgorithmus*). Der *URL-Resolver* liest die Anchor files und konvertiert diese von relativen URLs in absolute URLs, welche dann in *docIDs* gewandelt werden. Der Sorter bedient sich den durch die docIDs sortierten Barrels und generiert mithilfe der *wordID* den invertierten Index. Ein Programm namens *DumpLexicon* nimmt diese Liste zusammen mit dem vom Indexer produziertem Lexikon und

erstellt ein neues Lexikon, welches dann vom Searcher gebraucht wird. Der Searcher wird durch einen Webserver betrieben und beantwortet mit dem Lexikon (*DumpLexicon*), dem invertierten Index und dem *PageRank* die Anfragen (S.Brin u. L.Page, 1998).

Nun kann man sich die Frage stellen, was eine Suchmaschine gut macht. So sehr sie sich in Layout, Abfragearten, Anfragearten oder der Technik sind, so gleichen sie sich in dem Bestreben, das Informationsbedürfnis der Benutzer so gut wie irgend möglich zufrieden zu stellen. Alle Suchmaschinen bestehen aus drei grundlegenden Teilen. Einer Datenbank mit WWW-Dokumenten (1), einer Suchmaschinenoperationen auf einer oder mehreren Datenbanken (2) und einer Menge von Programmen, die die Darstellung des Ergebnisses auf dem Bildschirm berechnen (3). Die folgende Auflistung 4.2 zeigt einen Überblick über die drei Grundbausteine und deren Variablen, die unmittelbar Einfluss auf eine erfolgreiche Suche nehmen.

Um nun ein erfolgreiches Suchinterface gestalten zu können, ist es notwendig, sich den menschlichen Informationssuchprozess zu verdeutlichen. Zu diesem sehr komplexen kognitiven Prozess existiert eine Vielzahl von Theorien (Hearst, 2009). An dieser Stelle soll lediglich eine Betrachtung des *Standard Models* erfolgen. Schnell wird deutlich, dass bei dem Versuch die theoretischen Modelle auf praktische Webanwendungen zu münzen, es Parallelen aber auch Diskrepanzen gibt. (Andere Modelle wären: *Kognitives Modell*, *Dynamisches Modell (Berry picking)*, *Suche als strategischen Prozess*, etc.). Da es vorrangig aber um die Gestaltung von Recommendingfunktionen im Interface geht, ist die Beschränkung auf das Standard Model ausreichend. Die anderen genannten Modelle stellen weitere Ansätze des Informationssuchprozesses dar und basieren zu einem Großteil auf dem Standard Model. Für die Motivation eines Search Term Recommenders und anderer Recommendingfeatures sind diese allerdings zu explizit. Weiter noch, wird im Folgenden eine andere Aufteilung des Standard Models definiert, um eine Unterscheidung der Recommendingdienste vornehmen zu können.

„Information-seeking is a special case of problem solving. It includes recognizing and interpreting the information problem, establishing a plan of search, conducting the search, evaluating the results, and if necessary, iterating through the process again.“ (Marchionini, 1989)

Marchionini u. White (2008) beschreiben den Informationssuchprozess folgendermaßen. Zunächst braucht es die Erkenntnis eines Informationsbedarfs. Weiter akzeptiert man die Notwendigkeit einer Aktion, um sein Informationsbedürfnis befriedigen zu können. Um nach Informationen zur Problemlösung suchen zu können, muss zunächst eine Problemformulierung stattfinden. Anschließend gilt es die Information passend in einem Suchsystem formulieren zu können. Erhält man eine Ergebnismenge, muss diese nach potentiell tauglichen Informationen durchforstet werden. Sollte sich kein Ergebnis als brauchbar erweisen, muss das Problem umformuliert werden. Der letzte Schritt ist die Verwendung des Ergebnisses.

Teil der Suchmaschine	Variablen
<p><b>1. Datenbank mit WWW-Dokumenten</b></p>	<ul style="list-style-type: none"> <li>• <b>Größe der Datenbank:</b> <ul style="list-style-type: none"> <li>○ Wieviele Dokumente beinhaltet die Datenbank der Suchmaschine?</li> <li>○ Wieviel vom gesamten WWW wird durch die Suchmaschine erfasst?</li> </ul> </li> <li>• <b>Aktualität</b> <ul style="list-style-type: none"> <li>○ Die Datenbanken der Suchmaschinen bestehen aus Kopien von Webseiten und anderen Dokumenten, die von Crawlern und Spiders beim letzten Besuch erfasst wurden.</li> <li>○ Wie oft wird die Datenbank aktualisiert?</li> <li>○ Wie oft werden die vom Benutzer gesuchten Seiten aktualisiert?</li> </ul> </li> <li>• <b>Vollständigkeit</b> <ul style="list-style-type: none"> <li>○ Befindet sich in der Datenbank der Volltext, oder nur Teile der Seite?</li> <li>○ Ist jedes Wort indiziert?</li> </ul> </li> <li>• <b>Angebotene Dokumententypen</b> <ul style="list-style-type: none"> <li>○ Alle Suchmaschinen bieten Webseiten an</li> <li>○ Bieten sie zusätzlich PDF, Word, PowerPoint etc.?</li> </ul> </li> <li>• <b>Geschwindigkeit und Konsistenz</b> <ul style="list-style-type: none"> <li>○ Wie schnell ist die Suchmaschine?</li> <li>○ Wie konsistent? Bekommt man zu unterschiedlichen Zeiten, unterschiedliche Ergebnisse?</li> </ul> </li> </ul>
<p><b>2. Die Suchmaschinen Einsatzmöglichkeiten</b></p> <p>Alle Suchmaschinen bieten die Möglichkeit Suchbegriffe einzugeben. Aber was passiert im Inneren? Lassen sich vom Benutzer Beschränkungen vornehmen, die die Chancen erhöhen, das zu finden, was man sucht?</p>	<ul style="list-style-type: none"> <li>• <b>Basis-Suchoptionen und Beschränkungen</b> <ul style="list-style-type: none"> <li>○ Werden Anführungszeichen zur Phrasenbildung akzeptiert?</li> <li>○ Besteht die Möglichkeit einer OR-Suche (z.B. Synonyme)? Lassen sich nur einzelne Wörter mit OR verknüpfen, oder auch Phrasen?</li> </ul> </li> <li>• <b>Erweiterte Suchoptionen und Beschränkungen</b> <ul style="list-style-type: none"> <li>○ Ist es möglich Dokumente Domain übergreifend zu finden (org, edu, gov, etc.)? Ist es auf eine/mehre Domain beschränkt?</li> <li>○ Lässt sich eine Beschränkung auf ein oder mehrere Dokumententypen vornehmen (Pdf, Excel, etc.)?</li> <li>○ Lässt sich eine Beschränkung auf eine Sprache vornehmen?</li> <li>○ Wie leicht ist das Herausfiltern (Einschränkung) auf die aktuellsten Dokumente?</li> </ul> </li> <li>• <b>Generelle Beschränkungen und Features</b> <ul style="list-style-type: none"> <li>○ Was muss man tun um „normal“ oder mit „stop words“ zu suchen?</li> <li>○ Ist die Kapazität der Suchbegriffe oder der Suchkomplexität beschränkt?</li> <li>○ Besteht die Möglichkeit die „alten“ Suchbegriffe beizubehalten? Gibt es jedes Mal eine neue Suche?</li> <li>○ Kann man sich auf eine konsistente Ergebnismenge von Suche zu Suche von Tag zu Tag verlassen?</li> <li>○ Lässt sich die Suche und das Display individuell vom Benutzer einstellen?</li> <li>○ Gibt es einen Jugendschutz-Filter? Arbeitet er zuverlässig?</li> </ul> </li> </ul>
<p><b>3. Trefferanzeige</b></p> <p>Alle Suchmaschinen geben eine Menge von Ergebnissen zurück, von der die Suchmaschine, "meint", dass sie relevant sind. Wie gut kann sie sich in den Benutzer hineinversetzen?</p>	<ul style="list-style-type: none"> <li>• <b>Ranking</b> <ul style="list-style-type: none"> <li>○ Sind die Ergebnisse nach Popularität oder Relevanz geordnet? Beides?</li> <li>○ Werden Seiten mit inhaltlich der Suchphrase identischen Phrasen höher gerankt?</li> <li>○ Erhält man auch Ergebnisse, bei denen nicht alle Suchbegriffe vorkommen?</li> </ul> </li> <li>• <b>Display</b> <ul style="list-style-type: none"> <li>○ Sind die Suchbegriffe hervorgehoben? Werden Auszüge zur Seite angeboten, die das Warum des Matching untermauern?</li> <li>○ Gibt es andere Exzerpte der Seite? Vorschau?</li> </ul> </li> <li>• <b>Seite</b> <ul style="list-style-type: none"> <li>○ Wenn nur einige Seiten (pages) der WWW-Seite (site) angezeigt werden, sind es dann die mit den Suchbegriffen?</li> <li>○ Wie leicht ist es die gesamte WWW-Seite zu sehen?</li> <li>○ Kann das als bevorzugter Suchmodus gespeichert werden?</li> </ul> </li> </ul>

Abbildung 4.2: Was macht eine Suchmaschine gut? - Eigene Darstellung - siehe vergleichsweise Barker, 2003

Gerade der Vorgang der Umformulierung, oder besser, Reformulierung der Anfrage unterscheidet ältere Ansätze zur Beschreibung des Standardverfahrens. Im klassischen Information Retrieval formuliert der Benutzer sein Informationsbedürfnis und erhält als letzten Schritt sein gesuchtes Ergebnis (*exact match*). Die im folgenden Teil der Arbeit angesprochenen Systeme sind allesamt Systeme, die eine Suchanfrageformulierung, eine Durchsicht der Ergebnisse (Trefferanzeige) und eine Suchreformulierung anbieten. Man geht davon aus, dass der Benutzer nach der Anzeige der Trefferliste zu seiner ursprünglichen Anfrage noch Bedarf an einer Verfeinerung oder auch Veränderung seiner Suchanfrage hat. Die ursprüngliche Suche wird durch Browsingelemente ergänzt. Ein Suchprozess lässt sich so in zwei Schritte unterteilen. (Einfache) Suche und die Suchreformulierung durch gegebenenfalls Klicken von Einschränkungsmöglichkeiten, die vom System aufgrund der Suchanfrage und ihrer Treffer vorgegeben werden. Die nach dem Abschicken der Suchanfrage erstellte Trefferliste stellt damit nicht mehr den Endpunkt der Recherche dar, sondern einen Zwischenschritt, der zu weiteren Modifikationen der Anfrage einladen soll (Lewandowski, 2005).

Die Abbildung 4.3 zeigt abstrahiert den schematischen Ablauf einer interaktiven Informationssuche im klassischen Information Retrieval. Nach dem Einloggen oder dem ersten Besuch auf der Seite wird eine Suchanfrage vom Benutzer formuliert. Im nächsten Schritt wird nicht notwendigerweise eine Trefferliste ausgegeben, sondern der Benutzer kann aufgrund der Anzahl der Treffer entscheiden, ob er die Suchanfrage modifizieren möchte. Das dargestellte Modell beschreibt eine solange wiederkehrende Reformulierung der Suchanfrage, bis ein zufriedenstellendes Ergebnis gefunden wurde. Die Reformulierung erfolgt entweder nach der Ausgabe der Ergebnisanzahl bzw. der Durchsicht der Trefferliste oder aber nach der Ausgabe der Ergebnisse und deren Ansicht.

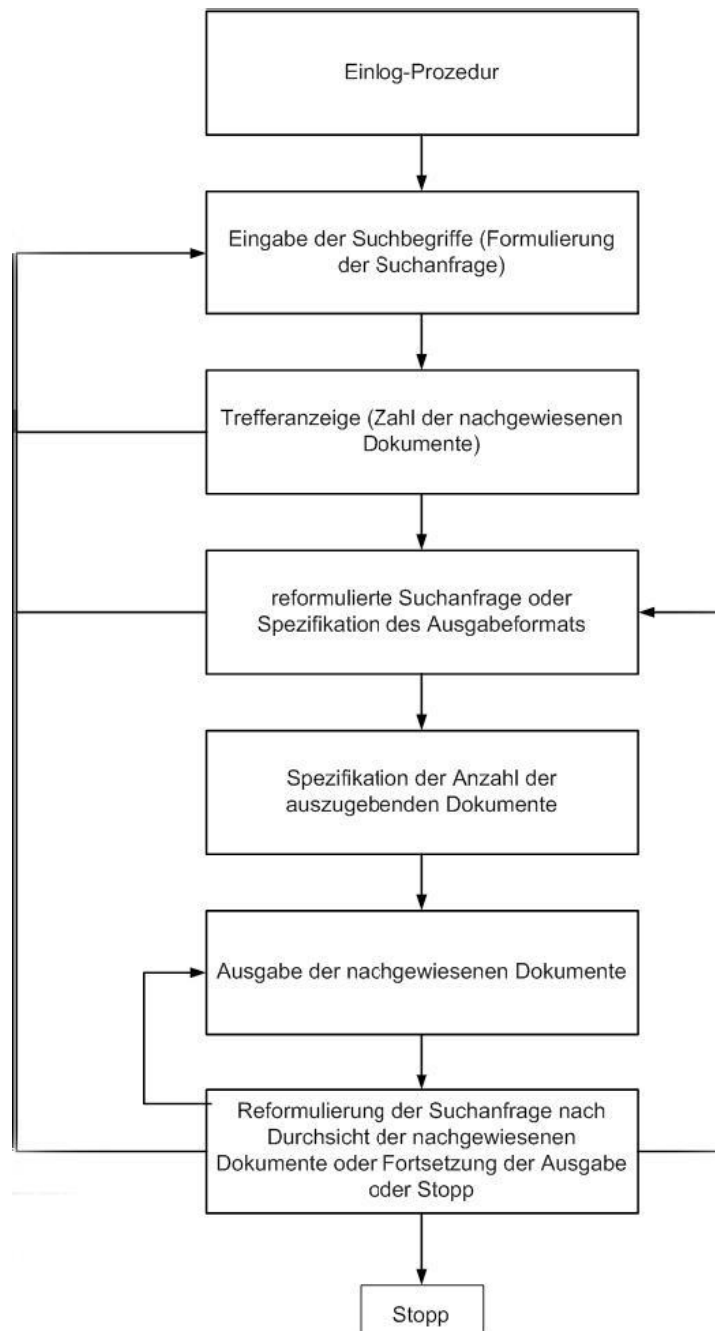


Abbildung 4.3: Informationssuchprozess nach Salton und McGill

## 4.2 Interfacedesign von Search-Term-Recommendierdiensten

Unter Search-Term-Recommendierdienste versteht man Applikationen, die dem Benutzer in der Formulierung seines Suchbegriffs unterstützen sollen. Dabei kann man *Pre-Search-Recommendations* und *Post-Search-Recommendations* unterscheiden.

Bei Pre-Search-Recommendations handelt es sich beispielsweise um Suchempfehlung während der Eingabe der Suchanfrage. In den meisten Fällen öffnet sich ein Fenster bzw. eine AJAX-Drop-Down-Menü mit Vorschlägen zur Komplettierung des Eingabebegriffs. Ziel ist es die Formulierung zu erleichtern und/oder Fehler zu korrigieren.

Post-Search-Recommendation sind Empfehlungen auf den Ergebnisseiten. Ein Beispiel wären die suchverwandten Empfehlungen am Ende der Trefferliste, realisiert durch Hyperlinks. Nach Lewandowski (2005), „laden“ diese Empfehlungen zur kognitiven Suche (Berrypicking) ein oder dienen ebenfalls der Fehlerkorrektur.



Abbildung 4.4: "Pre-Search-Recommendations"; Google

Wie beschrieben, besteht nach wie vor beim Benutzer der Wunsch nach einer schnellen Suchmaschine. Mit schnell, ist nicht nur die schnelle Ergebnislieferung gemeint, sondern die Schnelligkeit mit der der Benutzer in der Lage ist, die Suchmaschine zu bedienen. Darunter fällt die Formulierung der Anfrage, das Erlernen von Shortcuts, etc. . Um diesem Anspruch gerecht zu werden, verfolgen die Recommendingdienste die Grundidee einer Suchprozessbeschleunigung. Lässt man das Erlernen der Bedienung und Handhabung dieser Dienste außen vor lässt, erleichtern diese Applikationen die Suche spürbar in Qualität und Zeit( Bruza et al., 2004).

Wie im Teil „Das klassische Suchmaschinenmodell“4.1 und in Abbildung4.3 zu sehen ist der Vorgang der Informationssuche grob in drei Phasen zu unterteilen. Die erste Phase betrifft die (erste) For-





Abbildung 4.5: "Post-Search-Recommendations"; Google

mulierung der Anfrage. In der zweiten Phase werden entsprechend der Anfrage gefunden Ergebnisse präsentiert. Die Trefferanzeige. Der Benutzer durchforstet intellektuell die Ergebnisliste nach seinen gesuchten Informationen. Sollte der Benutzer fündig werden, wäre die Suche beendet. Das würde bedeuten, dass eine Anfragereformulierung, nicht zwangsläufig eintritt. Die dritte Phase schießt den einmaligen Prozess zu einem Suchzyklus, da der Benutzer gegebenenfalls wieder zu Phase 1 gelangt. Um genauer auf Recommenderdienste eingehen zu können, bedarf es einer anderen Unterteilung. Die Phase der Anfrageformulierung ist zu der Kategorie der Pre-Search-Recommenderdienste zu zählen, während die zweite und dritte Phase Teile der Post-Search-Recommenderdienste sind. Die Pre-Search-Recommendations nehmen bereits vor der eigentlichen Eingabe bzw. der Beendigung der Suchanfrage, Einfluss auf den Benutzer. Dies kann auf unterschiedliche Art erfolgen. Sei es, dass der Blick des Benutzers auf bestimmte Grafiken gelenkt wird, auf die er klicken soll, oder dass eine Komplettierung der Eingabe erfolgt. Um somit alle Möglichkeiten der Pre-Search-Recommenderdienste zu beschreiben, empfiehlt sich statt des Begriffs der Anfrageformulierung, der Begriff der Anfragespezifikation, welcher auch im weiteren Text dieser Arbeit verwendet wird. Die geteilte Betrachtung der Post-Search-Recommenderdienste in Trefferanzeige und Anfragereformulierung bleibt bestehen, da eine Anfragereformulierung nicht zwangsläufig eintritt.

## 4.3 Pre-Search-Recommendations

### 4.3.1 Anfragespezifikation

Bei der Anfragespezifikation des Informationssuchprozesses, drückt der Benutzer sein abstrakt konzipierten Informationsbedarf in einem Anfrageformat aus, mit der die Suchmaschine dann arbeitet. Dabei gibt es viele verschiedenen Arten, wie der Benutzer sein Informationsbedürfnis spezifizieren kann.

Anders als beim klassischen Information Retrieval, bei dem auf einen konkreten Informationsbedarf die Antwort (*exact matching*) ein Dokument oder im Fall des problemorientierten Informationsbedarfs eine Menge von Dokumenten zur Beantwortung einer Suchspezifikation zur Folge hat, ist die Spanne der unterschiedlichen Anfragearten bei der Suche im WWW weiter zu differenzieren (Lewandowski, 2005). Broder (2002) hat auf Basis einer Nutzerbefragung sowie *AltaVista* querylogs, die Anfragespezifikation in drei Arten unterteilt: *Navigationsorientiert*, *Informationsorientiert* und *Transaktionsorientiert*.

Bei navigationsorientierten Anfragen möchte der Benutzer eine bestimmte Website erreichen, von deren Existenz er bereits weiß oder deren Existenz er vermutet.

Dem klassischen Information Retrieval am ähnlichsten, sind die informationsorientierten Anfragen. Ziel ist das Finden thematisch passender Dokumente. Weitere Initiative ist lediglich beim Lesen und Durchschauen der Dokumente vom Benutzer verlangt. Wie schon erwähnt, ist die Spannweite von gestellten Anfragen im WWW sehr breit. Sie reicht von unspezifischen „Einwort“-Anfragen bis zu hochspezifischen Anfragen. Diese Besonderheit ist bei Systemen im klassischen Information Retrieval nicht der Fall, da diese in den meisten Fällen Verwendung bei Fachbenutzern finden.

Transaktionsorientierte Anfragen zielen auf eine Transaktion nach dem Auffinden einer bestimmten Seite ab. Die können beispielsweise der Kauf eines Produktes oder der Download einer Datei sein. Diese Art Anfragen sind ebenfalls eine WWW-Besonderheit und kommen nur in gesonderten Information Retrievalsystemen vor.

Das Ergebnis der Auswertungen nach Broder (2002) ergab, dass alle drei Arten von Anfragen nahezu zu gleich großen Teilen bei allen Anfragen vertreten waren. Dementsprechend sollte das beim Design eines Suchinterfaces oder einer Suchmaschine überhaupt, berücksichtigt werden. Nach Broder (2002) orientierte sich die erste Suchmaschinengeneration sehr stark am Information Retrieval. Die zweite Stufe, konnte durch Einbindung webspezifischer Daten sowohl informationsorientierte als auch transaktionsorientierte Anfragen bedienen. In der dritten Stufe, sollen transaktionsorientierte Anfragen unter Einbezug des Nutzers noch besser verarbeitet werden können.

Als Pendant zu der Aufteilung von Broder (2002), ist Hearst (2009) zu erwähnen. Hearst (2009) hingegen orientiert sich bei seiner Unterteilung mehr an der Eingabe selbst, als an der Benutzerintention. Die zwei wichtigsten Dimensionen im Anfragespezifikationsprozess nach Hearst (2009) sind:

- Die *Art der Information*, nach der der Benutzer sucht. Anfragespezifikation kann ein breites Spektrum von Sprachsätzen, über Schlüsselwörter und Schlüsselphrasen bis hin zu stark syntaxlastigen, sprachbasierten Kommandoanfragen umfassen.
- Der *Interface-Mechanismus*, mit dem der Benutzer interagiert, um nach den erforderlichen Informationen zu suchen. Dies beinhaltet Kommandozeileninterfaces, grafische Benutzungsoberflächen und Interfaces für navigierende Links.

## Textuelle Anfragespezifikation

Anfragespezifikationen über textuelle Informationen sind für gewöhnlich textueller Natur.

### *Volltextsuche*

Einige Probleme bei der Suche mit Suchsystemen, sind auf eine unvollständige Indexierung zurückzuführen. Beispielsweise könnte eine Indexierung für ein System, welches vorzugsweise Informationen zu Büchern bereitstellt, so aussehen: Autor, Titel, ISBN. Diese sehr spezifischen Informationen müssten dann beim Benutzer bekannt sein, um in der Datenbank fündig zu werden. Eine „grobe“ Suche auf einem Themengebiet wäre so nicht möglich. Dahingegen bietet die Volltextsuche eine Verbesserung, da eine Indexierung über den gesamten Inhalt des Werkes erfolgt. Somit lassen sich Relevanzwerte für die Suche berechnen und beispielsweise eine Kategorisierung auf gewisse Themenbereiche vornehmen. Die Volltextsuche hat sich als Norm bei Suchsystemen mit Suchen über textuellem Inhalt etabliert.

### *Keyword-Anfragespezifikation*

Die meist verbreitete Form der textuellen Anfragespezifikation, ist die Keyword-Anfragespezifikationen. Keyword-Anfragespezifikationen beinhalten eine Liste von ein oder mehreren Wörtern oder Phrasen. Diese werden dazu verwendet um nach Dokumenten zu suchen, welche die Wörter der Listen enthalten. Dabei können Keyword-Anfragen die unterschiedlichsten Formen der Sprache annehmen. Manche Keyword-Anfragen beinhalten Listen mit verschiedenen Wörtern und Phrasen, viele andere Listen bestehen aus geläufigen Nomen und wieder andere setzen sich aus syntaktischen Fragmenten mit z.B. Präpositionen und Verben zusammen. Seit den frühen Tagen des WWW werden statistische Erhebungen zur Länge der Anfragespezifikationen durchgeführt. Nach einer Studie von Jansen u. a. (2007) betrug die durchschnittliche Länge einer Anfrage 2,8 Terme. Frühere Suchmaschinen bedienten sich diesen statistischen Werten, um die Ergebnisse zu ranken. Schnell stellte sich aber heraus, dass der Be-

nutzer diese Rankingmethode für zu undurchsichtig bewertete<sup>2</sup> (Lake, 1998). Als geeignete Methode stellte sich die *Term-Konjunktion* heraus. Bei der Fülle der im WWW vorhandenen Dokumente war es ein Leichtes, Dokumente als Ergebnisse zu liefern, die alle angegebenen Terme (zum Beispiel drei) in Konjunktion enthielten.

#### *Vollständige Suchstatements*

Als eine weitere Art von textueller Anfragespezifikation, sind die *vollständigen Suchstatements* zu nennen. Ein konkretes Beispiel aus der Praxis wäre da die Suchmaschine *WolframAlpha*<sup>3</sup>. Bei Fragen wie der Art „Wie groß ist Barack Obama?“, wird in einer internen Datenbank versucht, die passende Antwort zu berechnen, und nicht wie gewöhnlich auf Dokumente im WWW mit passendem Inhalt zu verweisen. Andere Systeme dieser Art, machen sich die Struktur der Anfrage zu nutzen, indem sie nach Dokumenten suchen, in denen die gestellte Anfrage enthalten oder Nomenphrasen aus der Anfrage vorkommen.

Untersuchungen haben gezeigt, dass unerfahrene Benutzer rein intuitiv versuchen eine Anfrage in ganzen Sätzen zu formulieren. Ferner mussten Anwenderneulinge von herkömmlichen Suchmaschinen zunächst lernen, dass sie mit einer Ergebnismenge zu rechnen, und diese zu sichten hatten (Pollock u. Hockley, 1997).

Amüsant dabei zu erwähnen, dass die Suchmaschine *AskJeeves*<sup>4</sup> es ihren Benutzer erlaubte, die Anfragen in ganzen Sätzen zu stellen. Obwohl *AskJeeves* diese Funktionalität nie unterstützte. Dennoch aber machte sie sich die Sympathie der Benutzer zu eigen, die glaubten, dass es möglich sei (Hearst, 2009). Zusammenfassend ist zu sagen, dass alle aktuellen Systeme, die die Möglichkeit von Anfragespezifikation in vollständigen Sätzen erlauben, ihre Berechnungen aufgrund syntaktischer Regeln vornehmen. Mit mathematischen Formeln stellt man eine Mutmaßung zur gesuchten Information des Benutzers an.

---

<sup>2</sup>Benutzerfrage: “Warum braucht das höchst gerankte Dokument nur mindestens eins von drei Anfragentermen enthalten?”

<sup>3</sup><http://www.wolframalpha.com>

<sup>4</sup><http://www.askjeeves.com>

### Anfragespezifikation direkt über das Interface

Die meisten Suchsysteme verwenden zur Anfragespezifikation ein Interface mit einer Eingabebox und einem Aktivierungsbutton. Dabei kann die Beschriftung des Buttons von „Search“, „Grok“<sup>5</sup> bis hin zum einfachen nichttextuellen Lupenpiktogramm bei *Bing*<sup>6</sup> variieren. Alternativ zum Klick auf den Button, lässt sich die Suche mit der Betätigung der *Return*-Taste absenden. *Google* bietet beispielsweise noch einen zusätzlichen Button „Auf gut Glück!“ an, mit dem man nach Eingabe eines Zeichens direkt auf die erste Webseite der Ergebnismenge gelangt, die die Suche danach erzielt hätte. Interessante Untersuchungen nach Belkin u. a. (2003) haben gezeigt, dass die Größe der Eingabebox und die Anzahl der möglichen Eingabezeichen maßgeblich Einfluss auf die Anfragespezifikation des Benutzers nehmen. Je größer die Eingabebox und je größer die mögliche Anzahl der Eingabezeichen, desto länger waren die Eingaben. Weiter noch positionieren manche Interfacedesigner im Eingabefeld einen Hinweis, der angibt, welche Anfragespezifikation vom Benutzer verwendet werden soll (Keywords, ausformulierte Fragen, etc.).

Bei vielen Systemen stehen im Bereich der Eingabebox diverse Optionen zur Verfügung, mit denen sich die potentielle Ergebnismenge beeinflussen lässt. Der erfahrene Benutzer kann unter „Suchoptionen“ die Suche individuell einstellen oder unter verschiedenen Unterkategorien wie Bilder, Web, etc. eine Auswahl treffen. Beispielsweise bietet *Bing*<sup>7</sup> dem Benutzer die Möglichkeit unter „Shopping“, „Bildern“, „Videos“, „News“, etc. und einer Spracheinstellung („nur Deutsch“, „Seiten aus Deutschland“) zu wählen. Bei *Ebay*<sup>8</sup> lässt sich die Suchkategorie („Art“, „Baby“, „Books“,...) einem alphabetisch sortiertem Drop-Down-Menü einstellen.

Systeme, ohne eigene Datenbank importieren die Ergebnismengen. Die Resources und Domains sind in den meisten Fällen wählbar. Beispielsweise verwendet *Grokker*<sup>9</sup> die Ergebnisse von *Wikipedia*<sup>10</sup>, *Yahoo!*<sup>11</sup> und *Amazon Books*<sup>12</sup>. Weiter noch besteht die Möglichkeit einer reinen Keyword-Anfragespezifikation oder den Zeitraum der gesuchten Dokumente mit einem Schieberegler einzuschränken. Andere Formen der Vorabselektion bieten Systeme wie *Searchme*<sup>13</sup> oder *Newsmap*<sup>14</sup>.

*Searchme* bietet in einem Karteireitersystem aktuelle Informationen aus den Bereichen „Latest-

---

<sup>5</sup><http://www.grokker.com>(Grokker.com)

<sup>6</sup><http://www.bing.com>

<sup>7</sup><http://www.bing.com>

<sup>8</sup><http://www.ebay.com>

<sup>9</sup><http://www.grokker.com>

<sup>10</sup><http://www.wikipedia.org>

<sup>11</sup><http://www.yahoo.com>

<sup>12</sup><http://www.amazon.com>

<sup>13</sup><http://www.searchme.com>

<sup>14</sup><http://www.newsmap.jp>

News“, „Sports“, „Music“, „Featured“, „Entertainment“, „Videos“ an, die nach Popularität sortiert sind. Bei Klick auf die Reiter lassen sich die jeweiligen Informationen in den einzelnen Bereichen einsehen.

Ist bei den meisten Suchsystemen zu beobachten, dass die Startseite lediglich aus Logo, Eingabebox, Button und diversen Optionen besteht, findet man auf *Newsmap* bereits zu Beginn an eine mit aktuellsten Informationen ausgefüllte Benutzungsoberfläche vor (Genaueres in 4.5). Durch Klick auf die Rechtecke, gelangt der Benutzer zu den jeweiligen Informationen. Der Übergang zwischen Anfragespezifikation und Trefferanzeige ist fließend. Auf beiden Benutzungsoberflächen ist es dem Benutzer möglich, sich lediglich per Mausklick zu bewegen, zu suchen und zu finden (Browsing). So kann ein Informationsbedarf durch reine Interaktion mit dem System befriedigt werden, ohne dass eine herkömmliche Suchanfrage getätigt werden muss.

Ein Novum bietet die Suchmaschinen *Picollator*<sup>15</sup>. Wie schon ihr Werbeslogan vermuten lässt, „Search by Faces“, ist das System auf die Suche von Portraitfotos spezialisiert. Neben einer herkömmlichen Eingabe einer Anfragespezifikation ist es dem Benutzer möglich ein Bild vom eigenen Rechner hochzuladen oder die URL des Bildes anzugeben. *Picollator* sucht nach ähnlichen Bildern im WWW und rankt diese in einer Liste.

### **Dynamische Termempfehlung während der Anfragespezifikation**

Während schon seit längerem dem Benutzer, zusätzlich zur Trefferanzeige, Suchbegriffempfehlungen angeboten werden, erfahren Entwicklungen zur Termempfehlung während der Formulierung der Anfragespezifikation zunehmender Beliebtheit. Manche Systeme versuchen den Benutzer mit dynamischen Termempfehlungen noch vor dem Retrievalergebnis zu unterstützen. Andere Systeme empfehlen dynamisch Dokumente, die mit den bisherigen Eingabezeichen matchen. Dynamische Termempfehlungen versprechen eine sofortige Lösung zwischen der Formulierung der Anfragespezifikation und der Durchsicht einer langen Liste von Termempfehlungen.

Üblich sind dynamische Termempfehlungen mit *Prefix-Matching*. Das bedeutet, dass Empfehlungen (gegebenenfalls Komplettierungen) zur bereits erfolgten und/oder andauernden Eingabe ausgesprochen werden. Bei einer Eingabe von beispielweise „ap“, erhält man Empfehlungen der Art „apple“, „application“ oder „apologize“. - Gerade eben, die Begriffe die ebenfalls „ap“ als Präfix enthalten. Nach Eingabe des Buchstabens „b“ auf *Ebay*, öffnet sich ein Drop-Down-Menü mit Empfehlungen von „BMW“, „Bose“ etc.. Lautet die Eingabe jedoch „ba“, so erhält man Empfehlungen der Art „Bastler“, „Ray Ban“, „Baby“. Also nicht zwangsläufig Begriffe deren Präfixe der Eingabe gleichen, sondern Suchbegriffe, nach denen häufig auf *Ebay* gesucht wurde und die die eingegebenen Zeichen enthalten.

---

<sup>15</sup><http://www.picollator.com>



Abbildung 4.6: Deutsche Seite von Google

Die gängigste grafische Umsetzung ist das Drop-Down-Menü. Bei Eingabe im Suchschlitz, öffnet sich darunter eine Box, die Empfehlungen zur Eingabe enthält. Anbieter wie *Bing* weisen zusätzlich mit einem kleinen Pfeil am linken unteren Rand auf die Empfehlungen hin. *Ebay* bedient sich einer Lösung, bei der die dynamischen Termempfehlungen durch eine graue Linie von der eigentlichen Suchbox separiert aufgeführt werden. Dabei variiert die Anzahl der vorgeschlagenen Termempfehlungen von System zu System. Auf der deutschen Seite von *Google* werden (soweit möglich) immer zehn Terme oder Phrasen vorgeschlagen (bei *Ebay* sind es konstant sieben). Rechts neben den Termempfehlungen findet sich zu jeder eine Angabe zur potentiellen Treffermenge. Die amerikanische Seite verzichtet auf eine solche Angabe, stellt aber dafür die in den Empfehlungen wieder auftauchenden Eingabeterme in fettgedruckter Schreibweise dar.

Um eine detaillierter Analyse vorzunehmen, soll die DIN ISO EN 9241, die Norm für Mensch-Maschine-Interaktionsergonomie, kurz berücksichtigt werden. Insbesondere Teil 110, die Grundsätze der Dialoggestaltung.

Die Aufgabenangemessenheit kann man als erfüllt betrachten, da das Grundziel, die Informationssuchprozessbeschleunigung, sowie die Benutzerunterstützung gegeben sind. Da die meisten dynamischen Termempfehlungen als solche kenntlich gemacht sind, ist die Selbstbeschreibungsfähigkeit vorhanden.

Die meisten Systeme bieten diverse Einstellmöglichkeit, wie beispielsweise die komplette Aktivierung bzw. Desaktivierung der Recommendingdienste. Daher können die Steuerbarkeit und die Individualisierbarkeit als vorhanden betrachtet werden. Die Fehlertoleranz der Termunterstützungen ist zufriedenstellend. Sucht der Benutzer zum Beispiel nach dem Begriff „Pizza“, gibt aber „Pzza“ ein, so wird dies



Abbildung 4.7: US-Amerikanische Seite von Google

von den meisten Systemen mit Empfehlungen wie „Pizza Hut“, „Pizza“, etc. kompensiert. Desweiteren stehen in den meisten Fällen „Hilfe“-Links zur Verfügung, die Informationen und Erklärungen zur Handhabung der Termempfehlungsmechanismen enthalten.



## 4.4 Post-Search-Recommendations

### 4.4.1 Trefferanzeige

Die gängigste Art Suchergebnisse im Display darzustellen, ist eine vertikale Liste von Informationen, welche die gefundenen Dokumente zusammenfasst. Typischerweise besteht ein Eintrag in der Liste aus Dokumententitel und einem Auszug an Metadaten, wie Datum, Autor, URL, Länge des Artikels und einer Zusammenfassung von relevanten Teilen im Dokument. Die Darstellung eines Dokuments in dieser Form in einer Liste, wird *Search Hit* genannt. Die Sammlung von Informationen wird manchmal als *Document Surrogate* bezeichnet. Die Search Hits mit Hyperlinks sind sozusagen als Platzhalter oder Ersatz für die eigentlichen Dokumente mit den gesuchten Informationen zu verstehen. Dabei spielt die Qualität dieser Platzhalter eine entscheidende Rolle in der Relevanzbewertung der Benutzer. Selbst das höchstgelistete Dokument wird weniger frequentiert, insofern der Titel nicht informativ genug ist oder irreführend wirkt.

Um berechnen zu können, welche Form von Document Surrogates am ehesten zu Suchmaschinen mit bestmöglicher Usability dienlich sind, haben Grimes u. a. (2007) einige Untersuchungen durchgeführt. Von den Testpersonen wurde verlangt, sich bzgl. eines Themas für ein Listenelement aus der Trefferanzeige zu entscheiden. Dabei wurde das Klick-Verhalten der Benutzer mitprotokolliert und ausgewertet. Der Test motivierte sich weiter dadurch, dass die Benutzer über das Rankingverhalten und die Relevanzbewertungen des Systems im Vorfeld aufgeklärt wurden. Es wurden 10.000 Paare von Zusammenfassungen evaluiert, bei denen der hitA vor hitB in der Liste stand. Der hitB erhielt mehr Klicks bei folgenden Konstellationen (Auszug): hitB enthält eine Zusammenfassung, hitA nicht; Zusammenfassung hitB ist länger als Zusammenfassung hitA; Titel von hitB enthält mehr Anfrage-Matches, als Titel von hitA; etc.

- Wenn möglich sollten alle Anfrageterme im Surrogate auftauchen, und ihre Beziehung zur Webseite verdeutlichen.
- Wenn die Anfrageterme im Titel für den Hit vorkommen, ist ein Vorkommen in der Zusammenfassung nicht zwangsläufig notwendig.
- Die Länge und die Komplexität der URLs sollte reduziert werden. Weiter sollten die URLs so platziert werden, so dass ihre Beziehung zur Anfragespezifikation deutlich wird.
- Einen positiven Effekt auf das Klickverhalten hatte das Vorkommen der Begriffe: *official, and, tourism, attractions, sexy* and *information*. Wogegen *encyclopedia, wikipedia* and *free* einen negativen Effekt darauf hatten.

Wie oben erwähnt, bieten moderne WWW-Suchmaschinen in der Trefferanzeige Zusammenfassungen (*Summary, Snippet, Abstract*) zu den einzelnen Listenelementen an. Im Gegensatz zu älteren Vari-

anten, bei denen lediglich die ersten fünf Zeilen eines Dokumentes als Zusammenfassung dargeboten wurden (J. Kupiec, 1995), werden sogenannte *Keyword-in-Context-Extractions* (KWIC, auch *Anfrageorientiert*), zur Darstellung im Display verwendet. Dabei dienen Satzfragmente, vollständige Sätze oder Gruppen von Sätzen, die aus dem Volltext des Dokumentes extrahiert wurden, und Terme aus der Anfragespezifikation enthalten, zur Darstellung der Zusammenfassungen in den Surrogates. Standardabstracts beschränkten sich auf die Hauptthematik des Dokumentes. Anfrageorientierte Abstracts hingegen, fassen das Vorkommen der Anfragespezifikation und deren Verknüpfung im Dokument zusammen. So kann durch Betrachtung eines anfrageorientierten Abstracts, der Informationsbedarf befriedigt werden, ohne dass das eigentliche Dokument eingehend betrachtet werden muss.

Da lange Zeit die Speicherung von kompletten Dokumenten ein kostenintensives Unterfangen war, wurden lediglich die ersten fünf Zeilen eines Dokumentes gespeichert und für den Abstract verwendet. *Google* begann als erstes Unternehmen Volltext-Dokumente zu archivieren und den Inhalt für anfrageorientierte Zusammenfassungen zu verwenden. Heute sind KWIC-Zusammenfassungen zu einem Standard für WWW-Suchmaschinen-Ergebnisdisplays geworden (Hearst, 2009).

### Weitere Features bei der Trefferanzeige

Neben den Standard Metadaten wie, Titel, Autor, Datum, etc. sind noch andere Features zu nennen, die sich im Laufe der Zeit als sinnvoll für die Präsentation von Suchergebnissen herausgestellt haben. Die meisten WWW-Suchmaschinen stellen 10 Hits pro Ergebnisseite dar. Bei statistischem Ranking wird zusätzlich (in den meisten Fällen) ein numerischer Wert angegeben, der Aufschluss über die berechnete Relevanz gibt. Frühere Versuche von z.B. *Google*, 30 Hits pro Seite anzubieten, wurden aufgrund der ungleich höheren Ladedauer schnell wieder verworfen. Interessant zu beobachten, dass die 2009 an den Start gegangene WWW-Suchmaschine *Bing*<sup>16</sup> von Microsoft bei der Darstellung der Trefferliste bei der Suche nach Bildern sich nicht auf eine konkrete Anzahl an Hits pro Seite beschränkt. Die Ergebnisse werden rasterförmig als Thumbnails von oben nach unten geladen. Durch Scrollen können alle gefundenen Bilder durchgesehen werden. Die meisten grafischen Suchinterfaces bieten bei Klick auf Dokumententitel/Thumbnail, eine Vorschau des Dokumenteninhalts in einem neuen Fensters oder einer Box an. So wird bei *Bing* durch Mauszeiger-Rollover über die Video-Thumbnail, eine Vorschau abgespielt.

Ein weiteres Feature sind die *Sitelinks*. Diese Links befinden sich meist unter dem Titel und der Zusammenfassung der jeweiligen Hits und verweisen auf oft besuchte bzw. gesuchte Unterseiten. Das Feature basiert gänzlich auf „Benutzerklickstatistiken“ oder auf „Verweilzeiten“ auf den Unterseiten. Sucht man zum Beispiel bei *Google* nach „labs“, so bekommen man als obersten Hit *Google Labs* ausgegeben. Im selben Hit befinden sich dann Sitelinks zu „Trends“, „Code Search“, etc.

---

<sup>16</sup><http://www.bing.com>

## Visualisierung der Trefferanzeige

Eine Vielzahl von Ideen zur Visualisierung von Informationen wurde zur Darstellung von Retrieval Ergebnissen getestet. Der Großteil, aber überlebte nicht im Mainstream der Suchinterfaces. Neben dem beschriebenen „klassischen“ Aufbau mit Sucheingabebox und Suchbutton finden sich jede Menge anderer interessanter Visualisierungsansätze im WWW. Insbesondere Multimedia-Suchmaschinen, die Ergebnisse verschiedener Medientypen auf einer Oberfläche präsentieren, sind bei der Trefferanzeige experimentierfreudig.

Ein kurzer Blick auf konkrete Beispiele lässt das schnell erkennen. Weitere ausführlichere Beschreibungen zu aktuellen Systemen finden sich in „Die betrachteten Systeme“ 4.5.

*Newsmap*<sup>17</sup> ist eine WWW-Suchmaschine, die insbesondere auf die Präsentation und schnelle Verfügbarkeit von aktuellen Schlagzeilen und Nachrichten spezialisiert ist. Die Oberfläche ist farblich in Rechtecke unterschiedlicher Größe mit schlagzeilenartigen Beschriftungen unterteilt. Am unteren Rand befindet sich eine Legende zur Farbgebung. Beispielsweise enthalten rote Rechtecke Inhalte aus dem Bereich *World*. Je größer ein Rechteck, desto populärer ist die Schlagzeile und die weiterführenden Informationen dahinter. Klar ist, dass die Größe der Quader eine Relevanzcodierung ist, aber ob sich die Relevanz auf den Wert der Popularität (sehr oft gesuchte Informationen) bezieht, oder auf die Relevanz der Anfragespezifikation, ist schwer ersichtlich.

Einer anderen Form der Trefferanzeige bedient sich *Kartoo*<sup>18</sup>, bei der *Karten-Anzeige*. Die Hits sind in Form von Items (in dem Fall: Seitenvorschauen) unterschiedlicher Größe dargestellt. Je größer das Item, desto relevanter die enthaltenen Informationen. Die jeweiligen Items sind, insofern eine Beziehung untereinander besteht, miteinander verbunden. So erhält man bei der Suche nach „Bob Dylan“, den dazugehörigen Wikipediaeintrag als größtes Item und die direkten Verbindungen (Kanten) zur Seite eines Radiosenders, eines Musikwikis und der Homepage des Künstlers.

Die Suchmaschine *Grokker*<sup>19</sup> bietet zur Betrachtung der Trefferanzeige zusätzlich zur gerankten Liste eine *MapView* an. Die Dokumente sind gemäß der Kategorisierung in „Blasen“ dargestellt. Die Anzahl der enthaltenen Dokumente ist in der Größe der jeweiligen Blase codiert. Je größer die Blase, desto mehr Dokumente befinden sich in ihr. Dabei kann in die Blasen hinein- auf Dokumenten-Ebene, oder heraus auf Blasen-Ebene gezoomt werden. Zur Konfiguration der Ergebnisliste steht ein Schieberegler am linken Rand zur Verfügung. Die Trefferliste kann ohne Änderung der Anfragespezifikation vom Benutzer beeinflusst werden.

---

<sup>17</sup><http://www.newsmap.jp>

<sup>18</sup><http://www.kartoo.com>

<sup>19</sup><http://www.grokker.com>



Abbildung 4.8: Newsmap.jp

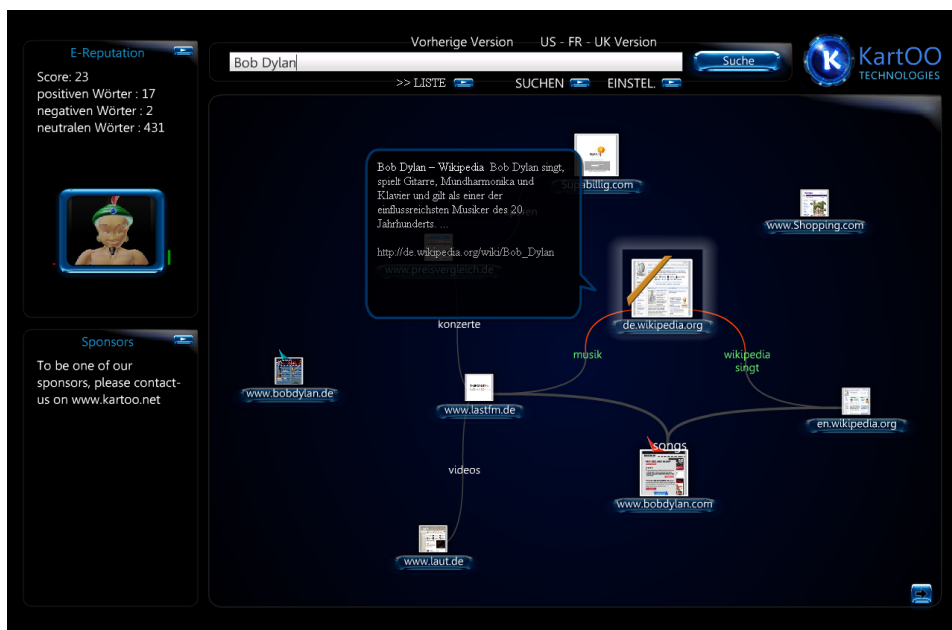


Abbildung 4.9: Kartoo - Karten-Anzeige

Abbildung 4.10: Grokker - „Blasen“1

*MyTag*<sup>20</sup> ist eine Taggingsuchmaschine, die ihre Daten von sechs Taggingsuchmaschinen unterschiedlicher Medientypen (Bibtexes, Fotos, Bookmarks,...) bezieht. Die Startseite erlaubt die Wahl der in die Suche involvierten Taggingsuchmaschinen (*Flickr*, *Youtube*, *del.icio.us*, *Connotea*, *Bibsonomy*, *BibTex*). Die Trefferanzeige erfolgt dann spaltenweise, wobei die Anzahl der Hits eingestellt und die Spaltenseiten einzeln geblättert werden können. Weiter findet auf der Benutzungsoberfläche neben der Ergebnisliste eine *TagCloud*, die sich aus allen Tags der angezeigten Ergebnissen berechnet. Die Tags werden gewichtet und je nach vorkommender Häufigkeit größer oder kleiner dargestellt. Beim Blättern der Spalten wird die *TagCloud* entsprechend angepasst. Für den Benutzer ist anhand der *TagCloud* ersichtlich, in welche thematische Richtung die Suche und die damit verbundenen Suchergebnisse tendiert. Die *TagCloud* kann direkt zur Anfragespezifikation verwendet werden. Der eigentliche Mehrwert erschließt sich nur für den angemeldeten Benutzer. Bei den meisten Taggingsuchsystemen, wie *Flickr* oder *del.icio.us* kann sich der angemeldete Benutzer persönliche Konten mit seinen Favoriten, Down- und Uploads anlegen. *MyTag* ermöglicht den Import aller Konten. Dadurch wird eine Personomy für jeden angemeldeten Benutzer berechnet. Zusätzliche Daten zur „Prägung“ der Personomy liefert das Klickverhalten (bei Klick auf einen Hit) des Anwenders. Eine personomyabhängige Sortierung der Trefferliste ist ebenfalls möglich. Weiter noch lassen sich durch einfachen Klick die eigenen Interessen (Personomy) bei der personalisierten Suche mit einbeziehen, oder lediglich die eigenen Daten durchsuchen (Braun u. a., 2007).

<sup>20</sup><http://mytag.uni-koblenz.de>

The screenshot shows the Grokker search interface. At the top, it says "See how Grokker can help your business" and "Selected Sources [2 of 3] Add/Remove". Below this, there are checkboxes for "Yahoo!", "Wikipedia", and "Amazon Books", with "Wikipedia" selected. The search term "obama" is entered in the search bar. The interface is divided into several sections:

- Left Panel:** "Refine Search" section with filters for "by keyword", "by date" (2007-12-08 to 2009-05-25), "by source", "by domain", and "Layout" options.
- Top Panel:** "Outline View", "Map View", "250 total results", "Zoom Back", "Expand View", "Working List (0 items)", "Email", and "Export".
- Center Panel:** A large circular bubble chart. A tooltip is visible over a bubble, showing details for "United States Senate career of Barack Obama". The tooltip includes a description, location, and source.
- Right Panel:** "Results: [1-5 of 5]" section with "Expand View" and "Expand Detail" options. It lists search results with titles like "Barack Obama presidential campaign, 2008", "Family of Barack Obama", "Public image of Barack Obama", "United States Senate career of Barack Obama", and "Political positions of Barack Obama".

Abbildung 4.11: Grokker - „Blasen“2

Die amerikanische Seite von *Google*<sup>21</sup> hat 2009 ebenfalls zwei Features zur Trefferanzeige und Unterstützung der Benutzer auf den Weg gebracht. Beim *Whonderwheel* befindet sich der Suchbegriff auf der Achse des Rades. An den Speichen, die um die Achse herum angeordnet sind, finden sich Suchtermempfehlungen, die nach *Google* in die Thematik des angegebenen Suchterms passen. Die *Timeline* stellt eine chronologische Verteilung der Suchergebnisse dar. So lässt sich ablesen, wie viele Seiten zu einem bestimmten Zeitpunkt für Anfragespezifikation relevant waren/sind. Per Klick lässt sich die Treffermenge auf einen Zeitraum beschränken. Ein Klick auf die *Timeline* oder auf eine, der Empfehlungen am *Whonderwheel* startet eine neue Suche und wird als neue Anfragespezifikation interpretiert.

Wie man sieht, ist der Übergang zwischen Trefferanzeige und Reformulierung der Anfragespezifikation desöfteren fließend. Im kommenden Abschnitt soll die Reformulierung der Anfragespezifikation dennoch gesondert und explizit betrachtet werden.

<sup>21</sup><http://www.google.com>



Abbildung 4.12: MyTag

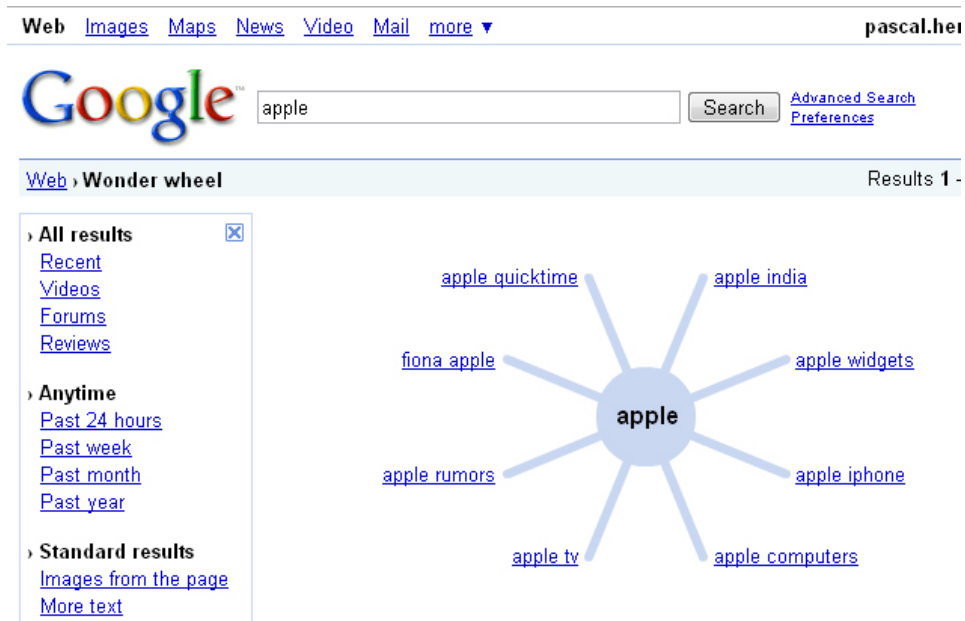


Abbildung 4.13: Google Wonderwheel

### 4.4.2 Reformulierung der Anfragespezifikation

Wie in „Das klassische Suchmaschinenmodell“ 4.1 beschrieben, besteht für gewöhnlich die Suchstrategie des Benutzers aus einer generellen Anfragespezifikation, also der Formulierung seines Informationsbedarfs, der Durchsicht der Suchergebnisse und, insofern die gesuchten Informationen nicht gefunden werden, aus der Veränderung der Anfragespezifikation um die Ergebnismenge zu verbessern. Dieser Zyklus wiederholt sich, bis der Benutzer die gesuchten Informationen gefunden hat, oder schlichtweg die Suche aufgibt. In den vorausgegangenen Kapiteln der Ausarbeitung wurden die Anfragespezifikation und die Trefferanzeige diskutiert. In diesem Teil soll der Schritt der Reformulierung der Anfragespezifikation betrachtet werden.

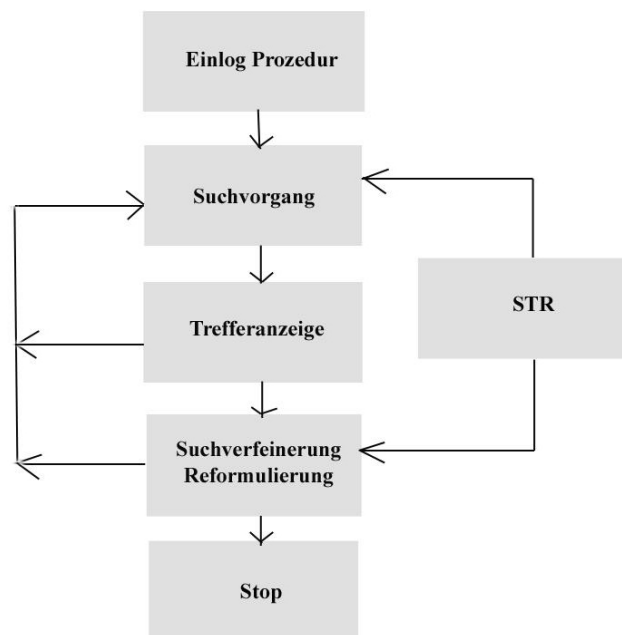


Abbildung 4.14: abstrakte Papierskizze - Eigene Darstellung



Die Skizze 4.14 stellt in sehr abstrahierter Form den Informationssuchprozess dar. Gut zu erkennen ist der sich gegebenenfalls wiederholende Zyklus der Formulierung und Reformulierung der Anfragespezifikation. Sehr grob lässt sich dieser Prozess in drei Phasen unterteilen. Eine Search Term Recommender Anwendung würde den Suchvorgang und die Suchverfeinerung durch Termempfehlungen unterstützen können.

Untersuchungen von WWW-Suchmaschinen Anfragelogs zeigen, dass es eine hohe Zahl an Reformulierungen der Anfragespezifikation gibt. Jansen u. a. (2005) untersuchten Logs, die über ein Zeitraum von 24 Stunden bei der Suchmaschine *Alta Vista 2002* protokolliert wurden. Dabei waren 72 Prozent der Suchsessions weniger als fünf Minuten lang. Die Zahl der Nutzer, die eine Reformulierung abgaben, belief sich auf 52 Prozent. Darunter befanden sich weiter 32 Prozent, die drei oder mehr Anfragen pro Session abgaben. Andere Studien kamen zu vergleichbaren Ergebnissen. So kann man sagen, dass trotz den Bemühungen während der Anfragespezifikation (Phase1) und der Trefferanzeige (Phase2), den Benutzer in der Informationsfindung zu unterstützen, die Reformulierung der Anfragespezifikation (Phase3) ein etablierter Bestandteil im Suchprozess bei den Benutzern ist. Nach Hearst (2009) u.a. weicht der Benutzer, hat er sich einmal für einen bestimmten Weg sein Informationsbedürfnisses auszudrücken entschieden, nur sehr selten gänzlich von der „Originalanfrage“ ab. Vielmehr versucht er diese vehement mit kleinen Variationen zu modifizieren.

„The user is subject to what psychologists call anchoring, i.e. the tendency to make insufficient adjustments to initial values when judging under uncertainty“. (Hertzum u. Frokjaer, 1996)

Beispiel: Russell (2006) beobachtete folgende Googlelogs zur Anfragespezifikation, wie viele Menschen bisweilen den neuen Harry Potter Band gekauft haben.

- Harry Potter and the Half-Blood Prince sales
- Harry Potter and the Half-Blood Prince amount sales
- Harry Potter and the Half-Blood Prince quantity sales
- Harry Potter and the Half-Blood Prince actual quantity sales
- ...

Um hilfreiche Alternativen zu bieten, gibt es verschiedene Entwicklungen, die auf den Reformulierungsprozess Einfluss nehmen. Geschätzte 10-15 Prozent der Suchanfragen enthalten Rechtschreib- oder Typografiefehler. Dementsprechend wichtig, ist ein Tool für Rechtschreibempfehlungen oder Rechtschreibkorrekturen. Für WWW-Suchmaschinen wurden sehr effektive Algorithmen zur potentiellen Fehlerdetektion entwickelt. Vor der Verwendung solch wirksamer Algorithmen, verglich eine Rechtschreibkorrektursoftware die Benutzereingaben mit vordefinierten Wörterbüchern. Desweiteren

war eine direkte Wortsubstitution nicht möglich. Mit der zunehmenden Nutzung des WWW wurde recht schnell deutlich, dass mit der Rechtschreibkorrektur der Anfrageformulierung es ein größeres Problem zu lösen galt. Der Großteil der Wörterbücher enthält nicht Firmennamen, Neologismen oder Phrasen, bestehend aus mehreren Wörtern. Zu den sich häufenden Schwierigkeiten kommt jedoch der Mehrwert der großen Datenbestände zum Nutzerverhalten hinzu. Mit den hohen Datenbeständen ist es möglich Rechtschreibempfehlungen durch das Vergleichen von Anfragen mit Anfragen ähnlicher Struktur, zu realisieren. Gemeinsamkeiten in Art und Weise, wie und welche Fehler gemacht wurden, lassen sich erkennen.

Eine Schlüsselfunktion in der Verbesserung der Rechtschreibempfehlungen war der Umstand, dass aus den Anfragelogs nicht nur die Rechtschreibbefehle, sondern auch die von den Benutzern selbstständig durchgeführten Korrekturen, extrahiert werden konnten. Gibt beispielsweise ein Benutzer den Begriff „Obama“ ein, korrigiert diesen in der darauffolgenden Anfrage zu „Obama“, so kann sich dieses „Verhalten“ ein Algorithmus zu nutze machen. Experimente mit Algorithmen, die die Korrekturen aufgrund von Anfragelogs berechneten, ergaben eine Ergebnisgenauigkeit von 88-90 Prozent bei 50 Prozent der fehlerhaften Anfragen. Bei WWW-Suchmaschineninterfaces findet man häufig solche Empfehlungen unmittelbar unter der Originalanfrage (aber über der Trefferanzeige; siehe *Google*)<sup>22</sup>. Die Empfehlung wird, für den Fall, dass der Benutzer den Fehler übersehen hat, am unteren Ende der Seite wiederholt. In den meisten Fällen bieten die Suchmaschinen Rechtschreibkorrekturen ohne zwingende Verwendung an. Schließlich kann es vorkommen, dass diese nicht mit der Intention des Benutzer übereinstimmen. Bei einer erfolglosen Suche ist nicht zwangsläufig ein Schreibfehler die Ursache. Beispielsweise sind einfach keine Ergebnisse zur Suche verfügbar. Sollte es sich aber um einen Eingabefehler handeln, wird der Benutzer sicherlich gerne die Empfehlung des Systems wahrnehmen. Um eine Balance dazwischen zu finden, bieten manche Suchmaschinen einige Hits mit der vermeintlich korrigierten Schreibweise verweben mit Hits der Originaleingabe an (Hearst, 2009), (Cucerzan u. Brill, 2004).

Die zweite wichtige Klasse der Dienste für eine Reformulierung der Anfragespezifikation, sind die automatisch empfohlenen Termvervollständigungen und Termerweiterungen (*Term Expansion*). Rechtschreibkorrekturen sind zwar ebenfalls Dienste der Anfragereformulierung, aber der Begriff Termerweiterung wird für gewöhnlich für Tools gebraucht, die alternative Wörter und Phrasen empfehlen. *Google* bietet seit kurzer Zeit suchverwandte Begriffe unterhalb der Hits an, die in die Thematik der Anfragespezifikation einzuordnen sind. Das Wort Termempfehlungen impliziert, dass sie ohne Benutzerinput generiert werden. Beispielsweise anhand von bestimmten Charakteristika der Treffermenge selbst, von Termen, die in den topgerankten Hits enthalten sind, einer Kombination aus beiden, aus einem "handgemachtem"Thesaurus oder mithilfe von Anfragelogs.<sup>23</sup>

---

<sup>22</sup><http://www.google.com>

<sup>23</sup>Um den hier gebrauchten Begriff Termempfehlungen noch einmal zu verdeutlichen. Es handelt sich nicht um die Empfehlungen vor oder während der Anfragespezifikation, auch wenn hier die gleiche Bezeichnung Verwendung findet. Gemeint sind die Empfehlungen, die dem Benutzer nach Abgabe der Anfrage und nach

Die meisten Usability Studien verzeichneten eine positive Benutzerresonanz bezüglich Termempfehlungen. Allerdings nur dann, wenn der Benutzer nicht dazu aufgerufen wurde, Relevanzurteile abzugeben oder aus einer sehr große Menge von vorgeschlagenen Termen, auszuwählen. Generell scheint es so, dass der Benutzer zur Reformulierung der Anfrage nicht aus einer eine Auswahl von multiplen Termen wählen möchte (Hearst, 2009).

Interfaces, die eine Reformulierung durch die Wahl eines einzelnen Terms erlauben (z.B. per Hyperlink), scheinen die bessere Lösung zu sein. Bei der Tagging-Suchmaschine *MyTag*<sup>24</sup> steht, wie schon erwähnt, neben der eigentlichen Trefferanzeige eine *TagCloud* zur Verfügung. Die TagCloud berechnet sich aus den am häufigsten vorkommenden Tags der angezeigten Hits und wird gewichtet dargestellt. Weiter werden die 7 ersten Tags von jedem Hit in der Trefferanzeige mit angezeigt. Durch Klick auf einen Term in der TagCloud oder auf ein Tag der Hits, wird dieser zur bereits eingegebenen Suchanfrage hinzugefügt. Anmerkung: Es handelt sich dabei nicht um eine Boolesche Verundung. Da *MyTag* lediglich eine Metasuchmaschine ist, wird die Suchanfrage so über die API an die involvierten Suchmaschinen weitergeleitet. Die von *Flickr*, *YouTube* etc. verwendeten Algorithmen, sind nicht veröffentlicht. Weiter noch ist es möglich einzelnen Suchbegriffe aus der Anfragespezifikation zu entfernen und somit eine neue Suche zu starten. Um eine gänzlich neue Suche starten zu können, müssen die Terme in der Eingabebox gelöscht werden (Braun u. a., 2007). Bei *Google* hingegen wird die Originalanfrage durch Klick auf eine Termempfehlung substituiert und eine neue Anfrage wird abgesendet.

Eine andere Art der Reformulierungsinformation boten White u. a. (2007) an: Die Anzeige von populären Zielwebseiten. Dabei speicherten sie über den Zeitraum von fünf Monaten tausende Suchaktivitäten von Benutzern. Diese Logs erlaubten ihnen die Serie der Aktionen zu rekonstruieren. Vom Besuch der Suchmaschinen Seite, zur Eingabe einer Anfrage, der Durchsicht der Treffer, der Besuch der Links bis hin zum Lesen der Webseite. Sie berechneten, wann solch ein *Session Trail* bei einer Stoppsite endete. Die Bedingung für eine Stoppsite war die Verweildauer auf ihr von mindestens 30 Minuten oder einer Aktivitätsänderung (z.B. Verwendung des Emailprogramms). Diese Stoppsiten wurden gespeichert und häufige Stoppsiten wurden entsprechend der Anfrage als Empfehlungen vorgeschlagen.

Die letzte, sehr bekannte Technik, um eine Anfragereformulierung zu unterstützen, die hier erwähnt werden soll, ist das *Relevance Feedback*. In seiner originalen Form beschreibt Relevance Feedback den Interaktionszyklus, bei dem der Benutzer die gefundenen Dokumente liest und diejenigen markiert, die relevant zu sein scheinen. Dadurch wird es dem Benutzer ermöglicht, die Größe der Treffermenge

---

Anzeige der Treffermenge angeboten werden.

<sup>24</sup><http://mytag.uni-koblenz.de>

und ihre Beschaffenheit zu verändern. In einer Variante, benutzt das System die Informationen durch die Markierung mancher Dokumente, um die Gewichtung für die Originaleingabeterme neu zu berechnen und neue Terme vorzuschlagen. Bei einer anderen Variante, schlägt das System dem Benutzer neue Terme vor, welcher dann daraus eine Untermenge auszuwählen hat, die seine Anfrage beschreibt. Die ursprüngliche Anfrage wird dann gelöscht und eine neue Treffermenge wird ausgegeben. Dokumente aus der alten Treffermenge können theoretisch in der neuen enthalten sein, aber mitunter in einer verschiedenen Rankingfolge. Für alle Relevance Feedback Techniken gilt, dass eine große Zahl an markierten Dokumenten, sich positiv auf das Resultat auswirkt (Harmann, 1992), (Lewandowski, 2005). Eine andere Methode, die als *Pseudo-Relevance-Feedback* bekannt ist, bedient sich weniger der Relevanzbewertung des Benutzers, sondern stuft die  $x$  ersten Dokumente als relevant ein und benutzt diese um die Anfrage mithilfe eines Relevance Feedback Algorithmus zu untermauern. Bedingung für einen korrekten Verlauf ist, dass die topranked Dokumente wirklich relevant sind.

## 4.5 Die betrachteten Systeme

Alle hier aufgeführten Systeme bedienen sich einem Suchinterface. Standardgemäß erfolgt zunächst eine Anfragespezifikation und das Absenden der selbigen. Was die Trefferanzeige, die Reformulierung der Anfragespezifikation und die Recommendingdienste angeht, gibt es sehr unterschiedliche Lösungsansätze.

Bei manchen Systemen ist die Frage nach Reiterer et al., 2007, wie viel Visualisierung wirklich nötig sei, durchaus berechtigt. Beispiel: Schreiben Sie sich eine Einkaufsliste auf! - Zeichnen Sie drei Äpfel, oder schreiben Sie „3 Äpfel“ auf?

Nach Reiterer et al. sind wir es gewohnt Listen zu lesen und zu interpretieren. So nett eine großzügige Visualisierung auch manchmal anzuschauen scheint, so lässt die Funktionalität der Informationssuche und vor allem der Informationsfindung hin und wieder zu wünschen übrig. Aber, ob mehr oder weniger Visualisierung dem Information Retrieval zu gute kommt oder nicht, ist mitunter Geschmacksache.

### 4.5.1 oSkope

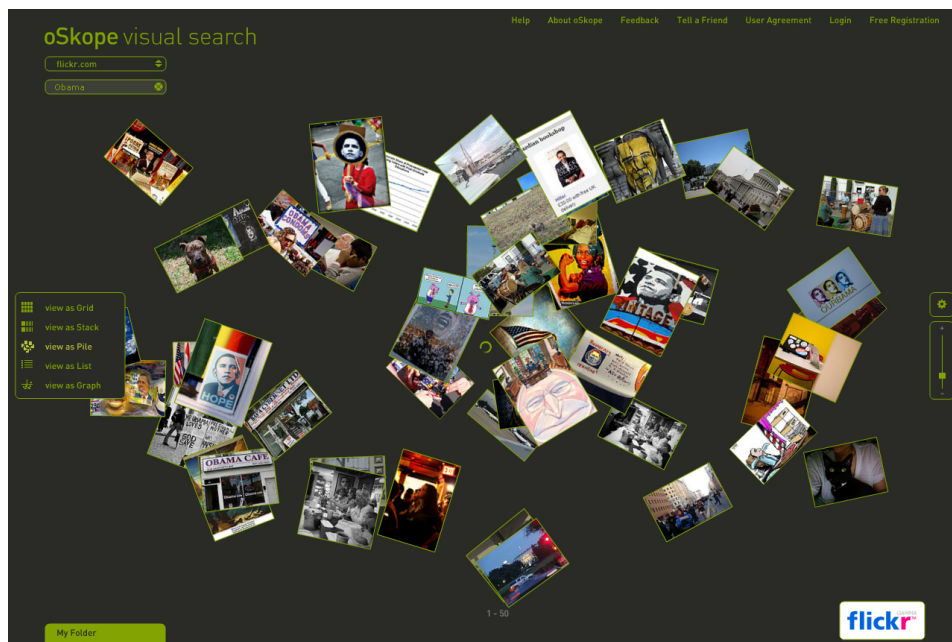


Abbildung 4.15: oSkope.com

*oSkope*<sup>25</sup> bezieht seine Informationen von verschiedenen Suchmaschinen wie: *Yahoo*, *Ebay*, *YouTu-be*, *Fotolia* oder *Amazon*. Das wahrscheinlich interessanteste Feature, sind die unterschiedlichen Ergeb-

---

<sup>25</sup><http://www.oskope.com>

nispräsentationen auf dem Bildschirm. Zur Auswahl stehen Raster, Stapel, „Haufen“, Liste und Graph. Bei Rollover über die Ergebnisse beispielsweise aus *Ebay*, erscheint ein kleines Vorschauenfenster mit auktionenrelevanten Informationen. Ebenso lassen sich Bilder in der Vorschau mit unterschiedlicher Auflösung darstellen. Mittels DragandDrop ist es dem Benutzer möglich, Gruppen nach eigenem Gusto zu bilden. Obwohl dem Benutzer ein Ordner zur Verfügung steht, in dem gefundenen Ergebnisse abgelegt werden können, sind die vorgenommenen Arrangements nicht speicherbar.

## 4.5.2 Kartoo

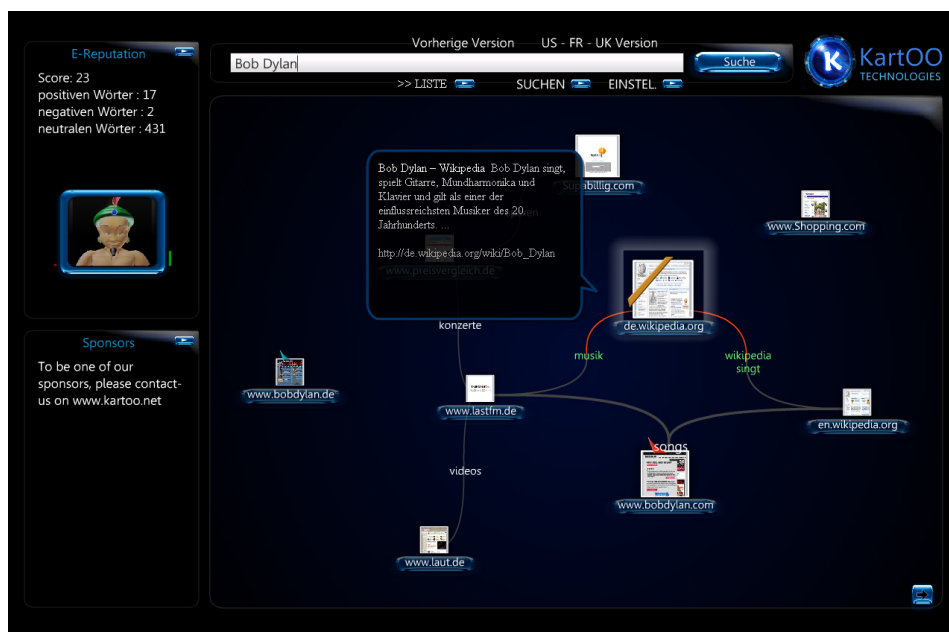


Abbildung 4.16: Kartoo.com

*Kartoo*<sup>26</sup> ist eine Meta-Suchmaschine. Nach der ersten Eingabe im Suchfeld erhält man eine Kartenansicht, die die Ergebnisse als Entitäten und relevante Wörter, sowie deren Verbindungen bzw. Verwandtschaften untereinander darstellt. Relevante Wörter werden benutzt um verschiedene Resultate in der Karte zu visualisieren. Weiter noch besteht die Möglichkeit diese durch Klicken der Suche hinzuzufügen und somit eine Reformulierung vorzunehmen. Zusammenhängende Informationen werden durch Kurven oder Linien verbunden, die in ihrer Darstellung an geografischen Landkarten erinnern. Dem Benutzer soll so der Zusammenhang der einzelnen Dokumente in der Ergebnismenge verdeutlicht und durch Interaktion näher in die Suche eingebunden werden. Sucht man beispielsweise nach „Madonna“, so erhält man die Homepage der Musikerin, den Wikipediaeintrag und einen MTV Link als größtdargestellte Ergebnisse. Als relevante Wörter sind „news“, „music“, „ringtones“ zu finden.

<sup>26</sup><http://www.kartoo.com>

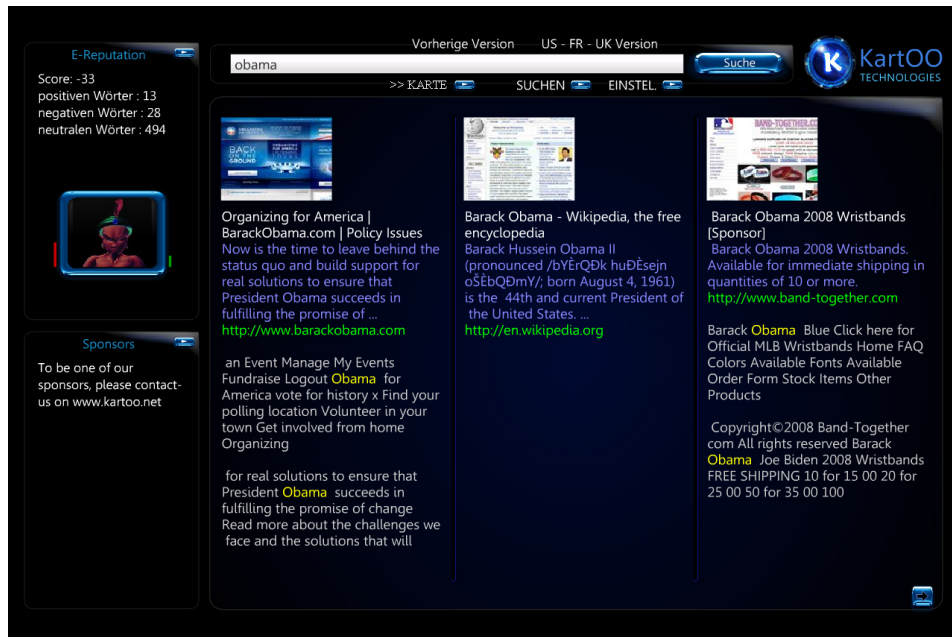


Abbildung 4.17: Kartoo.com

### 4.5.3 Grokker („ambient findability“)

Bei *Grokker*<sup>27</sup> findet man eine Vielzahl intuitiver Möglichkeiten, um entweder durch die Ergebnismenge zu navigieren oder diese zu verändern. Ein Beispiel ist die *Grokker's visual map*, welche auf der *balloon tree visual metaphor* basiert. In ihr werden die Ergebnisse in kreisförmigen „Blasen“ dargestellt. Je größer eine solche Blase ist, desto mehr Dokumente dieser Kategorie befinden sich in ihr. Für den Benutzer besteht die Möglichkeit in die Blasen bis auf Dokument-Ebene zu zoomen. Ein interessantes und nützliches Gimmick, ist ein Schieberegler in der linken Steuerungsleiste, mit dem sich die Ergebnismenge auf einen bestimmten Erscheinungszeitraum beschränken lässt. Desweiteren lässt sich die Trefferanzeige durch den Gebrauch von Filtern wie Datum, Autor oder Land dynamisch formatieren. Dementsprechend kann der Benutzer individuell den „Level of Detail“ im Ergebnisinterface bestimmen, in dem er mehr oder weniger Informationen zulässt. Die in *Grokker* realisierte Techniken lassen sich insbesondere bei der Durchsicht von großen Datenbeständen verwenden.

### 4.5.4 Searchme

*Searchme*<sup>28</sup> ist eine Suchmaschine, die einen besonders großen Schwerpunkt auf die Visualisierung legt. Ohne Eingabe, sind „News im Web“, „Sport“ und andere Kategorien der Suche verfügbar. Der horizontale Scroll (siehe *Apple*) dient zur Navigation durch die Webseitenvorschauen. Diese sind von

<sup>27</sup><http://www.grokker.com/>

<sup>28</sup><http://www.searchme.com>

See how Grokker can help your business News & Events | Blogs | Contact Us | Feedback | Help | Home

**groker** Selected Sources [2 of 3] [Add/Remove](#)  
 Yahoo!  Wikipedia  Amazon Books

**Outline View**  250 total results

**obama (250 results)**

- President (39)
- Political (25)
- News (14)
- United States (14)
- Community (14)
- Democratic (12)
- Law (17)
- Senate (12)
- Hawaii (15)
- Barack Obama (13)
- Election (11)
- Dunham (9)
- Nyanza Province (3)
- Videos (9)
- White House (7)
- Board of Directors (6)
- President-elect Barack Obama (6)
- New York (6)
- Victory (5)
- Health Care (4)
- Los Angeles (4)
- School (3)
- Campaign (3)
- Sen (3)
- Foreign Policy (3)
- Obama has also Written (3)
- John Dickenson (2)
- War (2)
- Debate (2)

**Results: [1-10 of 250]**  **Detail:**

**obama**

**Barack Obama - Wikipedia, the free encyclopedia**  
[Add to Working List](#) | [Ezst to del.icio.us](#) | [Bookmark](#) | [Email](#)  
 Hyperlinked encyclopedia entry provides an extensive personal and political profile of the U.S. President. ... Main article: Early life and career of Barack Obama ...  
[http://en.wikipedia.org/wiki/Barack\\_Obama](http://en.wikipedia.org/wiki/Barack_Obama) - Sonntag, 23. Mai 2009  
 Source: Wikipedia

**Joyce Foundation**  
[Add to Working List](#) | [Ezst to del.icio.us](#) | [Bookmark](#) | [Email](#)  
 ... Communities Project, and also from 1994 to 2002 on the board of directors of the Joyce Foundation. [27] Obama served on the board of directors of the Chicago Annenberg Challenge from 1995...  
[http://en.wikipedia.org/wiki/Joyce\\_Foundation](http://en.wikipedia.org/wiki/Joyce_Foundation) - Montag, 25. Mai 2009  
 Source: Wikipedia

**Hillary Rodham Clinton**  
[Add to Working List](#) | [Ezst to del.icio.us](#) | [Bookmark](#) | [Email](#)  
 ... In February 2007. After a close campaign in the 2008 Democratic Party presidential primaries against Hillary Rodham Clinton, he won his party's nomination, becoming the first major party African American candidate for...  
[http://en.wikipedia.org/wiki/Hillary\\_Rodham\\_Clinton](http://en.wikipedia.org/wiki/Hillary_Rodham_Clinton) - Montag, 25. Mai 2009  
 Source: Wikipedia

**The White House**  
[Add to Working List](#) | [Ezst to del.icio.us](#) | [Bookmark](#) | [Email](#)  
 Features statements and press releases by President Barack Obama as well as documents, an index of government information, history, and tour information.  
<http://www.whitehouse.gov/> - Sonntag, 24. Mai 2009  
 Source: Yahoo!

**2004 Democratic National Convention keynote address**  
[Add to Working List](#) | [Ezst to del.icio.us](#) | [Bookmark](#) | [Email](#)  
 ... crowded field in the March 2004 Democratic primary raised his visibility, and his prime-time televised keynote address at the Democratic National Convention in July 2004 made him a rising star nationally in...  
[http://en.wikipedia.org/wiki/2004\\_Democratic\\_National\\_Convention\\_Keynote](http://en.wikipedia.org/wiki/2004_Democratic_National_Convention_Keynote)... - Montag, 25. Mai 2009  
 Source: Wikipedia

**President Barack Obama**  
[Add to Working List](#) | [Ezst to del.icio.us](#) | [Bookmark](#) | [Email](#)  
 WhiteHouse.gov is the official web site for the White House and President Barack Obama, the 44th President of the United States. This site is a source for...  
[http://www.whitehouse.gov/the-press-office/president\\_obama/](http://www.whitehouse.gov/the-press-office/president_obama/) - Sonntag, 24. Mai 2009  
 Source: Yahoo!

**Barack Obama: News & Videos about Barack Obama - CNN.com**  
[Add to Working List](#) | [Ezst to del.icio.us](#) | [Bookmark](#) | [Email](#)

Abbildung 4.18: Grokker.com

all web video images music news shopping searchme life stacks > tools > preferences > about us >

**searchme:**

share: [twitter](#) [facebook](#)

Wikipedia is contained by people like you. Please donate today.  
[Call this page](#) | [History](#)

**Clint Eastwood**  
 From Wikipedia, the free encyclopedia

For other uses, see **Clint Eastwood** (disambiguation).

**Clint Eastwood** (born May 31, 1928) is an American film actor, director, producer and composer. He is primarily known for his tough guy, anti-hero acting roles in violent action films, particularly when paired with James Caan in the 1960s, 1970s, and 1980s. He has received five Academy Awards, five Golden Globe Awards, a Screen Actors Guild Award and five People's Choice Awards—including one for lifetime achievement. He has performed as the iconic man with his name in the Dollars Trilogy of spaghetti westerns, and as a reprobate hardy catalyst in the two Dirty Harry films he made him an enduring icon of masculinity. For his work in the films colorspan (1992) and Redline color body (2001) he received the Best Actor award for his performance in the Best Actor, and received nominations for Best Actor. These films in particular, as well others such as play dirty for me (1971), the outlaw joseph pater (1981), escape from alcatraz (1979), in the Line of Fire (1993), and gran Torino (2008), have all received great critical acclaim and commercial success. He has directed many of his films, and has also found success in others in which he has not acted such as Mystic River (2003). He also has an interest in politics. He served as Mayor of Carmel by the Sea, California, from 1983 to 1988.

**Clint Eastwood** is the only person to have starred in six blockbuster films for five consecutive decades, making him the longest-running movie star. <sup>[1][2]</sup>

**Contents** (click)

- Early life
- Early career
- 1950s
- 1960s
- 1970s
- 1980s
- 1990s
- 2000s
- Other projects
- Politics
- Personal life
- Relationships & family
- Director and producer
- Bibliography
- Awards and honors
- Academy Award nominations
- Golden Globe nominations
- Discography
- Albums
- Singles
- References
- Further reading
- External links

**Early life**

**Eastwood Clint bei Amazon**  
 DVDs - neu oder gebraucht!  
 Kostenlose Lieferung ab € 20  
[Amazon.de/dvd](#)

**Clint Eastwood bei eBay**  
 Clint eastwood. Rohraeweise  
 Angebote Clint eastwood? Ab zu  
 eBay!  
[www.ebay.de/Clint+Eastwood](http://www.ebay.de/Clint+Eastwood)

**Clint Eastwood**  
 Richtige Auswahl an Fanartikeln  
 Clint Eastwood im Angebot  
[TrendModule.org/Happfana](http://TrendModule.org/Happfana)

**Clint Eastwood**  
 at the film *Dirty Harry* in 1967

**Born:** [Clint Eastwood](#)  
 May 31, 1928 (aged 78)  
 San Francisco, California, U.S.

**Occupation:** film director, producer, actor, composer, author

**Years:** 1955–present

**Spouse(s):** [Nancy Johnson](#) (1953–1964)  
[Dina Ruiz](#) (1995–present)

**Domestic:** [Sandra Locke](#) (1975–1988)

**partner(s):** [Frances Fisher](#) (1990–1991)

Abbildung 4.19: searchme.com



ausreichender Auflösung und ermöglichen so einen Überblick über den Inhalt der Webseite. Ein Zoom in die Vorschau ist ebenfalls möglich. Die Terme der Suchanfrage, werden in den Vorschauen hervorgehoben (*Highlighting*). Weitere Suchkategorien, neben dem Eingabefeld, ermöglichen eine weitere Filterung der Ergebnisse. Der Benutzer kann alle Ergebnisse in individuell gestalteten Subkategorien, sogenannten *Stacks*, speichern.

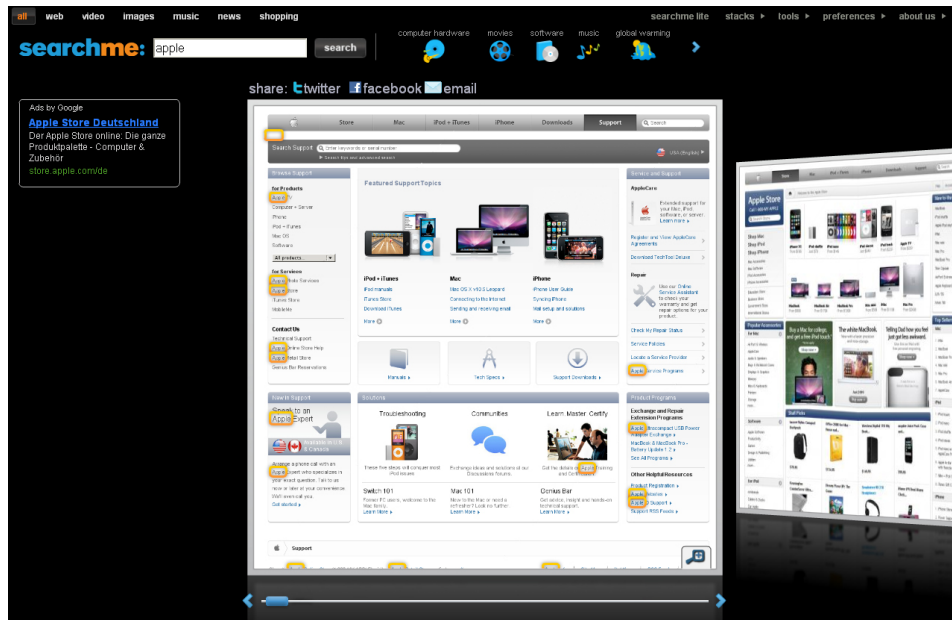


Abbildung 4.20: searchme.com

### 4.5.5 MyTag

*MyTag*<sup>29</sup> ist ein im Sommer 2007 im Rahmen eines Projektpraktikums, an der Universität Koblenz entwickeltes Portal. Die Idee hinter *MyTag* ist es verschiedene Taggingportale zu vereinen und durch diverse Features die Nutzungsmöglichkeiten zu erweitern. Ein Manko von bekannten Tagging-Suchmaschinen wie *Youtube*, *Flickr* etc., ist die Limitierung auf einen Medientyp. Ein weiteres Defizit stellt das Ranking der Ergebnisse dar. In den meisten Fällen dienen lediglich Popularitätswerte oder Aktualität zur Berechnung der Ergebnislisten. Ein personalisiertes Ranking bleibt außen vor. Mit *MyTag* wurde der Versuch unternommen, nicht nur diverse Taggingsuchmaschinen und die jeweiligen Profile auf den unterschiedlichen Systemen auf *MyTag* zu vereinen, sondern auch dem Benutzer von *MyTag* die Möglichkeit zu eröffnen, die gefundenen Ergebnisse automatisch nach seinen Benutzungsgewohnheiten zu ranken. Konkret bedeutet dies, eine Protokollierung und Auswertung des Nutzungsverhaltens beim angemeldeten Benutzer. Die Nutzung von *MyTag* erübrigt somit den Gebrauch von vielen anderen Portalen, und bietet dazu noch unterstützende Mehrwerte.

<sup>29</sup><http://mytag.uni-koblenz.de>

Abbildung 4.21: mytag.uni-koblenz.de

## 4.5.6 Google

Abbildung 4.22: Google.com (Google timeline)

*Google*<sup>30</sup>, die wohl bekannteste Suchmaschine, wartet mit interessanten Neuerungen in 2009 auf. Mitte Mai 2009 präsentierte die Entwicklungsabteilung des Unternehmens *Google Labs* neue Funktionen, die die Suche für die Anwender künftig einfacher und übersichtlicher machen sollen. Wer etwa ein bestimmtes Produkt sucht, bekommt mit der Option *Reviews* nur Seiten mit Kritiken oder Besprechungen angezeigt. Eine andere Möglichkeit bietet die *Timeline*. *Google* präsentiert eine Zeitleiste, auf der sich ablesen lässt, wie viele Seiten einen Suchbegriff zu einem bestimmten Zeitpunkt enthielten oder enthalten. Durch Klick auf die Timeline gelangt der Benutzer in das gewünschte Jahr und den

<sup>30</sup><http://www.google.com>

Monat der entsprechenden Dokumente. Ein weiterer interessanter Ansatz stellt das sogenannte *Wonderwheel* dar, welches den Suchbegriff in einer radähnlichen Grafik darstellt. Im Zentrum befindet sich der eingegebene Suchbegriff, und um diesen Kern herum werden Begriffe empfohlen, die nach *Google*, thematisch in die Kategorie des Suchbegriffs einzuordnen sind.

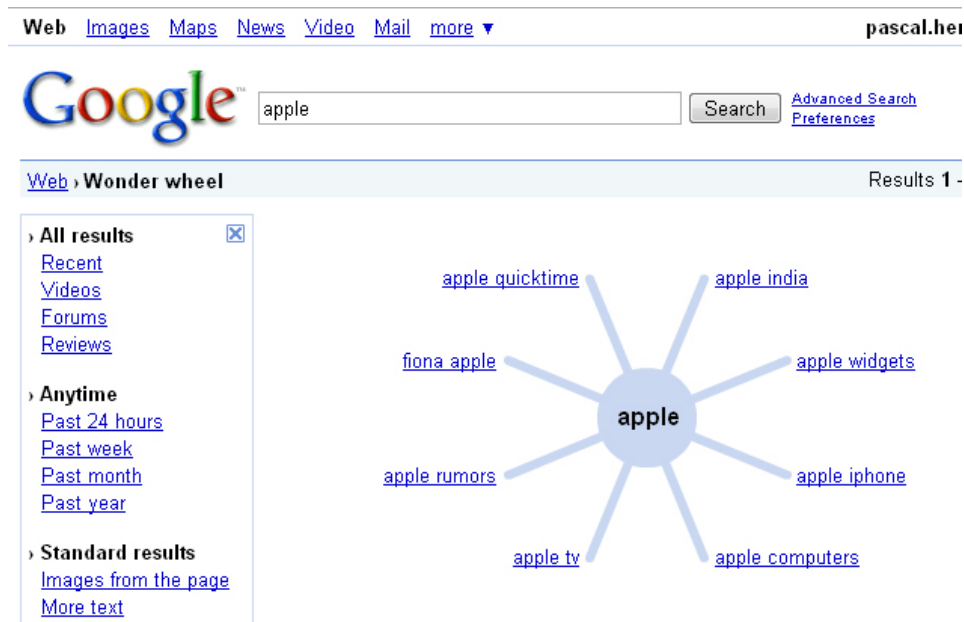


Abbildung 4.23: Google.com (Google Wonderwheel)

### 4.5.7 Bing

Die neue Internetsuchmaschine von Microsoft *Bing*<sup>31</sup>, arbeitet in den USA mit diversen Partnern zusammen und soll sich mit Flügen, Aktienkursen und Sportergebnissen auskennen. Die deutsche Version bezieht ihr Spezialwissen aus der Datenbank des Shoppinportals *Ciao*, welches insbesondere bei der Produktsuche Informationen zu Preisvergleichen und Nutzerbewertungen zum Tragen kommt. Ähnlich wie bei *Google* kann der Benutzer auf der Startseite von *Bing* neben einer globalen Suche auch verschiedene Suchkategorien auswählen und Suchoptionen einstellen. Darunter *Bilder*, *Videos*, *Shopping*, *News* und *Maps*. Sucht man beispielsweise nach einer konkreten Flugnummer, so erhält man Informationen um welchen Flug es sich handelt und ob er gegebenenfalls Verspätung hat. Bei der Suche nach Bildern und Videos erlaubt es *Bing*, die Suche mit wenig Aufwand einzuschränken - etwa nach Porträtfotos, Schwarz-Weiß-Bildern oder Dateigrößen. Bei der Videosuche durchforstet *Bing* bis zu neun verschiedene Videoquellen und listet diese in einer Voransicht auf. Wird der Zeiger über die Thumbnails bewegt, wird automatisch eine Voransicht abgespielt.

<sup>31</sup><http://www.bing.com>

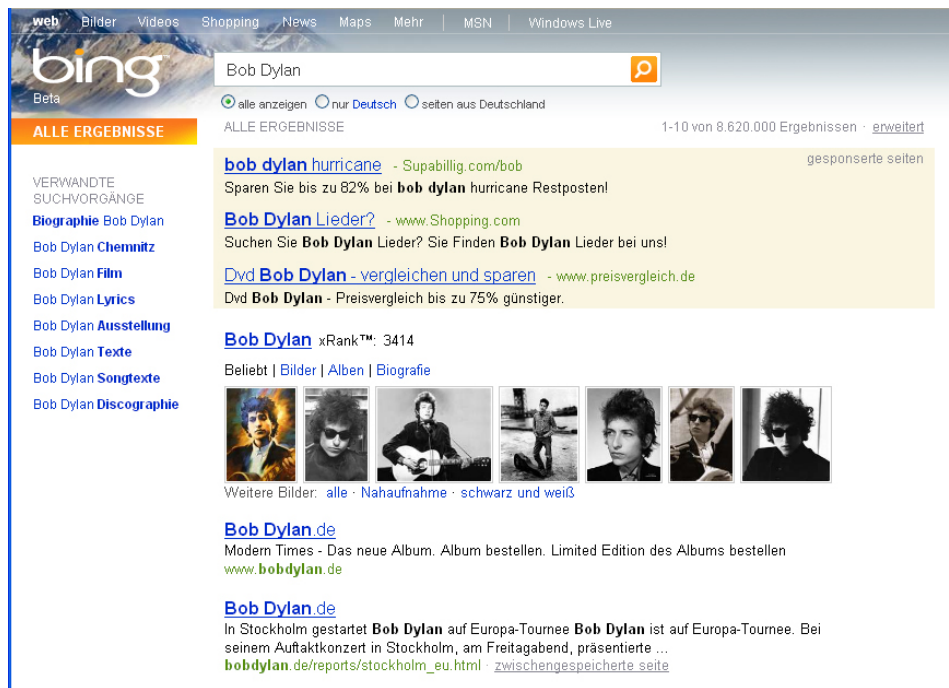


Abbildung 4.24: Bing.com

### 4.5.8 Wolfram Alpha

*Wolfram Alpha*<sup>32</sup> versteht sich als eine „Antwortmaschine“, die Antworten aus denen im Internet verfügbaren Informationen berechnet, statt wie es herkömmliche Suchmaschinen machen, lediglich darauf zu verweisen. Die Maschine interpretiert die Suchanfrage des Benutzers und sucht die passende Formel um eine Antwort zu berechnen. Grundlage ist die Software *Mathematica*. Sucht man beispielsweise nach zwei Unternehmen, so erhält man bei *Wolfram Alpha* ein Kurvendiagramm mit der Entwicklung der Aktienkurse. Bei *Google* würde der Benutzer einen Verweis auf Seiten erhalten, auf denen die Unternehmen erwähnt werden. Allerdings verschweigt *Wolfram Alpha* bei vielen Resultaten die Quellen und ist aufgrund der eigenen Datenbasis hinsichtlich mancher Thematik sehr beschränkt. Anders als bei *Google* und Co. findet eine qualitative Einschätzung des Betreibers bzgl. der Informationen der Datenbank statt und bedarf daher von Seiten des Benutzers eines großen Vertrauens.

### 4.5.9 Picollator („image driven search“)

*Picollator*<sup>33</sup> ist eine Suchmaschine basierend auf einer Patternerkennungs-Technologie. Sie erlaubt die Suche von Portraits. Die Vorlage kann entweder hochgeladen oder unter Angabe der URL verwendet werden. Picollator sucht dann im WWW nach Bildern, die Ähnlichkeiten mit der Bildvorlage haben.

<sup>32</sup><http://www.wolframalpha.com>

<sup>33</sup><http://www.picollator.com>



---

Gut gemalte Vorlagen sind ebenfalls zur Suche geeignet. Nebenbei besteht aber auch die Möglichkeit einer textbasierten Anfragespezifikation. Nach Angaben des Betreibers ist die Suchmaschine mit komplexen Technologien ausgerüstet, die mathematisch die Unterschiede beziehungsweise Gemeinsamkeiten von Bildern berechnen. Die Resultate lassen jedoch mitunter noch zu wünschen übrig. Es kann vorkommen, dass bei einer „weiblichen Bildvorlage“, „männliche Bilder“ zurückgeliefert werden.

## 4.6 Fazit State of the Art

In diesem Teil der Arbeit wurde der Informationssuchprozess mit WWW-Suchmaschinen beschrieben und auf den aktuellen Stand von Suchmaschineninterfaces an konkreten Beispielen erörtert. Zunächst wurde anhand des klassischen Suchmaschinenmodells und anderen Informationssuchmodellen ein Suchzyklus grob skizziert. Da das Hauptaugenmerk auf dem Design einer Recommenderanwendung liegt, wurde durch eine neue Unterteilung in Pre-Search-Recommendations und Post-Search-Recommendations der Suchvorgang mit der Priorität auf Recommendingdienste betrachtet.

Die Anfragespezifikation ist das Bindeglied zwischen dem Benutzerverständnis für seinen Informationsbedarf und des Informationsanbieters. Heutzutage wird die Anfragespezifikation in den meisten Fällen als eine Keyword-Anfrage realisiert. Der Benutzer formuliert seine Anfrage in Form von Keywords oder folgt Hyperlinks auf bekannten Webseiten. Im Teil „Pre-Search-Recommendations“<sup>4.3</sup> wurden die unterschiedlichen Arten der Anfragespezifikation beleuchtet und anhand aktueller Beispiele näher erläutert. Insbesondere wurde auf die unterschiedlichen Arten der Anfragespezifikation und die zahlreichen Neuerungen wie z.B. die dynamischen Suchtermempfehlungen eingegangen.

Das Präsentieren von Suchergebnissen ist eine sehr kritische Komponente im Informationssuchzyklus. Im Kapitel „Post-Search-Recommendations“<sup>4.4</sup> wurden am Beispiel einiger empirischer Studien zusammengefasst, welche Aspekte der Dokumente am besten in der Trefferanzeige Verwendung finden sollten. Weiter wurden diverse Features bei der Trefferanzeige und diverse unterschiedliche Varianten der Visualisierung der Trefferanzeige beleuchtet. Obwohl sich die Betrachtungsweise auf die Trefferanzeige im WWW bei WWW-Suchmaschinen in den letzten Jahren nur gering verändert hat, gab es innovative Neuerungen in Technologie und Design, welche dem Benutzer in seiner Informationssuche und der Entscheidung, welche Hyperlinks für ihn interessant sein könnten, effektiv unterstützen. Dies beinhaltet die Anzeige der Anfrageterme in den Hits, die Balance zwischen Länge und Informationsdichte in den Zusammenfassungen, grafische Umsetzungen und Kategorisierungen der Treffermenge und vieles mehr.

Bei erfolgloser Suche, kann ein Benutzer Schwierigkeiten haben, seine Suchanfrage neu zu formulieren oder eine alternative Möglichkeit zu finden. Dementsprechend sind Reformulierungs-Tools als äußerst effektive Bestandteile eines Suchmaschineninterfaces einzustufen. Wie im Kapitel „Post-Search-Recommendations“<sup>4.4</sup> eingehend beschrieben, gibt es sehr viele Herangehensweisen um eine Reformulierung zu erleichtern oder den Benutzer darin zu unterstützen. Beispielsweise werden Rechtschreibkorrekturen angeboten, die durch Klick auf die entsprechenden Hyperlinks eine neue Suche schnell ermöglichen. So gesehen, kann die Reformulierung der Anfrage an sich, als eine eigene Suchstrategie betrachtet werden. Denn nahezu 50 Prozent aller Suchsessions enthalten eine Art der Anfragereformulierung.

In den nun folgenden Kapiteln sollen die Studien und Entwicklungen beim Design eines Interface zur Integration des Search Term Recommenders berücksichtigt werden. Mittels Prototyping sollen zunächst einige Entwürfe erstellt werden. Dabei werden Papierprototypen genauso wie vorläufige implementierte Prototypen zur Verwendung kommen. Nach einer Entwurfsphase sollen diese in Usability-Tests, gemäß Usability-Engineering, überprüft werden. Die Testergebnisse der Benutzertests sollen die weitere Entwicklung des Search Term Recommenders erleichtern.



# 5 Designentwurf für den Search Term Recommender

In diesem Teil der Arbeit geht es um den Prozess des Designs eines Interaktionsinterfaces allgemein und konkret an (Papier-)Prototypen und Mockups mit stetem Blick auf die Entwicklung eines Interaktionsdesigns für den Search Term Recommender. Mit der Arbeit an dem Search Term Recommender wird der Versuch unternommen, ein Modul zu entwickeln, welches mittels Empfehlungen die Suche unterstützt. Anstatt einer technikzentrierten standardisierten Benutzungsoberfläche, wird ein gebrauchstaugliches, mit dem Benutzer interagierendes System benötigt. Nicht der Benutzer soll sich dem System oder in dem Fall dem Modul anpassen, sondern das System sich an den Benutzer.

## 5.1 User - Interface - Design

„Nicht die Nutzungskontexte schaffen Raum für die Nutzung von Computern, sondern Computer werden in die Nutzungskontexte eingebettet, ohne diese zu stören.“ (Herczeg, 2009)

Dies wird am besten durch die Software-Ergonomie und dem Interaktionsdesign realisiert. Das Interaktionsdesign gestaltet das interaktive System. Welche Qualität das System hinsichtlich bestimmter Zielkriterien hat, wird durch die Software-Ergonomie festgelegt (Herczeg, 2009). Bei dem Entwurf eines Benutzungsoberflächen (User Interface) spielen das Zusammenwirken von Interaktionsdesign, Software-Ergonomie und auch Software-Engineering bzw. Usability-Engineering eine entscheidende Rolle.

Eine solches Interface ist über das WWW global zugänglich und muss somit für eine breite Masse an Benutzern, verständlich und handhabbar konstruiert werden. Dabei gilt es mitunter das Alter, den kulturellen oder auch den fachspezifischen Hintergrund der potentiellen Benutzer zu berücksichtigen.<sup>1</sup> Der Designentwurf soll von Beginn bis zur endgültigen Gestaltung skizziert werden.

Dabei bietet sich zunächst ein Blick auf die Usability und Designvereinbarungen an, deren Einbezug bei einem Interfaceentwurf nach Krause, Hearst, Nielsen et al. unumgänglich ist. Eines der wichtigsten

---

<sup>1</sup>Dies gilt selbstverständlich für alle Arten von Softwareentwicklungen, die von Benutzer bedient werden sollen.

Qualitätsmerkmale eines Interfaces ist die Usability. Der Begriff soll in dem Fall dafür stehen, wie einfach ein System für den Benutzer zu gebrauchen ist. Beim Design eines (Such-)Interfaces gilt stets die Maxime der Einfachheit. Egal, um welche Art von Interface es sich dabei handelt, empfiehlt sich stets eine simple Struktur und eine gute Usability für die Seite. Gerade bekannte und erfolgreiche Seiten, die sich im WWW durchgesetzt und etabliert haben, glänzen weniger durch auffallende Farben, komplexen Visualisierungen etc, als durch klare und überschaubare Gestaltung.

Im Folgenden, beziehe ich mich konkret auf, Krause (1995), Shneidermann u. Plaisant (2004), Nielsen (2003) und Hearst (2009), die Komponenten der Usability definiert haben.

- *Erlernbarkeit*: Wie einfach ist es für den Benutzer, Grundkenntnisse beim ersten Besuch auf dem Interface zu erlernen?
- *Effizienz*: Wie schnell ist der Benutzer imstande, die gelernten Kenntnisse anzuwenden?
- *Merkbarkeit*: Wie schnell findet sich der Benutzer auf dem Interface wieder zurecht, nachdem er eine Weile das selbige nicht genutzt hat?
- *Fehler*: Wie viele Fehler macht der Benutzer? Wie beeinträchtigen diese Fehler? Und wie schnell lassen sich diese wieder beheben?
- *Zufriedenheit*: Wie zufrieden ist der Benutzer mit dem Interface? Wie gefällt ihm das Interface?

Um high usable Interfaces zu kreieren, ist die Betrachtung der Usability und dem User Interface Designs unabdingbar. Das Gebiet, welches Interface Design genauso, wie das Grundverständnis der Benutzerinteraktion vereint, nennt man *Human Computer Interaction, HCI*. Weiter noch empfiehlt sich bei der Betrachtung von Userinterfaces das *Human Computer Information Retrieval, HCIR* (Shneidermann u. Plaisant, 2004) (Marchionini u. White, 2008). Das HCIR hat sich weitgehend auf dem akademischen Researchgebiet manifestiert. Da im Bereich des Information Retrieval und dem Bereich der Human Computer Interaction desöfteren versäumt wurde disziplinübergreifend zu forschen und zu entwickeln, soll die HCIR Aspekte aus beiden Bereichen vereinen.<sup>2</sup> Die Entwicklung einer Designtechnik namens *User-Centered-Design*, welches als Hauptziel die Entwicklung eines handhabbaren Designs mit dem Benutzer im Mittelpunkt hat, orientiert sich ebenfalls stark am Gebiet des HCIR. Weiter geht es um die Integration menschlicher Intelligenz in den Systemen. Dies bedeutet im Einzelnen, dass auch die menschliche Fehlbarkeit bei der Steuerung mit einkalkuliert werden sollte.<sup>3</sup> Dabei hält

<sup>2</sup>Beispielsweise würde sich eine Sprachsteuerung bei der Human Computer Interaction einer großen Beliebtheit erfreuen, würde aber gegebenenfalls auf dem Gebiet des Information Retrieval nur mäßig Zustimmung erhalten. Hier gilt es eine für beide Anwendungsbereiche zufriedenstellende Lösung zu finden.

<sup>3</sup>Bei einer erfolglosen Suche soll dem Benutzer nicht die Kontrolle entzogen werden, indem er auf sich allein gestellt ist. Vielmehr geht es darum, Mechanismen (Suchtermkomplettierung, Alternative Suchtermempfehlungen, etc.) anzubieten, um den Benutzer näher an relevante Dokumente zu „führen“

sich der Designer ganz an die Vorstellungen, Wünsche und Bedürfnisse der potentiellen Benutzer. Ist das Design abgeschlossen, so wird dieses durch diversen Nutzertest in Theorie und Praxis evaluiert. Kritik und Evaluationsergebnisse fließen dann in das Redesign, so dass eine Art Zyklus von alternierenden Entwicklungs- und Testphasen entsteht (Software lifecycle).

Nicht nur zur Gestaltung von User-Interfaces, sondern auch zu Webseiten allgemein, existieren mehrere Richtlinien respektive Guides. Allerdings sollten diese Guides als eine Art „Kochrezept“ (Krause) zu verstehen sein. Zwar gibt es nach ausgiebigen Studien der menschlichen Kognition und Perzeption, Richtwerte (Bedingungen aufgrund der menschlichen Anatomie, kulturellen Hintergrund, Gewohnheiten, usw.), doch eine Universallösung bietet keiner der Guides. Shneidermann u. a. (1997), haben 1997 acht Designempfehlungen für allgemeine Such-Benutzer-Interfaces formuliert:

*Informatives Feedback anbieten (Sanduhr; sich füllender Balken):*

Ein wichtiger Punkt im Interfacedesign ist das Feedback bzw. die Rückmeldung an den Benutzer über den Status des Systems. Das bedeutet, dass der Benutzer visuell oder auch akustisch unmittelbar nach einer Aktion (z.B. Mausklick) über die damit verbundenen Konsequenzen und den weiteren Verlauf informiert wird. Als wichtige Feedback Indikatoren sind unter anderem folgende zu nennen.

Die Suchergebnisse sollten unmittelbar nach Suchanfragestellung angezeigt werden. In der Ergebnisliste selbst sollten die Suchbegriffe hervorgehoben werden. Moderne Suchmaschinen geben neben Titel noch zusätzliche Informationen wie URL oder eine textuelle Zusammenfassung an. Eine weitere effektive Form des Feedbacks im Display, ist die Möglichkeit die Ergebnisliste dynamisch nach verschiedenen Rankingkriterien (z.B. Datum, nach eigener Personomy (*MyTag*)) sortieren zu können. Nach Abgabe der Anfrage, haben sich sogenannte Term Suggestions oder auch Term Refinements als sinnvoll erwiesen. Unter diesen Feedbackindikator fällt auch der Search Term Recommender, bei dem vor oder während der eigentlichen Anfragespezifikation, eine Empfehlung ausgesprochen wird. Sehr verbreitet ist die Entwicklung, eine dynamisch erstellte Liste von empfohlenen Anfragebegriffen als Feedback zurück zu geben oder eine automatische Komplettierung des Begriffes während der Eingabe des Suchbegriffs vorzunehmen. Indikatoren, die Feedback zum Ranking der Trefferanzeige geben, sollten sparsam verwendet werden.<sup>4</sup>

*Unterstützung der Benutzerkontrolle:*

Hierbei geht es um die Opazität des Systems. Also, in wie weit die Kontrolle über die Vorgänge beim Benutzer liegt. Eine zu großes Ungleichgewicht in der Opazität („durchsichtige“ vs. „undurchsichtige“ Abläufe) kann für den Benutzer sehr schnell zu komplex werden. Viele aktuelle Systeme bieten für kundige Benutzer die Möglichkeit in „Erweiterten Einstellungen“ oder „Profieinstellungen“ an, das System individuell zu konfigurieren.

---

<sup>4</sup>Die Anzeige der Treffer in Form einer Liste dürfte mitunter Relevanzbeschreibung genug sein.

*Reduzierung der „Hauptspeicherzugriffe“ beim Benutzer:*

Statt Blanko-Eingabeschlitze haben sich z.B. Eingabeschlitze, mit Informationen über die resultierende Aktion bei der Benutzung, als Verbesserung erwiesen. Vermeintliche Unklarheiten beim Design sollten bereits im Vorfeld ausgeräumt werden. Der Benutzer sollte nicht erst darüber nachdenken müssen, wofür ein Eingabeschlitz zu gebrauchen ist und/oder, was aus der Benutzung folgt. Weiter ist es so, dass viele Benutzer, gerne auf die Informationen zurückgreifen, die sie schon in der Vergangenheit besucht haben (Jones u. a., 2002). Der *Chrome Web Browser* bietet beispielsweise ein Raster von Thumbnails der meistbesuchten Bilder des Benutzers an.

„A well-established principle of human memory is that it is often easier to recognize a word or name than it is to think up that word.“

Informationsstrukturen wie Links im WWW oder eine Tabelle mit dem Inhalt eines Buches, geben einen sehr guten Überblick über den Inhalt, und ermöglichen dem Benutzer die Informationen nach seinen Interessen zu navigieren und zu browsen. Dabei hat sich die Integration von Suche und Navigation als sinnvoll herausgestellt. Bei sehr großen Datenmengen ist eine Kategorisierung ebenfalls hilfreich.

*Angebot von Shortcuts für erfahrene Benutzer:*

Die Richtlinie, Abkürzungen oder sogenannte Shortcuts anzubieten, zielt auf die erfahrenen Benutzer. Denen wird dadurch eine schnellere und alternative Bedienung des Interfaces erlaubt. Das klassische Beispiel sind die *Keyboard-Shortcuts* (Kopieren, Einfügen, Löschen, etc. ). Zwar lassen sich Keyboard-Shortcuts auch bei Suchinterfaces anwenden, aber noch lohnenswerter erwiesen sich Shortcuts anderer Art. Eine Option für den angemeldeten Benutzer, wäre die Möglichkeit eines individuell gestalteten Interfaces. Eine weitere Art von Shortcuts sind gut definierte und gegebenenfalls vorhersagbare Informationsbedürfnisse. Bei der Suche nach „Wetter Bonn“, erhält man bei *Google* ohne Umschweife Angaben zur Temperatur und Witterung in grafischer Form. Somit bleibt der Weg über die Ergebnisliste und den jeweiligen Verweisen erspart.

*Reduzierung der Fehler; Simple Handhabung mit Fehlern:*

Leere Ergebnislisten sollten vermieden werden. Sollte es vorkommen, dass zur Anfrage des Benutzers keine Ergebnisse geliefert werden kann, so sollte eine „Hilfestellung“ von Seiten des Systems zur Verfügung stehen. In den meisten Fällen ist eine leere Ergebnisliste das Ergebnis einer unkorrekten und/oder suboptimalen Formulierung der Anfrage. Dann ist es am System, den Benutzer „auf den richtigen Weg zu leiten“.

*Konsistenz:*

Das System sollte stets erwartungskonform handeln.

*Berücksichtigung kleiner Details:*

Suchinterfaces müssen imstande sein, viele und komplexe Informationen gleichzeitig präsentieren zu können. Kleine Unterschiede entscheiden dabei über ein erfolgreiches oder unerfolgreiches Design. Untersuchungen haben gezeigt, dass der Großteil der Benutzer die ersten Treffer als relevanter einstufen und dementsprechend eher dazu neigten diese anzuklicken, als die folgenden (Joachims u. a., 2005). Weiter noch werden Buchstabierempfehlungen am Kopf der Seite vom Großteil der Anwender gerne einmal übersehen. *Google* reagierte darauf, und implementierte die Seite so, so dass oben die Frage: „Did you mean...?“ und am Ende der Seite die Frage in erweiterter Form wiederholt: „Did you mean to search for...?“ zu finden ist (Hurst, 2002, Sinba, 2005).

*Angemessenes Design; Ästhetik:*

Ästhetik liegt bekanntermaßen im Auge des Betrachters. Aber dennoch sollten unter Berücksichtigung der menschlichen Wahrnehmung, bestimmte Richtlinien berücksichtigt werden. So ist beispielsweise eine gelbe Schrift auf weißem Grund sehr schwer lesbar und eine zu überladene Seite, schreckt eher ab, als das sie einlädt, zu verweilen.

## 5.2 (Software) Prototyping

Prototyping bzw. Protoypenbau ist eine Methode der Softwareentwicklung, die schnell zu ersten Ergebnissen führt und frühzeitiges Feedback bezüglich der Eignung eines Lösungsansatzes ermöglicht.

Typischerweise simuliert ein Prototyp lediglich ein paar Aspekte der Features eines eventuellen Programms, und ist gegebenenfalls sehr von der finalen Implementierung verschieden. Für gewöhnlich werden Prototypen dazu verwendet, mit Benutzertest das Design und die Funktionalität zu evaluieren. Weiter können Prototypen beim Endnutzer Verwendung finden, um die Anforderungen zu beschreiben und zu prüfen, die vom Entwickler nicht berücksichtigt wurden. Prototypen können so auch sehr schnell einen Schlüsselfaktor in der Finanzierung der Entwicklung werden.

Protoypting hat verschiedene Ziele: Die Softwaredesigner und Entwickler können sich frühzeitig Feedback vom Benutzer einholen. Der Auftraggeber kann vergleichen, ob die Software mit denen im Vorfeld spezifizierten Anforderungen übereinstimmt und wie die Software aufgebaut ist.

Generell gliedert sich der Prototypingprozess in folgende Schritte:

1. Identifikation der Basis-Anforderungen. Zunächst müssen alle Anforderungen genau beschrieben werden. „Was geht rein, was kommt raus?“  
Details bezüglich Sicherheit, Robustheit usw., können hier noch vernachlässigt werden.
2. Entwicklung des Prototypen mit Benutzungsinterface.
3. Review. Die (End)Benutzer arbeiten mit dem Prototypen und geben eine Feedback dazu ab.
4. Nachbearbeitung und Erweiterung des Prototypen. Das Feedback wird dazu genutzt die Anforderungen und den Prototypen zu verbessern. Die Schritte 3 und 4 können sich gegebenenfalls mehrmals wiederholen.

Softwareprototyping hat sehr viele Varianten. Nahezu alle Methoden, auf konkrete wird im weiteren Verlauf noch eingegangen, basieren auf zwei Typen von Prototyping. *Throwaway Prototyping (Ended Prototyping)* und *Evolutionary Prototyping*.

Throwaway oder Rapid Prototyping beschreibt die Gestaltung eines Modells, welches nicht zwangsläufig Bestandteil der fertigen ausgelieferten Software ist. Nachdem primäre Anforderungen gesammelt wurden, wird ein simples Modell konstruiert, damit sich der Benutzer anhand des visuellen Beispiels, eine Vorstellung von den schließlich implementierten Anforderungen im fertigen System machen kann. Rapid Prototyping beinhaltet die Arbeit an vielen verschiedenen Stellen des Systems schon zu einem frühen Zeitpunkt. Die Methode ist von sehr informellem Charakter. Der wichtigste Faktor ist dabei die

Geschwindigkeit in der die Modelle verfügbar sind. Das Modell erhält dann einen Startpunkt, von dem aus es dann entsprechend der Auftraggeberressonanz weiterentwickelt wird. So ist es dem Benutzer stets möglich ein Feedback abzugeben, um somit bereits früh in der Entwicklungsphase Einfluss zu nehmen. Eine weitere Stärke besteht darin, dass im Allgemeinen bereits Interfaces designed werden. Interfaces sind die unmittelbare Schnittstelle zum Benutzer. Desweiteren ist es wesentlich einfacher die realisierten Anforderungen praktisch zu testen, als lediglich theoretisch zu bewerten.

Im Gegensatz zum Throwaway Prototyping, geht es beim Evolutionary Prototyping mehr um die Konstruktion eines robusten Prototyps in einer strukturierten Form. Das heißt, ein Projekt wird nie als abgeschlossen angesehen, sondern ständig weiterentwickelt. Da sich die Anforderungen an Softwaresysteme oft innerhalb kurzer Zeit verändern, besteht ein Vorteil darin, dass es dem Entwicklerteam möglich ist, Features hinzuzufügen oder Änderungen vorzunehmen, die während der Anforderungsspezifikation und der Designphase nicht bedacht wurden. Der Vorteil gegenüber Throwaway Prototyping besteht darin, dass man beim Evolutionary Prototyping funktionierende Systeme zur Verfügung hat. Obwohl nicht alle geplanten Features vorhanden sind, kann der bisherige Entwicklungsstand auf einer Interimbasis getestet werden, bis das fertige System ausgeliefert wird. Änderungswünsche an der Struktur, Design und Funktionalität müssen in der nächsten Version berücksichtigt werden. Weiter ist es im Evolutionary Prototyping möglich, dass sich die Entwickler des Projektteams, gemäß persönlichen Stärken auf die jeweiligen Arbeiten konzentrieren können, ohne stets im gesamten Entwicklungsprozess involviert zu sein. Das Gesamtprojekt wird in viele kleine möglichst unabhängige Einzelprojekte unterteilt. Jedes dieser Einzelprojekte wird in einem zyklischen Entwicklungsprozess entworfen und verbessert. Dabei wird nicht mehr zwischen der Entwicklung des Prototyps und des fertigen Produktes unterschieden. Evolutionary Prototyping kann genutzt werden, um die Machbarkeit eines Systems einzuschätzen. Dabei wird auf sehr spezifische Fragestellungen eingegangen und mögliche (Teil)Lösungen simuliert.

Weitere Prototyping-Methoden:

*Exploratives Prototyping* wird zur Bestimmung der Anforderungen und zur Beurteilung bestimmter Problemlösungen benutzt. Dabei wird sich auf die Funktionalität des Systems konzentriert und durch direkt Kommunikation mit dem Kunden, schnelle Ergebnisse erzielt. Hauptziel ist es nachzuweisen, dass die Spezifikationen und Ideen tauglich sind.

*Experimentelles Prototyping* findet seine Verwendung bei der Suche nach Möglichkeiten der Umsetzung, und wird daher zum Großteil zu Forschungszwecken benutzt. Mit einem experimentellen Prototyp wird eine umfangreiche Problemanalyse und Systemspezifikation durchgeführt. Die gewonnenen Ergebnissen können dann bei der Umsetzung des richtigen Produktes verwendet werden.

*Vertikales Prototyping.* Dabei wird ein ausgewählter Teil des Systems durch alle Ebenen hindurch implementiert. Vertikales Prototyping eignet sich besonders für Fälle, bei denen noch funktionalitäts- oder Implementierungsfragen ungeklärt sind.

Beim *Horizontales Prototyping* wird nur eine spezifische Ebene des Gesamtsystems realisiert. Diese Ebene wird möglichst vollständig dargestellt. Beispielsweise wird zunächst die GUI ohne tiefergehende Funktionalitäten realisiert. Diese kann dann dem Auftraggeber vorgelegt werden.

*Paperprototyping* ist ein sehr preiswertes, schnelles und technisch einfaches Verfahren für einen Usability-Test einer Software oder Internetseite. Ein Programm zu schreiben oder eine Internetseite zu entwerfen ist verhältnismäßig kompliziert, bei Paperprototyping aber wird das gewünschte Interface zunächst auf ein Blatt Papier skizziert. Auf Zuverlässigkeit und Leistungsfähigkeit des Produktes wird beim Paperprototyping nicht viel bzw. gar kein Wert gelegt. Eines der Hauptanwendungsgebiete des Paperprototyping ist das Brainstorming innerhalb des Entwicklerteams, um Ideen und Vorstellungen zu sammeln, wie ein Interface aussehen könnte. Das Interface wird hierfür Schritt für Schritt konstruiert. Um die Anwendbarkeit der des Interfaceentwurfs zu erproben, werden typische Situationen durchgespielt.

Mit Paperprototyping kann ein Anwendbarkeitstest schnell und kostengünstig durchgeführt werden. Die Testbenutzer bekommen eine auf Papier aufgemalten Prototyp vorgelegt und müssen damit interagieren. Eine andere Person übernimmt die „Rolle“ des Computers und reagiert gemäß den Aktionen der Testbenutzer so, wie es der Computer tun würde. Die Testbenutzer benutzen den Prototyp, indem sie die jeweiligen aufgemalten Buttons mit dem Finger berühren.

Die Vorteile von Prototyping sind: Die Anforderungen der Anwender können fortlaufend präzisiert und verifiziert werden. Damit sinkt das Risiko einer Fehlentwicklung. Unbeabsichtigte Wechselwirkungen zwischen einzelnen Komponenten des Produkts können früh erkannt werden. Die Qualitätssicherung kann früh eingebunden und der Fertigstellungsgrad besser verifiziert werden.

Nachteile sind: Prototyping verführt oft dazu, Anforderungen weder korrekt zu erheben noch sauber zu dokumentieren. Wenn der Prototyp lediglich als Basis für die folgende eigentliche Entwicklung dient, können zusätzliche Kosten und ein zeitlicher Mehraufwand entstehen.

(Gebhardt, 2000), (Sefelin u. a., 2003), (Stacy, 1997),  
<http://de.wikipedia.org/wiki/Prototyping> (Softwareentwicklung); Stand September 2009



## 5.3 Usability Testing von Userinterfaces

Vorallem im Webdesign und der Internetprogrammierung ist es wichtig, die Benutzungsfreundlichkeit des Interfacedesigns einer Seite zu testen. Versteht der Benutzer den Aufbau der Seite, und findet er die gewünschten Informationen? Reagiert die Seite bei Klick auf bestimmte Buttons nach der Vorstellung des Benutzers?

Überraschenderweise ist es sehr schwer, ein gut benutzbares neues Interface zu entwerfen und vielleicht noch schwieriger ist es, dieses auf Benutzungsfreundlichkeit hin zu testen. Der Tester muss die feinen Unterschiede im Design, der Optionen und die Motivation der Testbenutzer kennen. Darüber hinaus muss er über das Wissen verfügen, welche Faktoren auf das Ergebnis einer Usabilitystudie Einfluss haben können. Eine weitere Schwierigkeit besteht darin, dass bereits geringe Details im Design eines Interfaces einen starken Effekt auf die subjektive Reaktion der Probanden haben können. Gut gemeinte Vereinfachungen durch den Entwickler, können sich negativ auf das Verständnis auswirken.

Demnach stellt sich die Frage, was bei einer Interfaceevaluation gemessen werden soll. Im klassischen Information Retrieval evaluierte man die Verteilung von relevanten Dokumenten zu irrelevanten Dokumenten. Dieser Ansatz kann auch bei der Evaluation von Suchinterfaces verwendet werden, ist aber nur eine einzige Messkomponente unter anderen, überaus facettenreicheren Usabilitymessungen. Für gewöhnlich werden bei der Evaluation von Suchinterfaces drei Hauptaspekte der Usability berücksichtigt: *Effektivität*, *Effizienz* und *Zufriedenheit*, welche in der ISO 9241-11 definiert sind als:

- *Effektivität*: Genauigkeit und Vollständigkeit, mit denen der Benutzer die spezifischen Ziele erreichen kann.
- *Effizienz*: Der Verbrauch von Ressourcen in Abhängigkeit von der Genauigkeit und Vollständigkeit, mit denen der Benutzer die spezifischen Ziele erreichen kann.
- *Zufriedenheit*: Unabhängigkeit vom Discomfort, und positive Einstellung gegenüber der Benutzung des Produkts.

Dies sind die Kriterien, die idealer weise bei einer Suchinterface-Evaluation gemessen werden sollten. Dabei kann es vorkommen, dass sich die Priorität auf die verschiedenen Aspekte unterschiedlich verlagert. Testet man beispielsweise eine neue Technik für alternative Suchtermempfehlungen, würde die Verbesserung der Effizienz und die Reduzierung der Fehler mehr im Vordergrund stehen, als der Wiedererkennungswert der Technik. Manche Studien vernachlässigen bei der Evaluation die subjektive Meinung der Probanden zum Testprodukt. Dabei kann es in der Praxis einer der wichtigsten Messwerte sein. Schließlich wird ein unbeliebtes Interface auch nicht lieber durch eine gute Usability benutzt (Hornbæk u. Law, 2007).

Im Folgenden soll ein Überblick über die gängigsten Methoden zur Evaluation eines Interfaces von der

traditionellen Information Retrieval Evaluation, bis zu formellen und informellen Evaluationsmethoden beschrieben werden.

### 5.3.1 Standard Information Retrieval Evaluation

Im Allgemeinen werden im Information Retrieval die Evaluationen auf eine automatische Weise, ohne Einbezug von Benutzern durchgeführt. Man besitzt eine Dokumentenkollektion, definiert darauf eine Menge von Anfragen und man erhält daraufhin Dokumente, die bezüglich der Anfragenmenge als relevant eingestuft wurden. Rankingalgorithmen entscheiden dann weiter über die Rankingreihenfolge. Die häufigste Verwendung solcher Evaluationsmethoden findet sich im Textretrieval. Schon seit mehreren Jahren ist das Ziel der *Text Retrieval Conference (TREC)*, den State of the Art im Bereich des Information Retrieval mit definierten Testmethoden zu koordinieren. Beim sogenannten *ad hoc Retrieval-Track*, werden die Dokumentenkollektion, die Anfragen sowie die Relevanzkriterien von einer menschlichen Jury vorgegeben. Die teilnehmenden IR-Systeme werden entwickelt, eingefroren und alle mit den gleichen Vorgaben getestet, so dass vergleichbare Ergebnisse zustande kommen. Trotzdem bleibt eine Interaktion von Benutzern mit den Systemen weiterhin außen vor. Da aber die Prinzipien der HCIR nicht nur die Information Retrieval Interface Research beeinflusst haben, sondern auch die Test durch reale Benutzer motivierten, wird seit 1997 bei den Wettbewerben der TREC ein Interaktiver Track involviert. Dabei musste jede teilnehmende Gruppe mehrere Benutzer bestimmen, die eine Auswahl an Anfragen vornahm. Ziel war es, den Prozess der Suche durch den Benutzer, zusätzlich zu den Precision- und Recall-Ergebnissen in die Evaluation mit einfließen zu lassen (E.M. Voorhees, 2000).

### 5.3.2 Informelle Usability Evaluation

Testet man auf Probleme, sowie auf positive Aspekte insbesondere hinsichtlich des Designs eines Interfaces, so spricht man von informellem Usability-Testen. Typischerweise werden informelle Studien dazu verwendet, teilrealisiertes Design eines Interfaces zu testen, oder verschiedene Designkandidaten zu vergleichen. Größere Probleme können bereits in den ersten Evaluationsrunden durch wenige Probanden identifiziert werden (Nielsen und Landaur, 1993). Schnelles informelles Usability-Testen mit einer geringen Anzahl an Testern, ist auch unter dem Namen *Discount Usability Testing* bekannt (Nielsen, 1993).

Den Probanden werden in der frühen Phase der Designentwicklung *Low-Fidelity (Low-Fi)* Prototypen mit unterschiedlichen Designkonzepten dargeboten. Aufgrund der Schnelligkeit der Entwicklung und der Kosteneffizienz, greift man dabei sehr oft auf Paperprototyping zurück. Damit lassen sich die „Grundelemente“ des Designs besser testen, als mit Grafik-Design, wo die Gefahr einer Ablenkung durch Grafikgimmicks besteht. Nach den Low-Fi-Tests folgt in der Regel die Entwicklung von *High-Fidelity (High-Fi)* Versionen, bei denen mehr Interaktionselemente eingebaut werden. Diese werden

wiederum getestet. Nach jedem Testzyklus fließen die Ergebnisse in die Weiterentwicklung und in das Redesign der Prototypen. Auf die funktionstüchtige High-Fi-Version folgt eine Implementierung und weitere Tests mit einer größeren Zahl an Probanden. Zu der benötigten Anzahl an Versuchspersonen, gibt es unterschiedliche Meinungen. Nielsen (2000), veröffentlichte Ergebnisse, dass lediglich fünf Probanden dazu benötigt werden, um 85 Prozent der Usabilityfehler zu identifizieren.

Eine andere Form der informellen Evaluation ist die *Feldstudie*. Bei diesem Ansatz, werden die Testpersonen in ihre „natürlichen Umgebung“, beispielsweise an ihrem Arbeitsplatz besucht und ihre Verwendung des zu testenden Systems beobachtet. Dabei verfolgt man die Idee, dass sich die Testpersonen in ihrer gewohnten Umgebung realistischer verhalten, als in einer inszenierten Testumgebung (Grimes u. a., 2007). In einer Feldstudie kann beobachtet werden, wie das Interface benutzt wird, oder die Benutzer können befragt werden. Allerdings gestaltet sich das Testen von mehreren Designvariationen, aufgrund des Mehraufwands in einer Feldstudie, schwieriger, als in herkömmlicher Testumgebung (Hearst, 2009).

### 5.3.3 Formale Usability Evaluation und Kontrollierte Experimente

Formales Usability-Testen in Form von kontrollierten Experimenten ähnelt in der Vorgehensweise einer Feldstudie. Im Groben geht um die menschliche Benutzung des Interfaces und um berechnen zu können, ob ein Design gut oder schlecht funktioniert (und warum?). Weiter noch können kontrollierte Experimente, Aufschluss über die Verbesserung der Performanz eines existierenden Systems durch zusätzliche Features geben. Der Aufbau und die Struktur eines solchen Experiments können sehr komplex sein. Bei der Planung, Durchführung und der abschließenden Auswertung spielen viele unterschiedliche Faktoren eine Rolle. Detaillierte Ausführungen zum Aufbau und Design finden sich in G. Keppel (1992), R. Kohavi (2007), Kohavi u. a. (2008).

Eine klassische formale Usability-Studie ist beispielsweise die *Laborstudie*. Dabei werden die Probanden bei ihrer Tätigkeit mit der zu testenden Software durch ein einseitig durchschaubares Fenster beobachtet. Im Testraum befinden sich ein Schreibtisch, ein Stuhl, Kameras und ein Computer. Heutzutage ist es üblich, die Arbeitsschritte der Probanden per Video- und Audiomitschnitt zu protokollieren. Zusätzlich kann das Verhalten des Benutzers über eine Mikrofonanlage oder durch einen anwesenden Evaluierenden per Anweisungen beeinflusst, gelenkt oder unterstützt werden. In manchen Studien kommt zusätzlich Eye-Tracking-Technologie zur Verwendung. Dabei wird exakt berechnet, welche Teile des Bildschirms vom Benutzer betrachtet werden und wie intensiv. Alle diese Faktoren fließen in die Auswertung der Studie mit ein.

### 5.3.4 Longitudinal Studien

Um sich ein genaues Bild davon machen zu können, wie einzelne Benutzungspatterns eines Userinterface vom Benutzer angenommen werden, sind spezielle Studien über eine signifikante Zeitperiode erforderlich. Es ist wichtig, den Probanden die Nutzung in ihrer täglichen, routinierten Umgebung zu ermöglichen. Diese Art von Studie ist besonders sinnvoll bei der Evaluation von Search-User-Interfaces. Es erlaubt dem Evaluierenden, genauestens Änderungen im Umgang mit dem System zu beobachten. Wie die Probanden die Bedienung des Systems erlernen und wie sich die Benutzung über einen längeren Zeitraum verändert (Shneidermann u. Plaisant, 2006). Ein intensiveres Arbeiten mit der Software, ermöglicht den Benutzer realistischere Einschätzungen zur „Güte“ zu formulieren. Diese können beispielsweise einfach durch Fragebögen abgefragt werden. Sind die Probanden in einer Laborstudie für einen kurzen Zeitraum intensiv mit dem System beschäftigt, so kann das Benutzungsverhalten bei dem Testbenutzer einer Longitudinal Studie hingegen sehr flexibel beobachtet werden. Zum Beispiel kann die Benutzergruppe geteilt werden und jeder Teilgruppe wird ein anderes Interfacesetting zur Verfügung gestellt. Anschließend lässt sich beobachten, welche Settings von beiden Gruppen bevorzugt eingestellt werden.

### 5.3.5 Large-Scale Log-Based Usability Evaluation (Bucket-Testing)

*Large-Scale Log-Based Usability Testing* ziehen einen Nutzen aus einer großen Zahl von Besuchern auf einer Webseite. Hingegen früheren Vorgehensweisen, bei denen implementierte Software per CD oder DVD für die Verbraucher als eine zunächst fixe Version erhältlich war, entwickelte sich dieses Paradigma im WWW in eine andere Richtung. Die per CD ausgelieferten Versionen konnten nicht signifikant verändert werden und mussten folglich solange in Gebrauch bleiben, bis die nächste Softwareversion veröffentlicht wurde. - Dies mitunter einige Jahre später. Sehr viele Unternehmen veröffentlichen heute Produkte in „Beta“ bzw. „unfertigen“ Versionen mit dem deutlichen Verweis auf eventuelle Probleme. Und darauf, dass das System bis zur offiziellen Veröffentlichung noch diversen Veränderungen unterzogen wird. Manche Webseiten gehen sogar mit speziellen *Work-in-Progress-Versionen* an den Start. Dabei werden kontinuierlich Veränderungen während des Betriebs vorgenommen. Diese Dynamik der Webinterfaces erlaubt das Experimentieren mit unterschiedlichen Interfaces und verschiedenen Gruppen von Benutzern (Hearst, 2009).

Large-Scale Studien benutzen Log-Analysen und sind in der Industrie auch unter den Namen *Bucket-Testing*, *Split Testing* und *Parallel Flights* (R. Kohavi, 2007), (Kohavi u. a., 2008) bekannt. Dabei testet man neue Designentwürfe oder Features an einer stark frequentierten Webseite, die über das WWW erreichbar ist. Der einkommende User-Traffic wird per Zufall aufgesplittet, so dass einem Teil der Besucher das neue Design angezeigt wird. Dabei wird das Verhalten der Benutzer in Logfiles protokolliert<sup>5</sup>.

<sup>5</sup>Bei sehr stark benutzten Seiten lässt sich eine solche Studie schon binnen Stunden durchführen

Mit den Erkenntnissen über die Verhaltensänderungen der Benutzer, lassen sich Rückschlüsse auf die Qualität (Usability, Sympathie, etc.) des neuen Entwurfs schließen.

Bucket-Testing kann sowohl bei geringen Interfaceveränderungen, wie auch bei Large-Scale-Innovationen verwendet werden. Dabei empfiehlt es sich, die neue Version vorab an einer überschaubaren Menge von Probanden zu testen, um mögliche Fehler auszuschließen. Nach einem positiven Verlauf, können mehrere Benutzer in die Test-Bedingungen inkludiert werden (R. Kohavi, 2007).

Diese Art Studie lässt vornehmlich Analysen unter den verschiedenen Probanden zu. Die Benutzer sehen lediglich eine Version des Systems und sind nicht aufgefordert zwei Versionen miteinander zu vergleichen. Dies unterscheidet Bucket-Testing von formalen Studien. Da man weder explizites Feedback bekommt, noch die Benutzer eingehend befragt, erfordert es einen genauen Blick auf die Log-Files. Desweiteren bedarf es einer großen Benutzeranzahl um repräsentative statistisch Ergebnisse zu erhalten. Weiter ist Bucket-Testing durch den variierenden Benutzer-Pool limitiert. Der Benutzer-Pool ändert sich oder manche Benutzer löschen ihre Cookies, die zur Protokollierung der UserID verwendet werden.

## 5.4 Erster Interfaceentwurf für den STR und Benutzertest

Wie im State of the Art 4 zu sehen, gibt es mannigfaltige Realisierungen von Recommendingdiensten. Die Automatische Komplettierung bei der Formulierung der Anfrage, die TagCloud und die „Box“, mit suchverwandten Begriffen („Are you looking for...?“) - um nur einige zu nennen. Selbst bei der Beschränkung auf die Anfragespezifikation stehen viele technische Lösungen zur Auswahl.

In einer Brainstormingrunde haben wir uns (Philipp Schaer und ich) an den betrachteten Systemen orientiert und uns für die eine Realisierung des Search Term Recommender in einer Art *DropDown-Funktion* entschieden.

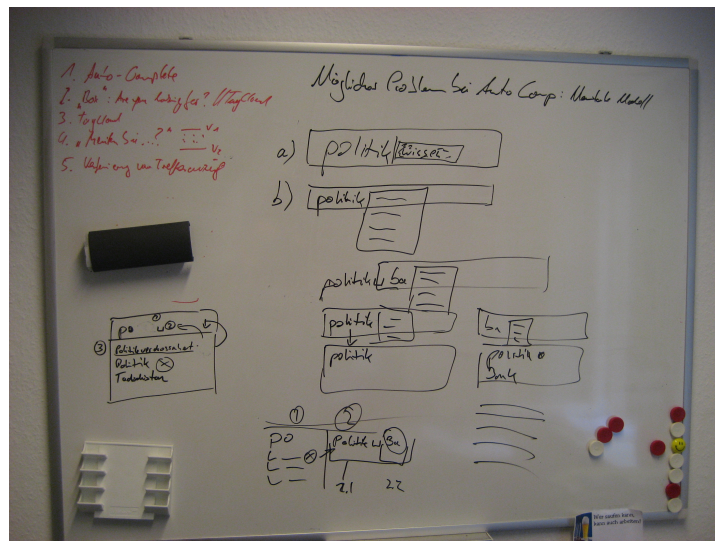


Abbildung 5.1: Brainstorming 1

### 5.4.1 Die DropDown-Funktion

Bei der Realisierung als DropDown-Funktion, öffnet sich bei der Eingabe eine DropDownBox, die Termempfehlungen zu den eingegebenen Begriffen enthält. Entscheidet sich der Benutzer für eine dieser Empfehlungen, wird durch Klick auf einen empfohlenen Term, die bisherige Eingabe durch diesen in der Eingabebox substituiert. Anders als die herkömmlichen Realisierungen mit automatischer Komplettierung (Vorschläge mit gleichen Präfixen) oder einer Rechtschreibkorrektur, werden durch den Search Term Recommender semantisch verwandte Terme empfohlen.

Bei dem Entwurf der DropDown-Funktion standen zwei Varianten zur Diskussion.

Im ersten Entwurf empfiehlt der Search Term Recommender nur semantisch verwandte Begriffe zum jeweils eingegebenen Begriff bzw. der bisher eingegebenen Eingabe. Konkret bedeutet dies, dass jeweils

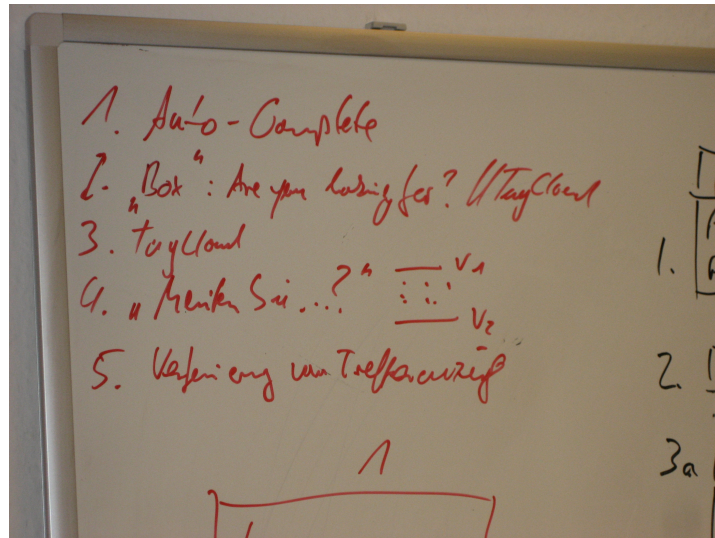


Abbildung 5.2: Brainstorming2

zur momentanen einzelnen Eingabe Termempfehlungen angeboten werden. Bereits eingegebene Suchterme werden nicht berücksichtigt. Eine Abgrenzung zwischen den einzelnen Termen erfolgt durch ein Leerzeichen.<sup>6</sup>

Im zweiten Entwurf erfolgt ebenfalls eine Abgrenzung der eingegebenen Terme durch ein Leerzeichen. Jedoch werden die Terme nach dem Leerzeichen außerhalb der Eingabebox platziert. Die DropDownBox würde fix unter dem Eingabefeld positioniert bleiben. Bei dieser Realisierung beziehen sich die Termempfehlungen auf die bereits eingegebenen Terme und die jeweils erfolgende Eingabe. Die eingegebenen Terme lassen sich durch Klick auf einen Löschbutton löschen, so dass sie bei den Empfehlungen nicht weiter berücksichtigt werden.

Die Aktivierung des Search Term Recommenders in der DropDownBox erfolgt durch Eingabe in das Eingabefeld. Auch hier stehen zwei Möglichkeiten zur Realisierung im Raum. Bei Möglichkeit 1, öffnet sich die DropDownBox nach dem Leerzeichen und enthält Termempfehlungen zur Eingabe (existierender Begriff)<sup>7</sup>, die durch Klick die Eingabe substituieren. Die Möglichkeit 2 sieht eine dynamische Veränderung der Termempfehlungen vor. Sobald eine Eingabe des ersten Zeichens erfolgt, öffnet sich die DropDownBox mit Termempfehlungen. Gemäß der weiteren Eingabe werden die Termempfehlungen dynamisch generiert. Nach abgeschlossener Eingabe oder Substitution durch eine Termempfehlung und Leerzeichen schließt sich die Box bis zur nächsten Eingabe. Unabhängig von der Sprache der Eingabe und Existenz des Begriffs, werden soweit möglich bereits Termempfehlungen angeboten.

<sup>6</sup>Grob am Interface skizziert, würde es bedeuten, dass die DropDownBox mit der Eingabe unterhalb der Eingabebox „mitwandert“.

<sup>7</sup>Falls der Begriff Teil des kontrollierten Vokabulars ist

## 5.4.2 Der Benutzertest

Um einen Prototypentest direkt am Benutzer durchzuführen, wurde sich für das schnelle und effiziente Paperprototyping entschieden. Gemäß den oben beschriebenen Ausführungen wurden zum DropDown-Interface mehrere Screens konstruiert, die die Funktionalität in den unterschiedlichen Varianten widerspiegeln. Die Gesamtzahl der Testbenutzer belief sich auf fünf. Dabei wurde darauf geachtet, Personen mit sehr guten und auch solche mit mäßigen Computer- und WWW-Kenntnissen in den Test mit einzubeziehen. Das Alter der Personen belief sich von 28 - 63 Jahren. Die Personen kamen jeweils einzeln in einem Vieraugengespräch mit den Prototypen in Berührung. Nach kurzer Instruierung wurde den Testbenutzern einige Aufgaben gestellt, die sie mit dem vorliegenden Interface zu lösen hatten. Dabei wurde insbesondere Wert auf das Verständnis der Personen bzgl. Funktionalität, Design und die Konsistenz des Programms gelegt.

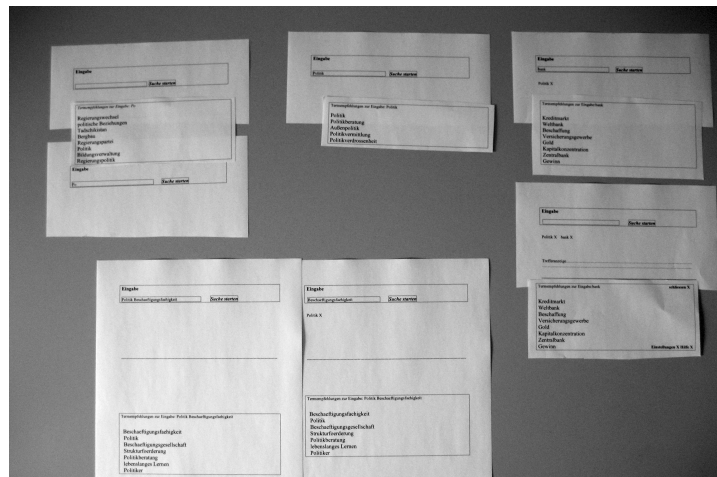


Abbildung 5.3: Paperprototyp; Screenauswahl

Weiterhin wurden unterschiedliche Variationen in Design des Prototypen sowie diverse Einstellungsmöglichkeiten getestet. Eine unterschiedliche Anzahl an Termempfehlungen (Die Termempfehlungen wurden von der Berechnung des *Mindserver*-Software übernommen), oder dem temporären Angebot eines „What is it about?“- Button etc., um hier nur einige zu nennen. Bei der Betätigung dieses Buttons wurde vom Versuchsleiter eine mündliche Erklärung hinzugefügt <sup>8</sup>.

<sup>8</sup>Ausgabe der Suchtermempfehlungen nach *MindServer*



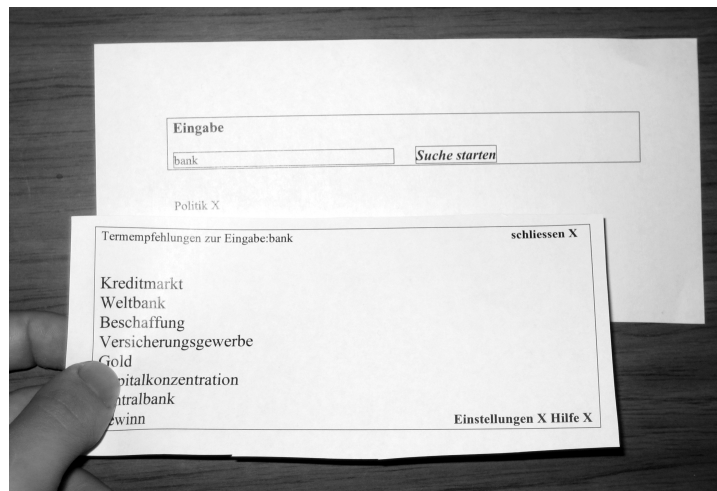


Abbildung 5.4: Paperprototyp1



Abbildung 5.5: Paperprototyp2

### 5.4.3 Die Aufgaben

Zunächst wurden den Testnutzern ein „Blanko“-Interface mit Eingabebox und Suchbutton vorgelegt. Die Aufgabe bestand darin, den Begriff „Politik“ einzugeben, ohne unmittelbar danach den Suchbutton zu betätigen. In der geöffneten DropDown Box sollte sich der Benutzer für den empfohlenen Begriff „Politik“ durch Klick darauf entscheiden. Bei der weiteren Eingabe von „ba“, sollte der Begriff zu „bank“ ausgeschrieben werden. Beim zweiten Durchgang sollten sich die Benutzer für den empfohlenen Term „Beschäftigungsfähigkeit“ entscheiden. Diese Durchläufe wurden entsprechend den beiden Entwürfen der Termempfehlungen, sowie der unterschiedlichen Aktivierungen des Search Term Recommenders mit den generierten Termen mehrmals wiederholt.

### 5.4.4 Das Ergebnis

Mit besonderem Interesse wurde die Reaktion der Testbenutzer auf die vorgeschlagenen Terme beobachtet. 4 von 5 Personen war die Technik einer DropDown-Box in Form von Rechtschreibkorrekturen bei Systemen wie *Google* oder als Empfehlung von oft gesuchten Begriffen bei *Ebay* bekannt. Einer Person ist die unterschiedliche Anzahl der Termempfehlungen und die teils unverständliche Beziehung zur Eingabe aufgefallen und hat dies durch Klick auf den „What is it about?“-Button moniert. Das Ranking der Suchtermempfehlungen war der Person unklar, und hat als einzige den Eingabebegriff als doppelt in den Empfehlungen „entdeckt“. Das Designkonzept wurde von drei Personen vollständig verstanden. Insgesamt favorisierten vier Personen die zweite Variante in Verbindung mit den dynamisch generierten Termempfehlungen. Drei Testbenutzer sprachen sich für eine übersichtlichere Erklärung und Einstellungsmöglichkeiten aus. Insgesamt stuften alle Probanden (nach anfänglichen Schwierigkeiten) den Mehrwert des Search Term Recommenders als positiv ein. O-Ton eines Benutzers: „Die Rechtschreibkorrektur finde ich doch eh über den Ergebnissen, wenn ich die Suche abgeschickt habe! Bei einer sehr speziellen Suche z.B. bei Versicherungen, kann der STR bestimmt hilfreich sein. Da kommt man meist nicht auf die Fachbegriffe.“

# 6 Implementierung STR-Prototyp als DropDown-Variante

Im folgenden Kapitel geht es um die praktische Realisierung eines Interaktionsdesigns für den Search Term Recommender Prototyp. Die in den vorausgegangenen Kapiteln betrachteten Themen, sowie die Ergebnisse des Paper-Prototyping-Usability-Tests sind mit in die Entwicklung eingeflossen. Im ersten nun folgenden Unterkapitel werden das zur Implementierung verwendete Framework, die Programmiersprache Groovy und andere involvierte Komponenten erläutert. Das zweite Unterkapitel beschreibt einen konkreten Blick auf den Quellcode und Algorithmen des Prototyps. Im dritten und letzten Unterkapitel, wurde anhand der Software ein weiterer Usability-Test an einer Gruppe von Probanden durchgeführt. Dabei sollen der Versuchsaufbau, die Durchführung und die Ergebnisse explizit betrachtet werden.

## 6.1 Grails, Groovy, Mindserver

### 6.1.1 Grails

Grails ist ein Framework zur Erstellung von Webanwendungen. Das Grails-Projekt wurde im Juli 2005 gestartet. Die Gründer waren Guillaume LaForge, Steven Devijver und Graeme Rocher. Angelehnt an *Ruby on Rails*, ist es höchst kompatibel zur Java-Plattform und läuft somit unter der *Java Virtual Machine*.<sup>1</sup> Grails baut auf unterschiedlichen etablierten Frameworks wie *Spring*, *Hibernate* und *SiteMesh* auf und verbindet diese mit der Skriptsprache Groovy. Grails erlaubt darüber hinaus die Nutzung der Vielfalt und Stabilität aller vorhandenen Java-Frameworks und -Bibliotheken. Mit Grails entwickelte Systeme bieten deshalb die gleiche Systemstabilität, die auch Java auszeichnet. Durch die nahtlose Integration zwischen Java- und Groovy-Code kann man bestehenden Java-Code in einer Grailsanwendung weiterverwenden. Grails verwendet *Grails Object Relational Mapping* (GORM), das auf Hibernate basiert, um Domainobjekte auf die Datenbank zu mappen. Grails stellt ein Plugin-Mechanismus bereit, der es erlaubt Funktionen für Grails einfach nachzurüsten. Viele Java Frameworks stehen als Plugin für

---

<sup>1</sup>Grails wurde ursprünglich unter dem Namen Groovy on Rails entwickelt.

Nachdem David Heinemeier Hansson, der Entwickler von Ruby on Rails, darum gebeten hatte, auf diesen Namen zu verzichten, wurde er jedoch nicht mehr verwendet

Grails zur Verfügung.

Als Open-Source Framework wird Grails von einer stets wachsenden Community weiterentwickelt. Die Hauptinitiation der Grails-Entwicklung war es das „coding by convention“ oder auch „convention over configuration“-Paradigma in Groovy mit einzubringen, um Java-Entwicklern eine Möglichkeit zur einfachen und schnell umfangreichen Web-Applikation zu bieten. Unter „convention over configuration“ versteht man das Konfigurieren von bestimmten Funktionen einer Applikation über ihren Ort und den Namen der Dateien. So finden sich beispielsweise Artefakte eines bestimmten Typs immer in dem gleichen Verzeichnis der Projektstruktur oder bestimmte Elemente haben immer den gleichen Namen. Zwar ist die Einhaltung einer bestimmten Ordnerstruktur unabdingbar, doch kann somit Konfigurationsaufwand eingespart und der Einblick in fremde Projekte erleichtert werden. Weiter noch sind bei Grails nahezu keine Konfiguration des Webservers oder der Applikation erforderlich. Eine in Grails erstellte Webanwendung lässt sich als Web-Archive (WAR-Datei) exportieren und kann so auf jedem Servlet-Container wie zum Beispiel dem *Apache Tomcat* installiert werden.

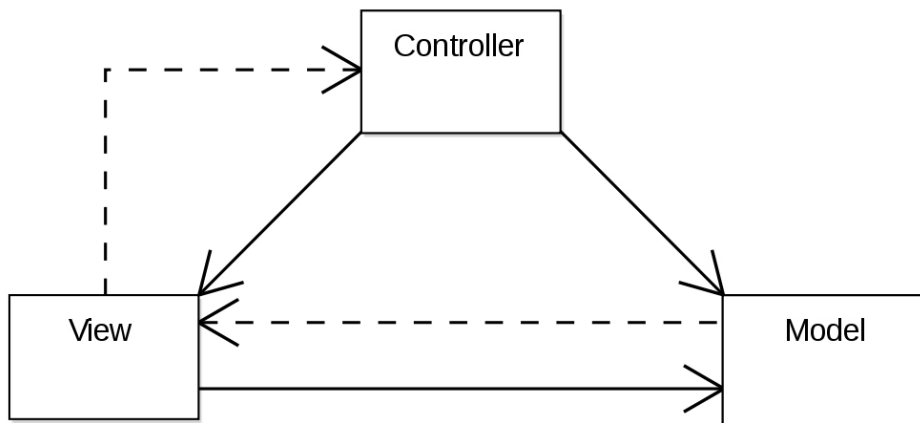


Abbildung 6.1: Model-View-Controller

Grails-Applikationen setzen das *MVC*-Konzept (Model-View-Controller) um. Ziel des Architekturmusters ist ein flexibler Programmentwurf, der eine spätere Änderung oder Erweiterung erleichtert und eine Wiederverwendbarkeit der einzelnen Komponenten ermöglicht. Eine Grails-Applikation besteht u.a. aus folgenden wichtigen Teilen:

- **Domains/Models:** Klassen zur Kapselung der Daten.
- **Services:** Methoden für die Business-Logik
- **Controllers:** Steuern die Abläufe in der Applikation. Defaultmäßig existiert ein Controller pro View.
- **Views:** Die einzelnen Webseiten (HTML, GSP).

## 6.1.2 Groovy

Groovy ist eine dynamisch typisierte Programmiersprache und Skriptsprache für die Java Virtual Machine. Das von James Strachan entwickelte Groovy basiert auf den Sprachen *Java* und *Python*. Die Website von Groovy <sup>2</sup> gibt eine der besten Definitionen von Groovy:

„Groovy ist eine agile, dynamische Sprache für die Java-Plattform mit vielen Fähigkeiten, die von Sprachen wie Python, Ruby und Smalltalk inspiriert sind, wodurch diese den Java-Entwicklern in einer Java-ähnlichen Syntax zugänglich werden.“

Groovy ist eng mit der Java-Plattform verbunden. Das gilt sowohl für die Implementierung (viele Teile von Groovy sind in Java geschrieben, der Rest in Groovy selbst), als auch für die Interaktion. Für Java-Entwickler sollte ein Umstieg auf Groovy nicht allzu schwierig sein, da Groovy mit stetem Blick auf die Belange von Java-Entwicklern entwickelt wurde. Groovy erlaubt eine drastische Reduzierung der Menge an Java-Code, der üblicherweise geschrieben werden müsste. Beispielsweise ist es nicht mehr notwendig *Iterator i = list.iterator()* zu schreiben, um eine Liste von Items zu durchlaufen. Ein simples *list.each* würde dasselbe bezwecken.

Java-Freundlichkeit bedeutet zweierlei: nahtlose Integration in die Laufzeitumgebung von Java (Java Runtime Environment) und eine auf Java abgestimmte Syntax. Groovy läuft innerhalb der Java Virtual Machine und nutzt die Java-Bibliotheken (beides zusammen wird als Java Runtime Environment oder JRE bezeichnet). Groovy ist nur eine neue Art, normale Java-Klassen anzulegen. Aus der Perspektive der Laufzeit ist Groovy Java mit einer zusätzlichen Jar-Datei als Abhängigkeit. Durch das hohe Maß an Kompatibilität, ist es kein Problem, Java aus Groovy heraus aufzurufen. Jeder Typ von Groovy ist ein Subtyp von *java.lang.Object*. Jedes Objekt von Groovy ist eine ganz normale Instanz eines Typs. Die Integration in umgekehrter Richtung ist ebenso möglich.

Argumente, die für eine Nutzung von Groovy sprechen und um Unterschiede zu Java verdeutlichen:

- *Prägnante, ausdrucksstarke, kurze und lesbare Syntax*: Einerseits ermöglicht das Java-Entwicklern die schnelle Einarbeitung. Andererseits heißt weniger Programmcode auch leichtere Wartung (und die damit einhergehende Produktivitätssteigerung) und weniger Bugs.
- *Ausgezeichnete Interaktion mit Java*: Groovy interagiert mit Java, und das nicht nur partiell, sondern vollständig und in beide Richtungen. Groovy kann problemlos sämtliche bereits entwickelten Java-Klassen im Klassenpfad wiederverwenden. Zudem ist es ohne weiteres möglich, Groovy-Programme von Java aus zu nutzen.
- *Eigenschaften einer dynamischen Sprache und Script-Fähigkeit*: Groovy erweitert Javas Klassenbibliotheken um etliche Funktionen, unterstützt Standardtypen und führt die bei Java nicht

---

<sup>2</sup><http://groovy.codehaus.org>

mehr vorhandene Operatorüberladung wieder ein. Javas Compiler muss im Voraus wissen, welche Datentypen zur Initialisierung von Variablen verwendet werden sollen. Groovy erlaubt es, den Inhalt einer Variablen via Late Binding nachträglich festzulegen.

Generell kann Groovy überall dort eingesetzt werden, wo auch Java zum Einsatz kommt. Sie eignet sich zur Entwicklung von Scripts, Servlets, Web-Applikationen, vollständige Anwendungen, Benutzeroberflächen, Datenverarbeitung (XML-Parsen) usw.

Koenig u. a. (2006) Staudemeyer (2007) Barclay u. Savage (2006)

### 6.1.3 MindServer

Im Jahr 2009 stellte die Firma *Recommind* bereits die 5.Generation der MindServer-Software vor. MindServer 5.0 ist ein patentiertes Verfahren, welches noch effizientere zentrale Informations-Kategorisierung und relevanzorientierte dezentrale Suche über alle Datenquellen und Datenformate in komplexen, auch standortübergreifenden IT-Systemen ermöglichen soll. *Recommind GmbH* ist eine der führenden deutschen Entwickler von intelligenten Suchmaschinen. Die MindServer-Software identifiziert unabhängig von Sprache und Themenbereich automatisch Konzepte, die ein Dokument beschreiben.

„Die fehlende Komponente beim Information Retrieval ist Kontext. Woher weiß ich, was der Inhalt eines Textes ist? Was macht genau diesen Text besonders wertvoll? Ist dieser Text für mich als individuellen User wichtig? Das alles sind Fragen, die von der überwiegenden Anzahl an Tools, die am Markt erhältlich sind, nicht einmal im Ansatz beantwortet werden können.“ Dr. Jan Puzicha, CTO und wissenschaftlicher Leiter der *Recommind*

Die MindServer-Software beinhaltet diverse Tools. Das *Administration Tool* dient zum Anlegen und zur Konfiguration von Projekten für unterschiedliche Quellen von Datenbeständen. Mithilfe des *Categorization Tool* können an den zuvor angelegten Projekten Feinabstimmungen vorgenommen werden. Weiterhin erfolgt in diesem Tool das Training für die Taxonomie, für die einzelnen Klassen oder für ausgewählte Klassen nach einem Trainingsplan. Für die Implementierung eines Search Term Recommender Prototyps als DropDown-Variante wurde der Trainingskorpus aus der vorausgegangenen Arbeit von Daniela Holl in ihre Diplomarbeit (Holl, 2009) weiter genutzt. Dabei wurden, die in den Projekten verwendeten Datenbestände, per Zufall aus der Sammlung *SOLIS*<sup>3</sup> extrahiert.

---

<sup>3</sup>SOLIS - Sozialwissenschaftlicher Informationsdienst. Eine eigene Datenbank des Informationszentrums für Sozialwissenschaften.

## 6.2 Implementierung

Die Ergebnisse aus dem ersten Paperprototyping-Benutzertests motivierten die technische Entwicklung eines Interaktionsdesigns für den Search Term Recommender in Form einer automatisierten DropDown-Variante. Der Großteil der Probanden äußerte sich positiv über diese Art der Integration. Dabei wurde insbesondere die Variante präferiert, bei der sich die Termempfehlungen explizit auf die der eingegebene Suche bezogen. Für die meisten der Testnutzer war die Technik und Funktionalität einer DropDown-Box von diversen Portalen bekannt und leicht verständlich.

Aufgrund der großen Grails-Community steht ein breitgefächertes Angebot an Plugins zur Verfügung. So wurde auch bei der Implementierung des STR-Prototyps als DropDown-Variante das frei verfügbare Plugin *RichUI* verwendet. Dieses Plugin beinhaltet unterschiedliche Packages und Skripte zur Erweiterung von Grails. Nach erfolgter Installation stehen verschiedene Möglichkeiten, wie beispielsweise die Erstellung eines Fotoalbums oder eines Kalenders zur Verfügung. Für die Realisierung des STR-Prototyps als DropDown-Variante war vornehmlich die *Autocomplete-Funktion* von Interesse.

Sie ermöglicht eine automatische Empfehlung zur Komplettierung einer Eingabe in einem Eingabefeld. Eine sich automatisch öffnende Ajax-DropDown-Box beinhaltet dann Komplettierungsvorschläge. Voraussetzung ist eine Datenbank mit Termen, die als Komplettierungsvorschläge verwendet werden können. Besteht beispielsweise die Datenbank aus „Peter“, „Paul“ und „Hans“, würde sich bei der Eingabe von „P“, „Peter“ und „Paul“ als Vorschläge in der Box wiederfinden.

### 6.2.1 Anforderungen an den Prototyp

Wie oben beschrieben, stellte das Plugin *RichUI* ein Grundgerüst für eine Entwicklung. Zunächst aber wurden diverse (Grund)-Anforderungen an den Prototyp erhoben.

- **A1 - Inhalt:** Der STR bietet Termempfehlungen, die im semantischen Kontext zur Benutzereingabe stehen.
- **A2 - Benutzbarkeit:** Die DropDown-Funktionalität ist sehr verbreitet und bekannt.
- **A3 - Erlernbarkeit:** Erlernen durch benutzen. Ggf. Erklärungen durch „What is it about?“, „Hilfe?“.
- **A4 - Robustheit:** Termempfehlungen für jede Benutzereingabe ausgeben.

Auf Basis dieser und den ermittelten Ergebnisse aus der Paperprototyping-Studie wurde mit Grails und der MindServer-Software ein Prototyp implementiert. Zunächst ist es wichtig zu erwähnen, dass es sich bei dem Prototyp weniger um eine komplette Anwendung handelt, als eine technische Realisierung der Paperprototyping-Testergebnisse und Anregungen! Primär sollte der Prototyp Verwendung im Usability-Test finden. En detail, sollte ein Prototyp mit einem Eingabefeld erstellt werden, der bei einer

Eingabe eines Zeichens durch den Benutzer aktiviert wird und Termempfehlungen vom Mindserver diesbezüglich in einer Ajax-DropDown-Box ausgibt. Ziel sollte sein, dass die Eingabe an Mindserver gesendet wird und die Rückgabeterme in der DropDown-Box als Termempfehlungen dargeboten werden. Weiter noch bestand die Anforderung an diverse Gestaltungen des Eingabefeldes.

Da schon im Paperprototyping-Test mehrere Optionen diskutiert wurden, wurde sich letztendlich für die Realisierung von drei Eingabefelder entschieden, um die verschiedenen technischen Variationen bei der Evaluation testen zu können. Um den Spagat zwischen Übersichtlichkeit und Vollständigkeit zu schaffen, sollten weniger die Termempfehlungen in ihrer Zahl beschränkt, und mehr eine sinnvolle Sortierung gefunden werden. Die Termempfehlungen sollten sich in Echtzeit gemäß der Benutzereingabe automatisch während der Eingabe generieren. Die Auswahl einer Termempfehlung in der DropDown-Box per Maus oder Keyboard sollte „kenntlich“ gemacht werden.

## 6.2.2 Funktionen des Prototyps

Es wurde ein Interaktionsdesignprototyp für den Search Term Recommender als DropDown-Variante erstellt. Das Interface bietet drei Eingabefelder. Im ersten Feld ist die Eingabe nur eines Eingabeterms möglich. Im zweiten Eingabefeld ist ebenfalls nur die Eingabe eines Terms möglich. Jedoch wird nach der Eingabe eine automatische Suche angeregt. Da in dem Fall keine Verlinkung zu einer Suchmaschine und einer Datenbank besteht, sind keine Ergebnisse zu erwarten. Das zweite Eingabefeld dient lediglich zur Überprüfung, inwieweit es die Versuchspersonen behakt, die Initiative aus der Hand zu geben. In dem Fall, wo der Benutzer mit der Suchtermempfehlung zufrieden ist, erspart der Automatismus Zeit. Besteht im anderen Fall, noch das Bedürfnis einer Änderung oder Verfeinerung der Eingabe, so ist ein übereiltes Abschicken irritierend. Die Möglichkeit einer beliebigen Verfeinerung der Anfragespezifikation bietet das dritte Eingabefeld. Hier ist es dem Anwender möglich, mehrere Terme auszuwählen und zu kombinieren. Diese werden dann mit einem „+“-Zeichen aneinandergereiht. Erfolgt eine Benutzereingabe in eines der drei Felder, wird diese automatisch in Echtzeit verarbeitet. In einer AJAX-DropDown-Box unterhalb des jeweiligen Eingabefeldes werden entsprechende Termempfehlungen dargeboten, die ausgewählt werden können und die bisher erfolgte Eingabe an Zeichen substituieren. Die Auswahl erfolgt entweder per Mausklick oder Keyboard. Die momentan ausgewählte Termempfehlung ist mit einem gelben Balken hinterlegt.

Um noch bessere Ergebnisse erzielen zu können, wurde eine ursprüngliche Anforderung an den Prototyp verändert. Bei der Eingabe eines Zeichens wird die Anfrage nicht unmittelbar an Mindserver gesendet. Zunächst führt der STR-Prototyp als DropDown-Variante eine Überprüfung des letzten Zeichens oder der vorletzten Cursorposition durch. Handelt es sich dabei nicht um ein Whitespace („Leertaste“), wird die bisherige Eingabe von der Methode *askTheSoz* behandelt. Diese generiert entsprechend der Anfrage eine Liste von Eingabevorschlägen aus dem Thesaurus der Sozialwissenschaften, so dass die Darstellung in der DropDown-Box, einer automatischen Eingabekomplettierungshilfe, bekannt von anderen Portalen, ähnelt.



„Der Thesaurus Sozialwissenschaften ist ein zentrales Instrument für inhaltsorientierte Recherchen in den Datenbanken SOFIS und SOLIS. Zurzeit enthält der Thesaurus ca. 11.600 Einträge, davon rund 7.750 Deskriptoren und 3.850 Nicht-Deskriptoren. Er deckt alle Disziplinen der Sozialwissenschaften ab; in gewissem Umfang enthält er auch Begriffe aus angrenzenden Bereichen, um sozialwissenschaftliche Arbeiten mit interdisziplinären, anwendungsorientierten oder multikulturellen Fragestellungen angemessen inhaltlich beschreiben zu können.“ Quelle: [www.gesis.org](http://www.gesis.org)

Listing 6.1: Thesaurus der Sozialwissenschaften - Auszug

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <!-- Thesaurus Sozialwissenschaften de_en online: 19-01-09 -->
3 <thesaurus>
4   <term type="ND">
5     <id>10010606</id>
6     <name>1. Weltkrieg</name>
7     <translation lang="en">
8       <name>World War I</name>
9     </translation>
10    <relations>
11      <USE>10002395</USE>
12    </relations>
13    <notationcode>5.2.00</notationcode>
14  </term>
15  <term type="DD">
16    <id>10010607</id>
17    <name>11. Jahrhundert</name>
18    <translation lang="en">
19      <name>eleventh century</name>
20    </translation>
21    <notationcode>5.2.00</notationcode>
22  </term>
23  <term type="ND">
24    <id>10228410</id>
25    <name>11. September 2001</name>
26    <translation lang="en">
27      <name>September 11, 2001</name>
28    </translation>
29    <relations>
30      <USK>
31        <targetTerm>10009217</targetTerm>
32        <targetTerm>10009576</targetTerm>
33        <targetTerm>10000668</targetTerm>

```

```
34         </USK>
35     </relations >
36     <notationcode >4.4.09 </notationcode >
37     <notationcode >2.9.04 </notationcode >
38 </term >
```

Der Thesaurus liegt als XML-Datei vor und wird bei Bedarf eingelesen. Anschließend wird eine Liste mit Termen erstellt. In dem Fall sind insbesondere die *names* von Bedeutung. Ergibt jedoch die Zeichenüberprüfung ein „true“ für ein Whitespace, so wird die Eingabe an Mindserver gesendet und in der Methode *askMindserver* verarbeitet. Die Arbeit erfolgt defaultmäßig auf einem bereits definierten und trainierten Projekt, namens *Solistraining*. Der Rückgabewert ist ebenfalls eine Liste von genau den Termen, die auch als Termempfehlungen in der DropDown-Box zum Tragen kommen. Genauer gesagt, werden lediglich die Termnamen in der Ausgabe dargestellt. Weiter noch besitzt jeder dieser Terme einen Konfidenzwert zwischen 0 und 1, der einen Rückschluss auf die Relevanz desselbigen zulässt. Je näher der Konfidenzwert an 1 liegt, desto relevanter ist der Term als kontextuelle, semantisch verwandte Empfehlung für die Eingabe des Benutzers. Gemäß dem Konfidenzwertes erfolgt intern eine Sortierung der Termempfehlungen in absteigender Reihenfolge unmittelbar vor der Ausgabe im Interface.

Diese Funktionalität einer „Fallunterscheidung“ mittels Whitespace wurde mit der Annahme entwickelt, dass eine direkte Termempfehlung von Mindserver nur bedingt Sinn macht. So ist es dem Anwender zunächst möglich, seine Anfrage vollends zu formulieren.<sup>4</sup>

Unterstützend durch die Komplettierungshilfe auf Basis des Thesaurus der Sozialwissenschaften. Das Whitespace steht sozusagen für den Abschluss der Anfragespezifikation und leitet den Unterstützungsmechanismus durch den Mindserver ein. So wird eine weitere Möglichkeit zur Reformulierung der Anfragespezifikation durch die Auswahl einer von Mindserver angebotenen „Alternative“, gegeben und wirkt somit dem Vagheitsproblem V1 entgegen.

### 6.2.3 Implementierte Methoden und Algorithmen

Im Folgenden werden die Methoden des entwickelten STR-Prototyps als DropDown-Variante beschrieben. Damit soll die Funktionalität jeder einzelnen grob skizziert werden.

Alle Methoden sind Teil der Klasse *DropDownController*. Die Klasse *RichUI* ist lediglich für das Interface und das Layout verantwortlich. *RichUI* ruft die GSP-Datei *index* auf. Das Interface wird durch HTML-Code und Groovy-Elementen beschrieben. Es werden drei Eingabefelder generiert und die Funktionalitäten mit den jeweiligen Klassen und Methoden verknüpft.

<sup>4</sup>Tests nur mit Mindserverunterstützung haben gezeigt, dass „zu frühes Eingreifen“ von Mindserver teils sehr undurchsichtige Termempfehlungen zur Folge hatte. Bei der Suche nach „Politik“ wurden bereits bei der Eingabe „Po“, Empfehlungen wie „Tadschikistan“, „Radioaktivitaet“ oder „Europapolitik“ ausgesprochen, die schon im Paperprototyping-Test Unverständnis bei den Probanden hervorriefen.

### The Search Term Recommender Show Case - Autocomplete DropDown

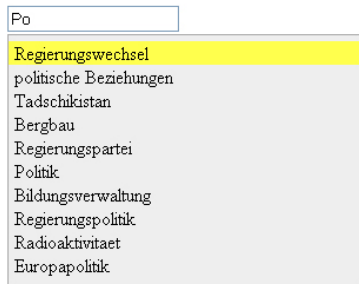


Abbildung 6.2: STR-Protoyp als DropDown-Variante

### The Search Term Recommender Show Case - Autocomplete DropDown

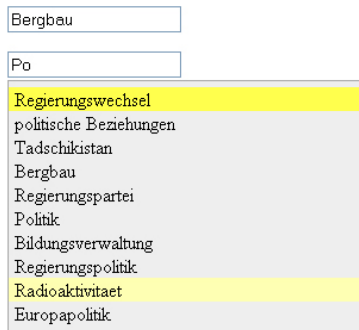


Abbildung 6.3: STR-Protoyp als DropDown-Variante

### The Search Term Recommender Show Case - Autocomplete DropDown



Abbildung 6.4: STR-Protoyp als DropDown-Variante

## searchAJAX

Erfolgt im Interface, bzw. im Eingabefeld eine Eingabe, so wird diese *Query* an *searchAJAX* übergeben.

Die Methode *searchAJAX* testet den letzten eingegebenen Character auf ein Whitespace. Dementsprechend wird die *Query* entweder an *askMindServer* oder *askTheSoz* übergeben. Ist der letzte Character kein Whitespace, so wird die Eingabe an *askTheSoz* weitergeleitet. Bei Betätigung der Leertaste durch den Benutzer wird die *Query* an *askMindserver* übergeben. Weiter erfolgt in *searchAJAX* das Rendering der XML-Daten. Die Methoden *askMindserver* und *askTheSoz* liefern jeweils eine Liste von kontrollierten Termen, *controlledTerms* und *compList*, zurück. Jeder Term der Liste *controlledTerms* besitzt einen String *term*, der *name* des Listenelements und einen Float *confidence* (Konfidenzwert), der *score* des Listenelementes. Jeder Term der Liste *compList* hat einen String *name*.

In *searchAJAX* wird der Name und der Konfidenzwert eines jeden Listenelementes ausgelesen und automatisch in der sich öffnenden DropDown-Box als Termempfehlungen zur jeweiligen erfolgten Eingabe ausgegeben.

Listing 6.2: searchAJAX

```
1 def searchAJAX = {
2
3     def query = params.query
4     def controlledTerms
5
6     println "QueryGetAtTest: ${query.getAt(query.size()-1).getChars()
7         [0].isWhitespace()}"
8
9     if (query.getAt(query.size()-1).getChars()[0].isWhitespace()){
10
11         controlledTerms = askMindserver("${query}%", "SolisTraining")
12     }
13     else {
14         controlledTerms = askTheSoz(query)
15     }
16
17     render(contentType: "text/xml") {
18         results() {
19             controlledTerms.each { controlledTerm ->
20                 result() {
21                     name(controlledTerm.term)
```

```

22         value ( controlledTerm . confidence )
23     }
24 }
25 }
26 }
27
28
29 }
30 }

```

## askMindserver

Die Methode *askMindserver* besitzt die Eingabeparameter *title* und *project*. Durch die Eingabe eines Whitespace wird die *Query* als *title* und defaultmäßig *SolisTraining*, der angelegte Trainingskorpus, als *project* an *askMindserver* übergeben. In *askMindserver* wird zunächst eine Instanz *mindserver* von der Klasse *MindserverConnector* erzeugt. Die Klasse *MindserverConnector* befindet sich im Projektverzeichnis *Groovy Source Packages* und stellt sozusagen die Verbindung zum Mindserver her. Die dazugehörigen Java-Dateien befinden sich im Projektverzeichnis *Libraries*. Weiter wird ein Objekt *xml* erzeugt und die Methode *getTerms* für *mindserver* aufgerufen. *GetTerms* ist eine Methode in der Klasse *MindserverConnector* und bekommt die Parameter *abstrct* und *title* übergeben. Die Methode *getTerms* liefert als XML-Rückgabe, *xmlResult*, ein XML-Dokument gemäß der Benutzereingabe.

Im Anschluss wird *xml* mit dem XML-Slurper geparsed und in *parsedXml* geschrieben. In *parsedXml* werden für jeden Term *name* und *score* extrahiert und in *MindserverTerm* geschrieben. Eine zuvor angelegte leere Liste *controlledTerms* wird schließlich mit *MinderserverTerms* gefüllt.

Die Liste *controlledTerms* wird nach aufsteigendem Konfidenzwert sortiert und zurück an *searchAJAX* übergeben, wo dann das Rendering in der DropDown-Box durchgeführt wird.

Listing 6.3: askMindserver

```

1  def askMindserver(title , project){
2      MindServerConnector mindServer = new MindServerConnector(project)
3      def abstrct = ""
4      def xml = mindServer.getTerms(title , abstrct)
5
6      log.debug(xml)
7
8      def parsedXml = new XmlSlurper().parseText(xml).declareNamespace(
9          rm:'http://www.recommind.com').'rm:annotation'. 'rm:taxonomy'
10     def controlledTerms = []
11
12     parsedXml.find{ it.@name == 'controlledVocabulary' }. 'rm:category' .

```

```
12         list().each{
13             def mindServerTerm = new MindserverTerm(term: it.@name,
14                 confidence: it.@score.toFloat())
15             controlledTerms.add(mindServerTerm)
16         }
17     controlledTerms = controlledTerms.sort{ it.confidence.toFloat() }.
18         reverse()
19     return controlledTerms
}
```

**askTheSoz**

Listing 6.4: askTheSoz

```
1 def askTheSoz(query){
2
3     def xml = new XmlSlurper().parse("thesoz.xml")
4     def compList = []
5     def terms = xml.term
6     terms.each{
7         println it.name
8         compList.add(it.name)
9
10    }
11
12    compList.sort()
13    return compList
14
15 }
```

Die Methode *askTheSoz* bekommt lediglich den Parameter *Query* übergeben. Zunächst wird in der Methode das Objekt *xml* definiert und mithilfe des XML-Slurpers die Datei „thesoz.xml“ eingelesen. Bei der Datei „thesoz.xml“ handelt es sich um den Thesaurus der Sozialwissenschaften im XML-Format. Weiter wird die zunächst leere Liste *compList* erstellt. Dem Objekt *terms* werden jeweils die einzelnen Terme von *xml* übergeben.

Im Anschluss wird über jeden Term von *terms*<sup>5</sup> iteriert und die Liste *compList* mit den in *terms* enthaltenen *names* beschrieben. In dem Fall sollen dienen lediglich die *names* aus dem Thesaurus der Sozialwissenschaften für die Funktionalität des STR-Prototyps als DropDown-Variante. Die anderen Informationen werden nicht gebraucht. Die *names* werden durch die Iterationen aus „thesoz.xml“ herausgefiltert.

Die Liste *compList* wird nach alphabetischer Sortierung an *searchAJAX* zurückgegeben.

---

<sup>5</sup>Der Thesaurus der Sozialwissenschaften enthält diverse Terme, die wiederum unterschiedliche Informationen wie *name* enthalten

## 6.3 Evaluation

Die Aufgabe des im folgenden beschriebenen Benutzertests war zunächst, die aus dem Paperprototyping-Benutzertest hervorgegangenen Ergebnisse anhand einer implementierten Version eines Prototyps zu testen. Ferner sollten die Resultate aus beiden Versuchsreihen miteinander verglichen werden, um zu prüfen, ob und inwiefern sich Unterschiede in den verschiedenen Testverfahren ergeben. Schließlich ging es darum, die Handhabung und den konzeptuellen Aufbau des Prototyps zu überprüfen. Weiter wurde noch eine Vergleichstestreihe von STR-Prototyp als DropDown-Variante und einschlägig bekannten Portalen, die AJAX-DropDown-Komplettierungshilfen und Recommendingdiensten anbieten, durchgeführt.

### 6.3.1 Testaufbau - Durchführung - Aufgaben

Für den Usability-Test wurde ein Testaufbau im Beobachtungslabor an der Universität Koblenz eingerichtet. So konnten die Tests mithilfe von adäquatem Equipment durchgeführt werden.



Abbildung 6.5: Beoblabor Uni Koblenz

Für die Usability-Testreihe wurde eine Gruppe von acht Versuchspersonen zusammengestellt. Aufgrund ihrer Erfahrungen im Umgang mit Retrievalsystemen, Suchmaschinen und Internetnutzung, konnten die Teilnehmer in Laien- und Expertengruppe unterschieden werden. Demnach zählte die Laiengruppe drei Versuchspersonen und die Expertengruppe war mit fünf Probanden besetzt. Die Unterteilung in zwei Gruppen wurde vom Versuchsleiter nicht offiziell mitgeteilt, um die Versuchspersonen nicht im Vorfeld ihrer Handlungen negativ zu beeinflussen. In einem kurzen Vorgespräch in der Gruppe über die Rechercheerfahrungen eines jeden einzelnen, wurde eine geheime Kategorisierung durch den



Versuchsleiter durchgeführt.

Anmerkung: Die Unterteilung der Probanden erfolgte auf subjektiver Meinung des Versuchsleiters. Die Expertengruppe setzte sich vornehmlich aus wissenschaftlichen Mitarbeitern der Institute für Informatik und Computervisualistik zusammen. Während die Laiengruppe ausschließlich aus Versuchspersonen aus „themafremden“ Instituten und Bereichen bestand.

Zu Beginn der Tests wurde in einem Gruppengespräch das Grundkonzept der durchzuführenden Tests erläutert und die technischen und räumlichen Umstände erklärt. Die Teilnehmern/-innen wurde anhand von Beispielen auf anderen Webseiten mit Recommendingunterstützungen auf die folgenden Tests eingestimmt und vorbereitet.

Während der Tests wurden der Bildschirm und verbale Äußerungen der Probanden per Videomitschnitt aufgezeichnet. Kommentare und Auffälligkeiten wurden vom Testleiter handschriftlich mitnotiert. Diese handschriftlichen Aufzeichnungen flossen in die Auswertung der Tests ein.

Nach Erklärung der Idee hinter dem System und dem Hergang der anstehenden Tests, anhand von Beispielen im Internet und diverser Tafelskizzen, wurde die Reihenfolge der Versuchspersonen festgelegt und immer nur jeweils ein Proband separiert mit den Tests beauftragt.

Anhand eines Zeitplans konnte gewährleistet werden, dass die für jede Versuchsperson vorgesehene Zeit nicht überschritten wurde, und das keine Gefahr der Beeinflussung durch die anderen Testteilnehmer/-innen bestand.

An einem Pretest nahmen alle acht Versuchspersonen teil. Der Pretest diente per se zur Erhebung von Daten, die dazu dienten, die Unterschiede zwischen Paperprototyping-Benutzertest und dem Benutzertest am implementierten Prototyp aufzuzeigen. Dabei konnten die Versuchspersonen Angaben dazu machen, ob sich beim Paperprototyping aufgetretene Unklarheiten, allein durch die Benutzung der implementierten Version beseitigen ließen.

Den Versuchspersonen wurde zunächst der Paperprototyping-Benutzertest vorgelegt. Im Anschluss wurden sie dann mit dem STR-Prototyp als DropDown-Variante am Bildschirm konfrontiert. Die Aufgaben waren analog zu den Aufgaben im vorausgegangenen Paperprototypingtest. Im Gegensatz zur vorausgegangenen Paperprototyping-Testreihe, gab es keine verbalen Hilfestellungen vom Versuchsleiter. Abschließend wurde in einem Vieraugengespräch mit dem Versuchsleiter eine 1:1 Gegenüberstellung von Paperprototyp und implementierter Version erörtert.<sup>6</sup>

An der zweiten Testreihe, dem Vergleichstest von STR-Prototyp als DropDown-Variante nahmen sieben der acht Versuchspersonen teil. Der Vergleichstest diente zur Erhebung der subjektiven Meinun-

---

<sup>6</sup>Auszug Fragen des Versuchsleiters:

Ist Ihnen der Umgang mit einer DropDown-Box bekannt?;

Gibt es Unklarheiten im Paperprototyping, die sich durch die interaktive Benutzung des implementierten Prototyps sozusagen von selbst geklärt haben?

Tendieren Sie mehr zu einem Paperprototyp-Test oder einem Test mit implementiertem Prototyp?



Abbildung 6.6: Usabilitytest1

gen der Versuchspersonen im direkten Vergleich des Prototyps mit Systemen wie *Google* und *Ebay*. Insbesondere lag der Fokus auf der Einschätzung der Güte, der Brauchbarkeit, den Vor- und Nachteilen, verglichen mit „etablierten“ DropDown-Applikationen.

Dabei hatten die Versuchspersonen unterschiedliche Aufgaben jeweils mit dem Prototyp und den genannten Portalen unmittelbar hintereinander zu lösen. Da für den Test kein Handout mit den wichtigsten Funktionalitäten bereitgestellt wurde, konnte nicht nur die im Vorgespräch angesprochenen Informationen angewandt und geübt werden, sondern es wurde auch gegebenenfalls vom Versuchsleiter eingegriffen, um Fragen zu beantworten oder auf Möglichkeiten hinzuweisen.<sup>7</sup> Dieses Vorgehen wurde

---

<sup>7</sup>(noch) nicht implementierte, aber durchaus geplante Features des Prototyps, wie beispielsweise Angaben zum Ranking der Termempfehlungen wurden vom Versuchsleiter mündlich als Optionen bereitgestellt.



Abbildung 6.7: Usabilitytest2

gewählt, um etwaige Benutzungsschwierigkeiten besser überprüfen zu können. Die Versuchspersonen hatten nach vorgegebenen Begriffen auf den diversen Portalen zu suchen und hatten dann Schulnoten von 1-3 für die Systeme zu vergeben. Weiter wurden die Probanden angehalten, nach bestimmten Begriffen zu suchen, und wurden dazu ermuntert Kommentare zu den Termempfehlungen, der Funktionalität, Handhabung sowie der konzeptuellen Idee abzugeben.

Folgende Kriterien galt es zu bewerten:

- Darstellung der Termempfehlungen
- Qualität der Termempfehlungen im Hinblick auf meine Suche
- Verständlichkeit der Termempfehlungen; Selbstbeschreibungsfähigkeit
- Hilfreich bei meiner Suche
- Visualisierung der Termempfehlungen
- Zukünftige Nutzung?

Im Anschluss an die Testreihe wurde den Versuchspersonen ein Fragebogen ausgehändigt, auf dem sie Gelegenheit hatten, den Prototyp zu bewerten. Um eine gegenseitige Beeinflussung der Versuchspersonen untereinander zu vermeiden, wurden die Fragebögen unter Ausschluss der anderen Teilnehmer/-innen und des Versuchsleiters anonym ausgefüllt. Weiter noch, wurden vom Versuchsleiter alle Fragebögen zunächst zusammengefaltet gesammelt, und erst bei Vollständigkeit ausgewertet.

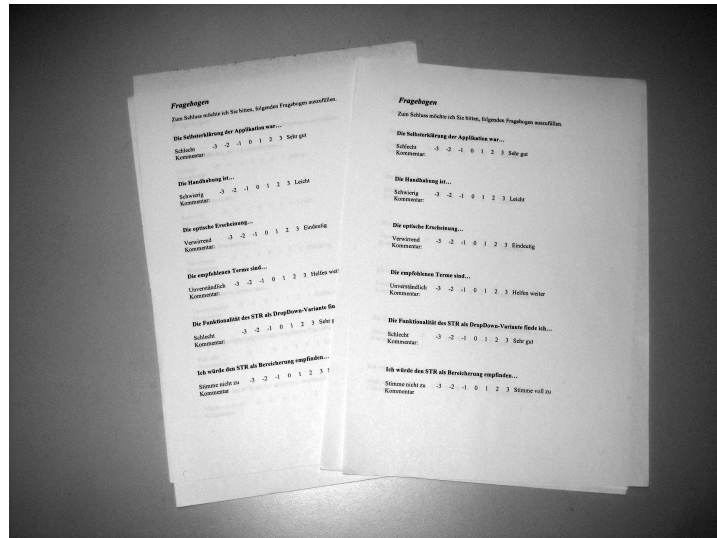


Abbildung 6.8: Benutzerbefragung; siehe Anhang

### 6.3.2 Ergebnisse

In diesem Abschnitt sollen die erhobenen Daten gesammelt und interpretiert werden.

#### Pretest

Mit dem Pretest galt es heraus zu finden, inwieweit sich die persönliche Interaktion mit dem Prototyp in „gewohnter Umgebung“, nämlich am Bildschirm, von der Verwendung des Paperprototyp unterscheidet.

Da es sich in diesem Fall um andere Versuchspersonen, als im ersten Paperprototyping-Test handelte, ließen sich gut Unterschiede beobachten. Insgesamt präferierten alle acht Teilnehmer/-innen den implementierten Prototyp. Nach dem Motto „Learning by doing“, kamen die Probanden mit dem Programm in Kontakt, und je nach Versuchsperson (Expertengruppe deutlich schneller), wurde der Umgang mit dem System schneller oder langsamer erlernt und schließlich von allen als intuitiver, als die Papiervorlage eingestuft.

Dennoch bezeichneten zwei Teilnehmer aus der Expertengruppe von sich aus, einen Paperprototyping-Test als eine schnelle und kostengünstige Alternative zur implementierten Version. Einer der beiden, konnte bereits eigene Paperprototyping-Erfahrungen mit einbringen.

Der Paperprototyp wurde durch Fingerzeig der Probanden gesteuert. Das abrupte Eingreifen des Versuchsleiters, der die Rolle des Computers übernahm, irritierte dabei zwei Teilnehmerinnen, da für sie die Funktion einer sich automatisch öffnenden AJAX-DropDown-Box völliges Neuland war. Nach ersten Berührungen mit dem Paperprototyp, war hingegen der Umgang am Bildschirm ein Leichtes.

Das Ergebnis des Test ist, dass jede Prototypingvariante für sich seine Vorteile (Kosten, Zeit, Zielumgebung) und Nutzen hat. Letztenendes empfiehlt sich eine Kombination beider. Die Verwendung eines

Paperprototyps im Vorfeld einer Studie am Bildschirm, führt die Versuchspersonen an die Thematik und einen abstrahierten Umgang mit dem System heran, und nimmt erste Berührungspunkte. Kurzum, lohnt sich der Entwurf eines Paperprototyps, da er sich insbesondere im Vorgespräch und der Einführung der Testteilnehmer/-innen, als äußerst sinnvoll gestalten kann.

### Vergleichstest

Beim Vergleichstest bestand die Aufgabe für die Probanden darin, den STR-Prototyp als DropDown-Variante mit den Seiten *Google* und *Ebay* zu vergleichen, da beide Seiten zum einen, eine mit dem Sucheingabefeld verknüpfte, automatische DropDown-Box und zum anderen, weitere Recommendingdienste anbieten. Wie im „State of the Art“ zu sehen, ist eine DropDown-Box lediglich für eine automatische Komplettierungshilfe gebräuchlich. In diesem Fall ging es in dem Test darum, diese dem STR-Prototyp gegenüberzustellen und die vermeintlichen Mehrwerte des STR-Prototyps aufzuzeigen. Um dem Test eine gewisse Richtung vorzugeben, und um eine bessere Auswertung vornehmen zu können, wurden den Versuchspersonen sechs Kriterien zur Bewertung vorgegeben. Kommentare der Versuchspersonen wurden handschriftlich dokumentiert und in die Auswertung einfließen lassen.

In Abbildung 6.9 sind in tabellarischer Form die Testergebnisse aufgelistet. Unter Kriterien finden sich die sechs vorgegebenen Kriterien, daneben sind die Bewertungen der sieben Testteilnehmer/-innen für die einzelne Applikation zu sehen. Die Durchschnittsnote errechnet sich aus dem arithmetischen Mittel aller sieben abgegebenen Bewertungen pro Kriterium. Die Zahl im Kasten unter der Spalte mit den Durchschnittsnoten ist das arithmetische Mittel aller Durchschnittsnoten. - Sozusagen das Gesamtergebnis.

Ein erster flüchtiger Blick auf die Bewertungsergebnisse legt die Vermutung nahe, dass dem STR-Prototyp als DropDown-Variante nur geringe Chancen für eine Integration auf diese Art und Weise einzuräumen sind. Beim genaueren Hinsehen wird allerdings deutlich, dass der Grundgedanke des STR-Prototyps von dem Großteil der Versuchspersonen erkannt und positiv aufgenommen wurde. Nach anfänglichen Schwierigkeiten, verstanden alle Testteilnehmer/-innen die Idee, semantisch verwandte Termempfehlungen anzubieten, als attraktives Novum. Insbesondere die Qualität der Termempfehlungen und die Güte der Unterstützung bei der Suche, durch die Neuerung der semantisch verwandten Termempfehlungen, wurde von den Versuchspersonen mit guten Bewertungen honoriert.

Die Ergebnisse verfestigen die Annahme, dass das schlechtere Abschneiden bei den anderen Kriterien auf die mangelnde Visualisierung und den fehlenden Bekanntheitsgrad zurückzuführen ist. Zum Zeitpunkt des Usability-Tests waren diverse Features noch nicht implementiert. Beispielsweise fehlten, aufgrund mangelnder Entwicklungszeit, diverse Steuerungsbuttons („What is it about?“, „Hilfe“, „Einstellungen“, etc.) und Visualisierungsmechanismen (Interfacevarianten). Die Optik des Interfaces

Kriterien	Prototyp-Bewertungen	Durchschnittsnote
<i>Darstellung der Termempfehlungen</i>	2;2;1;3;3;3;2	2,3
<i>Qualität der Termempfehlungen</i>	1;1;2;3;3;2;1	1,9
<i>Selbstbeschreibungsfähigkeit</i>	3;3;3;2;3;3;3	2,9
<i>Hilfreich bei meiner Suche</i>	2;1;1;2;3;3;2	1,9
<i>Visualisierung der Termempfehlungen</i>	3;3;2;3;3;3;3	2,9
<i>Zukünftige Nutzung?</i>	1;1;2;3;2;1;3	1,9
		2,3
Kriterien	Google-Bewertungen	Durchschnittsnote
<i>Darstellung der Termempfehlungen</i>	1;3;2;1;1;2;1	1,5
<i>Qualität der Termempfehlungen</i>	3;2;1;2;2;1;3	2
<i>Selbstbeschreibungsfähigkeit</i>	1;2;1;3;1;2;2	1,7
<i>Hilfreich bei meiner Suche</i>	2;2;2;3;1;1;3	2
<i>Visualisierung der Termempfehlungen</i>	1;1;1;1;2;2;2	1,4
<i>Zukünftige Nutzung?</i>	2;2;1;1;1;3;2	1,7
		1,7
Kriterien	Ebay-Bewertungen	Durchschnittsnote
<i>Darstellung der Termempfehlungen</i>	2;1;3;2;2;1;3	2
<i>Qualität der Termempfehlungen</i>	2;3;3;1;1;3;2	2,1
<i>Selbstbeschreibungsfähigkeit</i>	2;1;2;1;2;1;1	1,1
<i>Hilfreich bei meiner Suche</i>	3;3;3;1;2;2;1	2,1
<i>Visualisierung der Termempfehlungen</i>	2;2;3;2;1;1;1	1,7
<i>Zukünftige Nutzung?</i>	3;3;3;2;3;2;1	2,4
		1,9

Abbildung 6.9: Bewertungen durch die VP

wurde zunächst schlicht gehalten. Vielmehr stand die korrekte Funktionalität im Fokus. Wie schon erwähnt, handelte es sich beim Prototyp um keine vollständig ausgereifte Anwendung, sondern um eine vorläufige Fassung, die vornehmlich die Funktionalität beinhalten sollte, die auch im vorausgegangenen Paperprototyping-Test angeboten wurden. Ein Potential zur Verbesserung der äußeren Erscheinung besteht somit allemal. Ein weiterer Faktor, der für das schlechte Abschneiden des STR-Prototyps bei der Visualisierung verantwortlich gemacht werden könnte, ist die Expertengruppe. Dieser Personenkreis stellte mehr als die Hälfte der Probanden. Die Expertengruppe war mit Anwendungen dieser Art bereits im Vorfeld bekannt, hat diverse Erfahrungen und dementsprechend höhere Erwartungen und Anforderungen. Desweiteren bestach die Expertengruppe durch ein „schärferes Auge“, was etwaige eingebaute Fehler, wie Redundanzen in der Menge der Termempfehlungen anging. Insgesamt ließ sich auch ohne empirische Werte erkennen, dass die Expertengruppe wesentlich schneller mit der Materie vertraut und deutlich kritikfreudiger während des ganzen Tests war, als die Laiengruppe.

- Dennoch wurde mit Absicht nicht der Vergleich mit sehr bekannten und stark frequentierten Portalen gescheut, um Ergebnisse zu erhalten, die eine zukünftige Weiterentwicklung der Visualisierung in eine Richtung forcieren, die einen ähnlichen Erfolg wie bei den beiden Vergleichssystemen erhoffen lässt.

Ferner lassen das die Ergebnisse zur zukünftigen Nutzung vermuten. Sämtliche Versuchspersonen äußerten sich positiv zur Funktionalität des Prototyps. Fünf Probanden sprachen sich konkret für optische Verbesserungen und wünschenswerte Features aus. Beispielsweise wurde von zwei Versuchspersonen nähere Angaben zu jedem Term (Angaben zu potentiellen Treffern, Relevanz) in der Box, ähnlich wie bei *Google*, gewünscht. Ebenfalls an den Ergebnissen deutlich sichtbar, ist die mangelnde Selbstbeschreibungsfähigkeit der Termempfehlungen, die vorallem bei der ersten Nutzung, vom Großteil der Testteilnehmer/-innen moniert wurde. Der plötzliche Wechsel von Komplettierungshilfe von *TheSoz* zu den Empfehlungen von *MindServer* stifteten Verwirrung. Dementsprechend sind Überlegungen zur verbesserten Darstellung vom Übergang anzustreben. Allerdings konnten nach mehrfacher Nutzung, verbalen Hilfestellungen und Erklärungen von Seiten des Versuchsleiters, Unklarheiten ausgeräumt werden.

## Benutzerbefragung

Zum Ende des Tests wurde den Versuchspersonen ein Fragebogen vorgelegt, der anonym bearbeitet wurde. Da der Großteil der Versuchspersonen aus dem Bekanntenkreis des Versuchsleiters „rekrutiert“ wurden, sollte somit vermieden werden, dass bei der Beantwortung persönliche Rücksichtnahmen genommen wurden. Um dennoch eine Dokumentation von individuellen Zweifelsfällen gewährleisten zu können, bestand für die Teilnehmer/-innen die Möglichkeit ihre Antworten auf dem Fragebogen zu kommentieren.

Der Fragebogen bestand aus sechs Fragen, die mittels einer Skala von -3 bis +3 zu beantworten waren. Zu jeder Frage konnte ein Kommentar abgegeben werden.

Im Folgenden wird auf die einzelnen Fragen des Fragebogens eingegangen.

### *Die Selbsterklärung der Applikation*

Es wurde die Selbsterklärungsfähigkeit der Applikation abgefragt. Es ging darum, ob die Probanden mit oder ohne große Schwierigkeiten und Verständnisproblemen mit dem Prototyp interagieren konnten.

Alle Versuchsteilnehmer fühlten sich durch die mündliche Einführung im Gruppengespräch gut an die Thematik herangeführt (+3). Zwei Teilnehmer/-innen hatten nach eigener Aussage noch nie zuvor (bewusst) Kontakt mit der DropDown-Funktionalität.

Allerdings monierte der Großteil der Probanden (-2 - 0) die Semantik hinter den Termempfehlungen und den abrupten Wechsel nach dem Drücken der Leertaste. Ein Teilnehmer sprach sich explizit für die mündlichen Hilfen des Versuchsleiters aus.

### *Die Handhabung*

Die Handhabung fiel fünf Teilnehmern/-innen leicht (+2, +3). Zwei bewerteten die Handhabung mit 0 und -1. Bekannte Kritikpunkte waren die fehlende Erklärung der Termempfehlungen und der Wechsel zwischen „TheSoz“ und „MindServer“ bei Whitespace. Positiv wurde die schnelle Erlernbarkeit bewertet. Bereits im „2.Versuch“, spätestens im „3.Versuch“ fühlten sich alle Versuchspersonen mit der Funktionalität vertraut.

### *Die optische Erscheinung*

Nach den Bewertungen der Optik zu urteilen, gibt es für das Layout noch Verbesserungsbedarf. Sämtliche Bewertungen pendelten um den 0er-Wert. Zwei Teilnehmer/-innen machten deutlich (laut ihren Kommentaren), dass sie das Layout so verstanden haben, dass es primär um die Funktionalität ging. - Können sich aber noch eine „aufgepepptere“ Optik vorstellen.



*Die empfohlenen Terme*

Abgesehen von der noch ausbaufähigen Erklärungen zur Semantik, bewerteten sechs der sieben Versuchspersonen die Termempfehlungen als „gute Alternative“ und „interessante Neuerung“ (+2, +3). Ein Proband wünschte sich die Möglichkeit eines I/O-Schalters für den Search Term Recommender.

*Die Funktionalität des STR-Prototyps als DropDown-Variante*

Alle Testteilnehmer/-innen bewerteten die Realisierung des Search Term Recommender als DropDown-Variante als gut bis sehr gut. Zwei Versuchspersonen fanden den Gedanken zu Alternativen „...wie die Google Meinten Sie vielleicht - Geschichte...“ für lohnenswert oder eine andere optische Realisierung interessant.

*STR als Bereicherung*

Sämtliche Versuchspersonen stimmten der Frage, ob sie den STR-Prototyp als Bereicherung empfänden, voll zu (+2, +3). Neben diversen gewünschten Einstellungsmöglichkeiten oder Layoutalternativen, konnten sich alle eine zukünftige Nutzung des STR-Prototyps, „integriert in die herkömmlichen Suchmaschinen“, vorstellen.

### 6.3.3 Fazit Evaluation

Der Usability-Test zum STR-Prototyp als DropDown-Variante fand in einem Beobachtungslabor an der Universität Koblenz statt. Da zwei Teilnehmer aus zeitlichen Gründen nicht an diesem Test teilnehmen konnten, wurde ein extra Test in einer „face-to-face“-Testumgebung, außerhalb des Labors durchgeführt. Die Ergebnisse fanden hier keine explizite Berücksichtigung, um die angestrebte Homogenität der Tests nicht zu verfälschen.

Aus dieser Erfahrung lässt sich allerdings sagen, dass die Nutzung einer adäquaten Testumgebung entscheidende Vorteile bietet. Nicht nur das den Tests dadurch eine professioneller Note verliehen wird, und die Motivation der Versuchspersonen dadurch spürbar stärker war, sondern lässt sich somit auch eine störungsärmere Durchführung gewährleisten.

Der Test unterteilte sich in drei Teile. Einem Pretest, dem Vergleichstest und einer abschließenden Benutzerbefragung. Im Rahmen des Pretests wurde der direkte Vergleich zwischen Paperprototyping-Test und implementiertem Prototyp überprüft. Das Resultat war eine Empfehlung zur kombinierten Nutzung beider Prototypingverfahren. Der vorausgehende Gebrauch eines Paperprototyps vor einer Prototypstudie am Bildschirm, stellte sich als vorteilhafte Lösung heraus.

Der Vergleichstest warf zunächst einige Schatten auf den STR-Prototyp. Das Layout konnte im direkten Vergleich mit den anderen Portalen im Bereich Ästhetik wenig punkten. Allerdings erwies sich die Visualisierung als solche für den puristischen Test auf Funktionalität und Korrektheit als sinnvoll. Das Konzept, die Idee und die Integration als DropDown-Box ernteten ebenfalls Lob.

Die Benutzerbefragung förderte indes weitere Ergebnisse zu Tage, die aufgrund vermeintlicher Rücksichtnahme, sonst nicht hätten erhoben werden können. Das Ausfüllen der Fragebögen fand etwas zeitversetzt zum Vergleichstest statt und rundete die Testreihe ab. So hatten alle Testteilnehmer/-innen Gelegenheit, alles noch einmal Revue passieren zu lassen und einige Gedanken formulieren zu können. Insgesamt sind alle Versuchspersonen gut mit der Software zurechtgekommen. Probleme, die hier und da aufgetaucht sind, lassen sich mit überschaubarem Aufwand beheben. Die so gewonnen Erkenntnisse fließen bei der Weiterentwicklung bzw. bei der Rückkehr von der Evaluation zur praxisbestimmten Phase des Softwarelifecycle mit ein.

Die Auswertung der Testergebnisse lässt folgende Schlussfolgerung zu: Der Test am realen Benutzer ist allemal lohnenswert, wenn nicht unerlässlich, für die Entwicklung einer Software. Neben den empirischen Studien in vorausgegangenen Arbeiten, rentiert sich in diesem Fall ein Usability-Test am potentiellen Anwender besonders, um subjektive Meinungen und Anregungen einzufangen. - Fehler und Unklarheiten, die dem Entwickler aufgrund subjektiver Wahrnehmung nicht auffallen. Die durchweg positive Resonanz motiviert eine weiterführende Arbeit am Search Term Recommender. Insofern ein direkter Vergleich mit *Google und Co.* momentan noch etwas hinkt, ist das Potential der

Search Term Recommenders von allen Probanden erkannt worden.

# 7 Fazit und Ausblick

Im Rahmen der vorliegenden Ausarbeitung wurde anhand eines Search Term Recommender Moduls ein Interaktionsdesign entworfen, implementiert und am realen Benutzer getestet.

Da in vorausgegangene Arbeiten ein funktionstüchtiges Search Term Recommender Modul entwickelt wurde, galt es zunächst im State of the Art herauszuarbeiten, welche Recommendingdienste im Information Retrieval Verwendung finden und welche sich für die Realisierung einer Search Term Recommender Anwendung eignen könnten.

Im Kapitel Designentwurf für den Search Term Recommender wurden User-Interfaces, Prototyping und diverse Methoden des Usability-Testings betrachtet. Desweiteren wurde ein Designentwurf für den Search Term Recommender vorgestellt. Dieser wurde mittels Paperprototyping an einer Gruppe von Versuchspersonen getestet.

Die im Paperprototyping-Test erhobenen Ergebnisse flossen in die Implementierung des Designentwurfs ein. Dazu wurden in dieser Arbeit das verwendete Framework, die Programmiersprache sowie die algorithmischen Umstände genauer erläutert.

Der implementierte Designentwurf wurde in einer in einem Beobachtungslabor angelegten Usability-Studie überprüft. Dazu wurden der Versuchsaufbau mit seinen unterschiedlichen Testphasen und die Ergebnisse der Studie dargelegt.

Wie auch die empirischen Studien, bestätigten die Usability-Tests am Benutzer die Entwicklung eines Search Term Recommenders. Die Testergebnisse bestätigen die geleistete Arbeit und untermauern die These, dass ein Search Term Recommender Modul ein vielversprechender Ansatz zur Behandlung des Vagheitsproblems im Information Retrieval sein könnte. Auf dem Gebiet des Interaktionsdesigns von Information Retrieval Systemen, insbesondere aber auf mobilen Anwendungen (PDAs; z.B. *iPhone*), werden in Zukunft diverse Weiterentwicklungen zu erwarten sein. Neuerungen auf dem Gebieten Multimedia Search, Social Search, Multilingual Search oder Veränderungen der Keyword-Anfragen, hin zu Natürlichsprachlichen-Anfragemöglichkeiten.

Zwar begegnen sie dem Vagheitsproblem auf unterschiedlichen Wegen, eine gänzliche Lösung ist allerdings nicht zu erwarten. Eine Anfagespezifikation durch den Anwender bleibt unerlässlich. Dementsprechend könnte eine Search Term Recommender einen wertvollen Beitrag zur Entwicklung und Verbesserung von Retrieval Systemen leisten.

# Literaturverzeichnis

[Balabanovic u. Shoham 1997]

BALABANOVIC, Marko ; SHOHAM, Yoav: content-based, collaborative recommendation. In: *Communications of the ACM*, Nr. 40, S. 66-72, 1997

[Barclay u. Savage 2006]

BARCLAY, Kenneth ; SAVAGE, John ; KAUFMANN, Morgan (Hrsg.): *Groovy Programming: An Introduction for Java Developers*. 2006

[Becker 1976]

BECKER, G.C. ; BECKER, G.C. (Hrsg.): *The economic approach to human behavior*. Univ of Chicago Pr; Auflage: New edition, 1976

[Belkin u. a. 2003]

BELKIN, M. ; KELLY, D. ; G.KIM ; KIM, J.Y. ; LEE, H.J. ; MURESAN, G. ; TANG, M.C. ; YUAN, X.J. ; COOL, C.: Query length in interactive information retrieval. In: *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and development in information retrieval*, 2003

[Braun u. a. 2007]

BRAUN, Max ; DELLSCHAFT, Klaas ; FRANZ, Thomas ; HERING, Dominik ; JUNGEN, Peter ; METZLER, Hagen ; MÜLLER, Eugen ; ROSTILOV, Alexander ; SAATHOFF, Carsten: *Personalized Search and Exploration with MyTag / Universität Koblenz-Landau*. 2007. – Forschungsbericht

[Broder 2002]

BRODER, A.: A taxonomy of web search. In: *SIGIR Forum* 36(2) (2002), S. 3–10

[Cigan 2002]

CIGAN, Heidi: *Der Beitrag des Internet für den Fortschritt und das Wachstum in Deutschland / Hamburg Institute of International Economics*. 2002. – Forschungsbericht

[Clarke u. a. 2007]

CLARKE, C.L.A. ; AGICHTEN, E. ; DUMAIS, S. ; WHITE, R.W.: The influence of caption features on clickthrough patterns in web search. In: *In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and development in information retrieval (SIGIR'07)*, 2007, S. 135–142

[Cucerzan u. Brill 2004]

CUCERZAN, S. ; BRILL, E.: Spelling Correction As An Iterative Process That Exploits The Collective Knowledge Of Web Users. In: *Proceedings Of Conference On Empirical Methods In Natural Language Processing*, 2004

[Dumais u. a. 1998]

DUMAIS, S.T. ; FUMAS, G.W. ; LANDAUER, T.K. ; DEERWESTER, S. ; HARSCHMAN, R.: Using latent semantic analysis to improve access to textual information. In: *Proceedings of the SIGCHI conference on Human factors in computing Systems*, 1998

[Eibl 2003]

EIBL, Maximilian: Visualisierung im Document Retrieval. Theoretische und praktische Zusammenführung von Softwareergonomie und Graphik Design. 2. überarbeitete und erweiterte Auflage / IZ-Arbeitsbericht, Bonn. 2003. – Forschungsbericht

[E.M. Voorhees 2000]

E.M. VOORHEES, D. H.: Overview of the Sixth Text Retrieval Conference (TREC-6). In: *Information Processing and Management*, 2000, S. 3–35

[Foltz u. Dumais 1992]

FOLTZ, P.W. ; DUMAIS, S.T.: Personalized Information Delivery: An Analysis of Information Filtering. In: *Communications of the ACM* 35 (12), 1992

[Färber u. a. 2003]

FÄRBER, F. ; WEITZEL, T. ; KEIM, T. ; WENDT, O.: A model-based approach to recommending partners / Universität Mainz. 2003. – Forschungsbericht

[Furner 2002]

FURNER, Jonathan: On Recommending / University of California. 2002. – Forschungsbericht

[G. Keppel 1992]

G. KEPPEL, H. T.: Introduction to design and analysis / Freeman and Company. 1992. – Forschungsbericht

[Gebhardt 2000]

GEBHARDT, Andreas ; GEBHARDT, Andreas (Hrsg.): *Rapid Prototyping Werkzeuge für die schnelle Produktentstehung*. Hanser Verlag, München, 2000

[Grimes u. a. 2007]

GRIMES, C. ; TANG, D. ; RUSSELL, D.M.: Query Logs Alone are not Enough. In: *Proceedings of the 16th International Conference on World Wide Web (WWW'07)*, 2007

[Hearst 2009]

HEARST, Marti A.: *Search User Interfaces* / Cambridge University. 2009. – Forschungsbericht

[Herczeg 2009]

HERCZEG, Michael ; HERCZEG, Michael (Hrsg.): *Software-Ergonomie. Theorien, Modelle und Kriterien für gebrauchstaugliche interaktive Computersysteme*. Oldenbourg; Auflage: 3., 2009

[Hertzum u. Frokjaer 1996]

HERTZUM, Morten ; FROKJAER, Erik: Browsing and querying in online documentation: A study of user interfaces and the interaction process. In: *ACM Transactions on Computer-Human Interaction (ToCHI)*, 1996

[Holl 2009]

HOLL, Daniela: *Search Term Recommender auf Basis der Software MindServer*, Universität Koblenz - Landau, Diplomarbeit, 2009

[Hornbæk u. Law 2007]

HORNBÆK, K. ; LAW, E.L.C.: Meta-analysis of correlations among usability measures. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'07)*. ACM Press New York, NY, USA, 2007, S. 617–626

[Hutchins u. a. 1985]

HUTCHINS, E.L. ; HOLLAN, J.D. ; NORMAN, D.A.: Direct Manipulation Interfaces. In: *Human Computer Interaction Voll* (1985), S. 311–338

[J. Kupiec 1995]

J. KUPIEC, F. C.: A trainable document summarizer. In: *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and development in information retrieval (SIGIR'95)*. Seattle, WA, 1995, S. 68–73

[Jansen u. a. 2007]

JANSEN, B.J. ; SPINK, A. ; KOSHMAN, S.: Web searcher interaction with the Dogpile.com meta-search engine. In: *Journal of the American Society for Information Science and Technology* 58(5) (2007), S. 744–755

[Jansen u. a. 2005]

JANSEN, B.J. ; SPINK, A. ; PEDERSEN, J.O.: A Temporal Comparison Of Altavista Web Searching. In: *Journal Of The American Society For Information Science And Technology* 56(6) (2005), S. 559–570

[Joachims u. a. 2005]

JOACHIMS, T. ; GRANKA, L. ; PAN, B. ; HEMBROOKE, H. ; GAY, G.: Accurately Interpreting

Clickthrough Data As Implicit Feedback. In: *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and development in information retrieval (SIGIR'05)*, 2005

[Jones u. a. 2002]

JONES, W. ; DUMAIS, S. ; BRUCE, H.: Once Found, What Then? A Study Of Keeping Behaviors In The Personal Use Of Web Information. In: *Proceedings Of The American Society For Information Science And Technology*, 2002

[Koenig u. a. 2006]

KOENIG, Dierk ; LAFORGE, Guillaume ; GLOVER, Andrew ; KOENIG, Dierk (Hrsg.) ; LAFORGE, Guillaume (Hrsg.) ; GLOVER, Andrew (Hrsg.): *Groovy in Action*. Manning, 2006

[Kohavi u. a. 2008]

KOHAVI, R. ; LONGBOTHAM, R. ; SOMMERFIELD, D. ; HENNE, R.M.: Controlled Experiments on the Web: Survey and Practical Guide. *Data Mining and Knowledge Management*. 2008. – Forschungsbericht

[Krause 2003]

KRAUSE, Jürgen: Suchen und "Publizieren" fachwissenschaftlicher Informationen im WWW. In: *Medieneinsatz in der Wissenschaft: Tagung: Audiovisuelle Wissensmedien online; Informationsveranstaltung der IWF Wissen und Medien GmbH, IWF: Mesnchen, Wissen, Medien*, 2003

[Krause 2006]

KRAUSE, Jürgen: Shell Model, Semantic Web and Web Information Retrieval. In: *Ilse Harms and Heinz-Dirk Luckhardt and Hans W. Giessen (Hrsg.): Information und Sprache: Beiträge zu Informationswissenschaft, Computerlinguistik, Bibliothekswesen und verwandten Fächern. Festschrift für Harald H. Zimmermann*, K. G. Saur, München, 2006, S. 95–106

[Krause 2007]

KRAUSE, Jürgen: The Concepts of Semantic Heterogeneity and Ontology of the Semantic Web as a Background of the German Science Portals *vascoda* and *sowiport*. In: *International Conference on Semantic Web and Digital Libraries (ICSD-2007), 21-23 February 2007, Bangalore, India (2007)*, S. 13–24

[Krause 2008]

KRAUSE, Jürgen: Neuere Entwicklungen des Semantic Web im Vergleich mit den Digital Library Ansätzen von *vascoda* / *GESIS IZ*, Bonn. 2008. – Forschungsbericht

[Lake 1998]

LAKE, M.: Desperately Seeking Susan OR Suzie NOT Sushi. / *The New York Times*, September 3. 1998. – Forschungsbericht



[Lewandowski 2005]

LEWANDOWSKI, Dirk ; LEWANDOWSKI, Dirk (Hrsg.): *Web Information Retrieval. Technologien zur Informtaionssuche im Internet*. Dinges&Frick GmbH, Wiesbaden, 2005

[Mann 2002]

MANN, Thomas M.: *Visualization of Search Results from the World Wide Web*, University of Konstanz, Diss., 2002

[Mann u. Reiterer 2002]

MANN, Thomas M. ; REITERER, Harald: Evaluation of Different Visualization of WWW Search Results. / University of Konstanz. 2002. – Forschungsbericht

[Marchionini 1989]

MARCHIONINI, G.: Information-seeking strategies of novices using a full-text electronic encyclopedia / Journal of the American Society for Information Science,. 1989. – Forschungsbericht

[Marchionini u. White 2008]

MARCHIONINI, G. ; WHITE, R.W.: Find What You Need, Understand What You Find. In: *Journal of Human-Computer Interaction* 23(3) (2008), S. 205–237

[Mayr u. a. 2008]

MAYR, Phillip ; MUTSCHKE, Peter ; PETRAS, Vivien: Reducing semantic complexity in distributed libraries: Treatment of term vegueness an document re-ranking. / GESIS, IZ-Bonn. 2008. – Forschungsbericht

[Milic-Frayling u. a. 2004]

MILIC-FRAYLING, N. ; JONES, R. ; RODDEN, K. ; SMYTH, G. ; BLACKWELL, A. ; SOMMERER, R.: Smartback: supporting users in back navigation. In: *Proceedings of the 13th International Conference on World Wide Web (WWW'04)*, 2004

[Muramatsu u. Pratt 2001]

MURAMATSU, J. ; PRATT, W.: Transparent Queries: Investigation Users' Mental Models Of Search Engines. In: *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and development in information retrieval (SIGIR'01)*, 2001

[Nielsen 1993]

NIELSEN, J.: Usability Engineering / Academic Press. 1993. – Forschungsbericht

[Nielsen 2000]

NIELSEN, J.: Why You Only Need To Test With 5 Users / Academic Press. 2000. – Forschungsbericht

[Nielsen 2003]

NIELSEN, J.: Usability 101, 2003 / Academic Press. 2003. – Forschungsbericht

[Omar Alonso 2000]

OMAR ALONSO, Ricardo Baeza-Yates: Integration of Visualization with Search Engines. 2000. – Forschungsbericht

[Petras 2006]

PETRAS, Vivien: *Translating Dialects in Search: Mapping between Specialized Languages of Discourse and Docuetary Languages*, University of California, Berkeley, Diss., 2006

[Pollock u. Hockley 1997]

POLLOCK, A. ; HOCKLEY, A.: What's Wrong with Internet Searching. In: *D-Lib Magazine* (1997)

[Puzicha 2008]

PUZICHA, Jan: *Recommind: Mindserver / Recommind*. 2008. – Forschungsbericht

[R. Kohavi 2007]

R. KOHAVI, D. S.: Practical guide to controlled experiments on the web: listen to your customers not to the hippo. In: *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'07)*. ACM Press New York, NY, USA., 2007

[Resnick u. Bandos 2002]

RESNICK, Marc L. ; BANDOS, Jennifer: *Best Practices in Search Interface Design* / Florida International University. 2002. – Forschungsbericht

[Russell 2006]

RUSSELL, D.M.: How do Google searchers behave? Improving search by divining intent / Stanford University. 2006. – Forschungsbericht

[S.Brin u. L.Page 1998]

S.BRIN ; L.PAGE: *The Anatomy of a Large-Scale Hypertextual Web Search Engine* / Stanford University. 1998. – Forschungsbericht

[Schneider 2009]

SCHNEIDER, Rene: *OPACS, Benutzer und das Web* / Genfer Haute Ecole de Gestion. 2009. – Forschungsbericht

[Sefelin u. a. 2003]

SEFELIN, R. ; TSCHELIGI, M. ; GUKKER, V.: Paper Prototyping What is it good for? A Comparison of paper and Computer based Low fidelity Prototyping. In: *CHI 2003*, 2003

[Shneidermann 1998]

SHNEIDERMAN, B.: Designing the User Interface. Strategies of Effective Human-Computer interaction. 1998. – Forschungsbericht

[Shneidermann 2002]

SHNEIDERMAN, B.: Leonardo's Laptop - Human Needs and the new Computing Technologies / MIT Press. 2002. – Forschungsbericht

[Shneidermann u. a. 1997]

SHNEIDERMAN, B. ; BYRD, D. ; CROFT, W.B.: Clarifying Search: A User-Interface Framework for Text Searches. 1997. – Forschungsbericht

[Shneidermann u. Plaisant 2004]

SHNEIDERMAN, B. ; PLAISANT, C. ; SHNEIDERMAN (Hrsg.) ; PLAISANT (Hrsg.): *Designing the user interface: strategies for effective human-computer interaction*. Addison Wesley, 2004

[Shneidermann u. Plaisant 2006]

SHNEIDERMAN, B. ; PLAISANT, C.: Strategies for evaluating information visualization tools: multi-dimensional in-depth long-term case studies. In: *Proceedings of the 2006 conference Advanced Visual Interfaces (AVI'04), Workshop on Beyond time and errors: novel evaluation methods for information visualization*, 2006, S. 1–7

[Stacy 1997]

STACY, D.A.: Lecture notes on Rapid Prototyping / University of Guelph, Ontario. 1997. – Forschungsbericht

[Staudemeyer 2007]

STAUEMEYER, Jörg ; STAUEMEYER, Jörg (Hrsg.): *Groovy für Java-Entwickler*. O'Reilly Verlag, 2007

[White u. a. 2007]

WHITE, R.W. ; BILENKO, M. ; CUCERZAN, S.: Studying the Use of Popular Destinations to Enhance Web Search Interaction. In: *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and development in information retrieval (SIGIR'07)*, 2007

# A Anhang - Fragebogen

Im Anschluss an den Usabilitytest wurde den Probanden ein kurzer Fragebogen zum STR-Prototyp als DropDown-Variante zum Ausfüllen vorgelegt.

## Fragebogen

Zum Schluss möchte ich Sie bitten, folgenden Fragebogen auszufüllen.

*Die Selbsterklärung der Applikation war...*

Schlecht -3 -2 -1 0 1 2 3 Sehr gut

Kommentar:

*Die Handhabung war...*

Schwierig -3 -2 -1 0 1 2 3 Leicht

Kommentar:

*Die optische Erscheinung war...*

Verwirrend -3 -2 -1 0 1 2 3 Eindeutig

Kommentar:

*Die empfohlenen Terme waren...*

Unverständlich -3 -2 -1 0 1 2 3 Halben weiter Kommentar:

*Die Funktionalität des STR-Prototyps als DropDown-Variante finde ich...*

Schlecht -3 -2 -1 0 1 2 3 Sehr gut

Kommentar:

*Ich würde den STR als Bereicherung empfinden*

Stimme nicht zu -3 -2 -1 0 1 2 3 Stimme voll zu

Kommentar: