Johann Wolfgang Goethe-Universität
Frankfurt am Main

Fachbereich Biologie und Informatik (15)
Institut für Informatik
Lehrstuhl für Datenbanken und Informationssysteme

Diplomarbeit

vorgelegt von: Martin Klossek
E-Mail: martin@klossek3000.de

Betreuer: Peter Werner

Bearbeitungszeitraum: 27. Mai bis 27. November 2003

Erstprüfer: Herr Prof. Dott.-Ing. R. Zicari

Optimierung der Personalisierung im Internet durch Kollaboratives Filtern

Kurzfassung / Abstract

Mit dem World Wide Web sind der Bestand und die Verfügbarkeit von Informationen rapide angewachsen. Der Einzelne hat Schwierigkeiten, der Menge an Informationen und Wissen Herr zu werden und so der Informationsüberflutung zu begegnen. Dieses Problem wurde von Forschern und Technikern bereits frühzeitig erkannt und durch verschiedene Konzepte wie die intelligente Suche und die Klassifikation von Informationen zu lösen versucht. Ein weiteres Konzept ist die Personalisierung, die das bedarfsgerechte Zuschneiden von Informationen auf die Bedürfnisse des einzelnen Anwenders zum Ziel hat.

Diese Arbeit beschreibt dazu eine Reihe von Personalisierungstechniken und im Speziellen das Kollaborative Filtern als eine dieser Techniken. Bestehende Schwächen des Kollaborativen Filterns wie zu dünn besetzte Benutzerprofile und das mangelnde Erkennen von Änderungen im Benutzerinteresse im Verlauf der Zeit werden durch verschiedene Ansätze zu entschärfen versucht. Dazu wird eine Kombination mit Inhaltsbasierten Filtern und die Verbreiterung der Datenbasis bewerteter Ressourcen betrieben. Ziel ist die Optimierung der Personalisierung, so dass Anwender besser auf sie abgestimmte Informationen erhalten. Ein Teil der beschriebenen Ansätze wird zudem in einem prototypischen Informationssystem umgesetzt, um die Machbarkeit und den Nutzen zu prüfen. Dazu werden der auf der Java 2 Enterprise Edition aufbauende WebSphere Applikationsserver von IBM und die relationale Datenbank DB2 UDB aus gleichem Hause eingesetzt.

Schlüsselwörter: Personalisierung, Kollaboratives Filtern, Inhaltsbasiertes Filtern, Internet, WWW, Internetportal, Informationsüberflutung

–

The World Wide Web led to an enormous growth of information available worldwide. Users today have difficulties managing the huge amount of data and knowledge presented by internet search engines and information portals. Scientists and technicians have investigated this problem, called information overkill, and have found solutions such as intelligent information search and classification. Personalization, another concept to reduce information overkill, tailors information to the needs of individual customers.

This diploma thesis describes several technologies for personalization, all of them being well-established in both commercial and academic information systems. The main emphasis is put on collaborative filtering, a personalization technology that is widely used but still suffers from several weaknesses. The author discusses poorly equipped user profiles as well as poor recognition of changing user needs over time and shows different concepts to reduce the impact of these problems. The main goal is to further improve personalization efforts. Both combining collaborative and content-based filtering and broadening the available rating values are possible solutions. To prove feasibility and value selected concepts are implemented in an information system prototype that uses the IBM WebSphere Application Server based on Java 2 Enterprise Edition and the IBM DB2 UDB database.

Keywords: Personalization, Collaborative Filtering, Content-Based Filtering, Internet, WWW, Portal Website, Information Overkill

Ehrenwörtliche Erklärung

Ich versichere, dass ich diese Diplomarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Frankfurt am Main, 27. November 2003

Martin Klossek

Inhaltsverzeichnis

Vorwort

Die Computertechnik hat in den letzten fünfzig Jahren zu einem enormen Anwachsen von verfügbaren Daten und Informationen beigetragen. Informationsnetze und besonders das Internet haben diesen Effekt in den vergangenen zehn Jahren noch erheblich verstärkt und erlauben Menschen weltweit einen direkten Zugriff auf das gesammelte Wissen – sowohl für private als auch für berufliche Zwecke.

Neben der Verfügbarkeit ist auch die Produktion von neuen Informationen geradezu explodiert, da jeder Mensch mit einem Internetzugang Inhalte zu den verschiedensten Themen nicht nur abrufen, sondern auf einfachste Weise auch publizieren kann. Den Universalgelehrten wie Goethe, der einst das gesamte Wissen der Menschheit verinnerlicht hatte, gibt es schon lange nicht mehr, da der Einzelne dazu bei weitem nicht in der Lage wäre.

Glücklicherweise benötigt man für den privaten Alltag und die tägliche Arbeit nur einen kleinen Ausschnitt des verfügbaren Wissens. Die Schwierigkeit liegt jedoch darin, die tatsächlich erforderlichen Informationen effizient zu erlangen und so der Informationsüberflutung zu begegnen. Dazu wurden von Forschern und Technikern unterschiedliche Konzepte entwickelt, von denen eines die Personalisierung ist.

Durch Personalisierung passen sich Informationssysteme an die individuellen Interessen und Bedürfnisse des Anwenders an, um ihm die nötigen Informationen bedarfsgerecht zu präsentieren. In dieser Arbeit werden dazu eine Reihe von Verfahren vorgestellt und die Methode des Kollaborativen Filterns als besonders effektiv empfohlen. Da jedoch auch das Kollaborative Filtern – wie alle anderen Personalisierungstechniken – weit davon entfernt ist, perfekt zu arbeiten und dem Anwender nur die Informationen zu präsentieren, die ihn tatsächlich interessieren, werden im Rahmen dieser Arbeit einige Ansätze entworfen, wie die Personalisierung in Internetinformationssystemen optimiert werden kann.

Aufgabenstellung

Ziel dieser Arbeit ist es, zunächst eine breite Einführung in das Gebiet der Personalisierung im Internet zu geben und darauf aufbauend die Verfahren und Vorzüge des Kollaborativen Filterns gegenüber anderen, existierenden Personalisierungstechniken zu beschreiben. Ausgerüstet mit diesem Grundwissen sollen bestehende Schwächen des Kollaborativen Filterns ermittelt und Optimierungsmöglichkeiten aufgezeigt werden. Die Entwicklung eines prototypischen Informationssystems dient abschließend zur Erprobung der entwickelten Ansätze und baut dazu auf aktuellen Internet-Technologien auf.

Die Kapitel im Überblick

Die Arbeit ist in fünf Kapitel gegliedert, die aufeinander aufbauend mit jedem Kapitel eine weitere Spezialisierung von Themen aus vorangegangen Kapiteln vornehmen. Nach dieser Einführung im Vorwort werden im ersten Kapitel grundlegende Eigenschaften von Internet und World Wide Web vorgestellt, die für die Personalisierung maßgeblich sind. Dabei geht es darum, wie Benutzer mit Websites interagieren, wie ihre Interaktionen in Protokollen festgehalten werden und wie der Da-

tenschutz im Internet behandelt wird. Zudem wird ein Eindruck davon vermittelt, was Informationsüberflutung bedeutet und warum eine Lösung für dieses Problem nötig ist.

Im zweiten Kapitel wird beschrieben, wie eine Personalisierung in Informationssystemen erfolgen kann. Die verschiedenen personalisierbaren Elemente wie Inhalte, Produkte oder Navigation werden vorgestellt und anschließend technische Verfahren erläutert, mittels derer die Personalisierung möglich ist. Eines dieser Verfahren – Kollaboratives Filtern – wird in diesem Kapitel nur kurz aufgeführt, um es einordnen und vergleichen zu können. Im Dritten Kapitel erfolgt jedoch eine ausführlichere Behandlung. Dazu werden dort zunächst die grundlegende Arbeitsweise beschrieben, sinnvolle Einsatzgebiete gezeigt und Fallbeispiele anhand bestehender Systeme präsentiert. Im Anschluss werden unterschiedliche Algorithmen des Kollaborativen Filterns untersucht und bestehende Schwächen dieser Technik nebst Lösungsmöglichkeiten aufgezeigt.

Im vierten Kapitel werden aufbauend auf den gewonnenen Erkenntnissen drei Optimierungsansätze vorgestellt, die die Leistung von auf Kollaborativen Filtern beruhenden Personalisierungsstrategien steigern sollen. Dabei werden sowohl mathematisch-formale Strukturen aufgestellt als auch Hinweise auf eine Implementierung in Software gegeben. Im fünften Kapitel schließlich wird beschrieben, wie Ausschnitte der vorgestellten Optimierungsansätze in einer konkreten Implementierung eines kleinen, prototypischen Informationssystems vorgenommen wurden. Hierbei werden die verwendeten Techniken erläutert und zentrale Punkte der Implementierung näher betrachtet.

Abschließend wird der Inhalt der Arbeit und ihrer Ergebnisse zusammengefasst und ein Ausblick gegeben, an welchen Stellen Weiterentwicklungen vorgenommen werden könnten.

Verfügbarkeit und Technik

Das vorliegende Dokument wurde mit Microsoft Word XP nach der neuen Rechtschreibung geschrieben. Zum Formelsatz kam MathType 5.0 zum Einsatz und die Grafiken wurden größtenteils mit Microsoft PowerPoint 2003 erstellt. Die abschließende Konvertierung in PDF erfolgte mit Adobe Acrobat 6.0.

Die Arbeit sowie weitere Grafiken und der Programmcode der Implementierung können unter

http://www.klossek3000.de/cf/

abgerufen werden und stehen auch auf der beigefügten CD bereit. Darüber hinaus freue ich mich besonders über Anregungen und Kritik und bin per E-Mail an martin@klossek3000.de zu erreichen.

Danksagung

Die Diplomarbeit wurde von mir alleine erstellt, aber ohne die Unterstützung von helfenden Händen im Hintergrund wäre sie nicht möglich gewesen. Daher möchte ich mich an erster Stelle für die Unterstützung durch meine Eltern bedanken, die sie mir in der gesamten Zeit meines Studiums entgegengebracht haben und durch die das Studium möglich wurde.

Dank gilt auch meinem Betreuer Peter Werner, der immer wieder interessante Einwürfe und Korrekturen eingebracht hat. Zudem hat er mir erst das interessante Thema des Kollaborativen Filterns nahe gelegt. Weiterhin bedanke ich mich bei meinen Freunden Frank Bergmann, Martin Meedt, Marina Tzanova und Fabian Wleklinski für das Korrekturlesen der Arbeit und so manchen kritischen Kommentar.

1 Grundlagen

1.1 Historischer Abriss

Waren Informationen über die unterschiedlichsten Themen vor wenigen Jahren noch schwer oder gar nicht zu erhalten, so sind wir heute durch die Verfügbarkeit des Internets und seiner imposanten Entwicklung nicht selten mit Informationen überlastet. Wurden Informationen in grauer Vorzeit von Mensch zu Mensch mündlich weitergegeben, kamen später die Schriftform und handschriftlich gesetzte Bücher hinzu. Die Alexandrinische Bibliothek, die um 300 vor Christus in der ägyptischen Stadt Alexandria gegründet wurde, umfasste zu ihrer Blütezeit immerhin 700.000 Schriftrollen (siehe [Ber70]). Eine beachtliche Menge für das Zeitalter zwar, aber der Zugriff darauf war wenigen vorbehalten – sei es aus wirtschaftlichen Gründen oder wegen mangelnder Lesefähigkeit und Bildung. Es herrschte also keinesfalls Informationsüberfluss sondern eine allgemeine Knappheit an Information.

Der im 15. Jahrhundert aufkommende Buchdruck katapultierte die Verbreitung und Verfügbarkeit von Informationen in ein neues Zeitalter, da Informationen wesentlich schneller an größere Bevölkerungsschichten verteilt werden konnten und Bücher ihren elitären Charakter verloren (Informationen zu Gutenberg, dem Erfinder des modernen, westlichen Buchdruckes, unter [Gut1] und [Gut2]). Die kontinuierlich gestiegene Bildung der Bevölkerung und öffentlich zugängliche Bibliotheken beschleunigten den Zugriff auf und das Interesse an Informationen. Einige hundert Jahre später etablierten sich in der Mitte des 20. Jahrhunderts die Computertechnik und noch einmal wenige Jahre später schließlich das Internet, die beide zusammen zu einer unüberschaubaren Flut von verfügbaren Informationen geführt haben.

Der einzelne Mensch ist heute nicht mehr in der Lage einzelne Themenkomplexe vollständig zu erfassen, geschweige denn der gesamten Informationsflut im Internet Herr zu werden. Stattdessen begnügt man sich mit mehr oder weniger großen Ausschnitten des Wissens. Informationen sind jedoch zum unerlässlichen Antrieb von Wirtschaft, Staat und privatem Leben geworden. Populistisch wird daher gerne von der Informations- oder Wissensgesellschaft gesprochen, die die Industriegesellschaft beerbt hat (siehe [EK96], [Bul02]). Die Verwaltung und Verarbeitung von Informationen verlangt entsprechend nach intelligenten Konzepten, damit Menschen in der angebrochenen Informationsgesellschaft bestehen können und nicht von der Informationsflut überschwemmt werden.

1.2 Informationsüberflutung

Die gewaltige Menge an global verfügbaren Daten und Informationen, die in Computersystemen und Datenbanken verwaltet werden, überfordert den menschlichen Geist, da wir nur eine begrenzte Menge von Signalen und Daten verarbeiten können. Man spricht von der Informationsüberflutung oder dem Information Overkill. Selbst lokal in Personal Computern eingesetzte Festplatten haben heute schon ein ungeheures Fassungsvermögen, so dass die Orientierung in den dort gespeicherten Datenbeständen schwer fällt. Unternehmensdatenbanken, Wissensarchive oder Internet-

Suchmaschinen verwalten noch weitaus größere Datenmengen. Die Suchmaschine Google[1] beispielsweise hat gegenwärtig mehr als drei Milliarden Dokumente indiziert, andere Systeme wie AllTheWeb[2] stehen ihr kaum nach. Die gesamte im Internet verfügbare Menge an Daten ist aber noch deutlich umfangreicher, da längst nicht alle Websites von den Suchmaschinen erfasst werden und viele Informationsangebote nur einen Teil der Daten öffentlich bereitstellen (siehe [CC03]).

Die Informationsüberflutung beeinträchtigt eine sinnvolle und produktive Arbeits- und Lebensweise, da der Aufwand für die Filterung und die Suche rapide angestiegen ist und entsprechend die verfügbare Zeit für die eigentlich relevante Informationsnutzung schmälert. Im schlimmsten Fall kommt die Nutzung der nötigen Informationen durch die Kosten der Beschaffung vollständig zum Erliegen. Ein praxisnahes Beispiel ist die unerwünschte Zusendung von Werbe-E-Mails, kurz Spam genannt ([Mue99], [SF98]). Die Menge aller E-Mails wird so für den Anwender in solche mit nützlichen und gewünschten und solche mit weniger nützlichen bis gar unnützen Informationen aufgeteilt. Steigt der Anteil der unnützen Informationen in Form von Spam-E-Mails, muss der Anwender mehr Aufwand in die Filterung investieren und kann sich weniger den nötigen Aufgaben widmen. Aber Spam ist nur eine Form der Informationsüberflutung. Die Gunst der vielfältig verfügbaren hat sich so in eine Last der zu viel vorhandenen Informationen gekehrt.

Glücklicherweise muss der einzelne Mensch – sei es als Informationsarbeiter im Unternehmen oder im Privatleben – nur Zugriff auf einen Ausschnitt des gesamten weltweiten Wissens haben. Diese Sicht auf die angebotenen Informationen richtig zu finden, darin liegt die Schwierigkeit und möglicherweise auch die Kunst des Einzelnen. Eine nahe liegende Lösung wäre, das Rad der technologischen Weiterentwicklung zurückzudrehen und schlicht auf globale Datenbanken und Netzwerke zu verzichten. Genauso nahe liegend wie utopisch ist dieser Lösungsversuch jedoch. Zudem steht dann die Frage im Raum, ob ein solches Zurückdrehen der Entwicklung überhaupt wünschenswert wäre. Das liegt auch sicherlich im Ermessensspielraum jedes einzelnen Betrachters.

Eine zweite, alternative Lösungsvariante ist hingegen der Einsatz weiterer Technik und neuer Konzepte zur Lösung des Problems Informationsüberflutung. Die Personalisierung versucht hier Verfahren anzubieten, mit denen die gewaltige Menge an täglich auf uns einströmenden Informationen kanalisiert, gefiltert, sortiert und aufnahmegerecht zubereitet werden kann.

Betrachtet man die Entwicklung der wenigen zurückliegenden Jahre, in denen das Internet und das World Wide Web in weiten Bevölkerungsschichten der Industrieländer etabliert wurden, so kann man davon ausgehen, dass es in Zukunft noch weit mehr Daten und Informationen in Online-Form geben wird. Teilweise sind die Angebote bereits in traditioneller Form – beispielsweise in Büchern, Bibliotheken oder schlicht in der Überlieferung – vorhanden und werden nach und nach in eine für das Internet gerechte Form gebracht. Ständig entstehen aber auch neue Informationen, teils mit hohem, teils mit weniger hohem Nutzwert, da immer neue Menschen ins Netz strömen (siehe ARD/ZDF-Online-Studie 2002 [AZ03]). Man könnte daher die These aufstellen, dass es gegenwärtig noch viel zu wenige Informationen gibt, die digital verfügbar, global zugreifbar und einfach auffindbar sind. Umso mehr werden daher Methoden von allgemeinem Interesse sein, mit denen das individuell nötige Informationsfenster zugeschnitten werden kann.

1.3 Grundlagen von World Wide Web-Techniken

In den folgenden Abschnitten werden technische Grundlagen des World Wide Web (WWW) beschrieben, um die Basis für Überlegungen zur Personalisierung im Internet zu legen. Dabei geht es

[1] siehe www.google.com
[2] siehe www.alltheweb.com

am Anfang um grundlegende Konzepte wie die Arbeitsweise und den Zugriff auf Webserver und den Aufbau von Webseiten. Später werden dann die Protokollierung von Zugriffen und die Auswertung der Protokolle erläutert. Konzepte zur Überwindung von Schwachstellen in World Wide Web-Technologien wie Cookies oder HTTPS – Stichwort Zustandslosigkeit – werden ebenfalls behandelt und im Folgekapitel ein kleiner Exkurs zu Aspekten der Wahrung der Privatsphäre gemacht. Die Breite der Themen richtet sich dabei ganz nach dem Bedarf an Wissen, der für die Betrachtung der Personalisierung in den weiteren Kapiteln benötigt wird.

1.3.1 Architektur von Webzugriffen

Internetnutzer können von ihrem Arbeitsplatzrechner aus mit Hilfe von Webbrowsern auf Internetseiten zugreifen. Dabei fordert der Webbrowser die gewünschten Seiten von einem Webserver an. Findet der Server die Daten und hat der Benutzer die entsprechenden Zugriffsrechte, liefert der Server sie an den Webbrowser zurück. Häufig besteht eine solche Antwort aus verschiedenen Datenströmen wie HTML-Dateien, Bildern und anderen Elementen.

Bei jeder Anfrage sendet der Webbrowser einen eindeutigen Bezeichner des angeforderten Objekts (URI) und eine Reihe von Metadaten. Optional sind kleine clientseitige Datenspeicher – so genannte Cookies. Der eindeutige Bezeichner entspricht zumeist einem Pfad auf eine Datei oder auf eine Applikation im Webserver. Bei den Metadaten handelt es sich beispielsweise um Angaben wie Name und Fähigkeiten der Browsersoftware. Mit Cookies schließlich können kleine Mengen von Daten sitzungsbezogen oder dauerhaft im Webbrowser gespeichert werden. Darauf werde ich weiter unten noch ausführlicher eingehen.

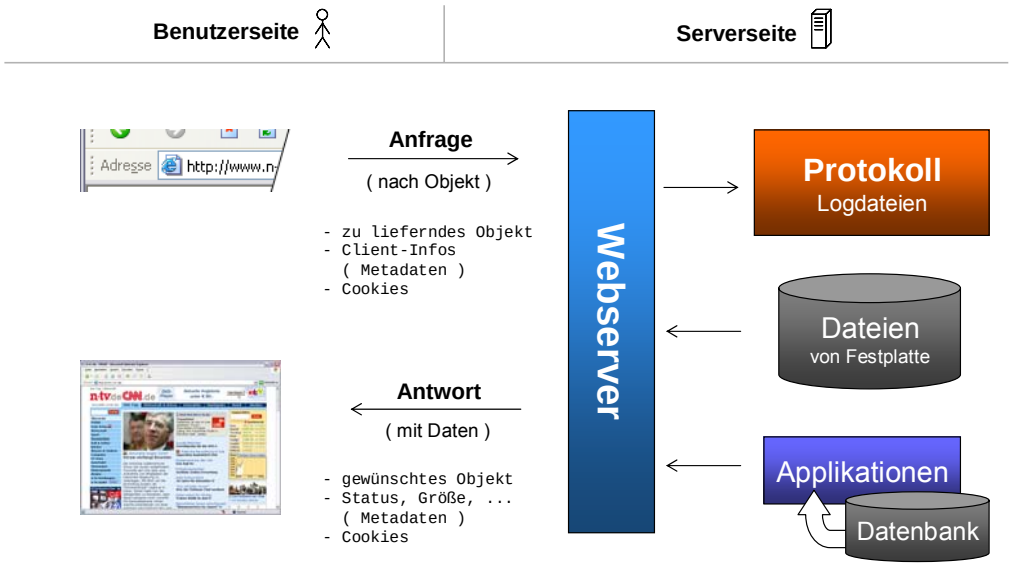

Abbildung 1 - Architektur von Webzugriffen

Der Webserver seinerseits liefert das angeforderte Objekt, soweit es vorhanden ist und keine Zugriffsbeschränkungen bestehen, und eine Reihe von Metadaten wie Größe, Zeitraum der Gültigkeit und Statuscodes. Optional ist ebenfalls die Rückgabe eines oder mehrerer Cookies. Bei den gelieferten Objekten kann es sich um beliebige Daten handeln – binär wie textuell. Metadaten und Cookies werden als Text übertragen (detaillierte Spezifikationen zu HTTP, dem Übertragungsprotokoll zwischen Webbrowser und -server, finden sich unter [IETF99] und z. B. [Wong00]).

Aus technischer Sicht bestehen sowohl Anfrage als auch Antwort aus einem oder zwei Datenblöcken. In jedem Fall wird der erste Datenblock als ein Header mit Textdaten geliefert. Hier werden

die Metadaten wie Statuscodes oder Dokumentgültigkeit und die Cookies übertragen. An ihn kann sich getrennt durch eine Leerzeile ein weiterer Datenblock anschließen, der die Nutzlast wie HTML-Dateien oder Grafiken aufnehmen kann. Im Falle von HTML- und Textdateien enthält der Block so Textdaten. Bei Grafiken und anderen binären Dokumenten enthält er Binärdaten.

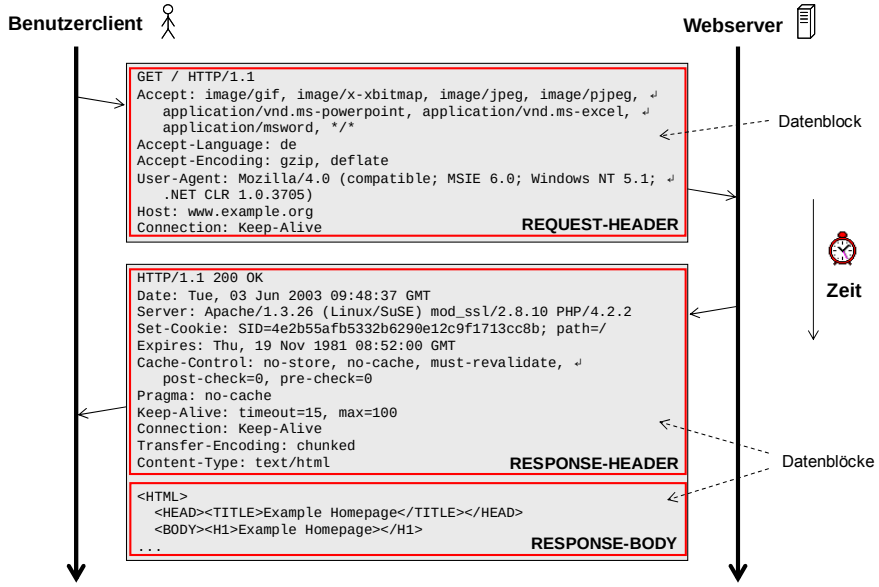

Abbildung 2 - Datenfluss zwischen Client und Webbrowser über HTTP

Die vom Client gesendete Anfrage enthält im Header einen Befehl, der den Webserver anweist, eine bestimmte Aktion auszuführen. Dazu gehören in erster Linie das Anfordern von Ressourcen mit dem Befehl GET oder nur das Anfordern von Metadaten ohne Nutzlast mit dem Befehl HEAD. Bei einer erfolgreichen Antwort des Webservers auf die GET-Anforderung folgt nach dem Header mit den Metadaten ein zweiter Datenblock mit der angeforderten Ressource. Beim HEAD-Befehl hingegen werden nur die Metadaten zurückgeliefert und es folgt kein weiterer Block.

Weitere Befehle sind das Übertragen von Daten vom Client auf den Server mit den Befehlen POST und PUT. Zudem gibt es eine Reihe zusätzlicher Befehle, die aber im praktischen Einsatz nur eine untergeordnete oder gar keine Bedeutung haben. Sie sind zwar in HTTP definiert, aber entweder gar nicht in handelsüblicher Webserversoftware implementiert oder werden von Websites nicht genutzt.

1.3.2 Repräsentation von Informationen im World Wide Web – Anatomie von Websites

Websites bestehen typischerweise aus einer Vielzahl von HTML-Dokumenten, die entweder physikalisch im Dateisystem des Webservers als Datei vorliegen oder dynamisch aus Skripten oder Anwendungen generiert werden. HTML-Dokumente enthalten vor allem Texte als Container für Informationen, aber auch Gliederungsanweisungen. Sie können mit Hyperlinks auf andere Dokumente sowohl innerhalb wie außerhalb der eigenen Website verlinken und so ein Netzwerk zwischen verschiedenen Informationseinheiten bilden. Daher auch die Namen HyperText Markup Language[3] und Hypertext Transfer Procotol.

[3] Die Spezifikation des W3C und weitere Infos finden sich hier http://www.w3.org/MarkUp/

Verlinkt werden kann auf andere HTML-Dokumente, reine Textdaten oder Binärdokumente, aber genauso können Grafiken eingebettet werden. Kurzum, jede im Internet frei verfügbare und durch ein URI identifizierte Ressource kann adressiert werden und so kann ein umspannend verknüpftes Netzwerk von Informationseinheiten aufgebaut werden. Der Webbrowser stellt Dokumente und Grafiken mit den Anweisungen von HTML dar. Zudem können Java-Applets, Flash-Filme und andere aktive Elemente eingebunden sein. Ferner besteht mit JavaScript und DynamicHTML die Möglichkeit, ebenfalls Interaktion innerhalb der Website anzubieten. Dies reicht von auf- und zuklappenden Menüs bis zu kleinen Anwendungen, die Berechnungen durchführen. Der Vorteil ist, dass diese Skripte nur im Webbrowser ablaufen und keine Verbindung mit dem Webserver nötig ist. Entsprechend werden sie schnell ausgeführt und bieten eine hohe Reaktionszeit.

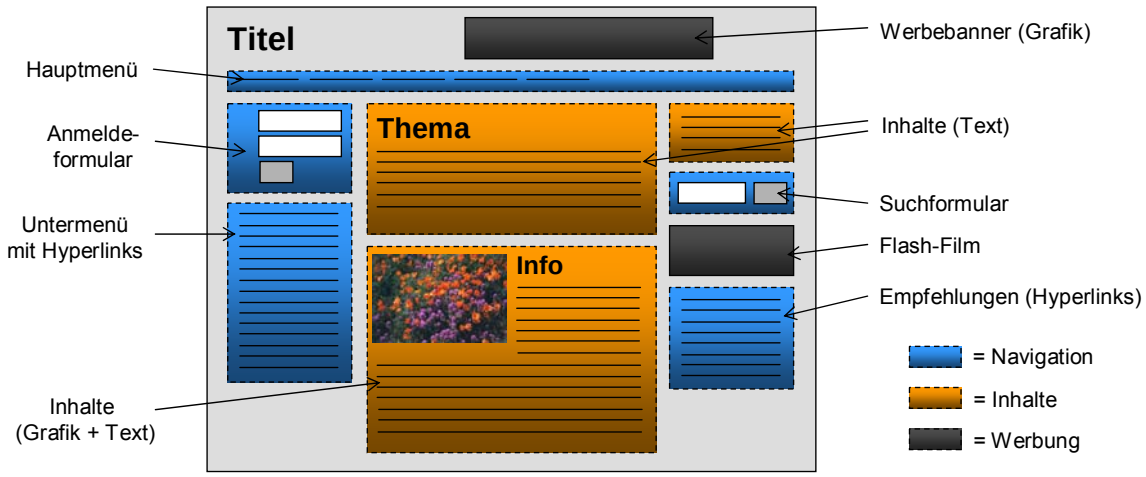

Abbildung 3 - Exemplarischer Aufbau einer Website

In Abbildung 3 sind alle Elemente einer umfangreichen Website zusammengestellt. Tatsächliche Implementierungen müssen natürlich nicht alle aufgeführten Bestandteile enthalten, aber die Grundzüge sind bei allen Websites gleich. So stehen in den Navigationsflächen Hyperlinks auf andere Inhalte und es gibt Interaktionsmöglichkeiten wie die Suche. Mit dem Anmeldeformular kann der Benutzer durch Eingabe seiner persönlichen Kennung und eines Passwortes Zugang zu geschützten oder personalisierten Bereichen der Website erhalten.

Die Inhaltsflächen können Texte, Grafiken, Auflistungen und Tabellen enthalten. Wesentlich ist, dass häufig mehrere Informationseinheiten auf einer angezeigten Seite dargestellt werden. Während die Gliederung durch HTML-Auszeichnungen erfolgt, kann das Layout wie Schriftgestaltung und Farbwahl durch Cascading Style Sheets (CSS)[4] eingestellt werden.

Sinnvoll ist, bei der Analyse von Inhalten oder der Auswertung von Zugriffsprotokollen (siehe Kapitel 1.3.5) zwischen den unterschiedlichen bereitgestellten Daten zu unterscheiden, da sie verschiedene Einsatzzwecke und Nutzwerte haben. Ein Webserver liefert zusammengefasst folgende zu unterscheidende Elemente aus

- statische HTML-Dokumente
- dynamisch durch Skripte und Anwendungen generierte HTML- Dokumente
- informationstragende Grafiken (wie Elemente zur Navigation, Schaubilder oder Werbebanner)
- clientseitige Applikationen wie Flash, Java-Applets oder ActiveX-Komponenten

[4] Das W3C bietet hierzu Informationen unter http://www.w3.org/Style/CSS/

- nicht informationstragende Grafiken und andere Elemente für Layout und Funktionalität (wie Cascading Style Sheet-Dateien mit der typischen .css-Endung oder clientseitiger Skriptcode wie JavaScript mit .js-Endung)
- binäre Daten wie PDF- und Office-Dokumente oder ZIP-Archive

Hierdurch wird deutlich, dass Websites aus einer Vielzahl von Dateien bestehen, die durch Hyperlinks miteinander verknüpft sind. Zum Aufbau von Websites sind zudem vielfältige Kenntnisse der genannten Technologien nötig.

1.3.3 Cookies

Mit Hilfe von Cookies können kleine Mengen von Informationen im Webbrowser des Benutzers gespeichert werden. Angemerkt sei an dieser Stelle, dass ein Webbrowser nicht nur auf Arbeitsplatzrechnern laufen kann, sondern auch in Kleinstcomputern wie Personal Digital Assistants (PDAs) und Mobiltelefonen. Cookies können bei all diesen Systemen eine Rolle spielen, da Daten mit ihnen entweder dauerhaft oder für den Zeitraum der Sitzung im Webbrowser gespeichert werden können. Mit ihnen kann die Zustandslosigkeit von HTTP auf logischer Ebene überwunden werden, indem man Schlüssel für Sitzungen oder Benutzerkennungen speichert. Die einzelnen Datenzugriffe bleiben im Gegensatz zu statusbehafteten Protokollen technisch dennoch unabhängig voneinander. Die Anwendung bekommt allerdings Statusinformationen, um beispielsweise einen wiederkehrenden Benutzer, der bereits einen Cookie mit seiner Benutzerkennung in seinem Webbrowser gespeichert hat, zu erkennen. Entwickelt wurde das Konzept der Cookies von der Firma Netscape, die Spezifikation findet sich hier [NCC99].

Bei einem einzelnen Cookie handelt es sich um eine Kombination aus Namen und zugeordnetem Wert sowie einigen wenigen, aber allesamt optionalen Metadaten. Bei den Werten oder Nutzdaten kann es sich um Benutzerkennungen, Präferenzen wie gewünschte Sprache und Layout oder um Sitzungskennungen handeln, je nach Bedarf der Website. Der Inhalt wird dabei immer url-kodiert abgelegt.

Feldname	
Name=Wert	die zu setzende Variable mit der Bezeichnung *Name* und dem Inhalt *Wert.* Dieses Feld ist Pflicht.
domain=Hostname	der *Hostname*, für die der Cookie zu setzen ist, also die Quelle des Cookies. Wird das Feld ausgelassen, wird der *Hostname* bzw. die IP-Adresse des den Cookie setzenden Servers verwendet.
expires=Verfallsdatum	*Verfallsdatum* des Cookies, nachdem der Browser die Daten im internen Speicher löscht. Das Datum wird mit Wdy, DD-Mon-YYYY HH:MM:SS GMT formatiert. Wird kein *Verfallsdatum* angegeben, ist die Lebensdauer auf die Sitzung des Browsers beschränkt.
path=Pfad	ein optionaler *Pfad*, der dem *Hostname* angehängt wird und die Quelle des Cookies weiter eingrenzen lässt.
secure	ein Flag ohne Parameter, dass das Setzen des Cookies nur bei sicheren Verbindungen – also über HTTPS – erlaubt

Tabelle 1 - Felder für die Übermittlung von Cookies im HTTP-Header

Ein Cookie wird zumeist auf dem Webserver erzeugt und an den Webbrowser zurückgeliefert. Bei jedem weiteren Zugriff sendet der Webbrowser alle mit der Website verbundenen Cookies an den

Webserver zurück. So findet ein ständiger Datenaustausch der Cookiedaten statt, der mit einem *Verfallsdatum* versehen werden kann. Transportiert wird der Cookie vom Server zum Browser als HTTP-Header, der folgendes Format hat und die Felder aus Tabelle 1 verwendet:

```
Set-Cookie: NAME=Wert; expires=Verfallsdatum;
            path=Pfad; domain=Hostname; secure
```

Der Umbruch ist hier nur kosmetischer Natur, bei der Datenübertragung wäre Set-Cookie eine Zeile. In der Gegenrichtung vom Browser zum Server wird bei jedem Zugriff auf die mit Hostname und Pfad spezifizierte Website der Cookie übermittelt, ebenfalls als HTTP-Header:

```
Cookie: NAME1=Wert1; NAME2=Wert2; NAME3=Wert3; ...
```

Der Webbrowser verwaltet dazu alle Cookies, die er von verschiedenen Webservern erhalten hat und löscht diejenigen, deren Verfallsdatum überschritten ist oder die nur für eine Sitzung gültig waren. Cookies können jedoch auch im Webbrowser selbst mit Skriptsprachen wie JavaScript erzeugt und verwaltet werden.

Um Missbrauch vorzubeugen sind Cookies weiterhin an bestimmte Websites und Verzeichnisse gebunden (*domain*- und *path*-Felder) und können auch nur eine beschränkte Menge von Daten speichern (wie eben die beschriebenen kurzen Kennungen oder auch Schlüsselwerte). Ein Webserver kann dabei mehrere Cookies gleichzeitig setzen. Ferner bieten gängige Webbrowser umfangreiche Cookieverwaltungen an, mit denen einerseits die gespeicherten Cookies eingesehen werden können und die andererseits für jede Website festlegen, ob Cookies gesetzt werden dürfen oder nicht. Bedarfsweise kann der Benutzer bei jedem Versuch, einen Cookie zu setzen, vom Browser gefragt werden, ob er dies zulassen will oder nicht.

Der Vorgang beim Setzen eines Cookies könnte exemplarisch folgendermaßen ablaufen (siehe z. B. [Wong00]). Denkbar wäre dazu, dass der Benutzer zunächst ein Formular mit seinem Vor- und Nachnamen auf der Website ausgefüllt hat und der Webserver daraufhin einen Cookie setzen will, um den Benutzer beim nächsten Besuch wieder zu erkennen. Das würde so aussehen:

```
Set-Cookie: firstname=Martin; expires=Wed, 31-Dec-2003 23:59:59 GMT;
            path=/; domain=www.klossek3000.de
Set-Cookie: firstname=Martin; expires=Wed, 31-Dec-2003 23:59:59 GMT;
            path=/; domain=www.klossek3000.de
```

Der Webbrowser würde dann die Cookiedaten bei jedem Zugriff auf die Domain mit Pfad „www.klossek3000.de/“ im HTTP-Header mitschicken und zwar bis zum Ende des Verfallsdatums am 31.12.2003.

```
Cookie: firstname=Martin; lastname=Klossek;
```

Typischerweise werden in Cookies auch Sitzungsschlüssel oder Benutzerkennung gespeichert. Das kann dann beispielsweise beim Transport vom Browser zum Server so aussehen, wobei hier eine Benutzerkennung und zwei Schlüssel für Sitzungen transportiert werden.

```
Cookie: CSLOGINNAME=klossek;
        skey=767CE9B4E8266EB370D18207151BE2-00-2392;
        CSSESSID=696b4431a3116fc14f1610155851d1
```

Wie beschrieben sind Cookies also recht harmlose Name-Wert-Paare und keine ausführbaren Programme beispielsweise. Darüber hinaus wurden Cookies von Anfang an deutliche Beschränkungen auferlegt (siehe [NCC99], [Wer00]), die von Webbrowser zu erfüllen sind.

- maximal 300 gespeicherte Cookies im Browser
- ein Cookie darf maximal 4 Kilobyte lang sein (Name und Wert inkl.)

- pro Server dürfen 20 Cookies gesetzt werden

Soll der insgesamt 301. Cookie gesetzt werden oder für einen Server der 21., dann soll der Cookie zuerst gelöscht werden, der am wenigsten benutzt wurde. Zwar können einzelne Softwarepakete in ihrer Implementierung davon abweichen, aber dennoch sind die Datenmengen überschaubar: maximal 1200 Kilobyte bei 300 Cookies zu 4 Kilobyte.

Zu erwähnen ist noch, dass viele Benutzer trotz dieser recht rigiden Sicherheitsvorkehrungen und der technisch passiven Eigenschaft von Cookies Bedenken gegen ihren Einsatz haben, da sie Datenschutzprobleme befürchten. Ganz auszuräumen sind diese Bedenken nicht, die Gefahren sind aber dennoch überschaubar, wie an verschiedenen Stellen in den folgenden Kapiteln aufgezeigt wird, wenn sich eine passende Gelegenheit dazu ergibt.

1.3.4 Sitzungen

Ruft ein Benutzer Informationen von einer Website ab, dann spricht man von einer Sitzung. In einer Sitzung werden verschiedene Informationen abgerufen und der Benutzer kann der Website über eine Anmeldemöglichkeit bekannt sein. Verlässt er die Website wieder oder ist für einen gewissen Zeitraum inaktiv, dann endet die Sitzung. Bei einem erneuten Zugriff, startet eine neue Sitzung.

Betrachtet man die Zugriffe eines einzelnen Benutzers auf das Informationsangebot einer Website, so kann man jede abgerufene Informationseinheit chronologisch aneinanderreihen – vom ersten bis zum letzten Besuch der Website. Dadurch entsteht eine aneinander gereihte und lange Kette von Zugriffen. Innerhalb dieser Kette finden sich aber Inseln von Zugriffen, da ein einzelner Benutzer Informationen nicht permanent sondern zusammenhängend nur zu bestimmten Zeitpunkten abruft – man spricht daher von einzelnen Sitzungen (in der Fachterminologie auch englisch Session genannt).

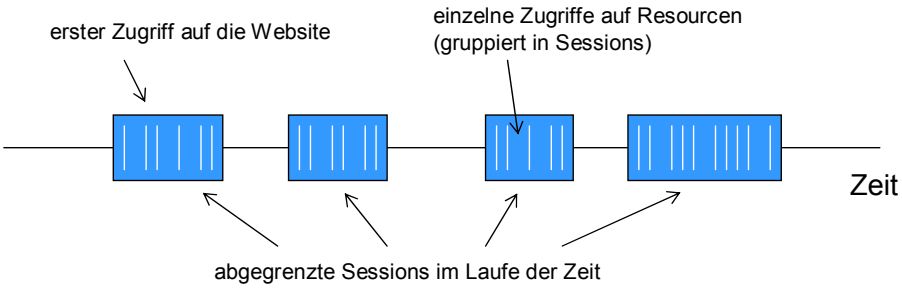

Abbildung 4 - Erläuterung des Sitzungsbegriffes

Mit einer Sitzung kann eine Reihe von Zuständen festgehalten werden, beispielsweise wer der zugreifende Benutzer ist, wann die Sitzung begonnen hat oder welchen Browser der Benutzer verwendet. Sogar der Warenkorb bei einem Online-Shop ist sitzungsbasiert. Auf Webserverseite wird dies häufig mit einem Datenspeicher realisiert, in den verschiedene Werte gelegt werden können, die jeweils einer Sitzung zugeordnet sind – ein Sitzungsspeicher für jeden Benutzer. Eine Sitzung muss im System aber nicht explizit abgebildet sein, sondern stellt als abstraktes Konstrukt prinzipiell eine Folge von Zugriffen eines Benutzers zu einem bestimmten Zeitpunkt dar. Jeder erste Zugriff eines Benutzers auf das Informationsangebot der Website zu einem bestimmten Zeitpunkt öffnet eine neue Sitzung und der letzte Zugriff in diesem Zeitrahmen schließt sie.

Jeder nach dem ersten erfolgende Zugriff auf eine Ressource bzw. konkreter eine Informationseinheit wird der aktuellen Sitzung zugeordnet, so dass eine Art Kurzzeitgedächtnis für das Benutzer-

verhalten aufgebaut wird[5]. Darüber hinaus können die historischen Sitzungen aus den Logdateien des Webservers extrahiert werden und sind ein Teil des Langzeitgedächtnisses des Benutzerverhaltens.

Da das World Wide Web-Protokoll HTTP keine Sitzungen unterstützt, wird jeder Zugriff eigenständig und unabhängig von anderen Zugriffen behandelt. Für die Verwaltung von Sitzungen ist die Identifikation und Sicherung von Zuständen aber essentiell, um einzelne Zugriffe als zusammengehörig gruppieren zu können. Im Laufe der Geschichte des World Wide Web haben sich daher verschiedene Verfahren etabliert, die hier kurz beschrieben werden.

- Cookies mit Session- oder Benutzerkennung
- URL-Rewriting mit Sessionkennung
- Benutzer-Authentifizierung (löst Sessionproblem nur partiell)
- HTTPS-Sitzung

Zwar nicht zur Sicherung von Zuständen während des Betriebes, aber zur Ableitung und Abgrenzung von Sitzungen bei der Logdateianalyse hat sich auch das Verfahren bewährt, eine Zeitspanne der Inaktivität – beispielsweise 25 Minuten – als Unterbrechung zweier Sessions zu interpretieren. Aufgrund der Zustandslosigkeit von HTTP ist diese Ableitung aber mit gewissen Unsicherheiten behaftet, die nur eine grobe Abgrenzung einzelner Sitzungen ermöglicht.

Bei der der Identifikation einer Sitzung hat sich ein Verfahren mit Einsatz von Cookies bewährt (siehe Kapitel 1.3.3). Dazu wird jeder neu gestarteten Sitzung eine Sitzungskennung *sid* zugeordnet. In der Regel ist das eine lange Zeichenkette mit Ziffern und alphanumerischen Zeichen. Ein Beispiel ist

```
sid = pzdlpx3dpqk2yx55mzo4uv
```

Nach dem Start der Sitzung schickt der Webserver einen Cookie mit der Sitzungskennung an den Browser. Bei eingeschalteten Cookies sendet der die Kennung fortan bei jedem Zugriff auf den Webserver mit und dort kann so der Zugriff als der Sitzung zugehörig erkannt werden. So lassen sich beispielsweise die Anmeldung eines Benutzers sichern und personalisierte Inhalte ausliefern. Der Nachteil ist, dass Cookies im Browser unterstützt und eingeschaltet sein müssen, was aus Datenschutzbedenken nicht immer der Fall ist.

Ein anderes Verfahren – das URL-Rewriting – benötigt keine Cookies und funktioniert daher immer. Dabei wird beim Start einer Sitzung durch den ersten Zugriff des Benutzers ebenfalls eine Kennung generiert. Der Webbrowser leitet die Anfrage an einen URL weiter, der dem ursprünglichen entspricht, aber zusätzlich die Sitzungskennung enthält. Bei jeder weiteren Anfrage schickt der Browser die Kennung so mit zum Server.

```
http://www.example.org/index.aspx?ID=96517
↓
http://www.example.org/pzdlpx3dpqk2yx55mzo4uv/index.aspx?ID=96517
```

Ein Nachteil ist hier, dass die URLs wesentlich länger und kryptischer werden. Speichert der Benutzer einen solchen URL als Bookmark, würde beim nächsten Abruf eine Sitzung suggeriert. Die Sitzung mit dieser Kennung ist dann aber sehr wahrscheinlich schon abgelaufen und der Webserver muss entsprechend reagieren und eine neue Sitzung mit einer neuen Kennung generieren.

[5] Der Begriff Gedächtnis ist natürlich sehr mächtig und das vorgestellte Konzept wird diesem – insbesondere dem menschlichen Gedächtnis – nicht gerecht, aber die Unterscheidung zwischen Kurz- und Langzeitgedächtnis verdeutlicht den Unterschied zwischen einer einzelnen Sitzung und der Gesamtheit aller historischen Sitzungen eines Benutzers

Ein weiteres Problem ist, dass alle Hyperlinks im jeweils ausgelieferten HTML-Dokument angepasst werden müssen und die Sitzungskennung enthalten. Wenn die Sitzungskennung nur bei einem Link fehlt, geht die Sitzung verloren. Die Linkanpassung ist bei statischen Seiten nahezu unmöglich und bedarf technischer Tricks wie Server Side Includes (siehe [Apa1]), die nicht immer verfügbar sind.

Weniger gebräuchlich aber dennoch möglich ist die Sitzungsidentifikation mit der HTTP-Benutzer- Authentifizierung und mit dem verschlüsselten Protokoll HTTPS. Die Identifikation mit einer Sessionkennung muss hier aber genauso implementiert werden. Zudem hat die Benutzer-Authentifizierung Sicherheitsprobleme, da Passwörter bei jedem Zugriff nur schwach verschlüsselt übertragen werden. HTTPS ist hingegen sicher, aber die Verschlüsselung kostet viel Rechenzeit und macht das Verfahren daher auch nur für bestimmte Bereiche interessant. Beispielsweise, wenn sowieso eine Absicherung wie im Homebanking nötig ist.

1.3.5 Protokollierung von Zugriffen

Jede Anfrage an den Webbrowser wird mit verschiedenen Metadaten wie Zeitpunkt des Zugriffs, Größe der übertragenen Daten und einigen weiteren protokolliert. Für die Beobachtung des Benutzerverhaltens sind insbesondere die informationstragenden Elemente interessant. Dazu gehören die meisten statisch wie dynamisch generierten HTML-Seiten sowie informationstragende Grafiken wie Schaubilder oder Werbebanner. Andere Elemente, die nur der Navigation, als Layoutelemente oder zur Verzierung dienen, sind für die Beobachtung des Benutzerverhaltens weniger nützlich und können als Rauschen bezeichnet werden (siehe Kapitel 1.3.2).

Charakteristisch für den Datenaustausch zwischen Webbrowser und Webserver ist, dass sich um ein HTML-Dokument mit textuellen Inhalten eine Reihe von Grafiken gruppieren. Genauso sind Dateien mit Cascading Style Sheets, JavaScripts oder anderen Formaten wie Flash anzutreffen. Informationen werden zudem in PDF- und Office-Dokumenten angeboten. Die Masse der Angebote basiert jedoch auf HTML-Dokumenten, die Verweise auf andere HTML-Dokumente sowie auf Grafiken und andere Formate haben.

Jeder Zugriff eines Webbrowsers auf einen Webserver mit dem Ziel, ein Informationsangebot wiederzugeben, ist daher in der Regel in verschiedene Dateien aufgeteilt, die nacheinander vom Webserver ausgeliefert werden. In der Protokollierung der Zugriffe wird häufig jeder Dateizugriff festgehalten, relevant für die Analyse des Benutzerverhaltens sind jedoch primär die protokollierten Zugriffe auf informationstragende Dateien.

Einschub: Flash-Applikation, Java-Applets und dynamische Seiten

> Probleme in der Protokollierung machen Flash-Applikationen und Java-Applets, die einmal ausgeliefert eigenständig auf dem Clientrechner ablaufen und in den seltensten Fällen weiter mit dem Webbrowser kommunizieren. In Folge werden die Benutzeraktionen nicht mehr protokolliert und dem Websitebetreiber fehlen die entsprechenden Daten. Der Einsatz solcher Anwendungen ist daher in Abhängigkeit von den Einsatzzwecken der Website zu bedenken.
>
> Ein weiteres Problem für die Protokollierung und Auswertung des Benutzerverhaltens liegt bei dynamisch generierten Seiten vor, bei denen Inhalte häufig über numerische Schlüsselwerte statt über sprechende Namen wie bei statischen HTML-Seiten adressiert werden. Hier ist je nach Anwendungsfall eine Abbildung von Schlüsseln auf sprechende Namen nötig (beispielsweise index.jsp?page=5743 statt information_mit_namen.jsp). Auch können sich hinter

> einem protokollierten Zugriff sehr viele Informationen verbergen, die dann nicht eindeutig identifiziert werden können.

Eine eigenständige Webserversoftware oder ein anderes Softwaresystem, das HTTP-Dienste anbietet, wie beispielsweise ein Applikationsserver, stellt die zentrale Schnittstelle zum Webbrowser auf dem Arbeitsplatzrechner dar. Entsprechend wird hier auch die Protokollierung der Zugriffe vorgenommen. Es gibt aber weitere Stellen im insbesondere bei großen Websites meist komplexen Zusammenspiel von verschiedenen Softwarekomponenten, an denen das Benutzerverhalten festgehalten wird. Protokollierung kann auch stattfinden in

- Applikationsservern
- Datenbanken (hier weniger nützlich, da auf zu niedriger Ebene)
- Client-Anwendungen (auch weniger nützlich für uns und proprietär)
- Protokollkomponenten in Content-Management-Systemen (CMS)

Applikationsserver oder andere Middleware, die als Laufzeitumgebungen für Webanwendungen und Skripte arbeiten, die ihrerseits dynamisch und meist aus Datenbanken gespeist Informationseinheiten zusammensetzen und an den Webbrowser ausliefern, können selbst jeden Zugriff protokollieren. Entweder zusätzlich zum Webserver oder auch eigenständig, wenn die Protokollkomponente des Webservers abgeschaltet oder nicht vorhanden ist. Ähnlich auch bei Content-Management-Systemen, die eine eigene Protokollierung integriert haben. Vorteil in beiden Fällen kann sein, dass die protokollierten Daten präziser sind, wenn mehr Informationen gespeichert werden, die nur in der Webanwendung, nicht aber im Webserver existieren (wie beispielsweise Informationen über den zugreifenden Benutzer). Zudem kann weniger Rauschen vorhanden sein, da nur wirklich relevante Zugriffe kodiert werden, wenn die Webanwendung die Bedeutung der Zugriffe kennt und zum Beispiel unwichtige Layoutelemente direkt herausfiltert.

In der Praxis ist jedoch vorrangig der Einsatz der Protokollkomponenten von Webserversoftware und ihren Logdateien anzutreffen. Ein über Jahre laufendes Monitoring der Firma netcraft (http://www.netcraft.com, siehe [NCL03]) gibt Auskunft darüber, welche Webserverprogramme am häufigsten eingesetzt werden. Zwar beziehen sich die Angaben nicht auf alle im Internet verfügbaren Webserver, aber dennoch dürfte die Datengrundlage repräsentativ sein, da gegenwärtig rund 40 Millionen Websites einbezogen werden (Stand April 2003). An führender Stelle steht dabei die Open-Source-Software Apache (rund 63% Verwendung) mit gewissen Abstand gefolgt von Internet Information Services oder Server – kurz IIS – von Microsoft (rund 28%). Weitere Programme aus den Anfangszeiten des World Wide Web sind zwar nach wie vor im Einsatz und werden auch teilweise weiterentwickelt, Ihre Bedeutung schwindet aber beständig und ihre Nutzung beschränkt sich teilweise auf Nischenanwendungen.

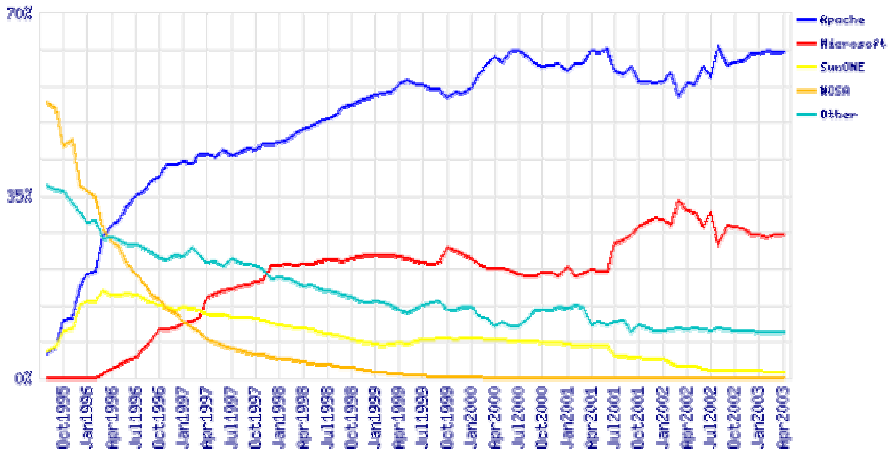

Abbildung 5 - Marktanteile von Netcraft beobachteter Server (08/1995 bis 04/2003)[6]

Aufgrund der Marktanteile wird sich im weiteren Verlauf in Bezug auf die Protokollierung von Zugriffen auf Apache und IIS beschränkt. Zum Vorteil des Anwenders arbeiten beiden Programme in dieser Hinsicht auch recht ähnlich. Dokumentation zur Konfiguration der Protokollierung von Apache findet sich unter [Apa2].

Protokollformate

Webserver wie Apache oder Microsoft IIS können alle Zugriffe auf Websites in Logdateien protokollieren. Dabei handelt es sich um Textdateien, in denen jede Zeile einen Datensatz enthält, der einen Zugriff auf die Website festhält. Zum Einsatz kommen dabei primär das *Common Log Format* (CLF, siehe [BLI97], [Lut95]) und eine erweiterte Version dieses Formats – als *Extended Log Format* bezeichnet (siehe [HB96]). Zudem gibt es proprietäre Lösungen der Webserversoftware.

Das *Common Log Format* wurde ursprünglich von NCSA (National Center for Supercomputing Applications) entwickelt, wird aber mittlerweile von allen gängigen Webservern unterstützt. Tatsächlich bieten aktuelle Webserver die Möglichkeit, das Format der Logdateien weitgehend frei zu definieren und das Protokollieren auch an externe Prozesse zu delegieren, beispielsweise um in einer Datenbank statt in einer einfachen Textdatei zu protokollieren oder um Vor- und Nachbereitungen auf den Logdaten durchzuführen. Je nach Anwendungszweck kann daher eine individuelle Konfiguration sinnvoll sein. Im Allgemeinen wird man jedoch das *Common Log Format* oder das *Extended Log Format* einsetzen, um auch zu zahlreichen Auswertungsprogrammen kompatibel zu bleiben, die sich auf diese gemeinsamen Formate stützen.

Festgehalten wird jeder Zugriff mit einer Reihe von Informationen wie angefordertes Objekt, Zeitpunkt der Anforderung oder Anzahl der gelieferten Zeichen. Ein kurzer Ausschnitt aus einer Logdatei:

```
141.2.114.181 - - [31/Mar/2003:00:01:25 +0200] "GET /b_wap.gif HTTP/1.1" 304 -
141.2.114.181 - - [31/Mar/2003:00:01:25 +0200] "GET /b4.gif HTTP/1.1" 304 -
217.233.253.250 - - [31/Mar/2003:00:01:32 +0200] "GET /start.php HTTP/1.1" 200 11517
217.233.253.250 - - [31/Mar/2003:00:01:32 +0200] "GET /banner/b_ab.gif HTTP/1.1" 200 5472
217.233.253.250 - - [31/Mar/2003:00:01:36 +0200] "GET /f/index.php?co=hp HTTP/1.1" 200 960
host4.przx.net - - [31/Mar/2003:00:01:37 +0200] "GET /portal/index.php HTTP/1.1" 200 6811
217.233.253.250 - - [31/Mar/2003:00:01:37 +0200] "GET /banner/b_ma.gif HTTP/1.1" 200 6857
217.233.237.118 - - [31/Mar/2003:00:01:44 +0200] "GET /news/news.php HTTP/1.1" 200 21414
217.233.253.250 - john [31/Mar/2003:00:01:45 +0200] "GET /bu/ HTTP/1.1" 200 1094
217.233.253.250 - john [31/Mar/2003:00:01:45 +0200] "GET /bu/bau.php HTTP/1.1" 200 17284
```

[6] Diagrammgrafik von der Website von Netcraft, siehe [NCL03]

Abbildung 6 - Exemplarischer Auszug aus einer Logdatei, Common Log Format (CLF)

Eine Zeile stellt einen Datensatz dar und enthält sieben Felder, die durch Leerzeichen getrennt sind. Dabei handelt es sich um:

Feldname	Funktion
remotehost	IP-Adresse oder falls verfügbar DNS-Name des anfragenden Rechners
rfc931	Eigentümer der Verbindung gemäß RFC931 (siehe [Lut95]), heute unverwendet
authuser	Name eines authentifizierten Benutzers
[date]	Zeitpunkt des Zugriffes bestehend aus Datum und Uhrzeit
"request"	Befehl an den Webserver inkl. angeforderter Ressource
status	Rückgabecode des Webservers abhängig vom Ausgang der Anforderung
bytes	im Erfolgsfall Größe der zurückgelieferten Datenmenge in Bytes

Tabelle 2 - Felder im Common Log Format

Mit dem Feld *remotehost* kann prinzipiell auf den anfragenden Rechner geschlossen werden. Leider lässt sich damit in Zeiten von Webproxyservern (mehrere Benutzer verbergen sich hinter einer gemeinsamen Adresse) oder Internet-Providern mit einem beschränkten Adressenvorrat nicht zwischen Benutzern unterscheiden. Das Feld kann daher nur als Indiz zur Unterscheidung zwischen Zugriffen von unterschiedlichen Rechnern verwendet werden, nicht aber als eindeutiger Beweis.

Gänzlich unbrauchbar ist das Feld *rfc931*, das für den Benutzernamen des Besitzers der Netzwerkverbindung vorgesehen war, mit dem man ursprünglich automatische Authentifizierungen vornehmen wollte (siehe [StJ85]). In der Praxis hat sich diese Technik nicht durchgesetzt und das Feld ist daher immer leer, was mit einem Minuszeichen "-" gekennzeichnet wird. Das gilt auch für alle anderen leeren Felder.

Interessanter dagegen ist das Feld *authuser*. Wurde auf Seiten des Webservers ein Zugriffsschutz für eine Ressource definiert via HTTP-Benutzer-Authentifizierung, dann muss sich der Anwender im Webbrowser mit Benutzername und Passwort authentifizieren (siehe Kapitel 1.3.4 Sitzungen). Bei erfolgreicher Authentifizierung gewährt der Webserver Zugang zu der gewünschten Ressource. Technisch sendet der Webbrowser bei jedem Zugriff die einmal abgefragte Benutzername/Passwort-Kombination an den Webserver, der seinerseits den Benutzernamen im Feld *authuser* in der Logdatei speichert. Anhand dieses Feldes kann also an sich eine sehr gute Unterscheidung zwischen Benutzern vorgenommen werden. Leider ist die Technik aus Sicherheitsgründen (die Passwörter sind anhand der schwachen Verschlüsselung bei unsicheren Verbindungen leicht zu ermitteln) nur bedingt einsatzfähig.

Bleiben noch die Felder für Datum des Zugriffes, für die Anforderung der Ressource, für den Status und für die Größe. Das Datumsfeld *date* besteht aus dem Tagesdatum und der Uhrzeit sowie einem Offset in Stunden und Minuten zur Greenwich Mean Time (GMT). Das Feld steht immer in eckigen Klammern, beispielsweise [31/Mar/2003:00:01:45 +0200].

Im Anschluss folgt das Feld für den eigentlichen Request, also das, was der Webbrowser vom Webserver anfordert. Das Feld ist immer in Anführungszeichen eingeschlossen und enthält exakt den Befehl gemäß HTTP-Definition, der zwischen Client und Server ausgetauscht wurde (siehe [IETF99]). Häufig sind hier als Befehle "GET" und etwas weniger häufig "HEAD" und "POST"

anzutreffen. Andere Befehle von HTTP sind eher exotisch und für die Zwecke der Logdateiauswertung und späterer Gewinnung impliziter Benutzerprofildaten weniger von Interesse.

Den Befehlen folgt im *request*-Feld eine Ressource, wobei es sich um einen Pfad zu einer Datei im virtuellen Verzeichnisraum des Webservers handelt. Dabei kann es sich sowohl um physikalisch vorhandene Dateien als auch um generierte Ströme von Daten handeln, die zum Zeitpunkt des Aufrufs erzeugt werden. Wichtig für eine spätere Logdateianalyse ist, dass die Dateiinhalte (seien es HTML-Dokumente, einfache Texte oder Bilder) anhand des Dateinamens und speziell der Dateiendung (wie .html oder .jpg) für eine Typbestimmung erkannt werden können. Auch der vordere Teil des Dateinamens und im Falle von dynamischen Inhalten angehängte Parameter (?parametername1=wert1&parametername2=wert2&...) spielen eine wichtige Rolle zur Identifizierung der angeforderten Inhalte. Beispielsweise können so auch Suchanfragen einer Website-internen Suchmaschine protokolliert werden, wenn der Suchausdruck in Parametern festgehalten wird (beispielsweise ?suchstring=suchausdruck). Angehängt an die angeforderte Ressource folgt schließlich die letzte Komponente des *request*-Felds, die Version des Protokolls, wie HTTP/1.1. Ein klassisches Beispiel für den vollständigen Inhalt dieses Feldes ist

```
"GET /bu/index.html HTTP/1.1"
```

Über den Ausgang der Ressourcenanfrage gibt das Feld *status* Auskunft. Hier ist in der HTTP-Spezifikation eine Reihe von definierten Statuscodes vorgesehen, beispielsweise für erfolgreichen Zugriff *200* oder für Nicht-Finden der gewünschten Ressource das allseits bekannte *404*. Eine genaue Auflistung und Beschreibung der einzelnen Statuscodes findet sich unter [IETF99]. Hier haben besonders die Datensätze Nutzen, die einen erfolgreichen Transfer einer Ressource signalisieren, weil man dann davon ausgehen kann, dass der Benutzer den Inhalt der Ressource erhalten und mit gewisser Wahrscheinlichkeit auch gesehen hat. Entsprechend kann in seinem Profil für jede erfolgreiche Sichtung ein Eintrag erfolgen und dieser als implizite Wertung der Ressource aufgefasst werden.

Zum Ende der Formatbeschreibung bleibt noch das Feld *bytes*, das im Fall einer stattgefundenen Übertragung schlicht die Anzahl der zurückgelieferten Daten gemessen in Bytes angibt. Hier hat es geringe Bedeutung, könnte aber zur Messung des Datentransfers pro Nutzer oder der gesamten Website herangezogen werden.

Extended Log Format

Eine Erweiterung zum *Common Log Format* stellt das so genannte *Extended Log Format*, manchmal auch *Extended Common Log Format* genannt, dar. Darin sind zwei weitere Felder am Ende der Feldliste enthalten, in denen für jeden Datensatz weitere Informationen stehen können.

Feldname	Funktion
"referer"	URL der verweisenden Ressource (falls vorhanden)
"user_agent"	Identifikation der verwendeten Clientsoftware (Browser, Suchmaschine, ...)

Tabelle 3 - Zusätzliche Felder im Extended Log Format

Im Feld *referer* wird die vom Benutzer chronologisch vorher besuchte Ressource angegeben. Dabei handelt es sich um den URL der besuchten Ressource, der auch außerhalb der vom Webserver bereitgestellten Website liegen kann. Liegt eine solche Vorgängerressource nicht vor, beispielsweise weil der Browser die Information nicht sendet oder der Benutzer schlicht mit einem leeren Webbrowser gestartet hat, enthält das Feld ein Minuszeichen "-". In jedem Fall wird der Inhalt mit doppelten Anführungszeichen umschlossen.

Gleiches gilt auch für das Feld *user_agent*, das die vom Webbrowser gesendete Kennung der Software enthält. Hiermit können der Typ und Hersteller des zugreifenden Browsers und in gewissen Grenzen auch seine technischen Fähigkeiten ermittelt werden. Beispielsweise steht ein Feldinhalt von "Mozilla/4.0 (compatible; MSIE 6.0; Windows 98)" für Microsoft Internet Explorer 6 unter Microsoft Windows 98. Damit lassen bei der Auswertung der Logdateien Aussagen über die technische Ausrüstung der Besucher machen und mögliche Optimierungen vornehmen. Indizierer von Suchmaschinen haben hier meist eigene Kennungen, wie "Googlebot/2.1 (+http://www.googlebot.com/bot.html)". Allerdings muss die Information nicht immer stimmen, da manche Browser oder Suchsysteme sich unter der Kennung anderer Software tarnen.

Interessanter für die hier angestrebten Zwecke ist das *referer*-Feld, da man damit Ketten von zusammenbesuchten Ressourcen ermitteln kann. Primär von Interesse sind Informationsinhalte. In der Fachterminologie der Logdateiauswertung werden diese Ketten Clickstreams oder Clickpaths genannt, da der Websitebenutzer für (fast) jeden neuen Datensatz in der Logdatei einen Mausklick vorgenommen hat (bzw. entsprechend die Tastatur bemüht, aber das ist für die einprägsame Begriffsbildung weniger hilfreich, daher die Mausaktion).

In der Auswertung können dann Rückschlüsse auf das Benutzerverhalten und die Interessenslage des einzelnen Benutzers gezogen werden, indem man zusammen- oder hintereinander besuchte Inhalte identifiziert. War ein Inhalt A für den Benutzer interessant und hat er danach zu einem Inhalt B verzweigt, könnte auch ein ähnlicher Inhalt C für ihn nützlich sein. Oder andere Benutzer haben ebenfalls zuerst A, danach B und schließlich noch einen weiteren Inhalt D besucht. Dann könnte man dem Besucher, der nur A und B besucht hat, auch D anbieten. Die gewonnenen Regeln können so zur Personalisierung dienen (siehe Kapitel 2.4.2 Regelbasierte Personalisierung und [Kloss01]).

Wie die Auswertung von Logfiles schließlich vonstatten gehen kann, wird im Kapitel 2.3.5 Logdateianalyse zur Ableitung impliziter, dynamischer Daten näher beschrieben.

Speichern von Cookies in der Logdatei

Sind Cookies im Webbrowser zugelassen und vorhanden (siehe Kapitel 1.3.3 Cookies), dann werden sie bei jedem Zugriff an den jeweiligen Webserver gesendet, der die Cookies ursprünglich ausgesetzt hat. Den Webserver kann man so konfigurieren, dass er zu jedem Datensatz in der Logdatei auch die Werte der vom Client gelieferten Cookies speichert. Damit lässt sich in der Auswertung des Logfiles beispielsweise eine Zuordnung der besuchten Informationen zu einzelnen Benutzern vornehmen und man kommt dem Ziel, Verhaltensdaten für Benutzerprofile zu erfassen, ein gutes Stück näher.

1.4 Schutz der Privatsphäre

Im Juli 1999 titelte der Spiegel mit „Digitale Vollkontrolle – Das Ende des Privaten“ [BS99] und drückte damit aus, dass die Daten der Bürger in vielfältigen Datenbanken gespeichert sind und das häufig in Unkenntnis der Betroffenen. Die Daten werden vielfältig verwendet, sei es zur Verbrechensbekämpfung, zur Marktforschung oder gar zur Wirtschaftsspionage und dazu bedurfte es nicht erst des Internets. Denn die Daten werden auf unterschiedliche Weise erhoben, von staatlichen Institutionen wie von privaten Unternehmen. Das Internet macht die automatische Erfassung von Daten jedoch ungleich einfacher, da die Kommunikation ohnehin digital erfolgt.

Ängste schürt es, wenn die verschiedenen Datenquellen zusammengeführt werden und der so genannte gläserne Bürger entsteht. Dann ist Diskriminierung bei gegenüber der Allgemeinheit ab-

weichenden Merkmalen sehr schnell möglich. Beispielsweise werden Kredite nur an bestimmte Klientel vergeben, Versicherungen verlangen zu hohe Prämien oder polizeiliche Maßnahmen treffen Unschuldige – häufig auch wegen ungenauen und fehlerhaften Daten. Den einfachsten Fall, dass Adressen ungefragt weitergegeben werden, kennt jeder aus der täglichen Praxis durch die ungefragte Zusendung von Werbung, sowohl on- wie offline. Im Folgenden werden daher einige Aspekte und Grundlagen der Datenschutzproblematik im Internet vorgestellt, die aber konzentriert sind, da sie in ihrer vollen Breite eine eigene Arbeit erfordern würden.

1.4.1 Allgemeine Probleme

Im Internet und World Wide Web werden aus technischen Gründen bei jedem Zugriff und jeder Sitzung Daten zwischen Server und Arbeitsplatzrechner ausgetauscht. Die Daten werden zudem nicht direkt in einer eigenen Leitung verschickt, sondern gehen über sehr verschiedene Systeme. Man spricht von Routing. Hier können also interessierte Stellen zuhören und Daten abfangen[7]. Aber auch der Zielserver arbeitet mit den empfangenen Daten und was er dort genau macht, entzieht sich der Kenntnis des Anwenders.

Zunächst wäre das nicht sonderlich schlimm und mit wenigen Risiken behaftet. Denn wenn man sich die standardmäßig versendeten Daten betrachtet, handelt es sich neben den Nutzdaten wie HTML-Dateien und Grafiken um Metainformationen wie IP-Adresse des aufrufenden Rechners, Zeitpunkt des Zugriffes oder Informationen über die Browsersoftware. Diese Daten sind jedoch technischer Natur und anonym. Ein Bezug zu einer Person ist zunächst nicht herzustellen.

Unangenehm wird es aber, wenn diese Daten mit personenbezogenen Daten verknüpft werden. Wenn beispielsweise genau gesagt werden kann, dass ein namhaft bekannter Benutzer, der in einer bestimmten Stadt und Straße wohnt, diese oder jene Telefonnummer besitzt und auch einige weitere Informationen über ihn bereitstehen, bestimmte Seiten einer Website aufgerufen, dort eine zeitlang verweilt und möglicherweise sogar Käufe getätigt hat. Werden diesen Informationen nicht weiter verwendet, sondern nur für den eigentlichen Zweck der Bereitstellung des Informationsangebotes oder der Abwicklung im Online-Shop genutzt, dann ist diese Datensammlung weniger schlimm – auch wenn ein ungutes Gefühl beim Benutzer bleiben kann, da er nicht weiß, was tatsächlich mit seinen Daten geschieht. Und hierin liegen dann die Risiken begründet!

Werden die Daten vom Websitebetreiber noch für andere Zwecke verwendet, beispielsweise für Marketingzwecke, zur Werbung oder zur Marktforschung, dann kann die Privatsphäre des Benutzers erheblich verletzt sein. Vor allem bei finanziellen oder gesundheitlichen Auskünften des Benutzers wird die Privatsphäre schnell gestört. Werden die Daten gar an Dritte weitergegeben, beispielsweise an Adressenhändler, was dann im schlimmsten Fall zu Spam führen kann, liegt ein klarer Missbrauch des Datenschutzes vor. Absicherung schafft hier, dass Websitebetreiber bei der Erhebung der personenbezogenen Daten genau erläutern, welche Daten sie erheben und was sie mit den Daten machen. Der Benutzer muss dann explizit die Möglichkeit haben, dieser Datenverwendung zuzustimmen. Ein Konzept, solche Datenschutzrichtlinien anzubieten, stellt das Platform for Privacy Preferences Project (P3P) des W3C-Konsortiums vor, das im Folgekapitel vorgestellt wird.

Aus gesetzlicher Sicht gab es lange Zeit nur vage Aussagen über den Datenschutz im Internet. Ursache war, dass das Medium noch relativ jung war und sich zunächst Erfahrungswerte heraus-

[7] Ein Grund, warum bei der Übertragung sensitiver Daten Point-To-Point-Verschlüsselung, beispielsweise mit SSL, im Webserver und Webbrowser genutzt werden sollte. Dann wird das „Mithören“ nahezu ausgeschlossen.

bilden mussten. Mittlerweile sieht es der Gesetzgeber aber so, dass die Bürger ein Recht auf informelle Selbstbestimmung haben und insbesondere selbst entscheiden können sollen, was mit ihren Daten in elektronischen Systemen passiert. Dazu zählen neben dem Internet auch Telefonnetze, speziell auch bei mobilen Telefonen. Diese informelle Selbstbestimmung wird als Teil der Menschenrechte betrachtet, genauso wie das Recht der freien Meinungsäußerung. Das Europäische Parlament und der Rat der Europäischen Union haben dazu eine Richtlinie herausgegeben. Sie gibt vor, wie mit personenbezogenen Daten in der elektronischen Kommunikation umgegangen werden soll und darf (siehe [EPREU02]). Die Richtlinie ist seit 1. November 2003 in Kraft und die EU-Mitgliedsstaaten müssen sie seitdem beachten. In ihr wird detailliert geregelt, wie mit personenbezogenen Daten sowohl was die Speicherung, Übertragung als auch die Verarbeitung angeht, umzugehen ist. Exemplarisch sei herausgegriffen, dass Cookies ausschließlich zu rechtmäßigen Zwecken eingesetzt werden dürfen. Ein solcher Zweck wäre die Überwindung der Zustandslosigkeit von HTTP und damit eine Verbesserung der technischen Möglichkeiten. Zudem wird verlangt, dass der Benutzer über das Setzen von Cookies informiert und ihm dargelegt wird, welche Daten gespeichert werden. Vor allem muss der Benutzer die Möglichkeit haben, das Setzen des Cookies abzulehnen. Sinnvollerweise sollte das Informationsangebot daher auch ohne die Verwendung von Cookies funktionieren (siehe [EPREU02], Absatz 25).

1.4.2 Platform for Privacy Preferences Project (P3P)

Ziel des Platform for Privacy Preferences Projektes (P3P)[8] vom World Wide Web Consortium (W3C) ist die Bereitstellung von technischen Möglichkeiten, damit Websites strukturierte und standardisierte Informationen über die Behandlung der Privatsphäre ihrer Benutzer und den Datenschutz machen können. Webbrowser und damit Website-Besucher sollen in der Lage sein, diese Informationen auf einfache Weise abrufen zu können, um zu erfahren, wie mit bereitgestellten persönlichen Daten umgegangen wird. Ein maschinenlesbares Format wird also benötigt. Ziel ist ferner, dass die Informationen von Software verarbeitet und geprüft werden können und so die Prüfung an Software delegiert werden kann. Beispielsweise durch Festlegung des Grades an Privatheit und Abschottung, den man als Benutzer wahren möchte.

Kurzum, P3P will dem Internetbenutzer Mittel an die Hand geben, um die Verwendung persönlicher Informationen durch Websites besser zu kontrollieren.

Zu beachten ist, dass das Projekt zwar Vorgaben definiert, wie die Datenschutz-Angaben auszusehen haben, aber keinerlei Methoden für die tatsächliche Sicherstellung ihrer Einhaltung vorgibt. Websitebetreiber, die also P3P-Zusicherungen bereitstellen, müssen sich zwar an die Vorgaben der Spezifikation halten, können ihre Zusicherungen aber dennoch missbräuchlich hintergehen. Ferner ist die Zusicherung auf die Website beschränkt und schließt nicht den Transport über Internet-Serviceprovider und Datenleitungen ein. Auch schreibt das Projekt – wie bei vielen W3C-Empfehlungen – nicht vor, wie Software aufgebaut sein sollte, die die Mechanismen des P3P verwendet. Wohl aber werden an verschiedenen Stellen Hinweise für eine mögliche Umsetzung gegeben.

P3P-Richtlinien

Im Folgenden wird der mangels einer aktuellen offiziellen Übersetzung vom W3C im englischen Original verwendete Begriff Policy mit Richtlinie übersetzt. Das Projekt hat eine Spezifikation (siehe [W3C02]) herausgegeben, die definiert, welche Syntax und Semantik die Richtlinien oder

[8] Übersichtsseite des Platform for Privacy Preferences Project (P3P) – http://www.w3.org/P3P

genauer die Richtliniendokumente haben sollen. Die Dokumente verwenden ein XML-Format mit Namensräumen für P3P-Elemente und enthalten Beschreibungen der von der Website benutzten und gesammelten persönlichen Daten[9]. Auch definiert die Spezifikation ein Verfahren, wie die Richtlinien-Dokumente mit Website-Ressourcen wie HTML-Dateien, Grafiken und Cookies verknüpft werden. Die Verknüpfung erfolgt mittels ebenfalls in XML gehaltenen Richtlinien-Referenzdateien, wobei insbesondere die Zusicherungen nur für diejenigen Ressourcen einer Website gelten, die mit einer oder mehrerer Richtlinien verknüpft sind. Der Website-Betreiber muss hier entsprechend sorgsam vorgehen.

Bei den verwendeten Daten kann es sich beispielsweise um die vom Webbrowser standardmäßig gesendeten Angaben wie Name der Browsersoftware und anderer Metadaten handeln, aber auch um Daten wie IP-Adresse des zugreifenden Rechners, Zeitpunkt des Zugriffs und anderer Felder, die in den Webserver-Logdateien protokolliert werden. Eine nächste Gruppe sind Daten, die durch Cookies ausgetauscht werden und zu einer Speicherung auf dem Arbeitsplatzrechner des Benutzers führen. Schließlich gibt es noch Daten, die Interaktion durch den Benutzer erfordern, wenn er ein Formular der Website bearbeitet. Hier können der Vor- und Nachname, die E-Mailadresse und die Telefonnummer betroffen sein. Mit einer P3P-Richtlinie kann eine Website zusichern, wie sie die verschiedenen Daten einsetzt. Ein Benutzer kann – assistiert durch den Webbrowser und in Abhängigkeit von der Richtlinie – entscheiden, wie viele Daten er der Website geben möchte.

Umgang von Webbrowser und Webserver mit P3P-Richtlinien

Auf Browserseite unterstützten Microsoft Internet Explorer 6 und Netscape Navigator 7 bzw. Mozilla die P3P-Richtlinien und bieten eine aufbereitete Anzeige an. Bei beiden Browsern ist die Cookie-Verwaltung mit der Richtlinie der Website verknüpfbar, soweit die Seite überhaupt über eine Richtlinie verfügt. Bei großen Websites ist das aber mittlerweile der Fall. Der Benutzer kann so einstellen, ob er das Speichern von Cookies von Seiten mit geeigneter Richtlinie gestatten will oder nicht.

Auf der Seite des Webservers wird die P3P-Richtlinien-Referenzdatei so abgelegt, dass sie von einem Benutzerclient einfach geladen werden kann. Eine Möglichkeit dazu ist, die Datei unterhalb des Wurzelverzeichnisses als /w3c/p3p.xml abzulegen. Bei der Website von IBM ist das z. B. mit http://www.ibm.com/w3c/p3p.xml der Fall. Der Vorteil hieran ist, dass die Referenzdatei und in Folge die P3P-Richtlinie bereits vor dem Laden jeder anderen Ressource eingesehen und so eine Entscheidung über die Preisgabe von Daten getroffen werden kann. Alternativ kann mit einem Link im HTML-Code oder mit einem HTTP-Header auf die Datei verwiesen werden.

Inhalt einer P3P-Richtlinie

Eine P3P-Richtlinie enthält zunächst die Kontaktdaten des Betreibers, also eine Art Impressum. Ferner werden alle verwendeten Daten und ihre Typen aufgelistet sowie eine Beschreibung abgegeben, wie diese Daten von der Website verwendet werden. Die Richtlinie enthält auch Hinweise, wie Datenschutzkonflikte gelöst werden sollen, beispielsweise über eine Servicehotline und einen URL auf eine menschenlesbare Form der Richtlinie, was mehr oder weniger den Allgemeinen Geschäftsbedingungen entspricht, jedoch auf Datenschutzsicht eingeengt ist. Elementar für Richtlinien ist, dass sie tatsächlich die Daten und ihre Verwendung aus technischer Sicht beschreiben und nicht nur eine Zusicherung in Form einer Datenschutzerklärung darstellen. Dem entspricht auch, dass alle Angaben in positiver Form erfolgen, also was die Website macht und nicht etwa, was sie nicht macht. Selbstverständlich müssen die Angaben in der Richtlinie wahr sein und alle

[9] Alternativ gibt es ein RDF-Schema für P3P-Richtlinien – http://www.w3.org/TR/p3p-rdfschema

Daten umfassen, die verwendet werden. Wie schon erwähnt, gibt es aber keine Mechanismen, die das tatsächliche Verhalten kontrollieren können.

Besonders interessant ist der Anteil, der beschreibt, wie die gesammelten Daten verwendet werden. Hier definiert die P3P Spezifikation eine Reihe von typischen Verfahren, die fast alle Möglichkeiten der Verarbeitung mit den gesammelten Daten abdecken und Interoperabilität zwischen verschiedenen Programmen und Diensten durch ein festes Vokabular ermöglichen. Die dynamischen Daten, die vom Benutzer geliefert werden, sind in folgende Gruppen unterteilt:

Datengruppe	Enthaltene Daten
clickstream	umfasst alle Informationen, die typischerweise in der Logdatei des Webservers gespeichert werden wie IP-Adresse des zugreifenden Rechners bzw. der Hostname, der angeforderte URL oder der Zeitpunkt des Zugriffs (siehe 1.3.5 Protokollierung von Zugriffen)
http	die beiden zusätzlichen Felder referer und useragent (siehe 1.3.5 Extended Log Format)
clientevents	nimmt Daten auf, die im Webbrowser aufgezeichnet wurden. Beispielsweise Mausbewegungen, JavaScript-Befehle, aber auch nicht visuelle Aktionen
cookies	steht für den Einsatz von Cookies, die vom Webserver zum Webbrowser und zurück transportiert werden
searchtext	bietet die Seite ein Suchformular an, über das Suchanfragen auf an den Webserver übertragen werden, fällt das in diese Gruppe
interactionrecord	sind solche Daten, die aufgrund der Interaktion des Benutzers mit der Seite ausgetauscht werden. So beispielsweise das Ausfüllen von Formularen, das Bewerten von Inhalten oder der Kauf von Produkten
miscdata	für alle von den oben genannten Gruppen nicht erfassten Daten greift diese Gruppe. Hier ist eine weitere Beschreibung anzugeben

Tabelle 4 - Gruppen dynamischer Daten für Richtliniendateien in der P3P-Spezifikation

Ferner wird definiert, wer die Daten erhält. Beispielsweise nur der Betreiber der Website oder ob sie auch weitergegeben werden. Daneben wird auch die Aufbewahrungszeit beschrieben, die nur für den Zeitpunkt des Zugriffs für eine bestimmte Zeit oder auf unbestimmte Zeit andauern kann. Nützlich ist auch das Element, das den Zweck der Datensammlung beschreibt. Hier können ebenfalls wie bei den Daten verschiedene vordefinierte Varianten aus einer Menge gewählt werden.

Zweck	Aufgabenbeschreibung
current	die Daten werden nur zur Verarbeitung des aktuellen Zugriffs benötigt, beispielsweise für eine Berechnung oder eine Suchanfrage
admin	wenn die Daten für administrative Zwecke, beispielsweise zur Websiteoptimierung gebraucht werden oder für Auslastungsstatistiken
develop	wenn Daten zur Verbesserung der Website herangezogen werden
tailoring	Daten werden nur zur Anpassung der Website für die aktuelle Sitzung benötigt. Für zukünftige Sitzungen ist keine Anpassung erlaubt. Beispielsweise für Empfehlungen anhand des aktuellen Warenkorbs gedacht.

pseudo-analysis	wenn die Daten zur Analyse des Benutzerverhaltens verwendet werden, aber eine anonymisierte Verarbeitung ohne persönliche Daten erfolgt. Stattdessen wird der Benutzer zum Beispiel durch eine zufällige Nummer ersetzt.
pseudo-decision	ähnlich wie die pseudo-analysis, aber hier wird nicht nur anonyme Analyse betrieben, sondern es werden auch Empfehlungen berechnet. Für Personalisierung relevant, wobei keine Identifikationsdaten wie Name, Telefonnummer oder Wohnort mit der Entscheidung verbunden sind. Beispielsweise können bei diesem Zweck Inhalte empfohlen werden.
individual-analysis	genauso wie pseudo-analysis, aber hier erfolgt die Analyse jedoch nicht anonym.
individual-decision	genauso wie pseudo-decision, aber hier erfolgt die Analyse und Entscheidungsberechnung jedoch nicht anonym. Beispielsweise fallen Kaufempfehlungen in Online-Shops unter diesen Zweck
contact	wenn die Daten für den Kontakt zum Benutzer genutzt werden dürfen, z. B. zum Verkauf von Produkten oder zur Information über neue Inhalte auf der Website. Schließt telefonische Kontaktaufnahme nicht ein! Auch eine direkte, angeforderte Kontaktaufnahme des Kunden wird hiermit nicht beschrieben, das wäre beim Zweck current der Fall
historical	Daten müssen aus gesetzlichen oder rechtlichen Gründen für eine bestimmte Zeit aufbewahrt werden.
telemarketing	wie contact, aber hier darf die Kontaktaufnahme ausschließlich über Telefon erfolgen. Eine direkte, angeforderte Kontaktaufnahme des Kunden wird hiermit ebenfalls nicht beschrieben, das wäre der Zweck current.
other-purpose	wenn die Daten für andere Zwecke genutzt werden sollen. Dann muss eine textuelle, natürlichsprachige Beschreibung erfolgen.

Tabelle 5 - Datennutzungszwecke für Richtliniendateien in der P3P-Spezifikation

Zusammenfassend dient eine P3P-Richtlinie also zur Verbesserung der Einsicht in das Datensammelverhalten einer Website. Der Sitebetreiber verpflichtet sich mit dem Schalten einer solchen Richtlinie, die Daten nicht für andere als die zugesicherten Zwecke zu verwenden. Das erhöht das Vertrauen beim Anwender. Eine Garantie, dass sich der Betreiber aber letztlich daran hält, gibt es dafür nicht und wäre technisch auch nicht umzusetzen. Hierzu wären eher unabhängige Gutachter mit regelmäßigen, stichprobenartigen Prüfungen oder gesetzliche Regelungen geeignet.

Im Anhang findet sich eine P3P-Richtliniendatei für das im Rahmen der Arbeit entwickelte Informationssystem MiniPortal. Die Richtlinie umfasst alle nötigen Elemente und lief fehlerfrei durch den P3P Validator des W3C[10].

1.4.3 Open Profiling Standard (OPS)

Bei Websites und besonders auch bei Online-Shops, die eine Anmeldung des Benutzers vorsehen, werden immer personenbezogene Daten erhoben und im Webserver gespeichert. Ziel ist, dass der Benutzer seine Daten bei mehrfacher Benutzung nicht wiederholt eingeben muss und beim nächs-

[10] P3P Validator des W3C zu finden unter http://www.w3.org/P3P/validator.html

ten Besuch wieder erkannt wird. Personalisierung, wie sie in den späteren Kapiteln vorgestellt wird, basiert auf diesem Grundprinzip.

Zu den erfassten Daten gehören Vor- und Nachname, Adressdaten wie Wohnort, Postleitzahl und Straße sowie Kontaktinformationen wie E-Mail und Telefonnummer. Häufig werden noch weitere Angaben wie Geschlecht, Geburtsdatum und persönliche Präferenzen, beispielsweise für den Inhalt der Seite, abgefragt. Letztlich handelt es sich immer um die gleichen Daten, die aber bei sehr vielen Websites eingegeben werden müssen. Idee des Open Profiling Standard (OPS) ist, dass diese Datenübermittlung standardisiert und automatisiert erfolgen kann, so dass der Benutzer beim erstmaligen Besuch neuer Websites keine Angaben mehr machen muss – Browser und Server handeln selbst die Übergabe der Daten aus. Der Zeitgewinn für den Benutzer wäre erheblich und die Websitebetreiber hätten präzisere und qualitativ hochwertigere Daten zur Verfügung. Jedoch dürften die Daten aus Gründen des Datenschutzes nicht an jede Website übermittelt werden, so dass hier ein Mechanismus für vertrauenswürdige Sites vorhanden sein muss (siehe [Mena00], Kapitel 6).

Angedacht wurde das System bereits 1997 von den Firmen Netscape, Firefly und VeriSign und beim W3C als Diskussionsnotiz eingereicht (siehe [HMSCM97]). Allerdings gab es auch recht schnell kritische Stimmen in Bezug auf den Datenschutz, da das Verfahren hier nur vage Vorgaben machte. Auch viel der Begriff der „Monster-Cookies“, da der Open Profiling Standard mit einem den Cookies ähnlichen Konzept arbeiten sollte, die Profildaten aber weitaus umfangreicher waren als gewöhnliche Cookies. Gleichzeitig wurde der negative Beigeschmack der Unsicherheit, der Cookies seit jeher begleitet, auf das neue Konzept übertragen (siehe [HZ97]).

Schließlich wurden die Ideen des Open Profiling Standard beim W3C in den Rahmen der Datenschutzgruppe Platform for Privacy Preferences Project (P3P) eingeordnet – und sind hier eingeschlafen. Auch in der Praxis hat sich das Verfahren nicht durchgesetzt, vor allem aus Sicherheitsbedenken, wenngleich ein solches oder ähnliches Konzept bei der Vielzahl an Websites, die Registrierung erfordern, durchaus nützlich wäre und vor allem bei der steigenden Zahl von Angeboten immer wichtiger werden dürfte.

2 Personalisierung

2.1 Motivation

2.1.1 Das Konzept der Personalisierung

Der Begriff der „Personalisierung“ findet seinen Ursprung bereits in der Offline-Ära, als Internet und World Wide Web noch nicht existierten bzw. für die Allgemeinheit unbedeutend waren. Die Internetfachwelt hat ihn aber begierig aufgenommen und in die eigene Terminologie integriert. Die traditionelle Personalisierung wird durch den Wunsch der Konsumenten motiviert, Massenprodukte zu einem individuellen Gegenstand zu machen: Einem einzigartigen Produkt, das nur ein Konsument besitzt und das sich durch individuelle Merkmale von der Masse abhebt. Typische Vertreter dieser Richtung sind die Gravur des Namens auf der Rückseite einer Armbanduhr, der eigene Name auf dem Portemonnaie und natürlich das individuelle Namensschild an der Haustür.

Das Produkt an sich bleibt bei diesem Verfahren in seiner eigenen Form bestehen. Der Konsument kann lediglich Nuancen verändern. Die Personalisierung im Internet geht hier wesentlich weiter, denn nicht nur der Name des Benutzers und Konsumenten kann personalisiert werden, sondern je nach System auch das eigentliche Produkt – nämlich die Art und Weise, wie Informationen dargeboten werden und vor allem welche Informationen überhaupt präsentiert werden. Personalisierung im Internet bedeutet die Anpassung von Informationen an die spezifischen Bedürfnisse eines Benutzers. Seien es Interessen, Darstellungsformen, Layouts – ein personalisiertes Informationsangebot wird sich an die individuellen Schwerpunkte eines jeden Benutzers anpassen. Auf der anderen Seite stehen dazu in Abgrenzung unpersonalisierte Angebote, die für alle Besucher dieselben Informationen liefern.

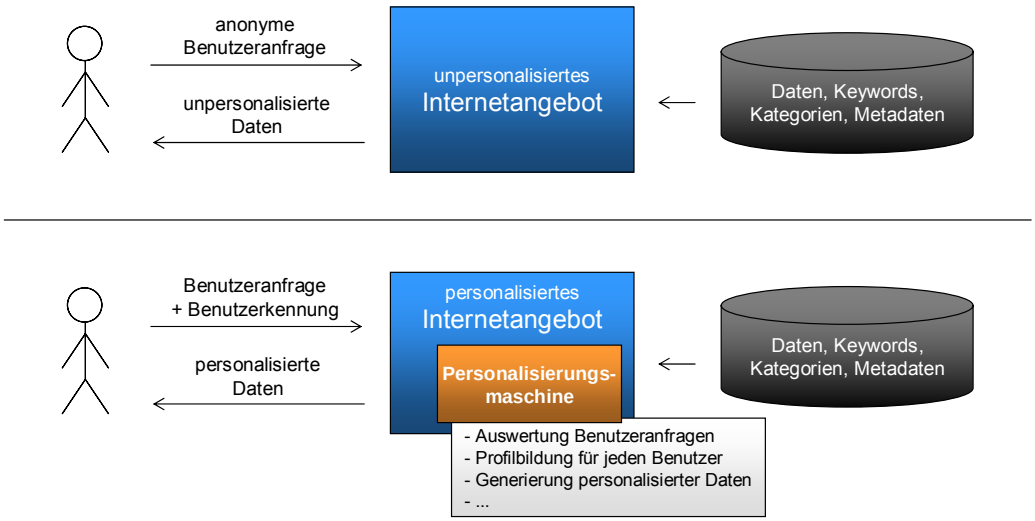

Abbildung 7 - Zugriff auf personalisierte und unpersonalisierte Angebote

Zwischen den beiden Konzepten personalisierter und unpersonalisierter Angebote stehen Systeme, die für eine festgelegte Zahl von Zielgruppen unterschiedliche Informationen bereitstellen. Die

Unterteilung der Benutzer erfolgt dabei jedoch nur in eine überschaubare Zahl von Gruppen. Beispielsweise in „Premium-Nutzer“ und „Standard-Nutzer“ oder in Anfänger, Fortgeschrittene und Experten. Eine echte Personalisierung in Form einer individuellen Ansprache eines einzelnen Benutzers wird aber nicht erreicht.

Einschub: Mass Customization

> Wurde Personalisierung in der traditionellen Offline-Welt eingangs auf die Gravur von Namen in diverse Produkte reduziert, sollte nicht unerwähnt bleiben, dass es in den letzten Jahren und Jahrzehnten auch in der produzierenden Industrie deutliche Weiterentwicklungen gegeben hat. Das Stichwort dazu heißt beispielsweise Mass Customization (siehe [Pine93])[11].
>
> Idee ist, dass Konsumenten weniger standardisierte Produkte kaufen wollen als solche, die an die eigenen Bedürfnisse und Lebenssituationen angepasst sind. Auch eine gesellschaftliche Abgrenzung zu Mitbürgern wird durch individuellere Produkte möglich. Primäre Beispiele hierfür sind Kleidung und Autos aber auch Personal Computer. Der Kunde möchte das zu erwerbende Produkt nicht nur aus einer Reihe von Produktvarianten auswählen, sondern individuell nach eigenen Schwerpunkten zusammenstellen. Zwar war und ist dies durch Handarbeit und spezielle Fertigung jederzeit auch in Zeiten der Massenproduktion möglich, die Umsetzung ist jedoch kostspielig und die Produkte tatsächlich eher exklusiver Natur.
>
> Mit Mass Customization jedoch können Produkte wie Kleidung und Automobile bis ins kleinste Detail vom Kunden konfiguriert und dann individuell produziert werden. Durch die Verwendung von Standardbaugruppen und modernen Produktionsverfahren bleiben die Produkte im Gegensatz zur wirklichen Individualfertigung dennoch erschwinglich.
>
> Nicht zu vergessen allerdings, dass die Produzenten ebenfalls eine Reihe von Vorteilen genießen und hierin sicherlich einer der Hauptantriebe für das Verfahren liegt. Beispielsweise wird nicht mehr "auf Halde" produziert, sondern nur nach Eingang von Kundenaufträgen. Langwierige und teure Lagerprozesse entfallen dadurch. Für die in dieser Arbeit behandelten Personalisierungsansprüche ist bei Mass Customization aber der Aspekt der Produktkonfiguration und Individualisierung besonders interessant.

Das Internet bietet sich als Partner solcher Mass Customization-Verfahren geradezu an, da hier vielfältige Personalisierungstechniken bereitstehen und dem Benutzer mittlerweile auch vertraut sind (siehe z. B. [Aar03]). Computerfirmen wie Dell[12] oder Apple[13] bieten in Ihren Online-Shops neben fertig zusammengestellten Produkten auch die Konfiguration individueller Pakete an. Der Kunde kann Computer nach seinen Vorgaben aufbauen und sie dadurch in Ausstattung und Preis erheblich variieren. Sie werden dadurch zwar „persönlicher“, aber nur in der Auswahl der Baugruppen, nicht im Produkt selbst – es handelt ich schließlich immer noch um Computer. Man spricht daher eher vom „Customizing“ der Produkte. Die Personalisierung im Internet geht jedoch noch deutlich weiter und wird insbesondere in Online-Shops rege eingesetzt.

Online-Shops können personalisiert werden, in dem jeder Benutzer beispielsweise eine individuelle Ansprache erfährt, bei jedem Besuch wieder erkannt wird und Tipps zu seinen bislang gekauften Produkten erhält. Auch die Präsentation neuer Produkte, die seinen Kaufinteressen entsprechen, ist ein wichtiges Element der Personalisierung. Wie diese Benutzerinteressen seitens des Anbieters ermittelt werden können, wird in den folgenden Kapiteln näher erläutert und stellt mit die größte

[11] Unter http://www.mass-customization.de/ findet sich eine umfangreiche Informationsseite der Technischen Universität München (TUM) zum Thema Mass Customization und Personalisierung
[12] siehe http://www.dell.com
[13] siehe http://store.apple.com/1-800-MY-APPLE/WebObjects/AppleStore/

Problematik der Internetpersonalisierung dar. Beachten sollte man jedoch, dass Kunde und Verkäufer unterschiedliche Ambitionen in Bezug auf die Personalisierung haben und es hier zu Konflikten kommen kann, wenn eine Seite die andere zu übervorteilen versucht. Exemplarisch seien hier zu neugierige Anbieter genannt, die zu viele Informationen von den Benutzern abfragen und diese vielleicht auch an andere Anbieter weiterveräußern. Ein Beispiel hierzu ist der Vorbehalt des Online-Buchhändlers Amazon[14], Kundendaten unter gewissen Umständen weiterzuveräußern, der immer wieder auf Kritik bei verschiedenen Interessensgruppen stößt (siehe [Heise00]). Diese und andere Problematiken im Interessenkonflikt zwischen Anbieter und Benutzer werden im Verlauf der Arbeit an verschiedenen Stellen noch näher erläutert.

Zunächst sollen aber noch weitere Einsatzgebiete vorgestellt werden, denn Personalisierung beschränkt sich nicht auf Online-Shops und E-Commerce. Internetangebote, die eine reichhaltige Zahl von Informationen anbieten, versuchen ebenfalls, Benutzern einen möglichst einfachen und effizienten Zugang zu den angebotenen Daten zu bieten. Je größer die verfügbare Menge an Informationen, desto schwieriger wird es für den Benutzer, an die von ihm tatsächlich gewünschten Inhalte zu gelangen. Kategorisierung und Suchmöglichkeiten helfen, die Informationsflut einzudämmen. Personalisierung erweitert und ergänzt diese Techniken, indem aktive Benutzer eines Internetinformationssystems ihre Interessen spezifizieren und dauerhaft ablegen können. Im Gegenzug erhalten sie eine maßgeschneiderte Sicht auf das Angebot und das „Rauschen“ uninteressanter Inhalte wird verringert. Die Suchzeiten können dadurch sinken und der Erfolg bei der Informationsbeschaffung steigen.

Eine ähnliche Problemstellung liegt in der Wissensverwaltung von Unternehmen. Mitarbeiter benötigen für Ihre tägliche Arbeit stets neue Informationen. Seien es leitende Angestellte, die Kennzahlen über die Unternehmensentwicklung abrufen oder Sachbearbeiter, die Informationen für die Bewältigung einer Aufgabe benötigen. Mittlerweile erfolgt der Zugang auf solche Informationen – also auf das Unternehmenswissen – häufig mittels eines Intranets. Nicht selten sind aber auch eigenständige, autonome Systeme im Einsatz, die wichtige Daten liefern. Mit Enterprise Information Portals (EIP) – oder deutsch Unternehmensportalen – wird versucht, die diversen Informationsquellen zu bündeln und unter einer Oberfläche bereitzustellen. Der Zugriff ist nicht zwingend nur auf Mitarbeiter beschränkt, auch Kunden und Zulieferer können über Zugangsmöglichkeiten verfügen. Das Portalsystem muss also eine Personalisierung nach Rollen zulassen und den Zugriff auf einzelne Informationseinheiten freischalten können.

Personalisierung kann hier aber noch deutlich weiter gehen, wenn jeder Portalnutzer die Arbeitsoberfläche des Systems und die vorhandenen Inhalte nach seinen Interessens- und Arbeitsschwerpunkten konfigurieren kann. Insbesondere bei großen Systemen mit vielen Daten kann die Informationsbeschaffung durch Personalisierung erleichtert werden, indem unwichtige Daten ausgeblendet werden. Zudem wird die parallele Erzeugung von gleichen Inhalten durch verschiedene Mitarbeiter reduziert, wenn sie bereits im Vorfeld Zugriff auf die benötigten Informationen haben, die von anderen Mitarbeitern schon zu einem früheren Zeitpunkt bereitgestellt wurden.

Wie die verschiedenen einführenden Beispiele gezeigt haben, bietet Personalisierung in Internetsystemen Optimierungsmöglichkeiten für den Zugriff auf Informationen unterschiedlichster Art. Genauso unterschiedlich sind auch die Zielsetzungen und Ansprüche, die Betreiber und Anwender in Bezug auf personalisierte Angebote haben. Entsprechend haben sich in den letzten Jahren auch verschiedene Konzepte zur Realisierung von personalisierten Systemen entwickelt und etabliert.

[14] siehe http://www.amazon.com oder http://www.amazon.de

2.1.2 Beweggründe für die Personalisierung von Internetangeboten

Man kann davon ausgehen, dass Betreiber von Internetangeboten beliebiger Art Interesse am Erfolg Ihrer Dienste haben. Die Bezifferung des Erfolges ist aber je nach Art des Angebots unterschiedlich. Bei Online-Shops steht die Steigerung von Umsatz und Gewinn bei hoher Kundenzufriedenheit im Vordergrund. Auch der Aufbau einer Marke kann im Mittelpunkt liegen. Werden unternehmensinterne Informationssysteme betrieben, sollen häufig die Produktivität der Mitarbeiter gesteigert und Kosten verringert werden. Wieder anders gelagerte Erfolgsziele haben Betreiber von Informationsportalen, die mit der Ware Information handeln. Hier spielt auch häufig die Schaltung von Werbung eine Rolle, um die Informationen selbst kostenlos oder zumindest kostengünstig anbieten zu können.

Informationsportale

Bei Informationsportalen mit einer großen Anzahl von Daten braucht der Benutzer Hilfsmittel, um an die gewünschten Informationen zu gelangen. Insbesondere bei intensiver und häufiger Benutzung erwartet er eine Unterstützung des Systems wie durch die Personalisierung, indem sich das Angebot an seine Interessen anpasst. Die Überwindung der Informationsüberflutung ist in diesem Zusammenhang nicht nur ein Komfortfrage, sondern in manchen Systemen essentiell, um das Angebot überhaupt nutzbar zu machen. Denn wenn die Menge der angebotenen Informationen zu groß ist, um sie von einem menschlichen Bearbeiter in für ihn angemessener Zeit zu bearbeiten, ist das Angebot für ihn ineffizient. Mit einem Zahlenbeispiel sei dies hier verdeutlicht: Das Presseportal pressrelations.de[15] liefert jeden Tag zwischen 20.000 und 30.000 neue Meldungen aus Presseagenturen sowie Pressemeldungen aus Public Relations-Abteilungen von Unternehmen (siehe [Zie01]). Ohne Personalisierungsfunktionen könnten die Nutzer dieses Systems das Angebot nicht sinnvoll nutzen, da sie die für Ihren Arbeitsbereich relevanten Meldungen nicht mehr oder nur mit sehr hohem Arbeitsaufwand ausfiltern könnten.

Wie im Beispiel beschrieben, dient Personalisierung in Informationsportalen zur Reduzierung der Komplexität. Die Menge der angebotenen Informationen in solchen Portalen muss erstens ausreichend groß sein, um verschiedene Themen inhaltlich abdecken zu können und eine sinnvolle Menge an Benutzern anzusprechen. Zweitens ist der einzelne Benutzer innerhalb eines solchen Systems in der Komplexität gefangen, da er nicht alle vorhandenen Informationen selbst abrufen kann, sondern nur einen Ausschnitt davon. Die richtige Wahl des Ausschnittes benötigt Hilfsmittel, um zu den gesuchten Informationen zu gelangen. Neben unpersonalisierten Verfahren wie Volltextsuche und Kategorisierung können so Personalisierungsmethoden die Komplexität reduzieren. Beispielsweise indem der Benutzer sein Interessensprofil spezifiziert und ihm bevorzugt Dokumente vorgeschlagen werden, die seinem Profil entsprechen (siehe [Ric00], [TO01]).

Ein weiterer Aspekt kommt hinzu: Vergleicht man ein Informationsportal mit einer traditionellen Bibliothek, so wird man in beiden mit reichhaltigen Informationen überschwemmt und die Recherchemöglichkeiten sind mehr oder weniger gut ausgeprägt. Ein Bibliothekar kann einem Besucher in der traditionellen Bibliothek möglicherweise Empfehlungen aussprechen, Fundstellen verraten und Tipps zu lesenswerten Büchern geben. Allerdings wird diese Fähigkeit mit der Spezialisierung des interessierenden Themengebietes und der Größe der Bibliothek erheblich abnehmen, da der Bibliothekar nicht auf allen Gebieten bewandert sein wird. Findet der Informationssuchende in der traditionellen Bibliothek die gewünschten Bücher nicht, wird er emotional eher unzufrieden sein. Gleiches gilt natürlich auch für den Suchenden in der digitalen Bibliothek oder im Informati-

[15] siehe http://www.pressrelations.de

onsportal. Hier stehen aber mit Personalisierung Methoden zur Verfügung, die die Wahrscheinlichkeit erhöhen, die richtigen Fundstücke zu finden. Die gespeicherten Präferenzen des Benutzers und das aktuelle Suchinteresse bieten eine bessere Datengrundlage, als sie ein fremder Bibliothekar haben kann.

Generell kann der Benutzer durch Personalisierungskonzepte emotional besser angesprochen werden, da sich das Informationsportal an seine Interessen und Präferenzen anpassen kann – seien es einfache Konzepte wie anpassbare optische Elemente auf der Webseite oder Personalisierung der Suche (siehe [IO03]). Umfang und Form der Personalisierung legt der Portalbetreiber aber letztlich selbst fest, wobei er von Budget und Nutzenerwartung geleitet wird. Im Gegensatz dazu ist es (zumindest gegenwärtig) unvorstellbar, dass sich das Bibliotheksgebäude der traditionellen Bibliothek an seine Besucher anpasst, indem sich Wände verschieben oder jeder Bibliothekar alle Besucher und ihre Präferenzen bereits am Eingang kennt.

Bietet ein Informationsportal sein Angebot für den Nutzer kostenlos an, dann wird häufig Bannerwerbung zur Finanzierung geschaltet. Die meisten Banner sprechen einen Großteil der Benutzer nicht an, da sie den Benutzerinteressen schlicht nicht entsprechen. Personalisierte Banner hingegen erlauben die Zuordnung von bestimmten Bannern zu bestimmten Benutzern und Wecken eher die Aufmerksamkeit. Solche benutzerorientierten Banner können teurer verkauft werden und liegen so im Interesse des Portalbetreibers.

> Schlüsselwörter: *Abmilderung der Informationsüberflutung, Reduzierung von Komplexität, Emotionale Ansprache, Personalisierte Bannerwerbung*

Suchmaschinen

Die großen Internetsuchmaschinen wie Google[16], AllTheWeb[17] oder AltaVista[18] suchen ebenfalls Konzepte gegen die Flut von Informationen, um dem Suchmaschinennutzer trotz steigender Datenmengen zufrieden stellende Suchresultate liefern zu können. Einfache Personalisierungskonzepte werden bei der Suchmaschine Google bereits umgesetzt. Hierzu zählen die Sprachwahl der Benutzungsschnittstelle, die Anzahl der zurückgegebenen Treffer pro Seite, die Sprache der zu suchenden Seiten oder auch ein Filter für jugendgefährdende Inhalte. Weitere und intelligentere Personalisierungstechniken könnten zukünftig eingesetzt werden, um einen schnelleren und präziseren Zugriff auf den Suchmaschinendatenbestand zu erhalten, wobei die zurückgelieferten Suchergebnisse an den Interessen des Benutzers ausgerichtet sind (siehe [Heise03]).

> Schlüsselwörter: *Abmilderung der Informationsüberflutung, präzisere und an individuelle Präferenzen angepasste Suche*

Communities

Internetangebote, die es Menschen verschiedener Interessen und Neigungen ermöglichen, miteinander in Kontakt zu treten, sind in besonderem Maße von Personalisierung abhängig. Communities[19] können Plattformen bieten, damit Benutzer gemeinsame Hobbys besprechen, sich über Produkte von Herstellern austauschen oder sich in einer professionellen Expertenrunde treffen können. Die Auswahl an Themen ist praktisch unbegrenzt und die aufgezählten spiegeln nur einen Ausschnitt wieder. Mitglieder haben je nach Angebot die Möglichkeit, mit anderen Teilnehmern per E-Mail oder Chat in Kontakt zu treten, per Diskussionsforum über verschiedene Themen Mei-

[16] siehe http://www.google.com
[17] siehe http://www.alltheweb.com
[18] siehe http://www.altavista.com
[19] Eine Definition des Begriffes Community findet sich unter: http://de.wikipedia.org/wiki/Community

nungen auszutauschen und möglicherweise Treffen in der Offline-Welt zu verabreden. Dazu muss jeder Benutzer einen Zugang zur Community erhalten und seine persönlichen Daten und Schwerpunkte in einem Profil festhalten.

Personalisierung kann dazu beitragen, dass Communitymitglieder mit gleichen Interessen vermittelt werden können (siehe [Kim01] insb. Kapitel 2 und 3). Bei technischen Communities können so auch Laien an Experten zu einem Themengebiet vermittelt werden und Experten untereinander in Kontakt treten.

Communities leben von ihren Mitgliedern und der Interaktion der Mitglieder untereinander. Die Betreiber müssen daher ein besonderes Interesse an der Bindung ihrer Mitglieder an eben diese Community haben. Personalisierungstechniken wie eine individuelle Homepage für jeden Benutzer, eine konfigurierbare Optik des Erscheinungsbildes und für alle Benutzer einsehbare Steckbriefe fördern die Bindung der Benutzer an ihre Community.

Schlüsselwörter: *Benutzerbindung, Finden von Gleichgesinnten*

Online-Shops

Betreiber von Online-Shops wollen genauso wie Händler in traditionellen Kaufhäusern ihre Umsätze und Gewinne steigern. Ein Weg dazu ist die Akquise neuer Kunden. Mittel für die Akquise sind je nach Branche und Größe des Händlers unterschiedlich. Werbung, guter Service und gute Platzierung im Markt sind primäre Instrumente dafür. Personalisierung kann in Online-Shops dazu beitragen, potentielle Kunden vom Kauf neuer Produkte zu überzeugen, wenn der Kundenanwärter neben den anderen Shopattributen auch von den Personalisierungsfeatures überzeugt werden kann. Grundsätzlich nützt Personalisierung aber eher bei einer längerfristigen Kundenbeziehung als bei Spontankäufen, da die verschiedenen Präferenzen und individuellen Wünsche des Kunden erst im Laufe der Zeit analysiert werden können.

Viel stärker als zur Neukundengewinnung kann Personalisierung also dazu beitragen, bestehende Kunden zu halten. Schließlich ist die Bindung bestehender Kunden ein zweiter wesentlicher Weg für die Steigerung von Umsatz und Gewinn. Weitläufig gesprochen ist es einfacher, einen bestehenden Kunden zu halten, als einen neuen zu akquirieren. Entsprechend wird der Betreiber eines Online-Shops besonderes Augenmerk auf die Bindung bestehender Kunden an seinen Shop legen. Personalisierung kann hier eine wichtige Rolle spielen.

Sei es, dass der Kunde zunächst beim Betreten des Shops persönlich mit Namen angesprochen oder ihm ein virtueller Berater zur Seite gestellt wird. Der Kunde fühlt sich durch die individuelle Ansprache mit hoher Wahrscheinlichkeit wohler als bei anonymer Abfertigung. Bei traditionellen Offline-Shops im Massengeschäft ist eine solche persönliche Ansprache eher selten der Fall. Der Online-Shop hingegen kennt die Kaufhistorie jedes einzelnen Kunden und kann daraus individuelle Empfehlungen für den Kauf von Produkten abgeben, die die bisher gekauften Artikel ergänzen oder aus einem für den Kunden ähnlich nützlichen Bereich stammen. Auch nach dem Kauf kann der Kunde auf der Shop-Website wichtige Informationen zu seinen Produkten finden. Für den Shopbetreiber ist der Online-Shop so auch ein Supportinstrument, das kostengünstiger als andere Verfahren wie eine Telefonhotline arbeiten kann, wenn bereits die Kaufhistorie des Kunden sowie seine allgemeinen Präferenzen bekannt sind und der Kunde seine Anfragen in Supportforen oder per E-Mailanfrage absetzt.

Zufriedene Kunden führen zu stetigeren oder gar höheren Umsätzen als unzufriedene, da sie selbst wieder kaufen werden und anderen von ihren positiven Kauferlebnissen erzählen. Personalisierung kann mit den beschriebenen Methoden zu zufriedeneren Kunden führen. Angemerkt sei aber, dass Personalisierung kein Allheilmittel für Online-Shops ist und harte Fakten wie Liefergeschwindig-

keit, Preise, Umfang des Sortiments, Übersichtlichkeit im Seitenaufbau und Verhalten bei Reklamationen primär entscheidend sind. Personalisierung optimiert die Kundenbindung und die Kaufvorgänge jedoch. Zum Beispiel dient das Empfehlen weiterer Produkte zum Cross-Selling: Ein Kunde wägt die Entscheidung für den Kauf eines Produktes ab oder hat sich bereits für den Kauf entschieden, betrachtet die technischen Daten, den Preis und die Abbildungen des Produkts. Zusätzlich werden ihm weitere Produkte vorgeschlagen, die das eigentliche Hauptprodukt ergänzen. Zu Schuhen können so Socken oder Schuhcreme vorgeschlagen werden, was übrigens auch kein Unbekanntes Vorgehen in der Offline-Welt ist!

Ähnliches passiert beim Up-Selling, wo dem Kunden ein höherwertiges und möglicherweise auch höherpreisiges Produkt empfohlen wird – beispielsweise statt eines einfachen Fahrrads ein Luxusmodell.

Tendenziell enthalten solche Empfehlungen noch keine persönliche Note. Integriert man jedoch die Präferenzen des Benutzers, so werden – um bei den Beispielen zu bleiben – nicht einfach Socken vorgeschlagen, sondern Socken in der Lieblingsfarbe des Benutzers soweit sie zum ausgewählten Schuhmodell passen. Die Wahrscheinlichkeit, dass der Kunde seiner individuellen Empfehlung folgt dürfte höher sein, als die eines unpersonalisierten Vorschlags (siehe [RK02], Seite xxi, römisch).

Personalisierung bietet aber noch weitere Verfahren an, um dem Kunden individuell interessante Produkte anzubieten. Mit der Kaufhistorie und Kollaborativem Filtern können auch Produkte als Empfehlungen präsentiert werden, die im aktuellen Kontext für den allgemeinen Betrachter unsinnig erscheinen. Für den individuellen Kunden können sie aber genau die gewünschte Ergänzung zu bisherigen Käufen darstellen. So klingt der gemeinsame Kauf von Bier und Babynahrung zwar verwunderlich, kann in der jeweiligen Kundensituation aber durchaus Sinn machen.

Neben dem Aussprechen von Kaufempfehlungen kann Personalisierung aber auch zu einer Anpassung des kompletten Online-Shops an die Kundenbedürfnisse genutzt werden. Im Stile eines „Mein Shop“, wie in Amazon[20] anbietet, können die dargebotenen Artikel und die Seitengestaltung ausschließlich den eigenen Interessen entsprechen (in Fall des Autors heißt der Shop im Shop „Martins Shop“). Weitere Features wie persönliche Wunschlisten, E-Mail-Benachrichtigungen über neue und veränderte Produkte, individuelle Newsletter und die schlichte Speicherung der Lieferanschrift führen zu einer noch stärkeren Bindung des Kunden an einen Online-Shop, wodurch auch die Bereitschaft sinkt, zu einem anderen Shop zu wechseln (siehe [IO03]).

In den vorangegangenen Absätzen lag die Ausrichtung eher auf dem Nutzen der Personalisierung für den Shop-Betreiber. In manchen Fällen wird der Kunde auch deutlich übervorteilt, beispielsweise wenn Empfehlungen benutzt werden, um Lagerbestände auszuräumen oder dem Kunden Produkte „angedreht“ werden sollen. Generell wird sich ein Online-Kunde in einem personalisierten Shop aber besser zurechtfinden und seinen Kaufbedürfnissen zufrieden stellender nachgehen können als in einem nicht an seine Interessen angepassten Online-Shop. Letztlich ziehen sowohl Anbieter als auch Kunde Nutzen aus personalisierten Websites.

> Schlüsselwörter: *Verkaufsförderung im Allgemeinen, Cross-/Up-Selling im Speziellen, Kundenservice, Kundenzufriedenheit, Kundenbindung*

Unternehmensportale

Wieder andere Beweggründe für den Einsatz von Personalisierung als Online-Shops haben Unternehmen, die Ihren Mitarbeitern Zugriff auf vielfältige Informationen mit Hilfe eines Unterneh-

[20] Amazons „Mein Shop“ unter http://www.amazon.de/exec/obidos/tg/stores/your/store-home/-/0/

mensportals bieten. In solchen Portalen finden sich neben trivialen Daten wie den Kantinenspeiseplänen und Telefonregistern auch komplette E-Mailarchive, Zugriffe auf die Mitarbeiterkorrespondenz, Reportingdaten, Richtlinien, Verträge, Produktbeschreibungen und vieles weitere. Zudem werden auch Anwendungen angeboten, wie das Einreichen von Urlaubsanträgen, interaktive Formulare für Beschaffungsmaßnahmen, die Anzeige von Kundenstammdaten und die generelle Abbildung von Unternehmensprozessen. Neben Mitarbeitern können auch Kunden und Zulieferer Zugriff auf Teile des Systems haben. Generell muss ein Rechtemanagement eingesetzt werden, um Informationen nur berechtigten Nutzern zugänglich zu machen – auch innerhalb des Unternehmens mit seinen verschiedenen Hierarchie- und Verantwortlichkeitsstufen (siehe beispielsweise [SAP02], [BS02]).

Ein Unternehmensportal hat das Ziel, möglichst viele Daten aus dem Unternehmen an zentraler Stelle und in gebündelter Form bereitzustellen. Die technische Umsetzung erfolgt primär mit Hilfe von Intra- und Extranet und dementsprechend mit Web-Standards. Die Masse der angebotenen Daten in Folge des immanenten Anspruch des Systems, nahezu sämtliche Unternehmensdaten zur Verfügung zu stellen, resultiert jedoch darin, dass Mitarbeiter bei der Informationsbeschaffung überfordert werden.

Der einzelne Mitarbeiter benötigt nur für die in der aktuellen Arbeitssituation zur Lösung der Aufgabe relevanten Informationen. Personalisierungskonzepte helfen dann, ihm maßgeschneidert die nötigen Informationen und auch Anwendungen bereitzustellen (siehe [DR02]). Zudem ist eine übergreifende Assistenz des Systems interessant, um den Mitarbeiter individuell über Termine, neue Dokumente und generell Veränderungen zu informieren (immer auf den Interessensausschnitt des Mitarbeiters ausgerichtet).

In der konkreten Umsetzung eines Unternehmensportals hilft Personalisierung Mitarbeitern, Zeiteinsparungen zu realisieren und so produktiver zu arbeiten. Durch die Festlegung von gewünschten Themen kann der Informationsarbeiter bestimmen, welche Informationen ihm in Listen und bei der Suche zur Verfügung stehen und über welche neuen Inhalte er informiert werden will. Durch die Konfiguration seines virtuellen Arbeitsbereichs im Portal kann er für ihn wichtige Elemente hervorheben – beispielsweise eine Informationsrecherche oder ein Tool zum Abfragen von Berichten – und weniger wichtige – wie im exemplarischen Fall den Anschluss an das Beschaffungssystem – ausblenden.

Zeiteinsparung bedeutet für das Unternehmen letztlich auch Kosteneinsparung. Hierin liegt ein Primärziel von Unternehmensportalen. So ist auch von Interesse, dass Mitarbeiter durch die Nutzung des Unternehmensportals und die Fokussierung auf ihren Themenausschnitt eher vermeiden, Inhalte mehrfach zu erzeugen, wenn andere Mitarbeiter den Inhalt bereits erstellt haben. Hier erfüllt das Portal eine Kommunikationsfunktion und durch die Personalisierung werden die Mitarbeiter über Änderungen in ihrem Themenbereich eher informiert, als wenn sie einer großen, unpersonalisierten und unüberschaubaren Informationsflut gegenüberstehen würden. Hierbei kann es auch vorkommen, dass Mitarbeiter mit anderen Mitarbeitern in Kontakt treten, die zwar aus anderen Abteilungen kommen, aber alle an ähnlichen Themen interessiert sind. Solche Kontakte lassen sich beispielsweise mit Techniken des Kollaborativen Filterns herstellen, die in Kapitel 3 noch näher betrachtet werden. Mit anderen Verfahren lassen sich ähnliche Inhalte zusammenführen und möglicherweise neues Wissen gewinnen. Dann führt das Unternehmensportal möglicherweise nicht nur zu Kosteneinsparungen sondern im Idealfall auch zu Wert- und Umsatzsteigerungen.

> Schlüsselwörter: *Zeiteinsparung, Vermeidung von Mehrfacherzeugung von Inhalten, Finden von Experten, Gewinnung von abgeleitetem Wissen*

2.1.3 Alternative Begriffe

Der Begriff der Personalisierung von Webseiten ist noch nicht allzu alt. Das World Wide Web ist einerseits selbst nur wenige Jahre alt, andererseits besteht der Bedarf nach Personalisierung erst, seitdem größere Websites aufgetaucht sind und größere Datenbestände online angeboten werden. Wie schon weiter oben erwähnt, wurde der Begriff aus dem „Personalisieren“ von Armbanduhren, Stiften und Frühstücksbrettern – dem Eingravieren von individuellen Schriftzügen und Bildern – ins Internet übernommen. Häufig wird Personalisierung genannt, wenn bestimmte Techniken zur Personalisierung eingesetzt werden. Aber das Konzept, das Erscheinungsbild und die Zusammenstellung von Inhalten an Benutzervorgaben anzupassen, ist auch ein Konzept oder Entwurfsmuster (Design Pattern).

Der Begriff „Personalisierung“ beschreibt also letztlich verschiedene Techniken und Konzepte, wobei alle die individuelle Anpassung von Systemen an Benutzer beschreiben. Alternativ trifft man in der Literatur auch auf Begriffe wie Customisierung [DR02], Adaption von Websites oder Adaptive Benutzungsschnittstellen [LWH03] bzw. deren englischsprachige Entsprechungen. Ebenfalls im Kontext von Personalisierung zu finden sind auch einzelne technische Methoden wie Empfehlungssysteme, bei denen dem Benutzer eine Liste von personalisierten Ressourcen eingeblendet werden [RK02].

2.2 Personalisierbare Elemente – Bausteine der Personalisierung

Die technischen Möglichkeiten im Zusammenspiel von Webservern, Anwendungen und Webbrowsern erlauben es, beliebige Daten zu personalisieren, die vom Webserver zum Browser geschickt werden. Diese Daten können verschiedene Formen annehmen und entweder tatsächliche Nutzdaten enthalten, beispielsweise Beschreibungen von Produkten, oder sie können eher eine Nebenrolle spielen. Hierzu zählen beispielsweise Werbebanner, die zwar wenig oder keine direkten Informationen tragen, welche im Zusammenhang mit den restlichen Inhalten der Website stehen, die aber dennoch jedem Benutzer individuell zugewiesen werden können und damit eine Komponente der Personalisierung darstellen.

2.2.1 Inhalte und Informationen

Die Begriffe Inhalte und Informationen sind zwar sehr allgemein gehalten, aber genauso vielfältig sind die verschiedenen Angebote des Internets. Jede auf einer Website angebotene Information kann in einzelne Blöcke zergliedert, kategorisiert und verschlagwortet werden. Weniger gut automatisch erfassbare Daten wie Bilder, Musik und Videos lassen sich durch Metadaten anreichern[21] und durch menschliche Experten klassifizieren.

Je nach Personalisierungstechnik können nur die für den Benutzer interessanten Informationen ausgegeben und die weniger interessanten ausgeblendet werden. Auch kann die Reihenfolge der Ausgabe gemäß der Priorisierung des Benutzers erfolgen, um so für ihn wichtige Informationen näher an sein Blickfeld heranrücken zu lassen. Einfache Techniken erlauben beispielsweise das Zusammenstellen von bevorzugten Nachrichtenkategorien oder Börsenkursen.

[21] siehe W3C Note zur Annotation von Bilddaten mit RDF, Quelle [LB02]

2.2.2 Dienstleistungen

Benutzer in personalisierten Informationssystemen können zur verstärkten Nutzerbindung bevorzugte Behandlungen erhalten. Hierzu zählen z. B. die frühzeitige Freischaltung von Informationen, bevor sie der breiten Masse an Benutzern zugänglich gemacht werden, Beratungsleistungen mit persönlichen Experten oder auch die Möglichkeit, in Diskussionsforen und Chatrunden Meinungen mit anderen Benutzern auszutauschen. Die Personalisierung kann die Nutzer unterteilen und Dienste nur einer registrierten Gruppe anbieten. Die Dienste selbst können darüber hinaus personalisiert sein, beispielsweise nach dem Grad der Nutzung. Aktive Nutzer können bevorzugt behandelt werden, während weniger aktive Nutzer nur Standarddienste im jeweiligen Kontext in Anspruch nehmen können. Auch können Beratungsleistungen durch persönliche Experten anhand der bekannten Präferenzen des Benutzers erfolgen.

2.2.3 Produkte

Ein klassisches Einsatzgebiet von Personalisierung im Internet sind Online-Shops, da die Präsentation der Produkte und sogar die Produkte selbst individuell auf die Bedürfnisse des Kunden abgestimmt werden können. Je nach den Präferenzen des Kunden können Produktangebote angezeigt werden, zu denen der Kunde vermutlich eine hohe Affinität aufweist. Die Basis für die Empfehlungen kann beispielsweise aus früheren Käufen oder Bewertungen des Kunden ermittelt werden.

Ähnlich zu realen Kaufhäusern und Supermärkten, in denen Aktionsartikel an exponierter Stelle in den Verkaufsräumlichkeiten positioniert werden, beispielsweise nahe der Kasse oder direkt am Eingang der Geschäfte, kann dies auch in Online-Shops erfolgen. Während der Platz in der Realität aber beschränkt ist und wenige Aktionsartikel für alle Kunden gleichermaßen platziert werden, kann in Online-Shops jeder Kunde durch Personalisierung auf ihn abgestimmte Aktionsartikel präsentiert bekommen. Die Wahrscheinlichkeit, dass ein Kunde einen solchen Artikel näher betrachtet oder gar kauft ist ungleich höher als bei der Präsentation für die Masse der Käufer in realen Geschäften. Diese Wahrscheinlichkeit steigt mit der Qualität der Abstimmung der angebotenen Produkte auf die Interessen des Kunden. Eine schlechte Abstimmung führt zu Frustration und vermutlich Nicht-Beachtung und damit Nicht-Kauf der angebotenen Ware. Eine gute Abstimmung wird den Nutzer hingegen eher zum Kauf animieren. Dadurch können auch im Internet so genannte Cross- und Up-Selling-Strategien erfolgen.

Hierbei geht es für den Händler zwar um den Verkauf von mehr (Cross) bzw. höherwertiger Ware (Up) als der ursprünglich vom Kunden geplanten, aber dem Kunden wird damit ebenfalls ein stärker zufrieden stellender Kauf ermöglicht. Cross- und Up-Selling sind daher eher als Beratungsleistung zu sehen, denn als „Kundennötigung“. Durch Up-Selling wird ein möglicher Fehlkauf verhindert und durch Cross-Selling können nötige Zubehörprodukte zum eigentlichen Produkt hinzugefügt werden. Dennoch darf nicht vergessen werden, dass vor allem der Händler durch höheren Umsatz davon profitiert.

Neben der Produkt-Platzierung kann im Internet auch die Personalisierung der Produkte selbst ermöglicht werden, wie eingangs im Einschub zu Mass Customization beschrieben. Websites bieten hier mit den technischen Interaktionsmöglichkeiten Konfigurationsmechanismen an, mit denen Produkte an die Bedürfnisse des Kunden angepasst werden können. Verschiedene Online-Shops wie Apple oder Dell, aber auch Smart für Automobile[22] demonstrieren das. Die eigentliche Produktion und Auslieferung der Waren erfolgt aber im Realen, so dass man hier von einer hybriden Personalisierung sprechen könnte.

[22] Produktkonfigurator der Firma Smart unter: http://www3.smart.com/CarConfigurator/ModelType.aspx

Die Individualisierung virtueller Produkte hingegen – wie Texte, Musik, Filme und Spiele – ist überraschenderweise noch wenig vorangeschritten. Aber gerade bei immateriellen Produkten ist eine Konfiguration wesentlich leichter als bei materiellen, industriell gefertigten Waren. Texte könnten aus einzelnen Bausteinen zusammengefügt werden, Musiksammlungen individuell aus einzelnen Songs zusammengestellt sein und auch Zusatzprodukte wie weitere Levels bei Computerspielen flexibler bereitgestellt werden. Virtuelle, per E-Mail versendete und an individuelle Vorgaben angepasste Grußkarten sind ein weiteres Beispiel.

2.2.4 Preise

Neben der Produktauswahl spielt in Online-Shops auch die Preisgestaltung eine wichtige Rolle für das Kaufverhalten der Benutzer. Allgemein müssen Preise in Online-Shops gegenüber der Konkurrenz in realen Geschäften bestehen können. Was dabei konkurrenzfähig ist, entscheidet letztlich der Kunde, wenn er einen unterschiedlichen Service, eine andere Produktvielfalt oder eine bessere oder schlechtere Verfügbarkeit der Ware geboten bekommt. Personalisierung kann zudem zwischen verschiedenen Kunden unterscheiden. Wie Stammkunden in traditionellen Offline-Geschäften können Vielkäufer auch in Online-Shops Rabatte und Sonderkonditionen erhalten. Dadurch wird eine Abgrenzung der Stammkunden zu erstmaligen oder weniger kaufbegeisterten Kunden vorgenommen. Denkbar wäre, die ausgewiesenen Preise je nach Benutzer unterschiedlich zu gestalten. Das ist aber nach [RK02] gegenüber anderen Kunden schwerer zu vermitteln. Dennoch ist eine Preisdifferenzierung möglich, wenn zwar allen Kunden der gleiche Preis ausgewiesen wird, aber die gewünschte Käuferschicht unterschiedliche Rabatte erhält. Dieses Konzept ist zudem nicht neu, sondern mit Skonti und Rabatten auch bei Offline-Geschäften erprobt. Wichtig ist aber die Untergliederung der Käuferschicht durch Segmentierung und – falls der einzelne Kunde besonders berücksichtigt wird – durch Personalisierung.

Eine weitere Möglichkeit zur Preisgestaltung liegt durch Personalisierung vor, wenn dem Kunden zu dem von ihm anvisierten Produkt noch ähnliche Waren angeboten werden. Dies kann, wie später noch gezeigt wird, mit verschiedenen Verfahren erreicht werden. Entscheidet sich der Kunde zusätzlich für den Kauf der speziell für ihn angebotenen Produkte (also Cross-Selling), dann kann ihm auch ein leicht rabattierter Preis geboten werden. Für den Händler sinkt dann zwar die Marge, aber der Umsatz steigt, da mehr Produkte verkauft werden.

2.2.5 Layout und Navigationselemente

Bei größeren Websites werden dem Benutzer eine Vielzahl von Texten, Optionen und Links angeboten. Nur ein Teil der Links, die zu Informationsangeboten oder Diensten führen, ist für den einzelnen Nutzer interessant. Insbesondere aktive Anwender werden daher Personalisierungsmöglichkeiten schätzen, die die Konfiguration von Menüs, Favoritenlisten und dargestelltem Inhalt zulassen.

Bei Menüs, die Hyperlinks auf Inhalte und Funktionen enthalten, können die angebotenen Inhalte je nach Kenntnisstand und Nutzungsdauer des Benutzers personalisiert werden. So erhalten erstmalige und unerfahrene Benutzer nur einen Teil der Funktionalität, um bei vielen Optionen nicht überfordert zu werden. Benutzer mit hoher Nutzungshäufigkeit erwarten hingegen die gleich bleibende Darstellung der Navigation von Nutzung zu Nutzung des Informationssystems. Expertennutzer, die viele Funktionen des Systems benutzten, möchten hingegen einen breiten und einfachen Zugriff auf alle Funktionalitäten. Solche Anforderungen treffen übrigens nicht nur für Websi-

tes und Informationssysteme zu, sondern auch für Arbeitsplatzanwendungen im Generellen (siehe [Shn02], Kapitel 16, speziell 16.5).

Wie bei Menüs kann auch bei Themen- und Inhaltelisten die Nutzungshäufigkeit der einzelnen Einträge beobachtet werden. Durch den Benutzer häufig genutzte Punkte werden dann an bevorzugter Stelle positioniert, während weniger oder nicht genutzte an peripheren Stellen oder gar nicht ausgegeben werden. Alternativ kann die Anordnung für alle Benutzer gleich sein, für den aktuellen Benutzer aber interessante Elemente können durch Piktogramme, Farben oder abweichende Schriftstile hervorgehoben werden.

Häufig bei Unternehmensportalen und Arbeitsumgebungen anzutreffen ist die Möglichkeit, das Layout des Informationssystems in gewissen Grenzen umzugestalten (angewendet in [SAP02]). Benutzer können individuell wählen, welche Funktions- und Informationsmodule sie wünschen und an welcher Stelle des Arbeitsbereichs sie angezeigt werden sollen. Auch die Farb- und Designwahl des Systems kann an die Benutzerpräferenzen angepasst werden, was bei gegenwärtig gebräuchlichen Desktopbetriebssystemen schon seit längerem möglich ist. Dadurch werden die Arbeitsbedingungen verbessert, da der Benutzer seinen eigenen Geschmack zur Geltung bringen kann und durch die Anordnung und Auswahl der für ihn wichtigen Module die nötigen Informationen im schnelleren und direkteren Zugriff stehen. Solche Anordnungs- und Layoutwahlmöglichkeiten bestehen auch bei großen öffentlichen Internetportalen und Communities wie beispielsweise Yahoo[23].

Abbildung 8 - Layoutkonfiguration (Mein Yahoo)

Mit eigenen Favoritenlisten kann der Benutzer interessante oder häufig genutzte Inhalte in Form einen Merkzettels speichern und bei regelmäßiger Nutzung sehr schnell durch einen statt vieler Mausklicks darauf zugreifen. Im Informationssystem muss es dazu bei der Anzeige der Inhalte in Listen- oder Detailform Möglichkeiten wie Hyperlinks oder Buttons geben, mit denen die aktuelle betrachtete Ressource in die Favoritenliste aufgenommen werden kann. Für den Systembetreiber ist diese Technik interessant, da die Inhalte der Listen aller Benutzer als Datenquelle für weitergehende Personalisierungstechniken dienen können.

[23] Layout- und Farbpersonalisierung siehe beispielsweise bei „Mein Yahoo" – http://de.my.yahoo.com

Abbildung 9 - Setzen individueller Lesezeichen für persönliche Navigation (tagesschau.de)

Ziel ist in allen Fällen, dass sich der Benutzer besser im System zurechtfindet, wohler fühlt und dadurch bessere Arbeits- oder Rechercheergebnisse erzielt. Bei Online-Shops geht es im Sinne von emotionaler Ansprache stärker um das Wohlbefinden des Kunden, mögliche Käufe und generell höhere Kundenzufriedenheit.

2.2.6 Ansprache des Benutzers

Ein einfaches aber wirkungsvolles – da persönliches – Merkmal zur Personalisierung ist die Ansprache des Benutzers mit ihm bekannten Informationen. Erstmalige Benutzer können Willkommensmeldungen und Hilfestellungen zur Systembenutzung erhalten, während häufige Benutzer z. B. beim fünfzigsten Besuch einen Dankesauspruch für die rege Nutzung präsentiert bekommen. Generell ist die Berücksichtigung unterschiedlicher Nutzungshäufigkeit sinnvoll, um so besonders aktive Benutzer speziell zu berücksichtigen und weniger aktive zu unterstützen. Vor allem bei Communities ist das anzutreffen (siehe [Kim01], Kapitel 4).

Aber auch die einfache Willkommensansprache mit dem eigenen Namen wie „Guten Tag Herr Meier“ oder die Gratulation zum Geburtstag sind nützlich, da der Benutzer damit mögliche Berührungsängste vor der anonymen Website abbaut und Vertrauen geschaffen wird. Nebenbei wird durch solche Merkmale die Begründung für die Eingabe der Profildaten erklärbarer.

2.2.7 Werbung

Vor allem öffentliche Informationsportale finanzieren sich zu einem Teil aus Werbeeinnahmen. Während die eigentlichen Inhalte wie Nachrichten, fachliche Artikel oder Unterhaltung kostenlos angeboten werden, dienen Bannereinblendungen mit Werbebotschaften als Einnahmequelle. Zufällig ausgewählte Banner erreichen nur einen kleinen Teil der Seitenbesucher. Die Klickrate – also der Anteil der Benutzer, die ein Banner sehen, anklicken und dem Linkziel folgen – ist gering, da das Banner thematisch oder optisch nur einen kleinen Teil der Benutzer anspricht.

Sind jedoch die Präferenzen eines Benutzers bekannt, können Bannerinhalte und Werbelinks geschaltet werden, die seinen Interessen vermutlich entsprechen. Der Seitenbetreiber kann also 1-zu-1-Marketing betreiben, da er jeden Besucher individuell anspricht. In Folge weckt die personalisierte Werbung eher Interesse beim Benutzer als eine Streuwerbung und darauf aufbauend steigt auch die Konversionsrate.

Das werbefinanzierte Fernsehen kann hiervon nur träumen... Bei Internet-Radiostationen hingegen ist mittlerweile auch ein 1-zu-1-Marketing möglich, wenn Werbung oder gar Inhalte personalisiert ausgestrahlt werden. Ein Beispiel für den Einsatz von Personalisierung bei Internet-Radiostationen ist unter [HF01] zu finden, wenngleich es hier um die Anpassung der Inhalte und nicht der Werbung geht. Der Schritt dorthin ist aber nicht allzu weit.

Aus Datenschutzsicht bedenklicher ist die Form der personalisierten Werbung, die von Werbefirmen praktiziert wird, welche als Dienstleister auftreten und auf vielen Websites Werbung schalten. Der einzelne Seitenbetreiber erhält einen Teil der Erlöse von den Bannerkunden, während ein anderer Teil beim Dienstleister verbleibt. Bedenklich ist der Einsatz, wenn die Bannerschaltung einen Cookie auf dem Rechner setzt, in dem Benutzerkennung oder -präferenzen gespeichert sind. Dann wird die Werbung zwar auf einer Website personalisiert, auf der die Banner des Dienstleisters geschaltet sind, aber auch auf anderen, die möglicherweise keinen Bezug zueinander haben. Zwar gelangen die Cookiedaten nur an den Bannerdienstleister, für den Benutzer erscheint es aber so, als ob die Daten ohne Benachrichtigung von einem Seitenbetreiber zu einem anderen gewandert sind, was nicht sonderlich vertrauenserweckend und darüber hinaus aus Datenschutzsicht fragwürdig ist.

2.2.8 Internationalisierung

Die Darbietung von Texten und Bildern in verschiedenen Sprachen ist nicht nur als ein Komfortmerkmal zu sehen, sondern durch sie werden vielen Benutzern die Inhalte sogar erst erschlossen. Zwar teilt ein Benutzer die gemeinsame sprachliche Präferenz mit vielen anderen Benutzern und sie stellt an sich keine individuelle Auslieferung von Inhalten dar, eher eine Gruppeneinteilung, aber dennoch ist die Internationalisierung als ein Teilaspekt der Personalisierung zu sehen.

Beispielsweise kann der Benutzer die bevorzugte Sprache in seinem Profil festlegen und arbeitet bei jedem Besuch des Systems mit der sprachlich abgestimmten Benutzungsschnittstelle. Das schließt die Sprache der Steuerelemente und Recherchetools ein. Der Suchmaschinenbetreiber Google ist hierfür ein gutes Beispiel, der anhand der IP-Adresse des zugreifenden Rechners die Sprache und das Herkunftsland sogar automatisch zu ermitteln versucht. Google bietet die Benutzungsschnittstelle in gegenwärtig 96 Sprachen an[24] – darunter auch eher humoristisch gemeinte Sprachen wie Latein und Klingonisch.

Ebenso ist die Bereitstellung von barrierefreien Aufbereitungen für körperlich eingeschränkte Menschen und die Voreinstellung darauf als mögliche Option eines Benutzerprofils denkbar.

2.2.9 Datenströme

Neben den inhaltlichen Bausteinen der Personalisierung spielt auch die technische Seite eine Rolle. Je nach Benutzer und verwendetem Computersystem können die Inhalte einer Website unterschiedlich ausgeliefert werden. Der Grad der nötigen Anpassung kann dazu sehr hoch sein, wenn eine heterogene Zielgruppe angesprochen wird. Benutzer mit älteren Rechnern und älterer Software benötigen anders aufbereitete Inhalte als solche mit neuen Systemen. Genauso haben Kleinstcomputer wie Persönliche Digitale Assistenten (PDAs) oder Mobiltelefone technische Beschränkungen in Bezug auf die Bildschirmauflösung, die Farbtiefe und die verfügbare Übertragungsbandbreite, die gesondert zu behandeln sind (siehe [Hjelm01]).

[24] Sprach- und Herkunftslandwahl bei Google unter http://www.google.com/language_tools?hl=en

Informationssysteme müssen die Inhalte so beispielsweise in verschiedenen HTML-Dialekten oder in WML für Mobiletelefone ausliefern. Auch personalisierte Web Services mit XML-Strömen sind denkbar, die noch einer Nachverarbeitung bedürfen, bevor sie dem Benutzer angezeigt werden. Zudem muss sich Personalisierung nicht auf das Web beschränken, sondern kann auch in E-Mails genutzt werden. Zielgerichtete, personalisierte E-Mails also als eine Alternative zur Massenabfertigung mit Spam.

Besonders bei Mobiltelefonen und mobilen Internetangeboten hat eine exakte Personalisierung einen hohen Nutzen, da die Bandbreite und die technischen Möglichkeiten extrem eingeschränkt sind. Jede zu viel nötige Navigation kostet erhebliche Zeit, verursacht unnötige Kosten und steigert die Anwenderunzufriedenheit. Mit mobiler Personalisierung könnten präferierte Menüpunkte und Interessen voreingestellt sein und so einen schnelleren Zugriff auf die gewünschten Informationen und Dienste erlauben.

2.3 Benutzerprofile als Personalisierungsgrundlage

Um Informationssysteme an die Bedürfnisse des Benutzers anpassen zu können, sind präzise Angaben über seine Präferenzen nötig. Je mehr Daten vorhanden sind, desto besser kann die Personalisierung erfolgen. Dabei ist zu definieren, wie die Präferenzen jedes Benutzers ermittelt, gespeichert und abgerufen werden können. In Informationssystemen geschieht dies mit dem abstrakten Konzepts des Benutzerprofils, das auf verschiedene Weisen implementiert werden kann.

Abbildung 10 - Schematischer Aufbau eines Benutzerprofils

In einem Benutzerprofil fließen unterschiedliche Datenströme zusammen, um ein gutes Bild der Benutzerpräferenzen zu ermöglichen. Hier erfolgt eine Unterteilung in statisch und dynamisch anfallende Daten sowie eine weitere dazu orthogonale Aufteilung in explizit und implizit gemachte Angaben.

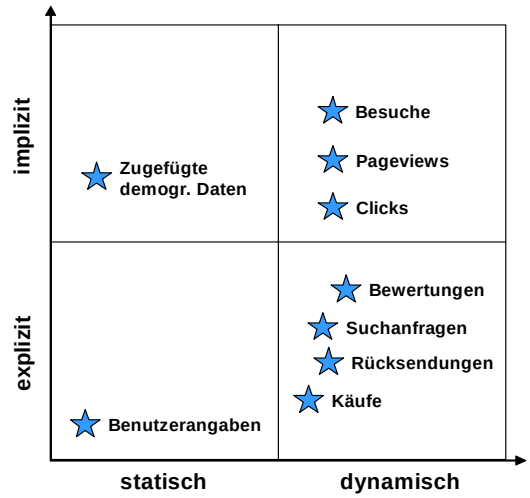

Abbildung 11 - Einordnung von Datenquellen in Benutzerprofilen

Statische Angaben werden einmal gemacht, treten in relativ kleiner Zahl auf und verändern sich vergleichsweise selten. Dynamische Daten hingegen werden kontinuierlich geliefert, verändern sich und werden aggregiert in das Benutzerprofil aufgenommen. Die Unterscheidung in explizit und implizit erfolgt, da Benutzer einerseits aus eigenem Antrieb und bewusst Daten bereitstellen – wie Angaben zur eigenen Person, z. B. Alter und Wohnort. Andererseits hinterlässt der Benutzer Datenspuren, wenn er ein Informationssystem benutzt. Diese können in implizite Profildaten umgewandelt werden. Die in Abbildung 11 aufgeführten Datenquellen werden dazu im Folgenden erläutert.

Individuell eingegebene und konfigurierte Interessen

Die einfachste Abfrage von Benutzerdaten erfolgt über Eingabeformulare mit Feldern zu verschiedenen Angaben wie Kontaktdaten, demographischen Daten oder einer Auswahl von Themengebieten. Sie können erhoben werden, wenn der Benutzer einen Zugang zum System einrichtet, um sich fortan anzumelden zu können.

Kontaktdaten sind vor allem interessant, wenn der Systembetreiber mit dem Benutzer in Kontakt treten möchte oder muss wie bei Online-Shops zur Auslieferung und für Rückfragen oder für Werbung. Eine möglichst große Zahl von Benutzern mit Kontaktdaten ist für viele Unternehmen ohne materielle Vermögenswerte das eigentliche Kapital. Relevante Felder hierfür sind z. B.

- Anmeldename, Passwort
- Vorname, Nachname, Namenszusätze
- Herkunftsland, Postleitzahl, Ort, Straße, Hausnummer
- Telefonnummern (Festnetz, Mobil, Fax)
- E-Mailadresse, Homepage, Instant-Messaging-ID

Problematisch ist, dass die Benutzer nicht immer gewillt sind, ihre persönlichen Abgaben herauszugeben. Falsche Eingaben sind daher durchaus möglich. Der Systembetreiber muss daher abwägen, inwieweit er die Daten überhaupt benötigt und wie viele er unbedingt abfragen muss. Sie stellen aber ein erstes Mittel zur persönlichen Ansprache auf der Website dar (siehe Kapitel 2.2.6 Ansprache des Benutzers).

Zum Aufbau von Regelbasierten Filtern und zum Marketing sind demographische Daten nützlich, mit denen Benutzer in Klassen unterteilt werden können (siehe 2.4.2 Regelbasierte Personalisierung). Hierzu zählen

- Geburtsdatum, Alter
- Geschlecht
- Familienstand, Kinder ja/nein, Anzahl der Kinder
- Region, Wohnort (z. B. durch PLZ kodiert)
- ausgeübter und erlernter Beruf, Schulbildung
- Einkommensklasse

Hier gilt ebenso, dass eine zu große Neugier des Betreibers zur Ablehnung oder gar zu bewussten Fehleingaben des Benutzers führen kann.

Schließlich können auch die präferierten Themen direkt über die Auswahl in Listen konfiguriert werden. Hierzu kann der Benutzer aus einer oder mehreren, auch hierarchischen Interessensgebieten auswählen. Die gewählten Themen werden im Benutzerprofil gespeichert und können als einfache Filter dienen (siehe 2.4.1 Checkbox-Personalisierung).

Von den individuell gemachten Angaben – speziell bei demographischen Daten und präferierten Themen – können Ableitungen vorgenommen und im Profil abgelegt werden. Beispielsweise können mit einigem Aufwand anhand der demographischen Daten und basierend auf Erfahrungswerten Kaufwahrscheinlichkeiten in Online-Shops ermittelt werden. In Folge werden Kunden dann Produkte empfohlen, die sie vermutlich kaufen werden. Aber auch andere Formen sind denkbar. So können anhand des Berufes oder der Schulbildung Artikel zu bestimmten Themen angeboten werden und dadurch eine Personalisierung erfolgen.

Externe Demographische Daten

Im Marketing- und Vertriebsbereich ist es üblich, Kundendaten mit externen demographischen Daten anzureichern. Dabei handelt es sich beispielsweise um Einkommensverhältnisse oder Kaufgewohnheiten einer Region, die von den vorliegenden Ortsangaben der Benutzer abgeleitet werden. Sie werden in Umfragen und anhand von Warenkörben ermittelt und können von darauf spezialisierten Dienstleistern eingekauft werden (siehe [Mena00], S. 314-339).

Solche Anreicherungen sind aber nur bedingt zutreffend, da sich in einer ärmeren Wohngegend auch einzelne reiche Bewohner verstecken können. Besucht ein solcher reicher Bewohner dann einen Online-Shop, werden ihm möglicherweise Produkte angeboten, die er definitiv nicht kaufen wird. Zur Verbreiterung der Datenbasis in den Benutzerprofilen können die Daten jedoch dennoch genutzt werden und können die Anzahl explizit abgefragter Angaben reduzieren helfen.

Bewertungen von Inhalt

Gezielte Aussagen über die Präferenz von einzelnen Ressourcen lassen sich über interaktive Bewertungen ermitteln. Auch textuelle Kommentare sind möglich, lassen sich aber nur schwer auswerten. Einfacher sind daher Bewertungen in numerischer oder symbolischer Form. Damit lassen sich sehr detaillierte Profile über die Vorlieben des Benutzers erstellen, die wiederum eine gute Personalisierung ermöglichen. Zudem werden die Bewertungen neben der eigentlichen Informationsaufnahme vorgenommen und dadurch nicht als störend empfunden. Manchen Benutzern liegt es sogar sehr daran, ihre Meinung abzugeben, insbesondere wenn die durchschnittliche Bewertung

öffentlich angezeigt wird und so jeder einen Beitrag zur Gesamtwertung leisten kann. Bewertungen gehören zu den expliziten, dynamischen Daten des Profils.

Zur Bewertungsabgabe setzt der Benutzer auf gängigen Websites zumeist einen diskreten Wert von 1 bis 5, wobei eine 1 niedriges Gefallen und eine 5 hohes Gefallen ausdrückt. Andere Wertebereiche und Stellvertreter wie Sterne sind ebenfalls anzutreffen. Hier stehen schlicht viele Sterne für hohes Interesse und niedrige für geringes. Zur Speicherung im Benutzerprofil werden die Angaben aber in leicht zu handhabende Zahlenwerte konvertiert.

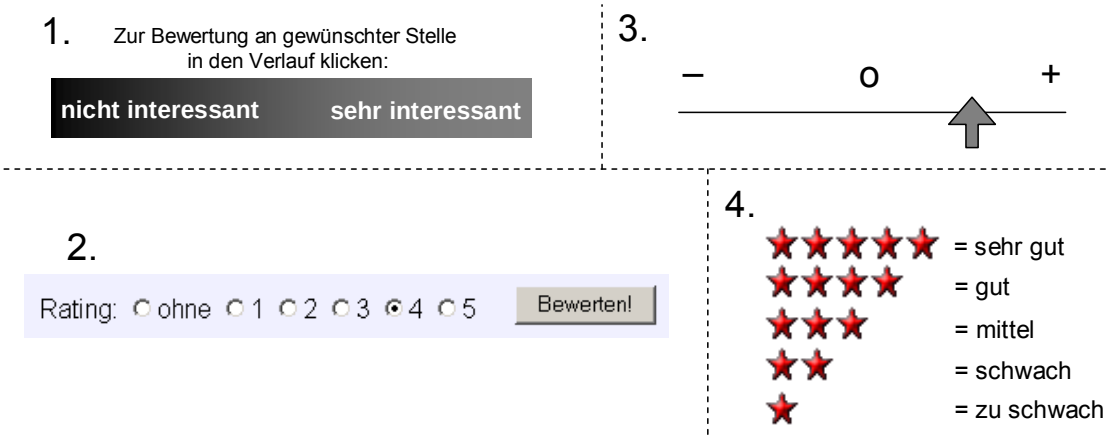

Abbildung 12 - Steuerelement zur Bewertung auf kontinuierlicher, numerischer Basis[25]

Wie in Abbildung 12 zu sehen, gibt es neben Abfragen in diskreten Bewertungsstufen (2. und 4.) auch solche, die auf einer kontinuierlichen Werteskala arbeiten (1. und 3.). Sie ermöglichen eine feinere Festlegung der Zustimmung zu einer Ressource. Letztlich sind aber alle aufgeführten Instrumente nur unterschiedliche Steuerelemente für die gleichen Daten.

Etwas anders dagegen sind Ja-Nein-Aussagen oder das einfache Wegklicken von Informationen bei Nicht-Gefallen. Sie erlauben nur eine binäre Aussage und sind daher sehr grob. Für den Benutzer sind sie allerdings einfach zu benutzen und haben daher ihre Daseinsberechtigung, da die Präferenz durch eine einfache binäre Entscheidung und anschließendes Anklicken festgelegt wird. Zur Wahl einer Bewertung auf einer Skala muss der Benutzer länger überlegen, da er mehr Möglichkeiten hat, und wird daher in der Summe weniger Bewertungen vornehmen.

Bei Online-Shops oder beim Inhalteverkauf kann der Benutzer nach dem Kauf zudem gefragt werden, ob er mit der erhaltenen Ware zufrieden ist und wie er sie bewertet. Das kann mittels sehr aufwändiger telefonischer Befragung, per E-Mail oder über einen Fragebogen auf der Website geschehen.

Beobachtung des Benutzerverhaltens

Daten aus der Beobachtung des Benutzerverhaltens sind die am einfachsten zu ermittelnden und am zahlreichsten vorhandenen Daten, da der Benutzer keinen besonderen Aufwand treiben muss: Er muss das Informationssystem schlicht benutzen! Alle seine Handlungen können als Datenquelle dienen und in sein Benutzerprofil einfließen. Hierzu zählen

- Anzahl Anmeldungen (gesamt, pro Zeitabschnitt), Verweilzeit im System
- Anzahl Ressourcenabrufe, Clickstreams, Dauer der Betrachtung

[25] Gesehen bei „Jester 2.0 - Jokes for *Your* sense of Humor" – http://shadow.ieor.berkeley.edu/humor

- Suchanfragen (Suchterme, gefundene Ressourcen)
- Käufe (Warenkörbe, Bestellungen) bei Online-Shops, Inhalteverkauf und Auktionen
- Rücksendungen bei Nicht-Gefallen der Ware
- Werbung, Bannereinblendungen

Die Erfassung der Einblendung von Werbung und Bannern für einen einzelnen Benutzer kann dazu dienen, diesem Benutzer in Zukunft andere Banner anzuzeigen, so dass wiederholte Werbung vermieden wird. Durch die Registrierung von Clicks auf Banner kann man zudem eine individuelle Erfolgsmessung der Werbemaßnahmen machen.

Alle Daten können entweder aus den Zugriffsprotokollen der beteiligten Systeme wie Webserver und Shopsystem mit Auswertungssoftware gewonnen und in extrahierter Form im Profil gespeichert werden. Oder die Daten werden direkt im Benutzerprofil erfasst, wenn die jeweilige Aktion wie der Abruf einer Ressource, die Anmeldung oder der Kauf eines Produktes erfolgt.

Kontext

Bei großen Informationssystemen kann je nach Teilbereich ein unterschiedliches Verhalten gegenüber dem Benutzer relevant sein. Beispielsweise sind in einer Kategorie andere Empfehlungen auszusprechen als in einer anderen. Im Benutzerprofil kann vorab eine Gliederung der Daten für die verschiedenen Bereiche der Website erfolgen. Ein Informationsportal mit angegliedertem Online-Shop kann z. B. im Informationsbereich andere Daten des Benutzerprofils benötigen als im Shopbereich.

Ferner hängt der Kontext auch von der Situation des Anwenders ab. Bei ortsbasierten Diensten in mobilen Systemen[26] kann je nach aktuellem Aufenthaltsort des Benutzers ein unterschiedliches Informationsangebot vorgeschlagen werden. Bietet das Informationssystem auch eine Website für den Zugriff von stationären Systemen, sind hierfür andere Profildaten relevant. Allgemeiner sind daher für temporär unterschiedliches Verhalten mehrere Bereiche im Benutzerprofil nötig, die für die jeweilige Situation angepasste Daten bereithalten.

2.3.2 Modell von Benutzerprofilen

Für jeden Benutzer *uid* aus der Menge der Benutzer *U* im Informationssystem wird ein Profil P_{uid} verwaltet. Dabei handelt es sich um eine abstrakte Datenstruktur, die für die verschiedenen im vorangegangenen Kapitel vorgestellten Daten unterschiedliche Speicherorte anbietet. Die physikalische Speicherung z. B. in Datenbanksystemen kann davon natürlich abweichen, um beispielsweise Performancevorteile zu gewinnen.

Die abstrakte Datenstruktur ist hierarchisch nach den Datenquellen sortiert. Je nach Informationssystem sind dabei nur ein Teil oder alle Ebenen mit Daten gefüllt.

```
class UserProfile {
    login {
        username;
        password;
    }
    private {
        contact {
            prename; lastname; ...
```

[26] Besser bekannt als „location based services“ bei Mobiltelefonen

```
            country; zip-code; city; ...
            email; homepage;
            ...
        }
        demographic {
            dateofbirth;
            profession; job;
            income;
            ...
        }
    }
    usage {
        common {
            numberoflogins;
            numberoffailedlogins;
            usagetime;
        }
        sessions {
            Liste der Sessions mit Datum und Dauer;
        }
        resources {
            für jede Ressource ein Datensatz {
                alle Clicks mit Datum;
                alle Käufe + Rücksendungen mit Datum;
                alle Einblendungen + Clicks auf Banner mit Datum;
            }
        }

        ...


        searches {

            für jede Suchanfrage Suchausdruck + Datum

        }
    }
}
```

Abbildung 13 - Abstrakte Datenstruktur des Benutzerprofils

Auf der Datenstruktur sind Lese- und Schreiboperationen zu definieren, um Abfragen der Profildaten für die Personalisierung zu ermöglichen und gegebenenfalls Änderungen vorzunehmen. Die Abfrage des Geburtsdatums von Benutzer 12 ist beispielsweise so möglich

```
dateofbirth := P12.private.demographic.dateofbirth;
```

Der Abruf der Folge von Aufrufen (Clicks) von Ressource 9392 durch den Benutzer 12 würde so aussehen

```
clicks_sequence := P12.usage.resources(9392).clicks;
```

Das Hinzufügen eines neuen Clicks auf Ressource 9392 zum Zeitpunkt *date* würde mit einer entsprechenden Funktion auf der Profildatenstruktur erfolgen

```
P12.usage.ressources(9392).add_click(date);
```

2.3.3 Speicherung und Anbindung

Die Speicherung von Profildaten ist im Wesentlichen das Speichern von Datensätzen mit je nach beabsichtigtem Einsatzzweck abweichenden Attributmengen. Einfache Profile sind einem Adressbuch ähnlich, komplexere enthalten zusätzlich weitergehende Angaben über die Interessen und Präferenzen von Benutzern wie in der vorangegangenen Datenstruktur vorgestellt. Zudem können Protokolle von Verhaltensdaten in aggregierter oder in Rohform hinzugezogen werden. In der Implementierung gibt es speziell zu berücksichtigende qualitative Anforderungen:

- flexibler lesender Zugriff für einzelne Profile und Profilmengen
- schreibender Zugriff im Vergleich zu lesendem Zugriff seltener
- großer Zuverlässigkeits- und Sicherheitsanspruch wegen Datenschutz und Wert
- hohe Zugriffsgeschwindigkeit

Der lesende Zugriff auf die Profildaten muss so möglich sein, dass er den höchst unterschiedlichen Personalisierungsalgorithmen ein großes Maß an Flexibilität erlaubt. So sind sowohl der Zugriff auf einzelne Profile als auch der Zugriff auf Mengen von Profilen mit Selektion und Projektion wichtig. Wird nur ein Personalisierungsverfahren eingesetzt, kann sich die Datenspeicherung aber hieran orientieren und spezielle Fähigkeiten integrieren wie die Aggregation von bestimmten Profildaten. Die volle Bandbreite an Möglichkeiten von Abfragesprachen wie SQL wird aber nicht benötigt.

Ändernde Zugriffe treten im Vergleich zu den lesenden weniger auf. Dennoch muss eine Aktualisierung der Profildaten – sei es durch explizite wie implizite Änderungen – möglich sein. Zudem sind Zuverlässigkeit und Sicherheit wichtige Aspekte, einerseits aus Datenschutzgründen, andererseits, da die Benutzerprofile einen großen Teil des Kapitals von Informationsportalen und Online-Shops darstellen.

Weiterhin ist eine hohe Abfragegeschwindigkeit nötig, da die Profildaten das unterste Glied in der Personalisierungskette darstellen. Alle Verfahren und Algorithmen der Personalisierung bedienen sich der gespeicherten Profile der Benutzer und benötigen daher einen sehr schnellen Zugriff. Besonders deswegen, da die Personalisierungsverfahren selbst sehr schnell sein müssen, um beim Anwender akzeptiert zu werden (siehe auch 3.4.5 Performance und Skalierbarkeit).

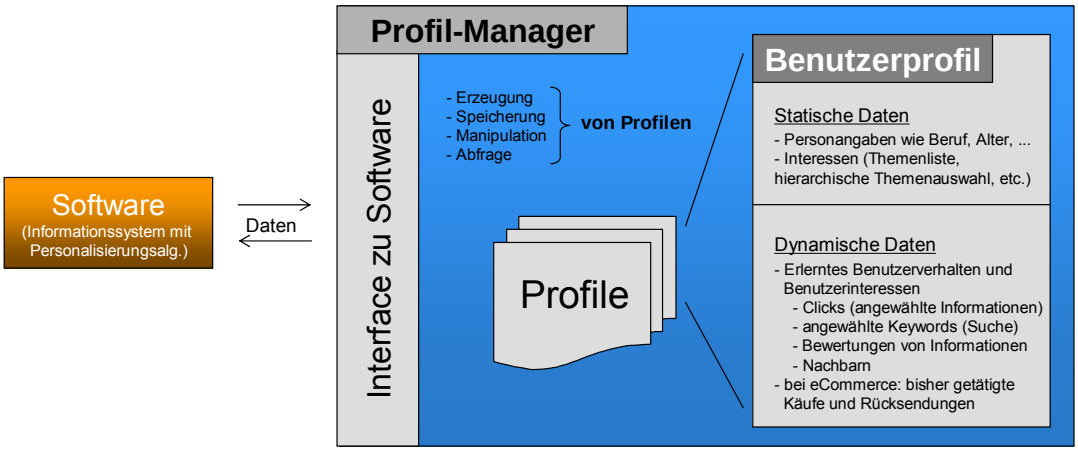

Abbildung 14 - Softwarebaugruppe Profil-Manager zur Verwaltung von Benutzerprofilen

Anhand der genannten Anforderungen kommen verschiedene Möglichkeiten zur Speicherung von Profilen in Anfrage. Dazu zählen relationale oder objektorientierte Datenbanken oder Verzeichnisdienste, die beispielsweise LDAP-konform sind (siehe [IETF95]). Um die nötige Geschwindigkeit zu erzielen, können die Profildaten auch verteilt gespeichert werden. Denkbar ist so, dass die Bewertungen physikalisch nicht bei den persönlichen Angaben wie Name und Wohnort sondern an anderer Stelle gespeichert werden.

Die Anbindung des Informationssystems an die mit einer der oben genannten Techniken implementierten Profilverwaltung kann entweder auf unterster Ebene via ODBC, JDBC oder ähnlichem erfolgen oder über vorgeschaltete Wrapper. Das ist letztlich eine Frage der Vorgaben bei der Entwicklung des Informationssystems. Die Profilverwaltung kann ferner vom eigentlichen Informationssystem abgetrennt und autark sein, was sich gut für die Lastverteilung bei großen Systemen eignet. Alternativ ist sie fest in das System integriert, was für die Implementierung einfacher ist, da die Personalisierungsalgorithmen in vielfältiger Weise auf die Profildaten zugreifen müssen.

2.3.4 Gewinnen von dynamischen Profildaten

Benutzer von Websites navigieren durch das ihnen dargebotene Informationsangebot, indem sie die in ihrem Webbrowser angezeigten Texte lesen und die Bilder betrachten und interpretieren. Stoßen sie beim Lesen auf einen Hyperlink, können sie ihn anklicken, um auf eine andere Seite zu gelangen und hier mit der Informationsaufnahme fortzufahren. Durch geschicktes Platzieren von Hyperlinks kann man Handlungen entsprechend provozieren und den Anwender leiten, was einem Kernelement der Personalisierungsbemühungen entspricht.

Interessant für die Personalisierung ist, welche Inhalte ein Benutzer betrachtet und wie lange er bei einem Text verweilt, denn das könnte als Grad für die Interessantheit der betrachteten Informationseinheit gewertet werden. Zudem können nacheinander betrachtete Ressourcen Aufschlüsse über die Verwandtheit von Inhalten geben. Man lernt damit, welche Interessen ein Benutzer haben könnte. Nützlich ist darüber hinaus, welche Werbebanner ein Benutzer betrachtet und welche er dann auch angeklickt hat – eine Spezialisierung der allgemeinen Benutzerbeobachtung.

Da der Benutzer der Website keine konkreten Angaben über seine Interessen macht, beispielsweise durch Eingabe von Schlüsselwörtern oder die Wahl von vorgegebenen Kategorien, sondern sein Interessensprofil durch das Besuchen und Lesen von bestimmten Seiten indirekt beschreibt, spricht man bei diesen Benutzerhandlungen von impliziten Profildaten.

Kauftransaktionen und Rücksendungen von nicht gewollter Ware sind in Online-Shops als dynamische, explizite Daten zu betrachten, da der Kunde aktiv die Ware bestellt oder zurückgesendet hat. Diese Daten können leicht aus den Transaktionsdatenbanken des Shopsystems gewonnen werden und werden hier daher nicht weiter betrachtet. Stattdessen geht es um die impliziten Daten, die aus den Protokollen des Webservers gewonnen werden können.

Wie gewinnt man implizite, dynamische Profildaten, um sie für die gewünschten Zwecke zu verwenden?

Blickt man einem Benutzer über die Schulter, dann könnte man mit einem Notizblock jeden angesehenen Text, jeden Mausklick und jedes eingeblendete Bild protokollieren. Beispielsweise in Form von Strichlisten. Mit der Stoppuhr könnte man zudem festhalten, wie lange eine Seite betrachtet wurde oder wie lange es dauert, bis ein Benutzer nach dem Betrachten einen Hyperlink mit der Maus anklickt und auf einen anderen Bestandteil des Informationsangebots verzweigt.

Anschließend kann man die Aufzeichnungen nach Zeit oder Benutzer sortieren und beurteilen, welche Seiten von besonderem Interesse für alle oder einzelne Besucher sind, welche Angebote zusammen besucht werden oder welche Bereiche keinen Benutzer anziehen.

Natürlich ist dies nur ein bildhaftes Verfahren. In der Realität von Webserver wird die Protokollierung des Benutzerverhaltens automatisch vorgenommen. Jedes ausgelieferte HTML-Dokument, jedes Bild und jeder Click werden auf dem Server im Protokoll festgehalten. Jeder Zugriff entspricht dabei einer Zeile oder aus Datenbanksicht einem Datensatz (siehe Kapitel 1.3.5 Protokollierung von Zugriffen). Die Protokolle stehen dem Websitebetreiber später zur Analyse zur Verfügung.

2.3.5 Logdateianalyse zur Ableitung impliziter, dynamischer Daten

Die Analyse von Logdateien läuft in zwei wesentlichen Schritten ab. Zunächst werden die rohen Protokolldaten gefiltert und jede Zeile, die keine nützlichen Zugriffsdaten enthält, wird aussortiert. Im zweiten Schritt werden die verbliebenen Zeilen den einzelnen Benutzern mit Hilfe der protokollierten Sitzungskennung zugeordnet und anschließend die Kennzahlen des Benutzerverhaltens ermittelt.

Bei der Vorverarbeitung wird jede Zeile zunächst gemäß des Protokollformates in einzelne Felder aufgespalten. Anhand des *request*-Feldes, das den Ort der angeforderten Ressource enthält, wird entschieden, welche Datensätze nützliche und welche unnütze Informationen enthalten (siehe Kapitel 1.3.5 Protokollierung von Zugriffen).

Nützliche Datensätze sind diejenigen, die durch den Abruf informationstragender Ressourcen erzeugt wurden. Dazu gehören HTML-Dokumente, die auf tatsächliche Ressourcen wie Texte verweisen, nicht jedoch Navigationsgrafiken, Layoutanweisungen, Framesets oder Startseiten. Ferner sind für die Gewinnung impliziter Daten nur diejenigen Zugriffe interessant, die eine einzelne Ressource geliefert haben, wie Detailansichten. Listenansichten sind weniger nützlich, da hier nicht klar ist, welche Ressource der Benutzer sehen wollte und der Informationsgehalt entsprechend niedrig ist. Anders bei Listen von Suchergebnissen, da hier der Benutzer einen Suchbegriff eingegeben hat und sowohl dieser als auch die gelieferten Ressourcen interessant sind. Zudem muss es sich um Zugriffe von angemeldeten Benutzern handeln, was man an der Sitzungskennung entweder am angeforderten URL im *request*-Feld oder am *cookie*-Feld erkennen kann (siehe Kapitel 1.3.4 Sitzungen).

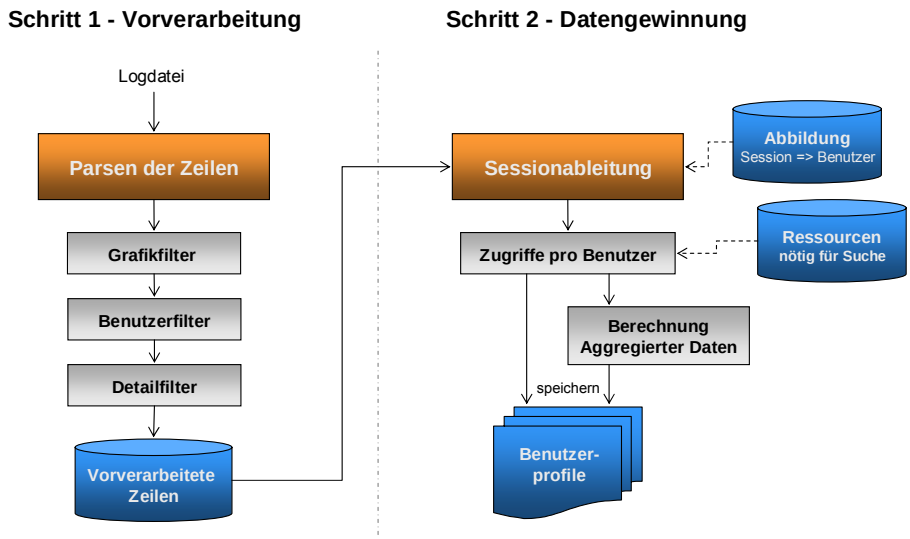

Abbildung 15 - Ablauf Logdateianalyse

Die verbliebenen Datensätze werden im zweiten Schritt jedem Benutzer anhand der Sitzungskennung zugeordnet. Dazu bedarf es einer im System gespeicherten Zuordnung von ehemaligen Sitzungskennungen zu Benutzern. Für jeden Benutzer kann dann eine Reihe von Sitzungen aus den Logdateien extrahiert werden, von denen jede eine Folge von Ressourcenabrufen enthält. Die Zugriffe werden anschließend durchlaufen und die gewünschten Daten und Kennzahlen bestimmt.

Der Abruf einer Detailseite mit einer Ressource kann direkt im Benutzerprofil gespeichert werden. Dazu wird die in 2.3.2 gezeigte Datenstruktur verwendet. Der Befehl für Benutzer *uid* und eine Ressource *iid* sieht für einen Zugriff mit dem Wert *accessdate* im *date*-Feld der Logdatei so aus:

```
P_iid.usage.ressources(iid).add_click(accessdate);
```

Schwierigkeiten kann manchen, dass die Zuordnung des angeforderten URL zu einer Ressource anhand ungünstiger Namensgebung nicht möglich ist. Bei einem exemplarischen Teil-URL von index.jsp?iid=432 für Ressource 432 ist das hingegen einfach möglich, da hier der Schlüssel für die Ressource direkt aus dem URL entnommen werden kann.

Bei den Datensätzen der Logdatei für die Suche ist das schwieriger. Hier wird typischerweise eine eigene Seite benutzt, der der Suchausdruck via Parameter übergeben wird, beispielsweise für die Suche nach XML.

```
search.jsp?searchterm=XML
```

Welche Ressourcen die Suche geliefert hat, ist daraus aber nicht ersichtlich. Daher muss bei der Logdateianalyse für jeden Datensatz mit der Such-URL im request-Feld die Suche intern durchgeführt werden. Alle oder ein Teil der gefundenen Ressourcen kann anschließend im Profil vermerkt werden. Zudem können dort auch die Suchausdrücke gespeichert werden, um über textuelle Informationen der vom Benutzer gesuchten Themen zu verfügen.

```
P_iid.usage.searches.add_search(searchterm, accessdate);
```

Durch die so vorgenommene Logdateianalyse sind die für Personalisierung interessanten Zugriffsdaten in die Benutzerprofile gelangt. Jetzt können noch aggregierte Werte wie die ihn Tabelle 6 dargestellten ermittelt werden. Die Tabelle erhebt jedoch keinen Anspruch auf Vollständigkeit.

Zwei Beispiele für diese aggregierten Daten sind die Summe aller Zugriffe auf eine Ressource durch einen Benutzer und die Verweildauer bei einer Ressource. Beide Kennzahlen können dazu

dienen, das Interesse eines Benutzers an einer Ressource auszudrücken. Eine niedrige Verweildauer und niedrige Besuchshäufigkeit der Ressource deuten auf geringes Interesse, während hohe Verweildauer und hohe Besuchshäufigkeit auf starkes Interesse schließen lassen. Allerdings ist diese Deutung fehlerbehaftet, da eine hohe Verweildauer auch daraus resultieren kann, dass der Benutzer zwischenzeitlich telefoniert oder eine andere Website besucht hat. Nichtsdestotrotz können die gewonnenen Daten grob als Anhaltspunkt dienen, wie das Interesse eines einzelnen Benutzers einzuschätzen ist.

Kennzahl	Formel
Abruf einer Ressource iid durch einen Benutzer uid (Click)	$c_i = iid$ mit $c_j.date$ = Datum des Abrufs
Sitzung eines Benutzers uid mit Abruf von Ressourcen iid (Sequenz)	$session_{uid} = (c_1, c_2, c_3, c_4,.., c_n.)$
Verweildauer eines Benutzers uid bei einer Ressource iid (Zeit)	$Verweildauer_{uid}(j) = c_{j+1}.date - c_j.date$
Anzahl Anmeldungen eines Benutzers uid	L_{uid} = # Anmelde-URL für Benutzer *uid*
Anzahl Sitzungen eines Benutzers uid	S_{uid} = # Sessionkennungen von Benutzer *uid*
Gesamtzahl der Abrufe einer Ressource iid durch einen Benutzer uid	$h_{uid}(iid) = \Sigma (c_j == iid)$
Durchschnittliche Verweildauer eines Benutzers uid bei einer Ressource iid	$\varnothing Verweildauer_{uid}(iid)$ $= (\Sigma Verweildauer_{uid}(iid)) / h_{uid}(iid)$

Tabelle 6 - Gewonnene Daten einzelner Benutzer bei der Logdateianalyse

Am Rande sei hier erwähnt, dass das Interesse an der Logdateianalyse neben diesen benutzerspezifischen Daten vor allem bei der Messung der Gesamtnutzung der Website liegt. So lassen sich die Gesamtabrufe aller Ressourcen ermitteln, die so genannten *Page Impressions*, und die Anzahl der Sitzungen, auch *Visits* genannt. Beide vermitteln einen Ausdruck von der Popularität der Website. Zudem kann analysiert werden, wie die Benutzer durch das Informationsangebot navigieren (Clickstreams) und ob es besonders häufig benutzte Pfade gibt. Weitere Informationen hierzu z. B. in [Mena00] oder in [Kloss01].

2.4 Verfahren zur Personalisierung

In den vorangegangenen Abschnitten wurde dargelegt, warum personalisierte Websites nützlich sein können, welche Elemente personalisiert werden und was als Datengrundlage verwendet wird. In den folgenden Abschnitten wird es um die Vorstellung der verschiedenen Verfahren gehen, mit denen Personalisierung möglich ist.

Wie in Kapitel 2.1.3 Alternative Begriffe beschrieben, gibt es unterschiedliche Begriffsschöpfungen für ähnliche Techniken. Im Folgenden werden die deutschsprachigen Titel bevorzugt, die in den meisten Publikationen verwendet werden (wenngleich auch dort nahezu ausschließlich in englischer Sprache).

2.4.1 Checkbox-Personalisierung

Mit Checkbox-Personalisierung ist die Anpassung der Arbeitsoberfläche und der Optik einer Website auf einfache Weise möglich. Die Auswahl gewünschter Themen, Inhalte, Layouts oder Farben erfolgt – der Name lässt es erahnen – per Checkbox. Dazu muss der Benutzer in den Einstellungen seines Benutzerkontos eine Reihe von Themen abspeichern. Neben Checkboxen sind auch andere Steuerelemente wie Listenansichten und Auswahllisten wie Comboboxen gebräuchlich. Entscheidend ist, dass der Benutzer die gewünschten anzuzeigenden Elemente des Systems direkt und explizit per Auswahl beeinflussen kann.

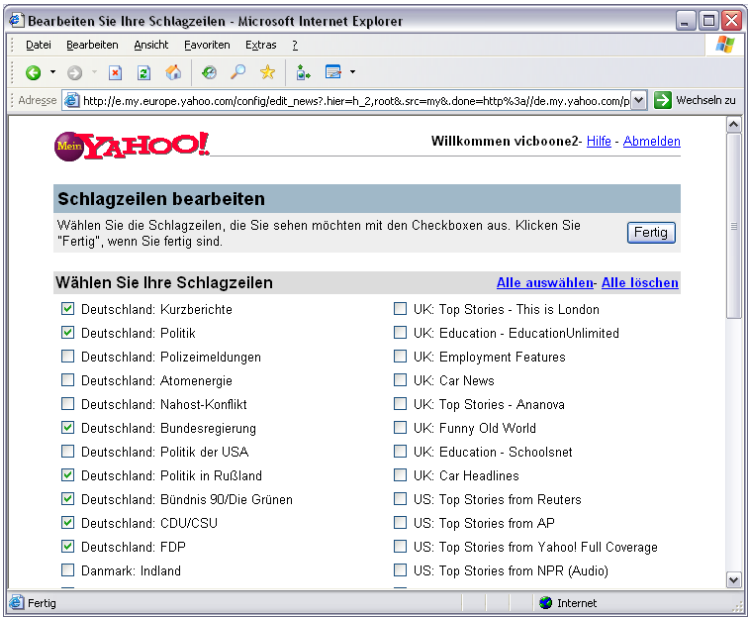

Abbildung 16 - Konfiguration der Nachrichtenschlagzeilen bei Mein Yahoo via Checkbox

Checkbox-Personalisierung wird in der Literatur auch als Anpassung oder Customisierung der Website für einzelne Benutzer bezeichnet (siehe [IO03]). Genereller wird auch von der Konfiguration von Software gesprochen. Große Softwarepakete wie Betriebssysteme oder Desktop-Textverarbeitungen bieten ebenfalls zahlreiche Möglichkeiten, nach den Wünschen des Benutzers angepasst zu werden. Zumeist ist das mit dem Ausfüllen von Einstellungen-Dialogen und darin enthaltenen Checkboxen möglich, beispielsweise für bevorzugte Tastaturkommandos, Farbeinstellungen oder aktivierte und deaktivierte Programmfunktionen.

Im Internet und World Wide Web hat sich Checkbox-Personalisierung als eine der ersten Formen von Personalisierung etabliert und wird dort beispielsweise bei den großen Internetportalen Yahoo als Mein Yahoo[27] oder bei Microsoft Network als MyMSN[28] angeboten. Auch Online-Shops wie Otto mit Mein Otto[29] oder Quelle mit Meine Quelle[30] bieten einfache Checkbox-Personalisierung an, beispielsweise zum themengerechten Beziehen von E-Mail-Newslettern oder Anlegen von Wunschlisten. Die Art und der Umfang der Konfigurierbarkeit sind jedoch von Anbieter zu Anbieter höchst unterschiedlich.

Die Installation für die Seitenbetreiber ist einfach, da die parametrisierte Generierung von Seiteninhalten lediglich von den jeweiligen Benutzerpräferenzen abhängig gemacht wird. Nicht ange-

[27] siehe „Mein Yahoo“ – http://de.my.yahoo.com
[28] siehe „MyMSN“ – http://www.msn.de/my.asp
[29] siehe „Mein Otto“ – http://www.otto.de
[30] siehe „Meine Quelle” – http://www.quelle.de

wählte Inhalte werden schlicht nicht angezeigt. Der Benutzer hat zudem die volle Kontrolle, da er jeden Inhalt und jedes zur Personalisierung angebotene Feature beeinflussen kann. Entsprechend ist die Ursache für die angezeigten und nicht angezeigten Inhalte sofort einsichtig, da sie dem gewählten Profil folgt. Besonders gut ist Checkbox-Personalisierung daher für die einfache und explizite Auswahl von Inhalten wie Börsenkursen, Wettervorhersagen und Nachrichten geeignet (siehe [Jen00]). Denn eine große Zahl an Möglichkeiten steht hier zur Verfügung, von denen der Benutzer nur einen kleinen Teil angezeigt bekommen möchte.

Der Hauptkritikpunkt an diesem Verfahren ist, dass bei einer großen Zahl an Informationen und Diensten für eine möglichst umfassende Konfiguration sehr viele Checkboxen zu aktivieren oder deaktivieren sind. Je genauer der Benutzer sein Profil spezifizieren möchte, desto mehr Angaben muss er über seine Interessen machen. Die Software bietet keinerlei Unterstützung für das Lernen der Benutzervorlieben, wie es andere, weiter unten behandelte Verfahren ermöglichen. Auch kann sich der Benutzer in seiner Privatsphäre gestört fühlen, wenn er ein umfangreiches Profil abgeben muss. Werden jedoch nur wenige Angaben gemacht oder ist überhaupt nur eine einfache Gliederung vorgesehen, ist das Profil zu ungenau und der Personalisierungserfolg bleibt gering, da Benutzer nicht ausreichend individualisierte Informationen erhalten.

Ein anderer Aspekt ist, dass die Checkbox-Personalisierung hochgradig von einem einmal spezifizierten Profil abhängig ist. Benutzer neigen jedoch dazu, ihr Profil nur selten anzupassen und arbeiten in Folge mit veralteten Angaben, wenn sich Geschmäcker oder Interessen im Laufe der Zeit verändert haben. Diese Form der Personalisierung sieht keine dynamischen, intelligenten Anpassungen des Benutzerprofils vor, beispielsweise durch die Beobachtung der Nutzungshäufigkeiten oder durch feingranulare Bewertungen einzelner Ressourcen. Durch mangelnde Anpassung an das Benutzerverhalten kann die Akzeptanz der personalisierten Website dann schwinden.

Problematischer ist noch, dass viele Benutzer gar nicht erst bereit sind, ein Profil in allen Einzelheiten zu spezifizieren, da der Aufwand je nach angebotenen Möglichkeiten erheblich ist. Werden zu wenige Angaben gemacht, ist der Nutzen der Personalisierung ebenfalls in Frage gestellt, da nicht ausreichend gefiltert wird. Manche Stimmen sagen daher, dass die Checkbox-Personalisierung als Konzept gescheitert sei, da sie nur von sehr wenigen Benutzern verwendet und akzeptiert wird (siehe [WCM01]). Nichtsdestotrotz ist das Verfahren als Basis- oder Einstiegslösung für Personalisierung sinnvoll, da Benutzerprofile mit persönlichen Daten und Präferenzen ohnehin in allen Verfahren benötigt werden und die Checkbox-Personalisierung sie als Grundlage bereitstellt.

2.4.2 Regelbasierte Personalisierung

Zentrales Element der regelbasierten Personalisierung oder genauer dem Regelbasierten Filtern, ist der Einsatz von Regeln, die beim Eintreten von bestimmten Merkmalen eines Benutzers zu Aktionen führen. Für jüngere, weibliche Benutzer können so beispielsweise anhand der Merkmale Alter und Geschlecht andere Informationen oder Produkte angeboten werden als für ältere, männliche Benutzer. Eine Regel hat die einfache Form

$$\text{wenn} \left(\textit{Bedingung} \text{ ist } \textit{wahr} \right) \text{dann } \textit{Aktion} \tag{2.1}$$

Bedingungen sind Abfragen von Attributen des Benutzerprofils wie demographische Daten (z. B. Alter, Geschlecht oder Wohnort), inhaltliche Präferenzen für bestimmte Themen oder bisheriges Verhalten im System (z. B. bisher angesehene Informationen und gekaufte Produkte). Auch aggregierte Daten wie die Zahl der Besuche innerhalb des letztes Monats, die Anzahl der Gesamtbesuche, der Zeitpunkt der letzten Anmeldung oder die Anzahl der Bestellungen können in Bedingun-

gen abgefragt werden. Bedingungen können darüber hinaus nach den Regeln der Booleschen Algebra zusammengestellt sein und so sehr detaillierte Abfragen des Benutzerprofils ermöglichen.

$$\text{wenn}\left(\textit{Bedingung}_1 \text{ ist } \textit{wahr} \wedge \textit{Bedingung}_2 \text{ ist } \textit{wahr} \right) \text{dann } \textit{Aktion} \tag{2.2}$$

Neben Daten des Benutzerprofils können auch kontextuelle Angaben wie die aktuelle Kategorie, die betrachtete Ressource, die letzten *n* angezeigten Seiten oder ähnliches einbezogen und so noch komplexere Bedingungen zusammengesetzt werden. Weiterhin können auch externe demographische Daten hinzugezogen werden, um Lücken in Benutzerprofilen zu schließen oder die Profilangaben zu erweitern. Beispielsweise können zu Ortsangaben Daten gekauft werden, die etwas über das verfügbare Einkommen aussagen (siehe [Mena00], Kapitel 7).

Treten die in der Bedingung einer Regel definierten Zustände für einen Benutzer innerhalb des aktuellen Kontextes ein, dann kann eine Aktion erfolgen. Bei Aktionen kann es sich um die Anzeige eines angepassten Textes, Offerten für Produkte oder Anpassung des Layouts handeln. Alternativ können Variablen gesetzt werden, die beispielsweise Wahrscheinlichkeiten für die Anzeige von bestimmten Nachrichtenartikeln repräsentieren und die an späterer Stelle zur Personalisierung der Anzeige ausgewertet werden.

$$\text{wenn}\left(\textit{Beruf} == \text{Politiker} \right) \text{dann Wk}\left(\textit{Nachrichtenanzeige} \right) = 95\% \tag{2.3}$$

Anhand ihrer Berufsgruppe könnten Benutzer unterschiedliche Ressourcen angeboten bekommen. Köche beispielsweise Rezepte und Serviertipps und Schreiner Bauanleitungen, Materiallisten und Modelle für Möbelstücke. Zugegeben ein einfaches Beispiel – werden die Regeln jedoch umfangreicher, kann das Informationssystem je nach Benutzerprofil spezifische Informationen bereitstellen und so ausgehend von wenigen Profildaten personalisieren.

Die Auswertung der Regeln im Programmcode ist sehr einfach, da programmiersprachliche Konstrukte benutzt werden können und die Daten der Attribute aus relationalen Datenbanken oder anderen Datenspeichern kommen. Die Performance ist entsprechend hoch. Beispielsweise kann so die Anzeige eines bestimmten Werbebanners für einen Benutzer von seinen Merkmalen und dem aktuellen Kontext abhängig gemacht werden.

```
public boolean canShowArticle( User user ) {

    if ( (user.age >= 18) && (user.jobs.contains( "developer" ))
        && (user.level > 10) && (this.notseen())
        && (this.category.contains( context.category )) )
    {
        return true;
    }
    else
    {
        return false;
    }

}
```

Abbildung 17 - Java-Programmcode zur Auswertung einer Regel

Der Codeausschnitt könnte einer Klasse zur Verwaltung eines Artikels entstammen. Das *context*-Objekt gibt Informationen über den aktuellen Kontext wieder, z. B. die aktuell vom Benutzer angezeigte Kategorie. Die Rückgabe ist entweder *true*, wenn der Artikel eingeblendet werden soll und *false*, wenn nicht.

Seinen Ursprung hat das Regelbasierte Filtern im klassischen Marketing. Hier versucht man anhand der Kaufhistorie aller Kunden und mittels Panelbefragungen das Verhalten der Kunden zu untersuchen. Ziel ist, das zukünftige Kaufverhalten eines einzelnen Kunden vorhersagen zu können, wozu sowohl die Art als auch die Anzahl der vermutlich zu kaufenden Produkte zählen. Kennt man die Präferenzen des einzelnen Kunden und zieht Kaufwahrscheinlichkeiten zu Rate, kann man geeignete Produkte besonders hervorheben. Verfolgt man dieses Bemühen weiter, führt es schließlich zu personalisierter Werbung, bei der alle Produkte auf den Kunden abgestimmt sind. Die Vorgehensweise beschränkt sich jedoch nicht nur auf den Marketing-Bereich. Auch in der Wissensverwaltung können in Abhängigkeit von vorangegangenen Informationsabrufen und dem allgemeinen Interesse des Kunden potentiell interessante Beiträge angeboten werden und so Zeit für den Sachbearbeiter gespart werden.

Welche Informationen, Dienstleistungen oder Produkte angeboten werden, soll mit Regeln ausgedrückt werden. Problematisch und auch der größte Nachteil des Regelbasierten Filterns ist die Schwierigkeit im Finden von geeigneten Regeln. Der Aufwand dazu kann beträchtlich sein, vor allem wenn ein sehr detailliertes Regelwerk erstellt werden soll. Die Aufgabe kann einerseits von menschlichen Experten erledigt werden, die über ein hohes Verständnis der jeweiligen Thematik und das nötige Domänenwissen verfügen. Beispielsweise kann ein Händler aus Erfahrung wissen, welche Produkte sich am besten zusammen verkaufen und die nötigen Regeln definieren.

Die Experten können aber andererseits auch maschinelle Hilfeleistung in Anspruch nehmen und so die Erstellung der Regeln vereinfachen. Typische Stichwörter sind hier Business Intelligence und Data Mining. Statistische Verfahren wie Regressionsanalyse sind aber ebenfalls möglich, soweit es der Anwendungszweck erlaubt. Im Internetbereich wird Web Usage Mining oder Web Log Mining als Spezialisierung des Data Mining verwendet, um Zugriffsdaten auf Websites und Kaufbewegungen in Online-Shops zu analysieren (siehe [Kloss01], [MAB00]). Die drei wesentlichen Verfahren dabei sind

- Segmentierung und Klassifikation
- Clusterbildung
- Assoziationsanalyse

Segmentierung und Klassifikation werden verwendet, um für einzelne Benutzer Verhaltensentscheidungen wie Kaufneigungen voraussagen zu können. Dazu wird das bisherige Verhalten der Benutzer, das beispielsweise in Logfiles gespeichert ist, zusammen mit ihren Benutzerprofilen von Neuronalen Netzen oder Entscheidungsbaumalgorithmen analysiert. Die Algorithmen liefern Regeln, die als Bedingungen die Merkmale der Benutzer verwenden und als Aktionen beispielsweise liefern, ob ein Shopbesucher kaufen wird oder welche Beträge er vermutlich ausgeben möchte (siehe [Mena00], Kapitel 1 und 3).

Die Clusterbildung dient zur automatischen Unterteilung der Benutzer in verschiedene Gruppen, die je nach Gruppenzusammensetzung unterschiedlich behandelt werden können. So kann es Cluster geben, die vorwiegend neue Benutzer enthalten und andere, die langjährige Nutzer umfassen, die das System intensiv benutzen. Auch eine Gliederung nach Themen ist möglich, so dass man Benutzer mit ähnlichen Interessen automatisch gruppiert. Neben Benutzern können auch Ressourcen in Cluster unterteilt werden. Bei der Darstellung einer Ressource lassen sich dann Ressourcen aus dem gleichen Cluster als ähnliche Ressourcen ausgeben (mehr zu Clusteralgorithmen siehe in Kapitel 3.2.2 Modellbasierte Algorithmen).

Mit der Assoziationsanalyse schließlich können gut Zusammenhänge zwischen Ressourcen und Inhalten aufgedeckt werden. Hierzu werden beispielsweise anhand von Zugriffsdaten gemeinsam oder nacheinander angezeigte Inhalte in Beziehung gesetzt. Die Annahme ist, dass Inhalte, die

zusammen abgerufen wurden, vermutlich auch miteinander verwandt sind. Zeigt ein Benutzer dann einen Artikel an, können ihm weitere ähnliche nahe gelegt und die Beziehungen in Regeln kodiert werden (siehe [Kloss01]).

Zwar liefern die maschinellen Regelextraktionsverfahren eine Hilfestellung bei der Definition der Regeln, aber letztlich müssen sie von menschlichen Experten geprüft werden. Der Aufwand ist generell hoch und das größte Manko des Regelbasierten Filterns. Änderungen im Benutzerverhalten können entsprechend nur schwerfällig berücksichtigt werden, da dies eine Neuausrichtung der Regeln zur Folge hat. Nützlich ist das Verfahren dennoch, da bereits mit relativ wenigen Informationen über einzelne Benutzer personalisierte Websites entwickelt werden können und der Aufwand in der Anzeigekomponente der Website durch die Verwendung der gut in Programmiersprachen zu implementierenden Regeln relativ gering ist.

2.4.3 Inhaltsbasierte Personalisierung

Formen des Inhaltsbasierten Filterns finden immer dann Anwendung, wenn der Inhalt von Ressourcen berücksichtigt wird. Entweder werden Ressourcen inhaltlich zueinander in Bezug gesetzt oder – was für Personalisierung relevanter ist – Benutzerprofile enthalten inhaltliche Präferenzen. Jedem Benutzer können dann solche Ressourcen angezeigt werden, die seinem inhaltlichen Profil entsprechen.

Der Benutzer kann seine Interessen wie bei der Checkbox-Personalisierung durch die Auswahl von Kategorien spezifizieren, vorausgesetzt die Ressourcen liegen in kategorisierter Form vor. Dann werden ihm Ressourcen geboten, die aus den von ihm gewählten Kategorien entstammen. Das ist der Top-Down-Ansatz. Der Nachteil ist jedoch die manuelle Konfiguration der Interessen durch den Benutzer, die aufwändig ist und daher zu geringer Anwenderakzeptanz führt (siehe Kapitel 2.4.1 Checkbox).

Eine zweite Variante ist, dass der Benutzer einzelne Ressourcen als für ihn nützlich oder weniger bewertet, was dem Bottom-Up-Ansatz entspricht. Das kann auf implizite oder explizite Weise erfolgen, wie in Kapitel 2.3 Benutzerprofile als Personalisierungsgrundlage gezeigt. Dann werden ihm Ressourcen vorgeschlagen, die den von ihm gewählten Ressourcen inhaltlich ähnlich sind. Hierzu ist keine Kategorisierung der Ressourcen nötig, aber die Ressourcenähnlichkeit muss ermittelt werden können.

In beiden Fällen kann die Wahl der Präferenzen auch gewichtet sein, so dass manche Themen stärker bevorzugt werden als andere. Dies kann durch numerische Bewertungen oder durch das Festlegen einer Reihenfolge ermöglicht werden. Die Ressourcen werden je nach Typ unterschiedlich verwaltet. So können Textdokumente durch Textanalyse und eine resultierende Menge von Schlüsselwörtern beschrieben werden. Binärdokumente wie Grafiken oder Musik werden durch Annotationen mit Attributen wie Größe, Titel oder Thema beschrieben. Formate wie das Resource Description Framework RDF können hierzu nützlich sein (siehe [LB02]). Während die Analyse von Textdokumenten recht gut maschinell möglich ist, müssen Binärdokumente häufig durch menschliche Sachbearbeiter beschrieben werden. Ein solches Attribut ist auch die Einstufung in nicht-jugendfreie bzw. jugendgefährdende Inhalte.

Der Abgleich zwischen Benutzerprofil und Ressourcenbestand erfolgt mit Methoden des Information Retrieval. Hier wird der Begriff des Filterns so beschrieben, dass die Benutzerpräferenzen mit den Dokumenten verglichen werden und die irrelevanten Ressourcen ausgefiltert werden. Übrig bleiben solche Elemente, die den Interessen entsprechen (siehe [BR99], Kapitel 2.3). Ein Prinzip ist das Vektormodell, das aufgrund seiner Einfachheit schnell umgesetzt werden kann. Dabei wird

jede Ressource *iid* als ein Vektor d_{iid} von Wörtern behandelt. Ebenso wird eine Anfrage *qid* an den Datenbestand *I* als Vektor q_{qid} von Wörtern dargestellt und zur Bestimmung der Ähnlichkeit von Anfrage und verfügbaren Dokumenten kann die Kosinusfunktion verwendet werden, die das Skalarprodukt und die Normen der Vektoren verwendet.

$$sim\left(\vec{d}_{iid}, \vec{q}_{qid}\right) = \frac{\vec{d}_{iid} \cdot \vec{q}_{iid}}{\left|\vec{d}_{iid}\right| \cdot \left|\vec{q}_{iid}\right|} \tag{2.4}$$

Als Elemente der Vektoren stehen Gewichte für die Wörter im Dokument oder der Anfrage. Das können einerseits binäre Werte sein mit einer 1, wenn das Wort im Dokument auftaucht und 0, wenn es nicht auftaucht. Andererseits können hier auch Worthäufigkeiten oder die Inverse Dokumentfrequenz stehen, wie im Information Retrieval als Optimierung üblich (siehe [BR99], Kapitel 2.5.3 oder [MMN01]).

Die Idee für die Personalisierung ist, dass die Präferenzen eines Benutzers *uid* als eine Anfrage und damit als Vektor von Wörtern betrachtet werden. Vergleicht man alle oder einen Teil der Dokumente des Datenbestandes *I* mit dem Wortvektor des Benutzers, erhält man sortiert nach dem Ähnlichkeitswert *sim* eine Reihenfolge der Dokumente, wobei diejenigen, die den Präferenzen sehr gut entsprechen, an erster Stelle stehen.

Statt den Wortvektoren kann für Ressourcen mit attributbasierten Beschreibungen auch eine Ähnlichkeit über den Attributen bestimmt werden. Das Prinzip ist ähnlich. Die Personalisierung über Kategorien arbeitet genauso, wenn man die Kategorien als Attribute zu einer Ressource auffasst, wobei eine Ressource auch mehreren Kategorien angehören kann. Dann sind dem Benutzer alle Inhalte zu liefern, die den Kategorien entsprechen, die er gewählt hat.

Nachteil der inhaltsbasierten Filterung ist, dass der Inhalt der Dokumente analysiert und z. B. mittels Schlüsselwörtern abstrahiert werden muss, um Ähnlichkeiten zu ermitteln. Für Textdokumente gelingt dies mit den Verfahren des Information Retrieval auf automatische Weise recht gut, bei Bildern, Filmen oder Musik ist es aber weitaus schwieriger bis nahezu unmöglich. Hier müssen dann menschliche Experten eine Verschlagwortung oder Kategorisierung vornehmen, was äußerst personalintensiv ist. Filme beispielsweise müssen mit Attributen wie Genre, Sprache und Länge sowie einer Inhaltsbeschreibung versehen werden, mit denen inhaltsbasiert gearbeitet werden kann.

Bei der Nachrichtensuchmaschine Paperball[31], die Nachrichten verschiedener Tageszeitungen auf ihren Online-Varianten durchsucht, indiziert und aggregiert, ist die Textanalyse hingegen gut möglich. Als Benutzer kann man Suchanfragen mit booleschen Ausdrücken durchführen, auf die hin die Suchmaschine alle passenden Zeitschriftenartikel liefert. Das Angebot ermöglicht zudem eine Personalisierung mit Kategoriewahl und die Wahl von präferierten Tageszeitungen für verschiedene Kategorien. Zudem ist das Speichern von Anfragen möglich – gewissermaßen Bookmarks – die bei Bedarf per Mausklick einfach aufgerufen werden können. Abbildung 18 zeigt eine solche gespeicherte Abfrage für eine Suchanfrage, die mehrere Suchbegriffe miteinander verknüpft.

[31] siehe http://www.paperball.de

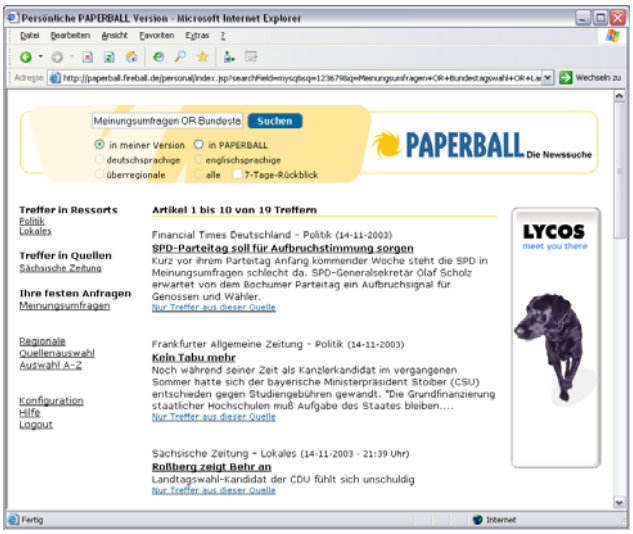

Abbildung 18 - Suchmaschine und Aggregator für Nachrichten (Paperball)

Günstig am Inhaltsbasierten Filtern ist allerdings, dass die Konfiguration der Benutzerinteressen sehr feingranular möglich ist und bei Online-Shops wirkliches 1-zu-1-Marketing realisierbar wird, da einem Benutzer Inhalte sehr individuell präsentiert werden können. Zudem verlangt dieses Verfahren den Benutzern vergleichsweise wenig Arbeit ab – der Aufwand liegt eher beim Systembetreiber, der den Datenbestand pflegen muss.

2.4.4 Kollaboratives Filtern

Personalisierung mit Kollaborativen Filtern beruht auf der Grundannahme, dass Ressourcen, die einem Benutzer gefallen genauso ähnlichen Benutzern gefallen. Die Ähnlichkeit zwischen Benutzern wird anhand von Ressourcenbewertungen ermittelt, so dass Benutzer mit gleichen Bewertungen für eine Menge von Ressourcen als ähnlich eingestuft werden. Einem Benutzer können so Informationen oder Produkte empfohlen werden, die von ähnlichen Benutzern als interessant oder gut beurteilt wurden.

Das Verfahren hat viele Vorteile gegenüber anderen Personalisierungstechniken. Durch die sehr feine Festlegung des Benutzerprofils durch Bewertungen einzelner Ressourcen wird wirkliches 1-zu-1-Marketing möglich, denn jeder Benutzer erhält nicht nur auf eine Gruppe, der er angehört, abgestimmte Informationen – wie z. B. beim Regelbasierten Filtern – sondern tatsächlich nur für ihn zusammengestellte.

Zudem ist beim Kollaborativen Filtern nur ein kleiner Aufwand für den Systembetreiber zur Einrichtung eines entsprechenden Systems nötig, da die Personalisierung über Bewertungen erfolgt, die von den Benutzern vorgenommen werden müssen. Das ermöglicht auch die Personalisierung schwer maschinell zu analysierender Ressourcen wie Grafiken, Filme oder Musik. Zudem spielt bei den letztgenannten Ressourcentypen auch der Geschmack des einzelnen Benutzers eine Rolle, der mit anderen Personalisierungsverfahren nur schwer zu erfassen ist. So kommt es dazu, dass beim Kollaborativen Filtern mitunter Empfehlungen generiert werden, die eher unerwartet sind, die für den Benutzer aber einen hohen Nutzwert darstellen. [RK02] zeigt viele Vorzüge Kollaborativen Filterns nebst Fallstudien auf.

Nachteilig ist allerdings, dass Bewertungen in ausreichender Zahl vorhanden sein müssen, damit das Verfahren gut arbeiten kann. Benutzer neigen jedoch eher dazu, nur wenige Ressourcenbewertungen abzugeben, so dass die Profile unvollständig bleiben können. Insbesondere bei großen In-

formationssystemen mit sehr vielen Datensätzen ist das problematisch, da dann entweder keine ähnlichen Benutzer bestimmt werden können oder viele Ressourcen ohne Bewertung vorliegen. Im Optimierungskapitel dieser Arbeit wird mit Lösungsansätzen näher hierauf eingegangen (siehe 4 Optimierung).

In Kapitel 3 Kollaboratives Filtern wird detailliert auf das Verfahren eingegangen und dabei Fallbeispiele und Algorithmen nebst Problemfeldern vorgestellt.

2.4.5 Hybride Verfahren

Da jedes der in den vorangegangen Abschnitten beschriebenen Personalisierungsverfahren individuelle Stärken und Schwächen hat, gibt es in der Forschung zu diesem Thema eine Richtung, die eine Optimierung der Personalisierung durch die Kombination von verschiedenen Verfahren anstrebt. Hierbei werden häufig Inhaltsbasierte und Kollaborative Filter miteinander verknüpft (siehe z. B. [Bau99] oder [FEBS02]). Aber auch Kombinationen anderer Verfahren sind möglich, so beispielsweise von Regelbasierten und Inhaltsbasierten Filtern.

Eine einfache Regel könnte zum Beispiel besagen, dass ein Benutzer mit einem bestimmten Beruf wie Softwareentwickler mit hoher Wahrscheinlichkeit gerne Artikel zu den Themen Java, Objektorientierung, Softwareentwurf und Design Pattern mag. Im Benutzerprofil könnten diese Themen als inhaltliche Präferenzen eingetragen sein und bei der inhaltsbasierten Personalisierung berücksichtigt werden.

$$\begin{aligned} &\text{wenn}\left(\textit{Beruf} == \text{Politiker} \right) \\ &\text{dann } \textit{Themenvoreinstellung} := \{\text{Java}, \text{Objektorientierung}, ...\} \end{aligned} \tag{2.5}$$

Der Vorteil an dieser Kombination wäre, dass der Benutzer nur die einzige Angabe zu seinem Beruf machen und nicht aus einer umfangreichen Liste von Themen wählen muss. Anhand der Erfahrungen mit anderen Benutzern des gleichen Berufes weiß der Regeldesigner, welche Themen diese Benutzer bevorzugen. Der Effekt ist also eine Voreinstellung präferierter Themen und ein schnellerer Personalisierungserfolg bei weniger Angaben durch den Benutzer. Bei einem rein Inhaltsbasierten Filter müsste der Benutzer erst eine langwierige Themenwahl über sich ergehen lassen und bei einem rein Regelbasierten wären die Auswahl von Themen zu starr und Änderungen durch den Benutzer nicht möglich.

2.4.6 Einbeziehung menschlicher Experten

Die Erfahrungen menschlicher Experten in der jeweiligen Wissensdomäne des Informationssystems können unterschiedlich genutzt werden. Bei Inhalts- oder Regelbasierten Filtern können sie sogar grundsätzlich nötig sein, um Informationen manuell zu klassifizieren oder Regeln aufzustellen.

Bei allen Verfahren können menschliche Experten aber dafür sorgen, dass die Personalisierungsergebnisse optimiert werden. Dazu ist einerseits eine Erfolgsmessung mit den entsprechenden Kennzahlen nötig – z. B. Umsatzsteigerungen in einem Online-Shop durch Einführung von Personalisierung oder eingesparte Arbeitszeit bei Unternehmensportalen. Andererseits können anschließend Korrekturen an den Personalisierungsverfahren vorgenommen werden.

Bei Regelbasierten Filtern können beispielsweise die Regeln anhand neuer Erfahrungen modifiziert und bei Inhaltsbasierten Filtern eine neue Zusammenstellung von Inhaltskategorien vorgenommen werden. Bei aus Betreibersicht automatischen Verfahren wie dem Kollaborativen Filtern

ist das schwieriger. Aber auch hier könnte durch Experten eine interne Neujustierung der Bewertungen erfolgen, wie sie im späteren Kapitel 4.1 Verbreiterung der Datenbasis angedacht wird.

2.4.7 Vergleich der Verfahren

Das perfekte Verfahren zur Personalisierung gibt es leider nicht. Genauso wenig wie Personalisierung selbst ein Allheilmittel für die Überwindung von Informationsüberflutung oder zur Steigerung von Umsätzen ist. Vielmehr hängt die Wahl des richtigen Personalisierungsverfahrens von den Anforderungen des Informationssystems und den angebotenen Ressourcen ab. Ein Online-Shop für Bekleidung hat andere Vorgaben als ein Informationsportal für Softwareentwickler.

Nichtsdestotrotz sind in folgender Vergleichsmatrix die vier hier beschriebenen Verfahren gegenübergestellt und nach verschiedenen Kriterien bewertet worden. Die Bewertung ist relativ grob und drückt mehr eine Tendenz als eine absolute Bestimmung aus. Verwendet wurde eine Skala von sehr niedrig, niedrig, mittel, hoch bis sehr hoch.

Kriterium	Checkbox	Regelbasiert	Inhaltsbasiert	Kollaborativ
Anwenderkomfort	mittel	sehr hoch	mittel bis hoch	hoch
Verständlichkeit	sehr hoch	mittel	hoch	niedrig
Anwendernutzen	mittel	mittel	sehr hoch	sehr hoch
Betreibernutzen	niedrig	sehr hoch	hoch	sehr hoch
Technischer Aufwand Website	niedrig	sehr niedrig	mittel	mittel
Technischer Aufwand Modell	sehr niedrig	mittel	hoch	niedrig oder sehr hoch
Benötigtes Domänenwissen	niedrig	sehr hoch	hoch	niedrig
Datenschutz	mittel	niedrig	hoch	hoch

Tabelle 7 - Vergleich der Personalisierungsverfahren

Beim Kollaborativen Filtern ist beispielsweise der Anwendernutzen sehr hoch, da häufig nützliche Empfehlungen ausgesprochen werden. Der technische Aufwand zur Umsetzung kann jedoch je nach Algorithmus unterschiedlich ausfallen. Bei Regelbasierten Filtern ist der Anwendernutzen nicht unbedingt hoch, wenn die Regeln für Gruppen von Benutzern statt für einzelne Benutzer erstellt wurden. Dann ist die Personalisierung nicht stark genug auf die Anwenderpräferenzen ausgerichtet. Der Aufwand für das Aufstellen der Regeln ist jedoch mittel hoch, wenn beispielsweise automatisierte Hilfsmittel genutzt werden. Natürlich können in der jeweiligen Implementierung abweichende Bewertungen für die Verfahren auftreten. Von der grundsätzlichen Perspektive her ergibt sich jedoch die Tendenz aus dieser Matrix.

2.5 Personalisierung vs. Datenschutz

Viele Internetnutzer stehen der Preisgabe ihrer persönlichen Daten wie Name, E-Mailadresse, Telefonnummer oder Geburtsdatum an Websites skeptisch gegenüber. Die Ursache dafür ist, dass die Nutzer einen Missbrauch ihrer Daten vermuten. Unbegründet sind solche Annahmen tatsächlich

nicht, wie man am Phänomen des E-Mail-Spams gut sehen kann (siehe [Mue99], [SF98]). Gibt man beispielsweise seine E-Mailadresse auf einer Website an, kann es durchaus passieren, dass man früher oder später ungefragt Werbeemails erhält, weil der Websitebetreiber entweder selbst Werbung verschickt oder die Adressen weiterverkauft. Hat er bei der Eingabe der E-Mailadresse eine Einwilligung vom Anwender durch einen entsprechenden Informationstext oder die Anwahl einer Checkbox eingeholt, dann ist diese Vorgehensweise rechtens. Schließlich hätte der Anwender auch auf die Einwilligung verzichten können. Weitere Verfahren wie das Prinzip, bei dem eine explizite Bestätigung einer Anmeldung z. B. durch die Antwort auf eine E-Mail des Betreibers nötig ist, sind auch möglich und weitaus seriöser.

Bedenklich ist, wenn die angegebenen Daten missbraucht werden. Bei der Weitergabe von E-Mailadressen und einsetzendem Spam ist das zumindest ärgerlich. Bei anderen Daten wie Kaufgewohnheiten, demographischen Daten oder gesundheitlichen Befunden kann die Weitergabe sogar gefährlich sein und massiv gegen den Datenschutz verstoßen. Mögliche Szenarien sind Diskriminierungen in Bezug auf Beruf, Gesundheit, persönliche Vorlieben und die Vergabe von Krediten bei den Stellen, die an die eigentlich nicht für sie bestimmten Daten gelangt sind. Bei fehlerhafter Datenübermittlung können sich besonders unangenehme Auswirkungen ergeben, wenn der angenommene Sachverhalt nicht mit den tatsächlichen Umständen des Betroffenen übereinstimmt (siehe [BS99] und Kapitel 1.4 Schutz der Privatsphäre).

Benutzer von Websites sind daher generell vorsichtig, ihre Daten preiszugeben, wobei der Grad an Vorsicht steigt, je intimer die Daten sind. Manche Anwender sind allerdings abhängig von den persönlichen Befindlichkeiten auch wesentlich freizügiger bei der Angabe ihre Daten. Hier spielen zudem kulturelle und regionale Unterschiede eine Rolle.

2.5.1 Gegensätzlichkeit Personalisierung und Privatsphäre

Das Problem des Datenschutzes für Personalisierung und Marketing ist, dass gute Kenntnisse jedes Benutzers nötig sind, um Informationen und Produkte individuell aufbereiten und anbieten zu können. Eine entsprechend umfangreiche Selbstbeschreibung der Websitebenutzer ist also von Vorteil, stößt aber bei diesen auf Skepsis. Mit der Erfassung und Analyse des Benutzerverhaltens wird der Benutzer zudem detailliert inspiziert und umfangreiche Profile über Vorlieben und Interessen können erstellt werden. Personalisierung und Datenschutz stehen daher in einem gewissen Konflikt zueinander, für den Lösungsmöglichkeiten gefunden werden müssen.

Ein Problem am Rande dieser Thematik ist, dass mögliche potentiell interessante Inhalte, die aber nicht im Benutzerprofil erfasst sind, in personalisierten Informationssystemen ausgeblendet werden. Manche Anwender befürchten daher, dass für sie interessante Informationen unterdrückt werden und wollen keine Personalisierung zulassen. Inwiefern dieser Sachverhalt tatsächlich kritisch ist oder ob nicht die Reduzierung der Informationsüberflutung vorteilhafter ist, muss sicherlich der Einzelne selbst entscheiden. Für den Anwender bestimmte Informationen und Erklärungen zur Arbeitsweise der Personalisierung können hier allerdings helfen.

2.5.2 Lösungsversuche

Mit einer Reihe von Maßnahmen können Datenschutzbedenken aus dem Weg geräumt werden. Generell ist es sinnvoll, mit Informationen über die Arbeitsweise des Informationssystems und der betriebenen Personalisierung Vertrauen zu schaffen. Wenn der Anwender weiß, was mit seinen Daten passiert, vertraut er einer entfernten Website eher, als wenn er über die Vorgänge im Dunkeln gelassen wird.

Hierzu können technische Standards wie die Richtlinien des Platform for Privacy Preferences Projektes (P3P) genutzt werden, die detailliert beschreiben, wie mit den persönlichen Daten des Anwenders umgegangen wird (siehe Kapitel 1.4.2). Einfache Erläuterungstexte an gut erreichbarer Stelle auf der Website bieten Vergleichbares. Eine Garantie, dass tatsächlich wie angegeben verfahren wird, hat der Anwender letztlich natürlich nicht.

Ein Impressum mit Angaben zur physischen Erreichbarkeit des Websitebetreibers trägt ebenfalls zur Vertrauensbildung bei und ermöglicht den Kontakt in Konfliktfragen. Allerdings ist die Bereitstellung eines Impressums ohnehin eine rechtliche Pflicht und keine optionale Variante. Die Übertragung von persönlichen Daten und Zugangskennungen durch Verschlüsselung (primär durch SSL[32]) gehört ebenfalls zum guten Ton und stärkt das Vertrauen des Anwenders in die Prozesse des Informationssystems.

Ferner kann eine Einhaltung der gemachten Datenschutzvorgaben durch Dritte überprüft werden. Hier gibt es beispielsweise mit TRUSTe[33] eine Organisation, die ein Logo bereitstellt, das auf Websites genutzt werden kann, die den Vorgaben der Organisation entsprechen und sich an Datenschutzlinien halten. Eine andere Vereinigung, die sich dem sicheren Einkaufen verschrieben hat, ist Trusted Shops[34].

Die bisher genannten Vorschläge waren recht allgemeiner Natur, können aber auch für die Stärkung des Vertrauens in personalisierte Websites genutzt werden. Damit könnte beispielsweise eine technische Variante unterstützt werden, die nur anonyme Daten im Informationssystem speichert. Ein Benutzer könnte durch einen Schlüsselwert identifiziert werden, an den die gespeicherten Präferenzen im Profil gekoppelt sind – beispielsweise die Bewertungen von Ressourcen für das Kollaborative Filtern. Persönliche Angaben wie Name und Kontaktdaten müssen jedoch nicht zusammen mit den Präferenzen und Vorlieben gespeichert werden.

Noch weitgehender wäre, gar keine Benutzerdaten im Informationssystem abzulegen. Um dennoch Personalisierung zu ermöglichen, könnten die Daten auf dem Arbeitsplatzrechner des Benutzers vorliegen. Sie würden dann bei jedem Zugriff z. B. mit Cookies übertragen und könnten dort temporär zur Personalisierung herangezogen werden. Modifikationen am Benutzerprofil würden zurück zum Benutzer gesandt, aber nicht in der Website gespeichert. Alternativ könnte die Personalisierung in Form einer Filterung erst auf dem Arbeitsplatzrechner stattfinden und es müsste gar kein Datenaustausch erfolgen. Verfahren wie dem Kollaborativen Filtern, die auf den Vergleich von Präferenzen verschiedener Benutzer ausgelegt sind, würde sich dieses Vorgehen allerdings verschließen, da nur die individuellen Daten lokal bekannt sind und nicht die anderer Benutzer.

Letztlich würde das aber den Personalisierungserfolg schmälern und den Anwender weniger zufrieden stellen. Technische Umsetzungsschwierigkeiten aufgrund mangelnder Unterstützung oder knapper Rechenzeit auf dem Arbeitsplatzrechner sowie Bandbreitenbeschränkungen sind weitere Probleme. Zudem ist auch anzuzweifeln, ob Websitebetreiber tatsächlich bereit sind, auf Benutzerprofile zu verzichten, da diese für sie z. B. für Marketingmaßnahmen einen deutlichen Wert darstellen. Einem Missbrauch von personenbezogenen Daten würde jedoch vorgebeugt.

[32] Eine übersichtliche Zusammenfassung zu SSL siehe http://en.wikipedia.org/wiki/Secure_Sockets_Layer
[33] Die Website der TRUSTe-Organisation unter https://www.truste.org
[34] Informationen zu Trusted Shops siehe http://www.trustedshops.de/

3 Kollaboratives Filtern

3.1 Motivation zur Idee des Kollaborativen Filterns

Die Personalisierung mit Kollaborativen Filtern erfolgt auf gänzlich anderem Wege als mit den typischen Verfahren wie inhalts- und regelbasierter Personalisierung, bei denen der Inhalt und die Zusammenhänge zwischen Ressourcen sowie die Eigenschaften von Benutzern analysiert werden (siehe Kapitel 2.4 Verfahren zur Personalisierung). Stattdessen wird die Kollaboration von Benutzern als Mittel verwendet, um dem einzelnen Benutzer Inhalte und Produkte anzubieten, die ihm vermutlich gefallen und oder ihn interessieren werden.

Die grundsätzliche Annahme beim Kollaborativen Filtern ist, dass Ressourcen, die einem Benutzer gefallen auch ähnlichen Benutzern gefallen werden. Kernelement ist daher die Betrachtung der Ähnlichkeiten zwischen Benutzern, wofür in den letzten zehn Jahren verschiedene Verfahren entwickelt und mit Erfolg eingesetzt wurden. Die populären Methoden und Algorithmen werden dazu später in 3.2 Eingesetzte Verfahren vorgestellt.

Der einzelne Benutzer drückt in Form von Bewertungen aus, welche Ressourcen des Informationssystems ihm gut gefallen oder für ihn nützlich sind und welche nicht. Eine Analyse der Inhalte ist dabei nicht erforderlich. Daher ergeben sich durch den Einsatz von Kollaborativen Filtern sowohl für den Websitebetreiber als auch für den Anwender eine Reihe von Vorteilen, durch die die Personalisierung im Internet optimiert werden kann.

- Neben Texten können Grafiken, Video, Musik und andere schwer automatisch zu klassifizierende Medien erfolgreich personalisiert werden.
- Die formal schwerlich quantifizierbaren Größe des Geschmackes und der Qualität können in die Personalisierung einfließen, beispielsweise für Präferenzen von Filmen.
- Der Systembetreiber hat vergleichsweise wenig Arbeit, da er selbst keine inhaltliche Klassifizierung vornehmen muss und die Arbeit des Bewertens auf die Schultern der Benutzer verlagert wird.
- Die Installation in bestehende Systemen kann gut erfolgen, da Kollaboratives Filtern in Form einer Blackbox unabhängig arbeiten kann, wenn Bewertungen von Ressourcen bereitstehen.
- Es erfolgt eine Personalisierung nach individuellen Präferenzen eines jeden Benutzers und nicht anhand der Zugehörigkeit zu einer Gruppe. Zudem sind die erstellten Benutzerprofile sehr genau auf den Benutzer abgestimmt, da eine Bewertung von einzelnen Ressourcen und nicht groben Themen oder Kategorien erfolgt.
- Neben der Personalisierung von Inhalten lassen sich durch Kollaboratives Filtern auch Menschen ähnlicher Interessen zusammenführen. Das ist für das Finden von Partnern und Experten nützlich.

Der Begriff des Kollaborativen Filterns tauchte in der Literatur erstmalig mit der Arbeit „Using Collaborative Filtering to Weave an Information Tapestry“ auf (siehe [GNOT92]). Ziel war dort der gefilterte Abruf von Newsgroup-Nachrichten, um für einen Benutzer interessante Diskussionen mitverfolgen zu können. Zum Einsatz kam eine SQL-ähnliche Anfragesprache, mit der Benutzer

die Inhalte aktiv filtern konnten. Im englischsprachigen Original wurde dazu der Begriff „Collaborative Filtering“ geprägt.

Zwei Jahre später folgte mit „GroupLens: An open architecture for collaborative filtering of netnews“ eine weitere Arbeit, die ebenfalls die Filterung von Newsgroup-Nachrichten beschrieb, aber jetzt eine automatische Variante des Kollaborativen Filterns einsetzte (siehe [RISBR94]). Das dort beschriebene Konzept der Informationsfilterung durch die Betrachtung ähnlicher Benutzer wurde fortan in verschiedenen Arbeiten auf das World Wide Web übertragen und wird von Forschern und Technikern weiterhin erweitert, verbessert sowie evaluiert. Neben dem Begriff „Collaborative Filtering“ wird alternativ auch „group filtering“ und „social filtering“ verwendet.

3.1.1 Grundlegende Arbeitsweise

Die Grundidee des Kollaborativen Filterns ist die Annahme, dass Benutzer, die ähnliche oder gleiche Bewertungen für Ressourcen abgeben, auch einen ähnlichen Geschmack haben. Anders ausgedrückt kann man sagen, Anwender, die Interesse an den gleichen Ressourcen haben, haben auch ähnliche Interessenschwerpunkte. In Folge kann man einem Benutzer Ressourcen zur Ansicht vorschlagen, die er zwar selbst noch nicht gesehen oder gar beurteilt hat, aber die ihm gleichende Benutzer bereits gesehen, beurteilt und für gut empfunden haben.

Insgesamt kann man so für alle Ressourcen eines Informationssystems eine Aussage darüber treffen, ob sie einen Benutzer möglicherweise interessieren oder ihm gefallen. Natürlich sind die Aussagen nicht frei von Fehlern und so wird es vorhergesagte Bewertungen geben, die nicht dem Empfinden des Benutzers entsprechen. Hierin liegt dann Optimierungspotential für kollaborative Filteralgorithmen. Nichtsdestotrotz sind die berechneten Vorhersagen weitaus besser als eine zufällige Ordnung der Ressourcen und eine zugrunde liegende zufällige Bewertung. Vergleiche zur Güte verschiedener kollaborativer Filteralgorithmen und zwischen zufälliger Auswahl wurden von verschiedenen Stellen angestellt (siehe u. a. [BHK98]). Weitere Informationen zur Qualitätsmessung finden sich unter Kapitel 3.3 Erfolgsmessung.

Elementar für Kollaboratives Filtern ist die Ausrichtung auf den Benutzer und nicht auf die zu verwaltenden Ressourcen. Im Extremfall bedeutet das, dass dem System über den Inhalt der Ressourcen nichts bekannt sein muss – im Gegensatz zu anderen Systemen, wie den inhaltsbasierten, bei denen eine gute Kenntnis des Datenbestandes nötig ist. Stattdessen werden Benutzerbewertungen zu den Ressourcen abgefragt und auf dieser Datenbasis gearbeitet.

Hierzu gibt es den Ansatz, dass der Benutzer eine explizite Bewertung zu einem Item angibt. Das kann ein diskreter numerischer Wert wie in Abbildung 19 dargestellt sein. Es werden jedoch auch andere Formen angewendet, wie eine kontinuierliche, numerische Skala oder textuelle Angaben wie von „sehr schlecht“ bis „sehr gut“. Innerhalb des Softwaresystems erfolgt jedoch eine Abbildung auf einen numerischen Wert.

Abbildung 19 - Steuerelement zur Bewertung mittels diskreter, numerischer Werte

Ein zweiter Ansatz sieht vor, dass der Benutzer implizit Wertungen durch sein Verhalten innerhalb des Informationssystems abgibt. Beispielsweise kann gezählt werden, wie oft er eine Ressource ansieht, wie oft er einen Button anklickt oder ob er nach einer Ressource sucht. Aus den gezählten Werten lassen sich Bewertungen über das Gefallen der betrachteten Ressourcen ableiten. Auch Warenkörbe und Bestellungen in Online-Shops können als implizite Bewertungen genutzt werden

(siehe Kapitel 2.3 Benutzerprofile als Personalisierungsgrundlage). Darüber hinaus sind Mischformen zwischen beiden Varianten möglich.

Allen Verfahren gemein ist, dass der Benutzer auf die eine oder andere Weise Bewertungen zu den angebotenen Ressourcen abgibt und vorzugsweise sogar möglichst viele Bewertungen vorzuweisen hat, da dadurch ein detailliertes Bild von seinen Interessen entsteht. Bei nur wenigen abgegebenen Bewertungen ist das Interessensprofil des Benutzers hingegen unscharf. Welche Mindestzahl von Bewertungen zu einem qualitativ hochwertigen Profil führen, ist jedoch von System zu System verschieden und hängt von der Art und Anzahl der Ressourcen ab. Bei manchen Informationssystemen wird bei der Neuanmeldung eines Benutzers eine Reihe von Ressourcen vorgelegt, die der Benutzer zunächst bewerten muss, um mit dem System zu arbeiten. Dadurch ergibt sich ein initiales Profil des Benutzers.

Jeder Benutzer verfügt also idealerweise über einen ausreichend großen Satz von Ressourcenbewertungen. Vergleicht man verschiedene Benutzerprofile miteinander, dann werden manche Benutzer eine Reihe von Ressourcen ähnlich gut bewerten. Andere Benutzer werden hingegen andere Mengen von Ressourcen als gut bewerten und wieder andere Gruppen werden genau diese Ressourcen als schlecht ansehen. So gibt es innerhalb der Benutzer verschiedene Gruppen, die ähnliche Präferenzen haben. In der Fachliteratur zu Kollaborativem Filtern wird auch von Nachbarschaft gesprochen, da man Benutzer mit ähnlichen Interessen als Nachbarn oder benachbart bezeichnet.

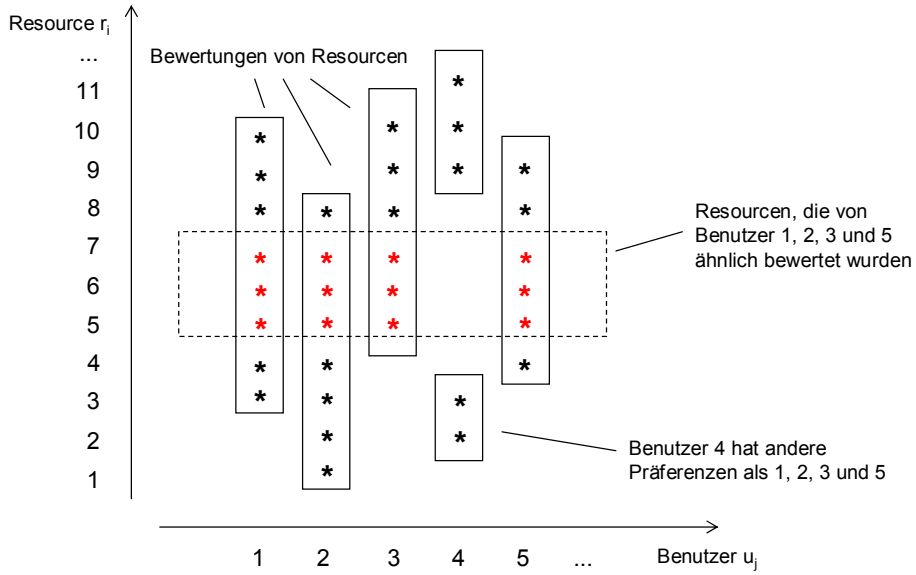

Abbildung 20 - Überlappung von Benutzerbewertungen

Hat ein Benutzer für eine Ressource keine Bewertung abgegeben – es fehlen also Daten an dieser Stelle – dann kann möglicherweise mit den Bewertungen ähnlicher Benutzer, die die fehlende Ressource bewertet haben, ausgeholfen werden. Dem Benutzer mit fehlender Bewertung wird durch das Kollaborative Filtern eine Bewertung berechnet, die auf den Bewertungen seiner Nachbarn basiert. Und hierin liegt die grundlegende Arbeitsweise des Kollaborativen Filterns begründet: Vorhersage von neuen Benutzerbewertungen durch zeitlich vorher getätigte Bewertungen.

Beispiel

> Ein einfaches Beispiel illustriert die Arbeitsweise des Kollaborativen Filterns. Beispielsweise könnten in einem fiktiven Informationssystem die Benutzer Christian, Daniel, Ralf und Stefa-

nie unterschiedliche Ressourcen auf einer Skala von 1 mit geringem bis 5 mit hohem Interesse bewertet haben.

	Christian	Daniel	Ralf	Stefanie
Einführung in XML	5	-	4	4
Cookies und Sicherheit	-	4	5	-
Datenbanken	2	5	2	4
Visualisierung	3	-	3	3
Java-Programmierung	-	1	-	2
Textverarbeitungstricks	5	1	4	-

Abbildung 21 - Exemplarische Ressourcenbewertungen

Anhand der gekauften Bücher wird die Ähnlichkeit der Benutzer bestimmt. In der Tabelle haben Christian und Ralf zwar nicht alle Ressourcen gemeinsam bewertet, aber doch für „Einführung in XML“, „Datenbanken“, „Visualisierung“ und „Textverarbeitungstricks“ sehr ähnliche Bewertungen abgegeben. Sie sind daher als ähnliche Benutzer zu sehen. Entsprechend kann Christian die bislang unbewertete Ressource „Cookies und Sicherheit“ empfohlen werden, weil sie Ralf sehr hoch bewertet hat.

Gleiches gilt auch für die ähnlichen Benutzer Daniel und Stefanie, die für „Datenbanken“ und „Java-Programmierung“ ähnliche Bewertungen abgegeben haben. Stefanie kann so „Cookies und Sicherheit“ nahe gelegt werden, weil es Daniel gut gefällt und Daniel kann von Stefanie „Einführung in XML“ empfohlen werden. Zudem haben auch Ralf und Stefanie recht ähnliche Präferenzen, aber keine so starken wie zwischen Stefanie und Daniel. Diese Ähnlichkeit würde daher weniger stark in die Empfehlung für eine bislang unbewertete Ressource einfließen.

Zur Berechnung der Bewertungen gibt es verschiedene Verfahren und Algorithmen, die in den folgenden Kapiteln erläutert werden. Allen reinen kollaborativen Filteralgorithmen gemein ist, dass sie nur auf Bewertungen aufbauen und nicht den Inhalt und die Beschaffenheit von Ressourcen analysieren. Zur Personalisierung eignet sich das Kollaborative Filtern vor allem, da ein individuelles Benutzerprofil als Grundlage für personalisierte Empfehlungen und die Sortierung von Ressourcen nach Interessensgrad verwendet werden kann.

3.1.2 Sinnvolle Anwendungsbereiche

Da Kollaboratives Filtern eine Basistechnik ist, die keine speziellen Vorgaben für die mit ihr verwalteten Inhalte macht, lässt es sich auf vielfältige Weise in Informationssystemen verschiedener Ausprägung einsetzen.

Einen besonders häufigen Einsatz erfährt das Kollaborative Filtern bei Online-Shops, um hier zu zufriedeneren Kunden und höheren Umsätzen zu verhelfen. Einem Benutzer können dabei individuelle Produkte angeboten werden. Eine Spezialisierung davon ist, zu einem Produkt eine Reihe von Ressourcen vorzuschlagen, die nach Meinung der Benutzer ähnlich sind. Bewertungen können in Online-Shops durch explizite Wertungen von Produkten auf einer Bewertungsskala oder implizit durch den Kauf eines Produktes erfolgen. Beim Kauf eines Produktes geht man davon aus, dass es dem Kunden gefällt und folglich wird eine positive Bewertung gesetzt.

Genauso gibt es Anwendungen in Informations- und Unternehmensportalen zur Personalisierung von fachlichen Informationen. Jedem Benutzer können individuell Vorschläge gemacht werden, welche Ressourcen er sich ansehen sollte. In der Umkehrung können Informationen unterdrückt

werden, die für den Benutzer von geringem Nutzen sind, und so zu einer Reduzierung der Informationsflut beigetragen werden. Auch das Finden von Experten oder an gleichen Themen interessierten Kollegen ist möglich.

Ein weiterer Einsatzbereich eröffnet sich in Vertriebsanwendungen, bzw. dem Customer Relation Ship Management (CRM), die nicht notwendiger ans World Wide Web angebunden sein müssen. Beispielsweise kann Kollaboratives Filtern auch in Call-Centern eingesetzt werden. Dort geht es darum, den Kunden gezielt Produkte anzubieten, die für sie möglicherweise interessant sind oder eine Ergänzung bisher gekaufter Produkte darstellen. Ähnlich wie im Online-Shop können die Empfehlungen aus dem Kaufverhalten ähnlicher Kunden gewonnen werden.

Prinzipiell sind Kollaborative Filter immer dort gut geeignet, wo Bewertungen von Ressourcen durch Benutzer möglich und verfügbar sind. Da keine Analyse der Inhalte nötig ist, eignet sich das Verfahren auch bei schwer maschinell zu erfassenden Ressourcen wie Grafiken und Filmen oder wenn eine Unterscheidung nach Geschmack und Qualität wie bei Musik erfolgen soll.

Letztlich handelt es sich beim Kollaborativem Filtern auch nur um eine der verfügbaren Personalisierungstechniken. Entsprechend sind die Einsatzmöglichkeiten genauso breit gestreut oder eingegrenzt wie für die Personalisierung generell (siehe dazu 2.1.2 Beweggründe für die Personalisierung von Internetangeboten).

3.1.3 Fallbeispiele

Der Online-Händler *amazon*[35] hat als einer der ersten das Kollaborative Filtern eingesetzt und implementierte eine umfangreiche Zahl verschiedener Empfehlungssysteme, um Kunden die angebotenen Produkte schmackhaft zu machen. Ziel ist dabei sowohl die Zufriedenstellung der Kunden als auch die Steigerung des Umsatzes, wozu Cross- und Up-Selling genutzt werden (siehe Kapitel 2.1.2 unter Online-Shops). Abbildung 22 zeigt einen Ausschnitt der Website, in dem unterhalb der Produktinformationen zu einem Buch Empfehlungen für alternative Bücher ausgegeben werden. Ein Kunde kann so dazu gebracht werden, entweder ein anderes Buch zu kaufen, das ihm besser gefällt, oder sogar ein zweites und drittes.

Abbildung 22 - Amazon.de: Empfehlungen ähnlicher Bücher zu einem angezeigten Buch

In Abbildung 23 wird ein Ausschnitt der *amazon*-Website gezeigt, in dem personalisierte Büchervorschläge eingeblendet werden, die dem Kunden vermutlich gefallen. Basis dafür ist die bisherige Kaufhistorie und die interaktive Verfeinerungsmöglichkeit, mit der uninteressante oder bereits im Besitz des Kunden befindliche Bücher abgewählt werden können. Betätigt man die entsprechende Schaltfläche, wird das jeweilige Buch aus der Liste entfernt und ein neues Buch erscheint.

[35] siehe http://www.amazon.de oder http://www.amazon.com

Abbildung 23 - Amazon.de: Persönliche Empfehlungen mit Verfeinerungsmöglichkeit

Neben Kollaborativen Filtern setzt *amazon* auch Inhaltsbasierte ein, da man an anderer Stelle im Shop Kategorien und interessante Themen vorgeben kann. Insbesondere das in Abbildung 22 gezeigte legendäre „Kunden, die dieses Buch gekauft haben, haben auch...“ ist mittlerweile von zahlreichen anderen Online-Shops kopiert worden.

Im *MovieLens*-Projekt[36] der Arbeitsgruppe *GroupLens Research* an der Universität von Minnesota (USA) können Filminteressierte Filme bewerten und erhalten eine Liste von neuen, vermutlich interessanten, Filmen zurück. Zum Einsatz kommt dabei ein Kollaborativer Filter, der die Empfehlungen anhand der Bewertungen ähnlicher Benutzer berechnet.. Vorteilhaft ist der Einsatz von Kollaborativem Filtern, da der Geschmack der Benutzer berücksichtigt wird, die Filminhalte aber selbst nicht analysiert werden müssen. Benutzer mit ähnlichen Genreinteressen dienen als Basis für die Berechnung der Bewertungen. Denkbar wäre ein kommerzieller Einsatz beispielsweise in einer Online-Videothek, um Kunden bei der Auswahl von zu leihenden Filmen unter die Arme zu greifen.

[36] Die Website des *MovieLens*-Projektes unter: http://movielens.umn.edu/

Abbildung 24 - MovieLens: Ausgabe von personalisierten Filmempfehlungen

SpamNet[37] ist ein Produkt zur Reduzierung der Informationsüberflutung mit E-Mail-Spam, das Kollaboratives Filtern einsetzt. Da eingehende Werbemails nicht für jeden Anwender Spam darstellen und unnütz sind, ist der Einsatz von starren Filter für alle Benutzer nicht sinnvoll. Mit Kollaborativen Filtern ist es aber möglich, dass Benutzer E-Mails als Spam klassifizieren, die von ähnlichen Benutzern ebenfalls als Spam klassifiziert wurden. Die Software sortiert solche E-Mails automatisch aus und bietet Steuerelemente, damit neue E-Mails entweder als Spam oder als Nicht-Spam bewertet werden können.

Abbildung 25 - SpamNet: Steuerelemente zur Klassifikation von E-Mails

Insgesamt wurden so laut Angaben auf der SpamNet-Website bei einer Nutzergemeinde von ca. 730.000 Personen schon gut 5600 Tage Arbeitszeit eingespart (Stand 24.11.2003).

3.1.4 Einordnung in die Architektur von Informationssystemen

Empfehlungsmaschinen mit kollaborativen Filteralgorithmen sitzen zwischen der Inhalteverwaltung und der Aufbereitung der Inhalte in der Benutzungsschnittstelle. Sie liefern eine personalisierte Sicht auf die in der Inhalteverwaltung bereitgestellten Ressourcen, indem sie die Selektion

[37] Die Website von *SpamNet* siehe unter: http://www.cloudmark.com/products/spamnet/

der gelieferten Ressourcen und deren Sortierung beeinflussen. Dabei bedienen sie sich als Datengrundlage der expliziten und impliziten Ressourcenbewertungen durch Benutzer. In einer mit *Bewertungsspeicher* bezeichneten Komponente werden die expliziten Bewertungen gespeichert, in einem *Benutzungsdatenspeicher* die Zugriffsdaten auf verschiedene Inhalte. Auch die Benutzerverwaltung und statisch gespeicherte Profildaten können zur Erzeugung von Empfehlungen herangezogen werden. Zu einer Nutzung der verschiedenen Datenquellen ist je nach Implementierung eine Aufbereitung und Verdichtung der Daten nötig, so dass die Algorithmen in der Empfehlungsmaschine mit jeweils einer Bewertung pro Ressource arbeiten können (Filterarchitektur siehe z. B. bei [FEBS02] und bei [SKKR00]).

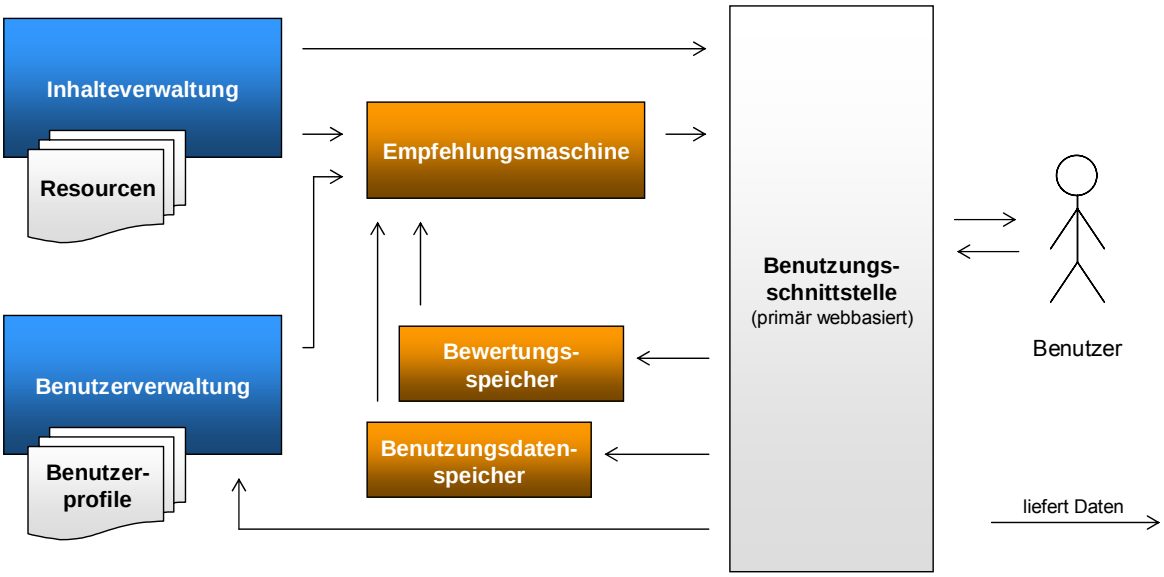

Abbildung 26 - Einbettung von Kollaborativen Filtern in das Informationssystem

Innerhalb der Empfehlungsmaschine können verschiedene Algorithmen zum Einsatz kommen, die unterschiedliche mathematische Techniken einsetzen. Auch eine Verbindung verschiedener Personalisierungstechniken wie Inhaltsbasierter Filter mit Kollaborativen Filtern ist möglich. Wie die Algorithmen aussehen können, wird in den nächsten Abschnitten beschrieben.

3.2 Eingesetzte Verfahren

Im Folgenden wird das dem Kollaborativen Filtern zugrunde liegende mathematische Modell zunächst formal erläutert. Zum Modell gehören einerseits die Daten in Form von Bewertungen und andererseits die Operationen auf diesen Daten. Anschließend werden die beiden grundlegenden Algorithmenklassen zum Kollaborativen Filtern betrachtet. Die Unterscheidung der beiden Klassen erfolgt in memory-based und model-based und wurde unter anderem von [BHK98] vorgenommen. Sie wird in der Literatur seitdem allgemein verwendet.

Daten

Die Datenbasis von Verfahren des Kollaborativen Filterns sind die Bewertungen von Ressourcen durch Benutzer, die abstrakt in einer Bewertungsmatrix *Ratings* zusammengefasst sind[38]. Für jede Ressource *iid* gibt es maximal eine Bewertung $rating_{uid,iid}$ durch jeden Benutzer *uid*. Eine Bewer-

[38] Statt Ressource können anwendungsabhängig auch andere Begriffe wie Item, Dokument oder Inhaltseintrag verwendet werden.

tung ist in diesem Modell ein numerischer Wert aus dem Intervall [0..1], wobei ein Wert von 0 eine geringe und ein Wert von 1 eine hohe Präferenz des Benutzers für die Ressource *iid* ausdrücken[39]. Eine unbewertete Ressource wird technisch durch einen Null-Wert repräsentiert.

$$Ratings = \left(rating_{uid,iid} \right) \text{ mit } rating_{uid,iid} \in [0,1] \tag{3.1}$$

In dieser Notation sind die Parameter eindeutige numerische Schlüssel des Benutzers und der Ressource, wobei als Eingabe nur solche Schlüsselwerte gestattet sind, die auf tatsächlich vorhandene Benutzer oder genauer deren Benutzeraccounts und vorhandene Ressourcen verweisen. Die Menge aller Benutzer sei *U* und die Menge aller Ressourcen *I*.

$$uid \in U \quad iid \in I$$

Formal stellt sich das Modell – es sei hier mit M_{CF} bezeichnet – als 4-Tupel aus den Benutzerdaten *U*, den Inhalten *I* sowie den Bewertungen *Ratings* und den Operationen dar.

$$M_{CF} = (U, I, Ratings, Operations) \tag{3.2}$$

Operationen

Die Verfahren des Kollaborativen Filterns bieten zwei grundlegende Funktionen auf den Daten an. Die erste Operation ist die Berechnung einer Bewertung für einen Benutzer und eine Ressource. Der Benutzer, für den die Berechnung durchgeführt werden soll, wird als aktiver Benutzer bezeichnet. Die zweite grundlegende Operation ist die Bestimmung einer Liste von Ressourcen, die möglichst gut den Benutzerinteressen entsprechen. So können z. B. die zehn voraussichtlich beliebtesten Ressourcen für den aktiven Benutzer geliefert werden. Zur Berechnung werden bei beiden Operationen sowohl die vom aktiven Benutzer bisher bewerteten Ressourcen als auch die Bewertungen der anderen Benutzer verwendet, also die Bewertungsmatrix *Ratings*.

Die erste Operation – sie sei mit Vorhersagefunktion bezeichnet – liefert zu einem gegebenen Benutzer und einer Ressource die vermutliche Bewertung. Die Rückgabe der Funktion entspricht von der Bedeutung den ursprünglichen, tatsächlichen Bewertungen und stammt ebenso aus dem gleichen Wertebereich – in dieser Arbeit also [0,1]. Der Benutzer, für den die Bewertung bestimmt werden soll, wird mit uid_a angegeben. Zu beachten ist, dass die Bewertung berechnet ist und eine Vorhersage der Benutzerpräferenz für die betrachtete Ressource widerspiegelt. Je nach Datengrundlage und Vorhersagealgorithmus ist die Qualität daher unterschiedlich.

$$pred(uid_a, iid) = predicted\_rating_{uid_a,iid} = ? \in [0,1] \tag{3.3}$$

Die Vorhersagefunktion *pred* kann als Blackbox aufgefasst werden, in die lediglich die Parameter eingegeben werden und die unabhängig von einer konkreten Implementierung eine Bewertung liefert. Für die Implementierung gibt es verschiedene Varianten, die in den folgenden Kapiteln vorgestellt werden. Bezieht man die Bewertungsmatrix mit ein, kann man die Funktionseingabe einfach um diese erweitern und formal schreiben:

$$pred(Ratings, uid_a, iid) = predicted\_rating_{uid_a,iid} = ? \in [0,1]$$

Die zweite grundlegende Operation ist die Bestimmung einer Liste von Ressourcen des Informationssystems, die möglichst gut den Interessen des aktiven Benutzers uid_a entsprechen. Sie sei im Folgenden Empfehlungslistenfunktion genannt. Denkbar wäre beispielsweise, die vermutlich zehn beliebtesten Ressourcen des aktiven Benutzers zurückzugeben. Die gelieferten Ressourcen sollten

[39] Es können auch andere Wertebereiche verwendet werden, beispielsweise [-1,1] oder [0,5]. Innerhalb des Informationssystems und Algorithmus' sollte aber mit einem einzelnen Wertebereich gearbeitet werden.

den Präferenzen des Benutzers in absteigender Reihenfolge entsprechen, so dass die erste Ressource am besten übereinstimmt. *n* bezeichnet in der folgenden Definition die Anzahl zurückzuliefernder Ressourcen.

$$toplist(uid_a, n) = (iid_1, iid_2, ..., iid_n) \tag{3.4}$$

Sinnvoll kann die Einschränkung auf eine Auswahl von Ressourcen $I_{selected}$ sein, wenn beispielsweise die Rückgabe einer Suchoperation nach den Präferenzen des aktiven Benutzers sortiert werden soll. Auch innerhalb eines Kontextes kann eine Einschränkung sinnvoll sein, wenn z. B. nur die Ressourcen einer Kategorie begutachtet werden. Hier bieten sich Verknüpfungsmöglichkeiten des Kollaborativen Filterns mit Inhaltsbasierten oder Regelbasierten Filtern an (siehe auch Kapitel 2.4 Verfahren zur Personalisierung).

$$toplist(I_{select}, uid_a, n) = (iid_1, iid_2, ..., iid_n) \tag{3.5}$$

Die allgemeine toplist-Funktion (3.5) entspricht (3.4), wenn man einfach I_{all} (alle Ressourcen des Informationssystems) als I_{select}-Parameter einsetzt.

Der Vollständigkeit halber kann die Empfehlungslistenfunktion auch die Bewertungsmatrix *Ratings* als Eingabe erhalten, da sie der Berechnung in allen Implementierungen zugrunde liegt. Auch hier ist die Operation eine Blackbox, in die die gewünschten Variablen eingegeben werden und die die Ergebnisse liefert. Die Bestimmung der Ressourcenliste bleibt der jeweiligen Implementierung überlassen.

$$toplist(Ratings, I_{select}, uid_a, n) = (iid_1, iid_2, ..., iid_n)$$

Hilfsoperationen

Neben den beschriebenen Vorhersagefunktionen werden von den Implementierungen weitere Operationen auf den Daten genutzt. Beispielsweise solche Hilfsfunktionen wie die Rückgabe einer einzelnen Bewertung oder die Rückgabe aller bewerteten Ressourcen eines Benutzers. Nützlich sind auch die Berechnung der durchschnittlichen Bewertung einer Ressource und die mittlere Bewertung eines Benutzers.

Die durchschnittliche Bewertung einer Ressource $mean\_resource\_rating_{iid}$ oder mrr_{iid} berechnet sich als Mittelwert aller Bewertungen der Benutzer U_{iid}, die die Ressource bewertet haben.

$$mrr_{iid} = mean\_resource\_rating_{iid} = \frac{1}{|U_{iid}|} \sum_{i \in U_{iid}} rating_{i,iid} \tag{3.6}$$

Diese durchschnittliche Ressourcenbewertung kann auch für unpersonalisierte Empfehlungen nützlich sein, wenn beispielsweise die zehn von allen Benutzern als am interessantesten eingestuften Ressourcen aufgelistet werden sollen. Ein Einsatz ist zudem möglich, wenn die Filter-Implementierung für die gewünschte Ressource und den gewünschten Benutzer keine Bewertung berechnen kann, beispielsweise wenn die Datengrundlage zu gering ist (siehe [HKBR99], Abschnitt 8).

Die mittlere Bewertung eines Benutzers über alle von ihm bewerteten Ressourcen I_{uid} hingegen beschreibt das Bewertungsniveau $mean\_user\_rating_{uid}$ oder kurz mur_{uid} eines Benutzers und wird analog zur durchschnittlichen Ressourcenbewertung berechnet.

$$mur_{uid} = mean\_user\_rating_{uid} = \frac{1}{|I_{uid}|} \sum_{i \in I_{uid}} rating_{uid,i} \tag{3.7}$$

Manche Benutzer bewerten beispielsweise eher generell positiv (nahe bei 1) und manche eher negativ (nahe bei 0), was mit dem Bewertungsniveau beziffert wird.

Nach diesen grundlegenden Vorarbeiten stellt sich jetzt die Frage, wie die möglichen Implementierungen der Vorhersagefunktion *pred* und der Empfehlungslistenfunktion *toplist* aussehen.

3.2.1 Speicherbasierte Algorithmen

Die memory-based oder speicherbasierten Algorithmen arbeiten für die Berechnung jeder Vorhersage auf der gesamten Bewertungsmatrix *Ratings*. Das ist insbesondere zur Abgrenzung zu den modellbasierten Algorithmen relevant, die weiter unten beschrieben werden und die mit einer Zwischenschicht rechnen.

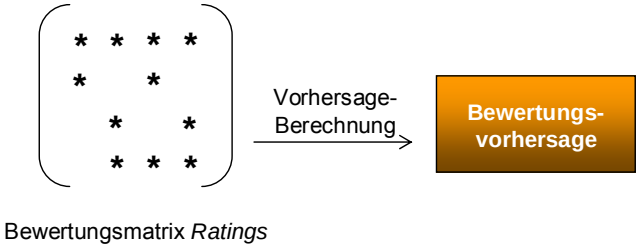

Abbildung 27 - Daten- und Kontrollfluss bei speicherbasierten Algorithmen

Bei der Berechnung suchen die Algorithmen in der Matrix zunächst eine Umgebung von ähnlichen Benutzern um den aktiven Benutzer, die auch Nachbarschaft genannt wird[40]. Die Ähnlichkeit zwischen zwei Benutzern $w(uid_1,uid_2)$ wird mit einer Metrik bestimmt. Die Unterschiede in den speicherbasierten Algorithmen liegen in erster Linie in der unterschiedlichen Ausgestaltung dieser Metrik. Anhand ähnlich bewerteter Ressourcen kann die Ähnlichkeit von Benutzerinteressen auf verschiedene Weise berechnet werden. Ein großer Wert von $w(uid_1,uid_2)$ entspricht einer hohen Ähnlichkeit – also hohen gemeinsamen Präferenzen über die bewerteten Ressourcen – und ein niedriger Wert einer geringen Ähnlichkeit. Auf Grundlage dieser Funktion lassen sich auch Benutzer ähnlicher Interessen in einem Informationssystem zusammenführen.

$$w(uid_1, uid_2) = \text{Ähnlichkeit zwischen Benutzer uid}_1 \text{ und uid}_2 \tag{3.8}$$

Grundidee ist, dass die Bewertung eines bislang durch den aktiven Benutzer unbewerteten Items vorhergesagt werden kann, indem die Bewertungen anderer Benutzer, die das Item bewertet haben, mit ihrer Ähnlichkeit zum aktiven Benutzer gewichtet werden. Mathematisch stellt sich das als gewichtete Summe der Bewertungen dar.

$$pred\left(uid_a, iid\right) = mur_{uid_a} + \kappa \sum_{uid \in U_{iid}} w\left(uid_a, uid\right)\left(rating_{uid,iid} - mur_{uid}\right) \tag{3.9}$$

Der Wert der Funktion entspricht einem Zugewinn oder einer Abnahme vom mittleren Bewertungsniveau des aktiven Benutzers. Entsprechend werden auch die Differenzen der vorhandenen Bewertungen der anderen Benutzer und ihrer jeweiligen Bewertungsniveaus herangezogen. Die Variable κ dient als Normalisierungsfaktor, so dass sich die Benutzerähnlichkeiten zu eins summieren und der Vorhersagewert nicht aus dem Wertebereich herausragt.

[40] Die speicherbasierten Algorithmen werden daher auch als nachbarschaftsbasierte Algorithmen bezeichnet (siehe beispielsweise bei [SKKR01]).

$$\kappa = \frac{1}{\sum_{uid \in U_{iid}} w(uid_a, uid)}$$

Charakteristisch an der Vorhersagefunktion bei speicherbasierten Algorithmen ist die Ähnlichkeitsbestimmung zwischen Benutzern, die mit distanz-, korrelations- oder ähnlichkeitsbasierten Metriken möglich ist. Mit I_{Both} werden im Folgenden diejenigen Ressourcen bezeichnet, die von beiden Benutzern bewertet wurden.

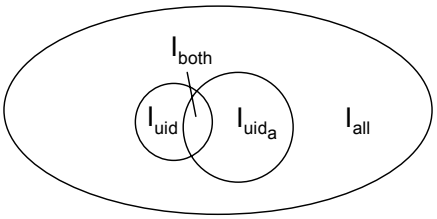

Abbildung 28 - Mengenüberlappung von gemeinsam bewerteten Ressourcen

I_{all} steht wie schon weiter oben verwendet für alle Ressourcen des Informationssystems und I_{uid_a} sowie I_{uid} jeweils für diejenigen Ressourcen, die vom aktiven Benutzer und vom Vergleichsbenutzer bewertet wurden.

Distanzbasierte Ähnlichkeitsfunktion

Die Ähnlichkeit zweier Bewertungsmengen kann beispielsweise mit der mittleren quadratischen Differenz[41] bestimmt werden.

$$w(uid_a, uid) = 1 - \frac{\sum_{iid \in I_{both}} \left(rating_{uid_a, iid} - rating_{uid, iid} \right)^2}{|I_{both}|} \tag{3.10}$$

Auch können ähnliche Funktionen wie die mittlere absolute Differenz verwendet werden, die statt den Quadraten die Beträge der Differenzen aufsummiert. Nachteil an dieser Ähnlichkeitsbestimmung ist aber, dass zwar die gemeinsamen bewerteten Ressourcen berücksichtigt werden, nicht jedoch deren Verhältnis an den Gesamtbewertungen der Benutzer. Möglicherweise stellen die gemeinsamen Bewertungen zweier Benutzer nur einen Bruchteil ihrer vollständigen Bewertungen dar. Dann würde dieses Ähnlichkeitsmaß einen falschen Zusammenhang zwischen den betrachteten Benutzern liefern.

Korrelationsbasierte Ähnlichkeitsfunktion

In vielen Papieren und Arbeiten zum Kollaborativen Filtern wird die Ähnlichkeit zweier Benutzer mit einer Korrelationsfunktion bestimmt. Dabei werden die gemeinsam vom aktiven Benutzer und vom Vergleichsbenutzer betrachteten Bewertungen verglichen. Wie bei den distanzbasierten Methoden wird auch nicht berücksichtigt, dass die Überlappung zweier Benutzer im Vergleich zur Anzahl der jeweiligen Gesamtbewertungen gering und daher wenig aussagekräftig sein kann. Allerdings wird in der folgenden Formel, die auch als Pearson-Korrelationskoeffizient bezeichnet wird und für das Kollaborative Filtern erstmalig beim *GroupLens*-Projekt verwendet wurde (siehe [RISBR94]), das durchschnittliche Bewertungsniveau der Vergleichsbenutzer einbezogen.

[41] In der englischen Literatur unter MSD (Mean Squared Difference) zu finden.

$$w(uid_a, uid) = \frac{\sum_{iid \in I_{both}} \left(rating_{uid_a,iid} - mur_{uid_a}\right)\left(rating_{uid,iid} - mur_{uid}\right)}{\sqrt{\sum_{iid \in I_{both}} \left(rating_{uid_a,iid} - mur_{uid_a}\right)^2} \sqrt{\sum_{iid \in I_{both}} \left(rating_{uid,iid} - mur_{uid}\right)^2}} \quad (3.11)$$

Die beiden Faktoren im Nenner der rechten Seite sind die Varianzen von Benutzer uid_a und *uid* über den gemeinsamen Bewertungen. Der Zähler entspricht der Kovarianz der Bewertungen.

$$\sigma_{uid,I_{both}} = \sqrt{\sum_{iid \in I_{both}} \left(rating_{uid,iid} - mur_{uid}\right)^2}$$

Die Ähnlichkeitsfunktion liefert bei auf [0,1] normierten Bewertungen und Mittelwerten einen reellen Wert zwischen minus eins und eins. In der Praxis hat sich diese Ähnlichkeitsbestimmung bewährt und liefert im Vergleich zu anderen Verfahren die besten Ergebnisse für speicherbasierte Algorithmen (siehe [BHK98]). Die Leistung kann mit einigen zusätzlichen Optimierungen noch verbessert werden, auf die an dieser Stelle aber nicht eingegangen wird.

Ein ähnlicher Ansatz ist der Spearman-Rang-Korrelationskoeffizient (siehe [HKBR99]), der statt einer kontinuierlichen Bewertung mit einem Rang von ganzen Zahlen wie 0 bis 5 als Bewertung arbeitet. Prinzipiell ist die Ähnlichkeitsformel identisch zu (3.11), lediglich die Bewertungen werden durch die entsprechenden diskreten Ränge ersetzt. Der Unterscheid ist eine möglicherweise einfachere Berechnung und ein größerer numerischer Abstand zwischen den möglichen Bewertungen, jedoch auch mit niedrigerer Genauigkeit durch die Diskretisierung.

Vektor-Ähnlichkeitsfunktion (auch Kosinusfunktion)

Ein anderes Konzept zur Ähnlichkeitsbestimmung kommt aus dem Information Retrieval. Ein Verfahren zum Vergleich von Anfragen und gespeicherten Inhalten ist das Vektormodell, das Dokumente als Vektoren interpretiert. Jede Komponente eines Vektors repräsentiert die Häufigkeit oder allgemeiner das Gewicht eines Wortes innerhalb des Dokuments (siehe [BR99], Kapitel 2.5.3). Dabei gibt es viele Worte, die nur in einem Teil der Dokumente auftauchen. Im Vektor ist die entsprechende Komponente dann null. Das Modell integriert also bereits die Behandlung von fehlenden Werten.

Die Ähnlichkeit zweier Dokumente kann mittels des Kosinus' im n-dimensionalen Raum bestimmt werden. Wobei n die Anzahl aller auftauchenden Worte ist und somit die Länge der Dokumentvektoren vorgibt. Anhand der Wortverteilung identische Dokumente liefern so eine 1 als Ähnlichkeit, während völlig entgegengesetzte Dokumente eine 0 zurückgeben.

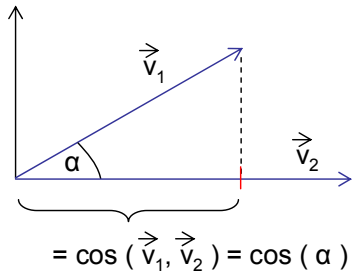

Abbildung 29 - Kosinus zwischen Vektoren zur Ähnlichkeitsbestimmung

Übertragen auf die Domäne des Kollaborativen Filterns sind die Benutzer als Dokumente zu interpretieren, die Worte als Ressourcen und die Bewertungen schließlich als Worthäufigkeiten. Für die Ähnlichkeitsfunktion ergibt sich daraus folgende Formel. Die Normen im Nenner sorgen dafür,

dass die gemeinsam bewerteten Ressourcen I_{both} im Vergleich zu allen Ressourcen der Benutzer uid_a und uid nicht bevorzugt gewichtet werden. Viele unterschiedliche Ressourcen und nur wenige gemeinsame führen also zur Abwertung und entsprechend niedrigeren Ähnlichkeitswerten.

$$w(uid_a, uid) = \frac{\langle rating_{uid\,a}, rating_{uid} \rangle}{\|rating_{uid_a}\| \cdot \|rating_{uid}\|} = \frac{\sum_{iid \in I_{all}} rating_{uid_a,iid} \cdot rating_{uid,iid}}{\sqrt{\sum_{j \in I_{uid_a}} rating_{uid_a,j}^{\,2}} \cdot \sqrt{\sum_{j \in I_{uid}} rating_{uid,j}^{\,2}}} \quad (3.12)$$

Im Information Retrieval-Bereich werden die Gewichte noch dahingehend optimiert, dass der Tatsache Rechnung getragen wird, dass besonders häufig in allen Dokumenten vorkommende Worte vermutlich eine geringe Aussagekraft besitzen. Worte, die hingegen nur in wenigen Dokumenten auftauchen und dafür innerhalb dieses Dokuments sehr intensiv, haben potentiell einen höheren Informationsgehalt. Dazu wird die so genannte inverse Dokumentfrequenz eines Wortes idf_{word} herangezogen, die auf der Häufigkeit der Worte in allen N Dokumenten basiert (siehe [BR99], Kapitel 2.5.3) .

$$idf_{word} = \log \frac{N}{n_{word}} \quad (3.13)$$

Mit n_{word} ist die Anzahl der Dokumente gegeben, die das Wort *word* enthalten. Ist das Wort in jedem Dokument des Datenbestandes enthalten, dann ist die inverse Dokumentfrequenz null. Tritt es jedoch nur in wenigen Dokumenten auf, ist die inverse Dokumentfrequenz entsprechend hoch.

In [BHK98] wird beschrieben, wie sich dieses Konzept auf das Kollaborative Filtern und die Benutzerähnlichkeit übertragen lässt. Ressourcen, die von vielen oder gar allen Benutzern bewertet wurden, haben der Annahme nach einen geringeren Informationsgehalt gegenüber solchen, die nur von wenigen Benutzern bewertet wurden. Analog zur inversen Dokumentfrequenz wird daher die Verwendung der inversen Benutzerfrequenz (oder englisch inverse user frequency, kurz *iuf*) vorgeschlagen.

$$iuf_{iid} = \log \frac{N}{n_{iid}} \quad (3.14)$$

Hier bezeichnet n_{iid} die Anzahl der Benutzer, die die Ressource *iid* bewertet haben und N die Anzahl aller Ressourcen im Informationssystem. Für die Ähnlichkeitsfunktion ergibt sich so eine leichte Modifikation, indem alle Bewertungswerte einfach mit der inversen Benutzerfrequenz multipliziert werden.

$$w'(uid_a, uid) = \frac{\sum_{iid \in I_{all}} iuf_{iid} \cdot rating_{uid_a,iid} \cdot rating_{uid,iid}}{\sqrt{\sum_{j \in I_{uid_a}} \left(iuf_j \cdot rating_{uid_a,j}\right)^2} \cdot \sqrt{\sum_{j \in I_{uid}} \left(iuf_j \cdot rating_{uid,j}\right)^2}} \quad (3.15)$$

In der Evaluierung auf Testdaten von [BHK98] ergeben sich so leichte Vorteile beim Einsatz von (3.15) gegenüber der Verwendung der einfachen Ähnlichkeitsfunktion (3.12).

Implementierung der toplist-Operation

Die *toplist*-Operation kann so implementiert werden, dass sie für alle Ressourcen einer vorgegebenen Menge die Bewertungen mittels der *pred*-Operation berechnet und die Ressourcen nach absteigender Bewertung sortiert zurückgibt. Es erfolgt also eine Sortierung und eine anschließende Selektion der ersten n Elemente. Dieses Verfahren bewahrt die Blackbox-Eigenschaft der *pred*-

Operation und kann daher unabhängig vom verwendeten Algorithmus eingesetzt werden. Die Vorauswahl der Ressourcenmenge kann beispielsweise für die aktuelle Kategorie erfolgen, also den Kontext, wenn die Ressourcen in Kategorien unterteilt sind, oder aufgrund eines anderen Filters, beispielsweise eines Inhaltsbasierten.

Die Blackbox-Eigenschaft der *pred*-Operation zu wahren ist der nahe liegende und einfach umzusetzende Weg. Bei großen Informationssystemen kann diese Vorgehensweise jedoch zu Performanceengpässen führen, selbst wenn die Vorauswahl der Ressourcen nur einem Teil des gesamten Datenbestandes entspricht. Günstiger wäre es daher, wenn man direkt von der Bewertungsmatrix ausgehend zu einer Liste von vermutlich *n* beliebtesten Ressourcen für den aktiven Benutzer gelangen könnte. Ein Ansatz dazu ist, die Nachbarbenutzer des aktiven Benutzers zu betrachten und die beliebtesten Ressourcen der Nachbarn sortiert nach Bewertung gewichtet mit der Ähnlichkeit zurückzugeben. Vereinfacht dargestellt erreicht dies folgende Funktion, wobei *top* die ersten *n* Elemente zurückgibt und *sort* die Sortierung vornimmt.

$$toplist_{neighbours}\left(I_{select}, uid_a, n\right) = top\left(n, sort\left(\bigcup_{uid \in neighbours(uid_a)} I_{select} \cap I_{uid}\right)\right) \qquad (3.16)$$

Die Nachbarschaft kann mittels eines Grenzwertes für die Benutzerähnlichkeitsfunktion ermittelt werden, so dass Benutzer mit einem Ähnlichkeitswert über dem Grenzwert als Nachbarn zählen und solche, die darunter liegen nicht. Auch Clusterverfahren zur Unterteilung der Benutzer in Gruppen mit ähnlichen Interessen sind zur Eingrenzung einer Nachbarschaft denkbar (siehe [SKKR00], [SKKR02]).

Bei der beschriebenen Vorgehensweise handelt es sich aber um eine Heuristik, da unter Umständen Ressourcen, die nicht in der Ausgangsmenge enthalten waren, weil sie von den Nachbarn des aktiven Benutzers als nicht beliebt eingestuft wurden, möglicherweise für den aktiven Benutzer doch interessant sind und besser bewertet werden. Auftreten kann das Problem beispielsweise an den Rändern der Nachbarschaft, wenn dort Benutzer mit besonders beliebten Ressourcen anzutreffen sind, diese Benutzer aber nicht mehr zur Nachbarschaft des aktiven Benutzers gezählt werden. Sind die Bewertungen innerhalb der Nachbarschaft auf eher niedrigem Niveau, erhält der Benutzer entsprechend weniger beliebte Empfehlungen und die als besonders gut beurteilten, aber außerhalb der Nachbarschaft liegenden Ressourcen werden unterdrückt. Die Wahl der Nachbarschaftsgröße ist also für die Qualität der Empfehlungen wichtig. Je größer die Nachbarschaft aber ist, desto höher ist auch der Rechenaufwand. Hier ist für die Umsetzung in Informationssystemen je nach Vorgaben ein entsprechender Kompromiss zu schließen.

3.2.2 Modellbasierte Algorithmen

Die modellbasierten Algorithmen des Kollaborativen Filterns berechnen zunächst ausgehend von den Bewertungen ein Modell und speichern es in einer geeigneten Repräsentationsform ab. Für eine Vorhersage wird dann das Modell nach dem gewünschten Wert befragt, wobei das erzeugte Modell in Form von Regeln, Entscheidungsbäume oder Clusteraufteilungen repräsentiert werden kann. Auch andere Verfahren des maschinellen Lernens und der Statistik sind denkbar, beispielsweise eine Sichtweise als Klassifikationsproblem mit neuronalen Netzen oder Support-Vektor-Maschinen.

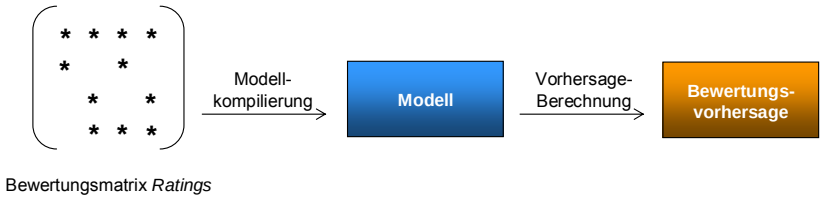

Abbildung 30 - Daten- und Kontrollfluss bei modellbasierten Algorithmen

In allen Algorithmenausprägungen geht es darum, Zusammenhänge zwischen den Ressourcen und den Benutzern anhand der Bewertungen zu erkennen und daraus Muster abzuleiten. Beispielsweise können Benutzer mit ähnlichen Interessen in Cluster gruppiert werden. Die Vorhersage einer Bewertung erfolgt dann in Abhängigkeit von der Clusterzugehörigkeit des aktiven Benutzers. Oder die Bewertung von Ressourcen mit bestimmten Werten impliziert die Bewertung anderer Ressourcen mit entsprechenden Werten. Mit den bisherigen Bewertungen eines Benutzers kann davon ausgehend die Bewertung für eine bislang unbewertete Ressource vorhergesagt werden.

Formal wird hierzu von [BHK98] vorgeschlagen, einen probabilistischen Ansatz zu verfolgen[42]. Betrachtet man die vorhergesagte Bewertung als eine Zufallsvariable $Rating_{uida,iid}$, dann kann diese n diskrete Werte wie ganze Zahlen von 0 bis 5 annehmen (der Einfachheit halber wird hier mit diskreten und nicht kontinuierlichen Zufallsvariablen gearbeitet). Die Berechnung der vorhergesagten Bewertung für einen aktiven Benutzer und eine konkrete Ressource erfolgt dann als Erwartungswert über der Zufallsvariable $Rating_{uida,iid}$, wobei die Wahrscheinlichkeiten für die Bewertung der Ressource *iid* unter der Annahme erfolgen, dass bereits m Bewertungen des aktiven Benutzers vorliegen.

$$\begin{aligned} pred\left(uid_a, iid\right) &= \mathrm{E}\left(Rating_{uid_a,iid}\right) \\ &= \sum_{i=0}^{n} \mathrm{Wk}\left(Rating_{uid_a,iid} = i \,\middle|\, Rating_{uid_a,k} \text{ mit } k \in I_{uid_a}\right) \cdot i \end{aligned} \tag{3.17}$$

Mit

$$m = \left| I_{uid_a} \right| \text{ mit } I_{uid_a} = \left\{ iid \,\middle|\, \text{Resource } iid \text{ wurde von Benutzer } uid_a \text{ bewertet} \right\}$$

Die Aufgabe ist jetzt, die bedingten Wahrscheinlichkeiten für die vermutete Bewertung der Ressource *iid* in allen n möglichen Ausgängen unter dem Vorwissen zu berechnen, dass der Benutzer bereits m viele Ressourcen mit $Rating_{uida,k}$ bewertet hat. Dazu gibt es verschiedene Ansätze. Ein Ansatz ist, die Benutzer in Cluster zu unterteilen, jeden Benutzer einem Cluster zuzuweisen und auf Basis der Verteilung im Cluster eine Bewertung zu berechnen. Ein anderer Ansatz ist, die Bewertungsberechnung als ein Klassifikationsproblem aufzufassen und für jede Ressource *iid* einen Regelsatz zu ermitteln, mit welcher Wahrscheinlichkeit sie für den aktiven Benutzer einen der diskreten Werte 1 bis n annimmt.

Clusterverfahren

Im Bereich des maschinellen Lernens oder des Data Minings gibt es verschiedene Verfahren, um eine Menge von Objekten in eine feste Zahl von Clustern zu unterteilen. Ziel der Verfahren ist, möglichst homogene Objekte innerhalb eines Clusters zusammenzufassen und gleichzeitig einen

[42] Allerdings sind im abstrakten Konzept des modellbasierten Vorgehens nicht nur probabilistische Formalismen möglich. Entscheidend ist vielmehr die Generierung oder Kompilierung eines Modells mit anschließender Speicherung, so dass Anfragen an das gespeicherte Modell möglich sind.

hohen Abstand zwischen den einzelnen Clustern zu erhalten, also eine hohe Heterogenität zwischen den Clustern zu erreichen.

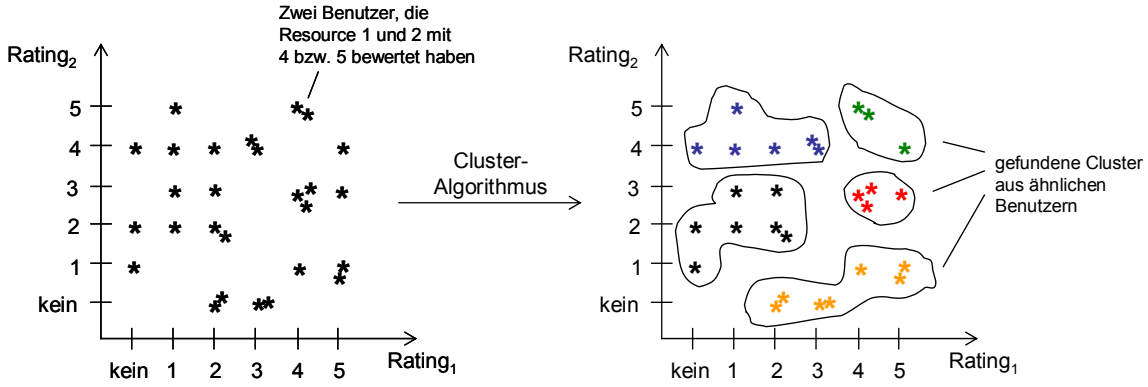

Abbildung 31 - Zerlegung von Benutzermengen in Cluster

Gemessen wird der Abstand mit einer Metrik, die folgenden Anforderungen genügen soll und verschiedene Ausprägungen haben kann – wie die Distanzfunktion aus Formel (3.10).

$$d(x,y)=0 \Leftrightarrow x=y \qquad d(x,y)=d(y,x) \qquad d(x,y)+d(y,z)\le d(x,z) \tag{3.18}$$

Algorithmen wie der k-Mittelwerte-Algorithmus oder Verfahren des hierarchischen Clusterns ordnen jedes Objekt einem einzelnen Cluster zu (siehe beispielsweise [Run00], Kapitel 5.6 und [WF01], Kapitel 6.6). Im Fall des Kollaborativen Filterns können die zu unterteilenden Objekte Benutzer oder Ressourcen sein, die in ähnliche Cluster gruppiert werden sollen, wobei hier nur auf die Unterteilung von Benutzern eingegangen wird. Idee dahinter ist, dass es im Informationssystem Gruppen oder Klassen von Benutzern gibt, die ähnliche Präferenzen haben, und diese Gruppen sich durch einen Clusteralgorithmus aufdecken lassen.

Ein Benutzer *uid* wird dabei als ein Vektor $Rating_{uid}$ von Bewertungen $rating_{uid,iid}$ repräsentiert. Der Vektor entspricht einer Zeile der Bewertungsmatrix *Ratings*. Zur Berechnung von Vorhersagen kann man für jeden Cluster einen Repräsentanten bestimmen – also einen Vektor von Ressourcenbewertungen, der entweder dem durchschnittlichen Benutzer des Clusters entspricht oder dem Median des Clusters. Anhand der Repräsentantenbewertungen kann dann eine Bewertung für die gewünschte Ressource berechnet werden. Alternativ kann auch – um im probabilistischen Modell zu bleiben – die Häufigkeit für jede diskrete Bewertung (hier 1, 2, 3, 4 oder 5) anhand der Verteilung im Cluster ermittelt und dann nach Formel (3.17) der Erwatungswert berechnet werden.

Einen anderen Ansatz verfolgt das Bayes'sche Clustering (siehe [WF01], S. 245). Hier wird zwar auch angenommen, dass es eine feste Zahl *k* von Clustern gibt, auf die die Benutzer zu verteilen sind, allerdings gehört ein Benutzer mit einer unterschiedlichen Wahrscheinlichkeit jedem Cluster und nicht nur einem einzigen an. Der Vorteil an dieser Sichtweise ist, dass es mitunter schwierig sein kann, geeignete Cluster zu finden und einen Benutzer zu einem einzigen Cluster zuzuordnen. Die wahrscheinlichkeitsbasierte Mehrfachzugehörigkeit zu einem Cluster entschärft das Problem, da ein Benutzer gewissermaßen zu mehreren Clustern gehört. Die Zugehörigkeit zu einem Cluster wird mit

$$\mathrm{Wk}(C=c) \tag{3.19}$$

ausgedrückt, wobei das kleine *c* für einen der *k* Cluster steht. Die Wahrscheinlichkeit für die Zugehörigkeit eines Benutzers zu einem Cluster *c* ist abhängig vom Bewertungsvektor des Benutzers. Es wird angenommen, dass die Bewertungen der einzelnen Ressourcen stochastisch unabhängig

voneinander sind, um den Satz von Bayes verwenden zu können. Allerdings ist diese Sichtweise durchaus anfechtbar, da die Bewertungen von ähnlichen Ressourcen je nach Inhalt und Semantik der Ressourcen möglicherweise doch zusammenhängen. Allerdings werden diese möglichen Abhängigkeiten beim reinen Kollaborativen Filtern nicht berücksichtigt. Hier wäre ein Ansatzpunkt für eine Verknüpfung mit Inhaltsbasierten Filtern denkbar. Aber zurück zur Wahrscheinlichkeit eines Benutzers, zu einem Cluster zu gehören unter Annahme der stochastischen Unabhängigkeit der *m* vorliegenden Bewertungen:

$$\begin{aligned}\mathrm{Wk}_{uid}\left(C=c \middle| Rating_{uid}\right) &= \mathrm{Wk}\left(C=c \middle| rating_{uid,1}, rating_{uid,2}, ..., rating_{uid,m}\right)\\ &= \frac{\mathrm{Wk}\left(rating_{uid,1},...,rating_{uid,m} \middle| C=c\right)\cdot \mathrm{Wk}\left(C=c\right)}{\mathrm{Wk}\left(rating_{uid,1},...,rating_{uid,m}\right)}\\ &= \frac{\mathrm{Wk}\left(C=c\right)\cdot \prod_{j=1}^{|R|}\mathrm{Wk}\left(rating_{uid,j} \middle| C=c\right)}{\mathrm{Wk}\left(rating_{uid,1},...,rating_{uid,m}\right)}\end{aligned} \tag{3.20}$$

Zur Berechnung der Vorhersage einer Bewertung wird der Erwartungswert mit Wahrscheinlichkeiten für jeweils einen möglichen, diskreten Ausgang der Bewertung herangezogen. Diese Wahrscheinlichkeiten lassen sich berechnen, wenn man die Clusterwahrscheinlichkeit des Benutzers heranzieht und folgende Formel verwendet (siehe auch [BHK98], Absatz 2.3.1).

$$\mathrm{Wk}_{uid}\left(C=c \wedge Rating_{uid}\right) = \mathrm{Wk}\left(C=c\right)\cdot \prod_{j=1}^{|R|}\mathrm{Wk}\left(rating_{uid,j} \middle| C=c\right) \tag{3.21}$$

Gesucht sind nämlich diejenigen Wahrscheinlichkeiten, für die alle bekannten Bewertungen des Benutzers fest sind und für die gesuchte Ressource jeder mögliche diskrete Wert eingesetzt wird. Das wären dann beim hier verwendeten Wertebereich die Ausgänge 1, 2, 3, 4 und 5.

$$\begin{aligned}&\mathrm{Wk}_{uid,iid}\left(C=c \wedge Rating_{uid} \wedge rating_{uid,iid}=1\right)=?\\ &\mathrm{Wk}_{uid,iid}\left(C=c \wedge Rating_{uid} \wedge rating_{uid,iid}=2\right)=?\\ &\qquad\ldots\\ &\mathrm{Wk}_{uid,iid}\left(C=c \wedge Rating_{uid} \wedge rating_{uid,iid}=5\right)=?\end{aligned} \tag{3.22}$$

Vorzunehmen ist noch eine Gewichtung nach der wahrscheinlichen Clusterzugehörigkeit. Wie berechnen sich aber diese Wahrscheinlichkeiten? Dazu wird in der Literatur ([WF01], [BHK98]) vorgeschlagen, den EM-Algorithmus[43] zu verwenden ([DLR77]). Der Algorithmus geht von einer anfänglich zufälligen Zuordnung der Benutzer auf die Cluster aus, berechnet dann die Mittelwerte und Varianzen der Verteilungen in jedem Cluster und justiert die Zugehörigkeit der Benutzer zu den Clustern neu. Die Justierung wird mehrfach durchlaufen, bis das Qualitätsmerkmal nur noch geringe Verbesserung aufweist. Anschließend wird abgebrochen und für jeden Benutzer *uid* mit seinem Bewertungsvektor $Rating_{uid}$ steht eine Wahrscheinlichkeit für die Verteilung auf die Cluster bereit, welche ja in (3.22) benötigt wird. Als Qualitätsmerkmal wird eine Likelihood-Funktion *quality* verwendet, die mit steigender Güte der Cluster – Homogenität im Cluster und Heterogenität zwischen Clustern – anwächst. Steigt die Funktion nur noch schwach an, ist ein Maximum erreicht und der Algorithmus bricht ab.

[43] Das „E“ steht für Erwartung und das „M“ für Maximierung.

$$quality = \sum_{uid=1}^{|U|} \log\left(\sum_{j=1}^{k} \mathrm{Wk}\left(C = j\right) \cdot \mathrm{Wk}\left(Rating_{uid} \middle| C = j \right) \right) \tag{3.23}$$

Während des Laufs aktualisiert der Algorithmus kontinuierlich die Mittelwerte μ_c und Varianzen σ_c der *k* Cluster. Auch die Verteilung der Cluster Wk (C=c) wird stets aktualisiert. Die Wahrscheinlichkeit eines Attributs und zusammen eines Benutzers einem gegebenen Cluster anzugehören ist der Schlüssel.

$$\mathrm{Wk}\left(rating_{uid,iid} \middle| C = c \right) = \frac{1}{\sqrt{2\pi}\sigma_{iid,c}} \cdot e^{-\frac{\left(rating_{uid,iid} - \mu_{iid,c}\right)^2}{2\sigma_{iid,c}^2}} \tag{3.24}$$

Zusammengefasst läuft der Algorithmus dann wie folgt ab:

- → Wähle zufällige Werte für Mittelwert μ_c und Varianz σ_c^2 jedes Clusters sowie die Verteilung der k Cluster Wk (C=c)
- → Wiederhole solange, bis Zuwachs von *quality* < Grenzwert
  - → Berechne alle Clusterwahrscheinlichkeiten Wk (C=c | Rating_{uid}) für jeden Benutzer uid
  - → Aktualisiere damit Werte für Mittelwert μ_c und Varianz σ_c^2 jedes Clusters sowie die Verteilung der Cluster Wk (C=c)
- → Verwende Clusterwahrscheinlichkeiten Wk (C=c | Rating_{uid}) zur Vorhersage von Bewertungen

Abbildung 32 - Ablauf des EM-Algorithmus

In [BHK98] wird vorgeschlagen, die Clusterzahl so zu wählen, dass die Likelihood-Funktion *quality* maximiert wird. Dazu muss man verschiedene Clusterzahlen wie 2, 5, 10 und 25 nacheinander durchrechnen und das berechnete Modell mit der besten Anzahl auswählen. Dieses Verfahren wird in einem umfassenden Algorithmus namens AutoClass vollzogen (siehe [CS96]).

Unvorteilhaft ist allerdings, dass das Finden der besten Cluster mit zunehmender Benutzerzahl sehr viel Rechenzeit in Anspruch nimmt. Der Modellkompilierungsprozess ist also eher träger Natur. Sind die Wahrscheinlichkeiten jedoch berechnet und gespeichert, kann die Vorhersage der Bewertung zügig berechnet werden.

Problematisch ist auch der Speicherbedarf, wenn zu viele Attribute in Form von Ressourcenbewertungen vorliegen. In konkreten Implementierungen ist dann eine Reduzierung auf eine feste Zahl der bei allen Benutzern beliebtesten Ressourcen sinnvoll.

Klassifizierungsverfahren

Ein anderer Ansatz ist, die Bewertungsberechnung als ein Klassifikationsproblem aufzufassen. Gegeben der Benutzer hat *m* Ressourcen mit entsprechenden diskreten Werten 1 bis *n* bewertet oder eine Bewertung vorgenommen, dann lässt sich für die gewünschte Ressource *iid* eine Zuordnung zu der Bewertungsklasse 1 bis *n* vornehmen. Dabei wird für die Zugehörigkeit von *iid* zu einer der Klassen eine Wahrscheinlichkeit bestimmt und nach (3.17) der Erwartungswert gebildet.

Die Wahrscheinlichkeiten für die Zugehörigkeit der gewünschten unbewerteten Ressource zu einer Klasse lassen sich durch eine Analyse der Zusammenhänge alle Ressourcenbewertungen der Bewertungsmatrix *Ratings* ermitteln. In [BHK98] wird vorgeschlagen, dazu ein Bayes'sches Netzwerk einzusetzen. Jede Ressource wird darin durch einen Knoten repräsentiert und ein zugehöriger Lernalgorithmus versucht, Beziehungen zwischen den Knoten zu ermitteln. Hier wird nicht weiter auf diesen Lernvorgang eingegangen, da er einerseits recht aufwändig ist und andererseits im weiteren Verlauf der Arbeit auch nicht benötigt wird. In [Heck95] findet sich eine detaillierte Beschreibung des Algorithmus' nebst Beispielen.

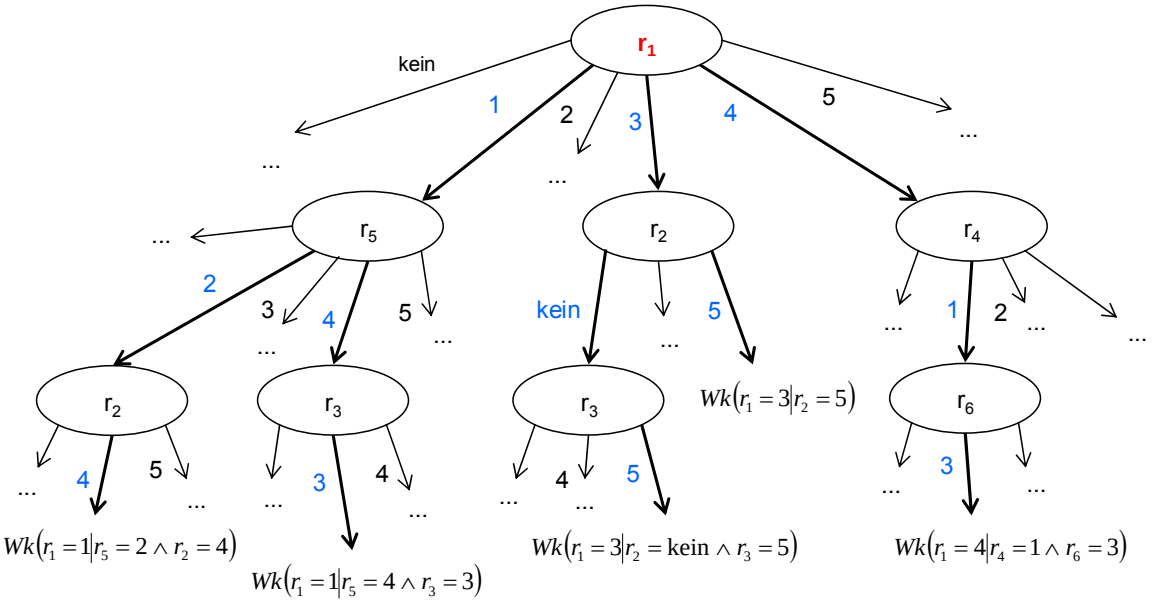

Abbildung 33 - Entscheidungsbaum für Klassifikation von Bewertungen

Wesentlich ist noch, dass der Lernalgorithmus für jede Ressource einen Entscheidungsbaum generiert. In Abbildung 33 ist ein solcher beispielhafter Baum für eine Ressource r_1 dargestellt. Folgt man den Ästen des Baumes, gelangt man zu den gesuchten Wahrscheinlichkeiten für eine komplette Ressourcenbewertung eines Benutzers, wobei tatsächliche Werte und keine Bewertungen auftreten können. Jeder Knoten des Baumes repräsentiert eine Ressource, wobei Ressourcen mehrfach auftauchen, so dass in jedem Ast alle Ressourcen anzutreffen sind. Jede Weggabelung zur Erlangung der gesuchten Wahrscheinlichkeit ist von den vorhandenen Bewertungen des Benutzers abhängig.

Der Vorteil an diesem Verfahren ist, dass die Modellgenerierung und damit die Generierung der Entscheidungsbäume zwar zeitlich aufwändig ist, die Berechnung von Bewertungen jedoch sehr schnell vonstatten geht.

3.2.3 Hybride Techniken

Mit kombinierten Verfahren, die in der Literatur auch als hybride Techniken bezeichnet werden, wird versucht, die Vorteile von speicherbasierten und modellbasierten Algorithmen zu nutzen, dabei aber die Nachteile zu umgehen.

Bei speicherbasierten Techniken sind die nötigen mathematischen Operationen vergleichsweise einfach. Die Berechnung einer einzelnen Bewertung wird aber immer langsamer, wenn mehr Ressourcen und Benutzer vorhanden sind und viele Bewertungen vorliegen. Ursache ist, dass die spei-

cherbasierten Algorithmen auf der gesamten *Ratings*-Matrix operieren und hier ähnliche Benutzer errechnen.

Bei den modellbasierten Verfahren ist der Kompilierungsschritt hingegen aufwändig und mathematisch nicht trivial. Der Aufbau des Modells dauert dadurch entsprechend lange und Änderungen an den Bewertungen werden nur träge erfasst. Die Berechnung einer einzelnen Bewertung erfolgt aber schnell und skaliert auch bei vielen Bewertungen gut, da nicht der gesamte Datenbestand sondern nur das generierte, kompakte Modell als Berechnungsgrundlage dient.

In [SKKR01] beispielsweise wird vorgeschlagen, statt der Bestimmung ähnlicher Benutzer mit der Kosinusfunktion oder dem Pearson-Korrelationskoeffizienten (siehe 3.2.1) die Ähnlichkeit der Ressourcen anhand ihrer Bewertungen zu ermitteln. Zu jeder Ressource wird eine feste Zahl *k* von anderen, ähnlichen Ressourcen ermittelt und in einer Tabelle gespeichert. Das entspricht der Modellgenerierung, die mit den Ähnlichkeitsmaßen sehr schnell erfolgen kann und beispielsweise einmal pro Tag ausgeführt wird.

Zur Berechnung einer Bewertung für eine Ressource werden in der Tabelle alle zu ihr ähnlichen Ressourcen abgerufen. Das ist mit Datenbanktechnik unter Verwendung von Indizes sehr effizient möglich. Die eigentliche Berechnung erfolgt dann als gewichtete Summe derjenigen ähnlichen Bewertungen von Ressourcen, zu denen der aktive Benutzer eine eigene Bewertung abgegeben hat. Das ist die Schnittmenge der gespeicherten ähnlichen Bewertungen und derjenigen, die der Benutzer tatsächlich bewertet hat. Die Zahl *k* der ähnlichen Ressourcen sollte daher nicht zu klein gewählt sein, damit die Schnittmenge nicht allzu oft leer ist, da sonst keine Berechnung erfolgen kann.

Vorteilhaft an diesem Ansatz ist, dass das Verfahren sehr gut skaliert, da durch das Nachschlagen ähnlicher Bewertungen in der Tabelle und das Berechnen der gewichteten Summe nur wenig Rechenzeit verbraucht wird. Es eignet sich daher auch gut für den Einsatz in großen Systemen mit vielen Ressourcen und Benutzern.

3.2.4 Kombination mit anderen Techniken

Ein weiteres Mittel zur Überwindung von Schwächen der kollaborativen Filteralgorithmen ist eine Kombination mit anderen Personalisierungstechniken, vor allem dem Inhaltsbasierten Filtern. Eine Verbindung mit Regelbasierten Filtern ist ebenso möglich (siehe Kapitel 2.4.5). Vor allem geht es darum, bei ungenügend vorhandenen Ressourcenbewertungen alternative Datenquellen bereitzustellen. Die Schwachstellen werden in Abschnitt 3.4 noch näher beschrieben. Mit [Bau99] und [MMN01] liegen zwei Arbeiten vor, die Inhaltsbasierte und Kollaborative Filter miteinander kombinieren.

Ein Konzept dazu ist, prinzipiell mit einem Kollaborativen Filter auf der Bewertungsmatrix *Ratings* zu arbeiten. Unbewertete Ressourcen fließen nicht in die Berechnung ein. Die fehlenden Werte können jedoch durch einen Inhaltsbasierten Filter bestimmt werden, wozu ein Benutzer beispielsweise eine Reihe von interessanten Themen festlegt. Die Bewertung einer Ressource wird anhand ihrer Übereinstimmung mit den definierten Themen berechnet. So wird auch mit der Ähnlichkeitsbestimmung im dritten Optimierungsansatz in Kapitel 4.3 Verknüpfung mit Inhaltsbasierten Filtern verfahren.

3.3 Erfolgsmessung

Zur Erfolgsmessung von kollaborativen Filtersystemen werden in der Literatur zwei unterschiedliche Konzepte verwendet (siehe beispielsweise [MRK97] oder [BHK98]). Das erste Verfahren prüft die statistische Genauigkeit von Vorhersageberechnungen und somit die Qualität der Vorhersage. Das andere Verfahren liefert quantitative Aussagen über den Nutzen einer Empfehlung. Grundannahme hierbei ist, dass Ressourcen mit einer hohen Bewertung nützlich für den Benutzer sind, wohingegen schwach bewertete Ressourcen weniger nützlich eingestuft werden.

Bei beiden Verfahren müssen signifikante Mengen an Testdaten in Form von Bewertungen vorliegen, damit eine aussagekräftige Erfolgsmessung möglich ist. Die Testdatenmenge TD besteht aus 3-Tupeln, die jeweils für einen Benutzer und eine Ressource eine Bewertung enthalten. Im in dieser Arbeit verwendeten Modell sind das ganzzahlige Werte von 1 bis 5 und deren Abbildungen auf das reellwertige Intervall [0,1]. Zur Erfolgsmessung wird die Testdatenmenge zunächst zufällig in einen Arbeitsteil WD für die Verwendung in den kollaborativen Filteralgorithmen und einen Teil ED zur Evaluierung unterteilt. Bei den modellbasierten Algorithmen wird das Modell mit den Arbeitsdaten trainiert.

Anschließend können die eigentlichen Testläufe stattfinden. Dazu werden alle Benutzer-Ressource-Kombinationen der Evaluierungsmenge ED durchlaufen und für sie die vorhergesagten Bewertungen nach der *pred*- und nach der *toplist*-Operation berechnet. Die errechneten Ergebnisse werden mit den tatsächlichen Bewertungen aus ED verglichen und so der Erfolg festgestellt.

Die statistische Genauigkeit der Berechnungen kann mit Hilfe des mittleren absoluten Fehlers beurteilt werden. Je näher die berechneten Bewertungen bei den tatsächlichen Bewertungen liegen, desto kleiner fällt der Fehler aus. *pred(uid,iid)* berechnet die Vorhersage für *uid* und *iid*, während $rating_{uid,iid}$ die tatsächliche Bewertung aus den Evaluierungsdaten ist.

$$\mathrm{MAE} = \frac{\sum_{(uid,iid,rating_{uid,iid})\in ED} \left| pred(uid,iid) - rating_{uid,iid} \right|}{\left| ED \right|} \tag{3.25}$$

Alternativ kann die mittlere Abweichung für jeden Benutzer einzeln ausgerechnet und dann über die Abweichungen aller Benutzer gemittelt werden. Ebenso kann der mittlere quadratische Fehler berechnet werden. Unabhängig von der jeweiligen Ausprägung ist die Aussage aber gleich: Diejenigen Algorithmen, die näher an den von den Benutzern tatsächlich vorgenommenen Bewertungen liegen, sind die besseren.

Einen anderen Aspekt der Erfolgsmessung beleuchtet das zweite Verfahren oder besser Konzept. Hier wird nicht direkt die berechnete Bewertung einer Ressource betrachtet, sondern ob für den Benutzer nützliche Ressourcen vorgeschlagen werden. Dieser Ansatz lehnt sich an die Qualitätsparameter Precision und Recall aus dem Information Retrieval an (siehe [BR99], Kapitel 3.2.1 und [BHK98]). Recall ist der Prozentsatz an gelieferten Ressourcen in Bezug auf alle relevanten Ressourcen im Datenbestand. Optimal wäre dabei, dass alle relevanten Ressourcen geliefert werden. Precision hingegen ist der Prozentsatz der tatsächlich gelieferten und relevanten Ressourcen an allen gelieferten Ressourcen. Werden beispielsweise auf eine Suchanfrage hin zwar viele relevante Ressourcen geliefert, aber gleichzeitig auch viele nicht passende, dann sinkt der Precision-Wert.

Beim Kollaborativen Filtern liefert die *toplist*-Operation eine nach absteigender Bewertung sortierte Liste von Ressourcen. Lässt man die *toplist*-Operation alle für einen Benutzer in der Evaluierungsmenge ED enthaltenen Ressourcen sortieren, kann diese Sortierung mit der Reihenfolge der Ressourcen nach ihrer tatsächlichen Bewertung verglichen werden. Werden die Ressourcen, die in

der Evaluierungsmenge hoch bewertet sind, am Anfang der berechneten Liste geliefert, ist der Algorithmus gut. Als Grenzwert zur Unterscheidung von hoch und niedrig bewerteten Ressourcen kann die Standardbewertung bzw. in Englisch der default vote herangezogen werden. Typischerweise liegt er in der Mitte des Wertebereichsintervalls. In dieser Arbeit ist die Standardbewertung d = 0.5, die auch bei fehlenden Bewertungen herangezogen werden kann, so dass empfohlene Ressourcen, die in der Evaluierungsmenge nicht enthalten sind, in Formel (3.26) vernachlässigt werden.

Ein weiterer Aspekt bei der Betrachtung der Nützlichkeit einer Empfehlungsliste ist, dass Benutzer eher die ersten als die letzten Elemente verwenden und als hilfreich empfinden. In [BHK98] wird die Nützlichkeit der später in der Liste auftauchenden Elemente exponentiell abgewertet. Mit α kann der Grad der Abwertung gesteuert werden, wobei die Abwertung umso stärker eintritt, je höher der Wert ausfällt. Ein Wert von 5 beispielsweise entspricht die Annahme, dass das fünfte Element noch mit Wahrscheinlichkeit ½ abgerufen wird. Zusammengesetzt ergibt sich so eine Formel für die Qualität und Nützlichkeit einer für einen Benutzer individuell berechneten Ressourcenliste.

$$toplist\_quality_{uid} = \sum_{i}^{n} \frac{\max\left(rating_{uid,iid(i)} - d, 0\right)}{2^{(i-1)/(\alpha-1)}} \tag{3.26}$$

Als Gütekriterium für den gesamten Datenbestand wird die gewichtete Summe der *toplist_quality*-Werte für alle Benutzer der Evaluierungsmenge ED berechnet. Die Gewichtung erfolgt durch die Division der Summe derjenigen *toplist_quality*-Werte, die sich ergibt, wenn man die jeweils am höchsten bewerteten Ressourcen der Evaluierungsmenge für diesen Benutzer verwendet.

$$toplist\_quality\_all = \frac{\sum_{uid \in U_{ED}} toplist\_quality_{uid}}{\sum_{uid \in U_{ED}} toplist\_quality\_eval_{uid}} \tag{3.27}$$

In der Literatur wird für Qualitätsmessungen in Form von Nützlichkeitsbewertungen auch die Receiver Operation Characteristic (ROC) verwendet (siehe beispielsweise [MMN01] oder [HKBR99]).

3.4 Schwierigkeiten des Kollaborativen Filterns mit Lösungsmöglichkeiten

Die zu Eingang dieses Kapitels beschriebenen Vorzüge des Kollaborativen Filterns machen es für die Personalisierung von Webseiten interessant. Insbesondere die Möglichkeit, ohne die Kenntnis des Inhaltes Empfehlungen für die Anzeige von alternativen Informationen aussprechen zu können, ist hervorzuheben. Auch die Benutzerzentriertheit und das Bewerten von Ressourcen sind spezifische Merkmale des Kollaborativen Filterns. Allerdings sind diese Fähigkeiten und Besonderheiten auch Ursache für eine Reihe von Schwierigkeiten, die bei Anwendungen des Kollaborativen Filterns auftreten. Im Folgenden werden daher die elementarsten dieser Probleme herausgegriffen und wenn möglich Verbesserungsvorschläge aufgeführt.

3.4.1 Neuer-Benutzer-Problem

Neu im Informationssystem angekommene Benutzer sind zunächst unbekannt und haben noch keine Ressourcenbewertungen abgegeben. Die Algorithmen für das Kollaborative Filtern greifen nicht, da keine Ähnlichkeit zu den anderen Benutzern des Systems ermittelt werden kann oder der neue Benutzer noch keinem Cluster zugeordnet werden kann. Auch andere Probleme sind je nach verwendetem Algorithmus möglich.

Abhilfe schafft, den Benutzer bei der erstmaligen Verwendung des Systems eine Anzahl von Ressourcen bewerten zu lassen. Die Anzahl kann sich proportional zum Umfang des gesamten Datenbestandes verhalten. Im MovieLens-Projekt[44] müssen beispielsweise zunächst 15 Filme bewertet werden, bevor der Benutzer mit dem System arbeiten kann.

Eine andere Möglichkeit ist die Einbeziehung von Profildaten, die der Benutzer beim Anlegen seines Zuganges angegeben hat. Beispielsweise demographische Angaben wie Alter und Wohnort oder die Auswahl von inhaltlichen Präferenzen. In Kombination mit Regel- oder Inhaltsbasierten Filtern könnte so die Zeit bis zur Verfügbarkeit erster Bewertungen – implizit wie explizit – überbrückt werden.

Bei beiden Ansätzen dienen vorab erfragte Daten als Lösung für das Neuer-Benutzer-Problem.

3.4.2 Kaltstart-Problem

Wird ein Informationssystem installiert und dem anvisierten Personenkreis zugänglich gemacht, sind die Informationen zwar bereits vorhanden, es liegen aber noch keine Bewertungen durch die Benutzer vor. Vor allem die ersten Benutzer des Systems werden auf Kollaborativen Filtern basierende Personalisierungskonzepte nicht sinnvoll nutzen können, da insgesamt zu wenige Bewertungen vorliegen – viele Ressourcen wurden vermutlich noch gar nicht bewertet.

Bei den nachbarschaftsbasierten Algorithmen wird es dadurch schwierig, benachbarte Benutzer des aktiven Benutzers zu finden, da die Schnittmengen der gemeinsam bewerteten Ressourcen häufig leer sein werden. Bei den modellbasierten Algorithmen wird es ebenfalls schwierig sein, aufgrund der dünnen Datenbasis Zusammenhänge zwischen Benutzern und Ressourcen zu erkennen.

Im Laufe des Betriebes steigt die Zahl von Bewertungen jedoch an und die Algorithmen können greifen. Bei neu hinzugekommenen Ressourcen besteht jedoch wieder das Problem, dass für diese keine Bewertungen vorliegen. Man spricht vom Erster-Bewerter-Problem (bzw. First-Rater-Problem), das auch für spezifischere Ressourcen gilt, die zwar schon länger im System sind aber bislang von keinem Benutzer bewertet wurden. Man könnte zwar auf die Idee kommen, diese Elemente aus dem System zu entfernen, da sie nicht genutzt werden. Aber möglicherweise gibt es aktuelle Benutzer, die die entsprechenden Ressourcen noch nicht gesehen haben oder zukünftige Benutzer, die diese interessant finden und eine Wertung vornehmen (siehe [MMN01]).

Stehen weitere Informationen zu den Ressourcen zur Verfügung, beispielsweise Stichworte oder Zusammenfassungen, können sowohl das Kaltstart- als auch das First-Rater-Problem gemildert werden, wenn Inhaltsbasierte Filter hinzugezogen werden. Für die unbewerteten Ressourcen können dann Empfehlungen ausgesprochen werden, die auf einer inhaltlichen Analyse beruhen.

[44] Siehe „movielens – helping you find the right movies“ – http://movielens.umn.edu

3.4.3 Dünner Datenbestand

Noch genereller als das Kaltstart-Problem ist die „Sparsity“ oder der „dünne Datenbestand“ in der Bewertungsmatrix. Da große Informationssysteme tausende oder gar Millionen von Ressourcen enthalten, aber viele Benutzer im Vergleich dazu nur sehr wenige Bewertungen abgeben, ist es schwierig, Benutzer mit gemeinsam bewerteten Ressourcen zu finden. In [SKKR00] wird aufgeführt, dass Benutzer in großen Systemen wie Amazon.com weit unter 1% der Ressourcen bewertet haben. Bei einer Million Büchern wären das immerhin schon 10.000 Bewertungen. Eine Zahl, die wohl nur die eifrigsten Benutzer erreichen werden.

Schwierig ist es zudem, von der Ähnlichkeit zweier Benutzer A und B und der Ähnlichkeit der Benutzer B und C auf die Ähnlichkeit von A und C zu schließen – also die Transitivität. Denn wenn A und B ähnlich sind, haben sie eine Reihe von gemeinsam bewerteten Ressourcen vorzuweisen. Genauso haben B und C eine Menge gemeinsamer bewerteter Ressourcen. Bei sehr vielen Ressourcen und einem dünnen Datenbestand kann aber nicht gefolgert werden, dass die Mengen gemeinsam bewerteter Ressourcen gleich sind. Die Transitivität ist also nur mit hoher Unsicherheit gegeben.

Für die Problematik wurden in der Fachwelt Lösungsvorschläge in verschiedenen Variationen vorgeschlagen. Eine der Hauptrichtungen ist, wo möglich nicht nur Kollaborative Filter sondern auch Inhaltsbasierte einzusetzen. Diese Kombinationen füllen auf die eine oder andere Weise die Lücken, die durch fehlende Bewertungen auftreten (siehe z. B. [MMN01] und [Bau99]). Eine weitere Herangehensweise ist, neben expliziten Ressourcenbewertungen auch das Verhalten des Benutzers im Informationssystem zu beobachten und so implizite Bewertungen zu erhalten. Dadurch gibt es potentiell mehr Bewertungen als wenn nur explizite verwendet werden, da Benutzer nicht interaktiv vorgehen müssen sondern das System wie gewohnt benutzen und die Bewertungen im Hintergrund generiert werden (siehe z. B. [HF01]). Eine dritte Richtung ist, die personalisierten Empfehlungen nicht anhand der Benutzer- sondern der Ressourcenähnlichkeit zu berechnen. Dabei werden zumindest alle bewerteten Ressourcen zueinander in Bezug gesetzt und die Anzahl von möglichen Ressourcen, die für einen Benutzer in Frage kommen, wird erhöht (siehe [SKKR01]).

3.4.4 Geringe Nachvollziehbarkeit für Benutzer

In personalisierten Informationssystemen mit Kollaborativen Filtern erhalten die Benutzer Listen der besten *n* Ressourcen, Suchergebnisse können nach den Präferenzen sortiert werden oder zu einzelnen Ressourcen werden die vorhergesagten Bewertungen in Form einer Nützlichkeit ausgegeben. Im Vergleich zu Inhalts- oder Regelbasierten Filtern ist die Erklärung, warum gerade diese Ressourcen empfohlen wurden, nicht so einfach zu vermitteln. Bei Inhaltsbasierten Filtern wählt der Benutzer beispielsweise eine Reihe von Themen und erhält bevorzugt Ressourcen zu den gewählten Themengebieten. Der Grund für die Empfehlung wird dem Benutzer durch die sichtbaren Inhalte deutlich und die Anzeige der gewählten Themen visualisiert die Benutzerpräferenzen.

Ein Bewertungsvektor wie beim Kollaborativen Filtern über alle vom Benutzer bewerteten Ressourcen als Zahlenkolonne hingegen ist dafür wenig hilfreich, da keine Abstraktion auf sofort einsichtige Themengebiete oder Schlüsselwörter stattfindet. Auch der Zusammenhang zwischen einer Empfehlung, die anhand der Ähnlichkeit des aktiven Benutzers zu anderen Benutzern ausgesprochen wird, ist nicht erkennbar, da der zugrunde liegende Algorithmus nicht erfasst werden kann. In Folge kann die Akzeptanz für die empfohlenen Ressourcen niedriger sein als bei Inhalts- oder Regelbasierten Filtern. Vor allem auch, wenn weniger sinnvolle Empfehlungen gegeben werden

oder wenn Empfehlungen aus Themenbereichen erfolgen, die zwar für den Benutzer interessant sind, aber mit denen er nicht gerechnet hat, weil sie weniger offensichtlich sind.

Zudem kann der Prozess beim Sammeln von impliziten Bewertungen zu einem Misstrauen gegenüber dem Personalisierungssystem führen, da die Privatsphäre bei Benutzern gestört wird. In allen Fällen kann das Vertrauen in die gebotenen Empfehlungen und die personalisierte Informationsaufbereitung gesteigert werden, wenn der Benutzer mit Daten versorgt wird, die die Entscheidung für die angebotenen Ressourcen belegen. Ein solches Erklärungsinstrument ist die Ausgabe eines Histogramms der Bewertungen der benachbarten Benutzer für eine vorhergesagte Ressourcenbewertung.

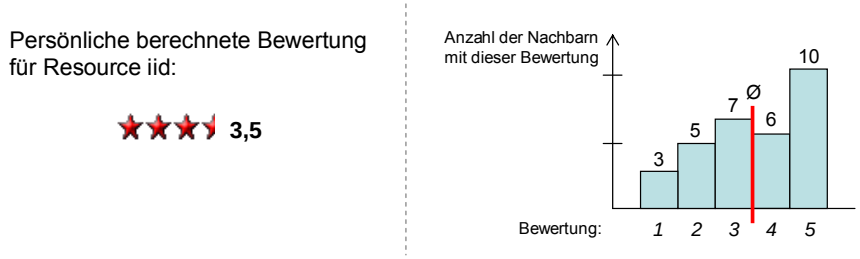

Abbildung 34 - Histogramm der Bewertungen der Nachbarn

Daneben werden in [HKR00] auch weitere Erklärungsverfahren vorgestellt und in Experimenten mit Testbenutzern gegeneinander abgewägt, wobei die Erklärung via Histogramm als besonders nützlich eingestuft wurde.

3.4.5 Performance und Skalierbarkeit

Schnelle Antwortzeiten sind im Web generell unabdingbar. Klicken Benutzer auf Links, sollte der Seitenaufbau je nach Zielseite in annehmbarer Zeit erfolgen, da die Benutzer auf ein solches Verhalten konditioniert sind. Bei einfachen Listenausgaben oder Detailansichten sind Zeiten unter einer Sekunde sinnvoll, bei Suchanfragen oder dem Anstoßen von Prozessen wie dem Versenden von E-Mail toleriert der Benutzer hingegen längere Reaktionszeiten. Überschreitet die Zeit zum Aufbau der Seite aber einen je nach Benutzer individuell leicht variierenden Schwellwert, vermutet der Benutzer Fehlfunktionen der Website. In letzter Konsequenz kehrt er dem Webangebot sogar den Rücken zu, was bei Online-Shops nicht im Sinne des Händlers ist. Oder er kann im Falle eines Informationssystems seine Arbeiten nicht zufrieden stellend durchführen.

Die Zeit zur Berechnung der unpersonalisierten Ausgabe der Webseitendaten muss bereits im nötigen Zeitrahmen liegen. Die kollaborativen Filteralgorithmen brauchen jedoch zusätzliche Rechenzeit, um die Personalisierung der Inhalte vorzunehmen. Entsprechend sind die Algorithmen sehr effizient zu programmieren. Für eine kleine Anzahl von Ressourcen und Benutzern ist das weniger problematisch. Bei großen Datenmengen müssen die Algorithmen jedoch gut skalieren können. Modellbasierte Algorithmen haben in dieser Beziehung einen Vorteil, da die Modellgenerierung zwar langsam ist, die Abfrage einzelner Bewertungen aber schnell erfolgt. Speicherbasierte Algorithmen müssen jedoch bei großen Datenmengen Ähnlichkeiten zwischen sehr vielen Benutzern berechnen und große Teile der Bewertungen verarbeiten.

Abhilfe schafft das Speichern von vorberechneten Daten in einem Cache. Die nötigen vorhergesagten Berechnungen mit der *pred*- und *toplist*-Operation (siehe Kapitel 3.2 Eingesetzte Verfahren) können aus dem Cache bedient werden. Die Werte für den Cache können im Batchbetrieb zu lastärmeren Zeiten berechnet werden. Wie bei allen Cache-Lösungen muss allerdings berücksich-

tigt werden, wann eine Aktualisierung erfolgen muss, weil die Daten im Cachespeicher nicht mehr gültig sind. Beim Kollaborativen Filtern passiert das, wenn neue Bewertungen gemacht werden. Allerdings muss nicht bei jeder Änderung eine Aktualisierung erfolgen, da einzelne neue Bewertungen die berechneten Bewertungen nicht wesentlich beeinflussen müssen. Eine zeitlich verzögerte oder von der Anzahl der Änderungen abhängige Aktualisierung des Caches ist möglich. Dabei kann auch eine nur partielle Änderung des Caches vorgenommen werden.

3.4.6 Schwache Reaktion auf Änderungen im langlebigen Benutzerinteresse

Ändern sich die Präferenzen des Benutzers im Laufe der Zeit, werden diese Änderungen in den Bewertungen und in Folge auch in den empfohlenen Ressourcen nicht sichtbar. Insbesondere wenn das Interesse an ehemals für interessant befundenen Themen schwindet, wird das Informationssystem ohne weiteres nicht darauf aufmerksam. Neue Interessen hingegen werden durch implizite wie explizite Bewertungen zeitnah erfasst.

Eine Interaktionsmöglichkeit stellt das wiederholte, explizite Bewerten von Ressourcen dar, da hiermit die Wertung auch nachträglich verändert und an neue Präferenzen angepasst werden kann. Da Benutzer in Informationssystemen generell wenige Bewertungen abgeben, erscheint dieses Verfahren nur bedingt nützlich – wenn auch nicht überflüssig. Denkbar wäre daher alternativ, explizite und implizite Bewertungen zu kombinieren und zu verfolgen, ob ehemals explizit gut bewertete Ressourcen im zeitlichen Verlauf noch verwendet und für gut befunden werden. In [HF01] wird dieser Ansatz vorgeschlagen, aber nicht ausgeführt. Weitere Überlegungen zu diesem Ansatz daher in Kapitel 4.2 Einbeziehung von Zeit.

3.4.7 Geringe Behandlung von Ausreißern und abweichendem Verhalten

Ein eher generelles Personalisierungsproblem ist, dass Ausreißer im Benutzerverhalten nicht erkannt oder überbewertet werden. Insbesondere bei impliziten Bewertungen, die jeden Aufruf einer Ressource erfassen, wird eine versehentlich angewählte Ressource überbewertet. Auch aus Unachtsamkeit oder Unkenntnis fehlerhaft vorgenommene explizite Bewertungen ohne Korrekturmöglichkeit durch den Benutzer führen zu falschen Bewertungen. Hiergegen gibt es vermutlich wenige Lösungsansätze, die Problematik ist allerdings auch eher unproblematisch, da der Benutzer letztlich selbstständig entscheidet, welche Bewertungen er vornimmt.

Schwerwiegender ist das Problem, dass ein temporär anderes Verhalten eines Benutzers nicht gut erfasst werden kann, da die Bewertungen und das Benutzerprofil ein einziges Bild des Benutzers abgeben. Ein Beispiel für einen solchen Vorfall sind personalisierte Online-Shops. Nutzt ein Benutzer den Shop in erster Linie für berufliche Buchkäufe, ist ein entsprechendes fachliches Profil erlernt worden. Möchte der gleiche Benutzer jedoch ein Buch für den privaten Bedarf erstehen, personalisiert der Shop schlecht und liefert im Extremfall gar keine geeigneten Artikel, obwohl sie im Buchbestand vorhanden wären. Eine triviale Lösung wäre das temporäre Abschalten der Personalisierung, ein anderer Weg die Bereitstellung von Unterprofilen mit unterschiedlichen Präferenzen. Das Problem gilt jedoch nicht nur für kollaborative Filteralgorithmen sondern allgemein für alle Personalisierungsverfahren.

4 Optimierung

Nachdem in den vorangegangenen Kapiteln die Grundlagen von Informationsportalen und Online-Shops dargelegt und Personalisierungsstrategien mit Kollaborativen Filtern vorgestellt wurden, stellt sich jetzt die Frage, wie Optimierungen daran in praktischer Umsetzung vorgenommen werden können.

Personalisierung dient gemeinhin als Mittel, Effizienzsteigerungen zu erreichen. Ziel ist dabei, dass bei Informationsportalen ein schnellerer Zugriff auf die gewünschten und relevanten Informationen möglich wird. Dies führt zu einer Zeitersparnis und höheren Arbeitszufriedenheit. Vor allem steht der Anwender im Vordergrund, während der Betreiber des Systems einen sekundären Nutzen zieht. Hier stehen stellvertretend Arbeitgeber und Portalbetreiber. Bei E-Commerce-Anwendungen wie Online-Shops und Werbung – Stichwort Bannerschaltung und Newsletter-Marketing – wird vordergründig die Befriedigung von Kundenbedürfnissen angestrebt. Hier spielt jedoch viel stärker der Nutzen für den Systembetreiber, also den Händler oder Vermarkter, eine Rolle.

Einfache Personalisierungskonzepte wie Checkbox-Personalisierung erlauben die grundlegende Anpassung des Informationssystems an die Bedürfnisse der Benutzer. Auch die Umsetzung für den Betreiber ist vergleichsweise einfach, da er die Erzeugung der typischerweise ohnehin dynamisch generierten Websiteinhalte mit einigen weiteren Parametern ausstattet, die von den Benutzerprofilen der Anwender bestimmt werden. Anders sieht es bei den komplexeren Personalisierungskonzepten wie dem Regel- oder Inhaltsbasierten Filtern aus. Hier sind umfangreiche Vorarbeiten für eine sinnvolle Generierung der Regeln oder eine Kategorisierung der Inhalte erforderlich. Automatische Verfahren wie Data Mining und Text Mining unterstützen den Betreiber zwar, aber erfordern trotzdem einen hohen manuellen Aufwand für die Pflege der Systeme.

Personalisierung mit Kollaborativen Filtern hingegen kann wie in den vorangegangen Kapiteln dargelegt unabhängig und unbeaufsichtigt arbeiten. Sowohl die direkte Integration von Grund auf beim Aufbau neuer Informationssysteme als auch eine spätere Nachrüstung sind möglich und können so nahezu jedes System personalisierbar machen. Hier liegt eine der großen Stärken von Kollaborativem Filtern. Allerdings erhält man diesen Vorzug nicht umsonst. Schlechtere Qualität bei dünnem Datenbestand, neuen Benutzern und neuen Inhalten sind wie beschrieben die Hauptprobleme (siehe 3.4 Schwierigkeiten des Kollaborativen Filterns mit Lösungsmöglichkeiten). Im Folgenden wird es daher darum gehen, wie ein Teil der Probleme beim Kollaborativen Filtern gelöst werden kann und bessere Personalisierungserfolge erzielt werden. Hier werden folgende Konzepte bearbeitet:

- Konzept 1: Verbreiterung der Datenbasis
- Konzept 2: Einbeziehung von Zeit zur Berücksichtigung von Interessensänderungen
- Konzept 3: Verknüpfung mit Inhaltsbasierten Filtern

Die Konzepte greifen auf unterschiedliche Weise ineinander und versuchen so zusammen eine Optimierung der Personalisierung im Internet durch Kollaboratives Filtern zu erreichen. Die Konzepte werden im Folgenden vorgestellt und enthalten neben mathematischen Veranschaulichungen auch Hinweise zur praktischen Umsetzung.

4.1 Verbreiterung der Datenbasis

Beim Kollaborativen Filtern stellen die Bewertungen von Ressourcen durch Benutzer die Datengrundlage dar, nach der ähnliche Ressourcen berechnet werden. Man geht davon aus, dass Benutzer, die Ressourcen ähnlich bewerten auch ähnliche Präferenzen haben. Bei klassischen Algorithmen wird für die Bewertung nur eine Datenquelle herangezogen. In der Literatur wird überwiegend von explizit durch den Benutzer vorgenommenen Bewertungen ausgegangen, die durch Auswahl eines numerischen oder symbolischen Wertes entstehen (siehe 2.3 Benutzerprofile als Personalisierungsgrundlage). Alternativ wird ausschließlich auf implizite Bewertungen zurückgegriffen, die aus der Anzeigehäufigkeit der Ressourcen berechnet werden.

In beiden Fällen werden Informationsquellen unbeachtet gelassen, die unter Hinzunahme zu einem präziseren Bild der Benutzerpräferenzen führen würden. In Folge bleiben viele Ressourcen unbewertet, obwohl Benutzer implizite oder explizite Bekundungen über ihren Nutzen abgegeben haben. Dieser Optimierungsansatz geht daher davon aus, dass prinzipiell alle Aktionen der Benutzer – explizite wie implizite – wertvoll sind und berücksichtigt werden sollten. Ziel ist, dass die Anzahl der bewerteten Ressourcen durch die Verbreiterung der Datenbasis quantitativ ansteigt. Die Vorhersageberechnungen sollen so als Ergebnis qualitativ verbessert werden.

Bei aktiven Benutzern fällt die Verkleinerung des dünnen Datenbestandes zwar zunächst weniger ins Gewicht, da sie sowieso mehr explizite Bewertungen vornehmen und sie insgesamt mehr mit dem System interagieren. Bei seltener Nutzung steigt jedoch die Zahl der Informationen im Benutzerprofil deutlich, wenn die im System hinterlassenen Datenspuren berücksichtigt werden. Letztlich profitieren alle Anwender, da die verschiedenen Algorithmen mit einer dichteren Bewertungsmatrix arbeiten können und die Idee des Kollaborativen Filtern gerade die ist, dass die Bewertungen anderer Benutzer zur Berechung von Vorhersagen für einen einzelnen Benutzer dienen.

Zu beachten ist, dass der Optimierungsansatz Modifikationen an der Datenquelle und nicht an den Filteralgorithmen vornimmt. Ziel ist dabei, dass die Eingabe für die kollaborativen Filteralgorithmen nach wie vor ausschließlich aus Bewertungen besteht und die Algorithmen als Blackbox behandelt werden können. Vorteilhaft ist daran, dass man den gewünschten Filteralgorithmus je nach Anforderung leicht austauschen kann.

4.1.1 Datenquellen

Je nach Informationssystem bieten sich verschiedene Datenquellen zur Verwendung an. Die Datenquellen können nach ihrer Nützlichkeit und Aussagekraft für eine Gesamtbewertung einer Ressource durch den Benutzer unterschieden und sortiert werden. Zu bedenken ist dabei, dass die Annahmen bezüglich der Aussagekraft generell mit Unsicherheit behaftet sind. Die Modellparameter sind sinnvollerweise nach empirischen Tests zu justieren, wovon hier jedoch abgesehen wird, da dies den Rahmen dieser Arbeit sprengen würde. Stattdessen wird mit den naheliegendsten Annahmen gearbeitet.

Allgemein nützlich sind explizite Bewertungen auf einer numerischen oder kategorischen Skala. Sie besitzen eine hohe Aussagekraft, da der Benutzer hier aktiv und bewusst eine Interessensbekundung vornimmt. Entsprechend sollten sie in einem Profil nicht fehlen. Leider neigen Anwender dazu, sie zu ignorieren. Einerseits besteht eine psychische Barriere, eine konkrete Aussage zu treffen. Andererseits bedeutet das explizite Bewerten eine Mehrarbeit durch zusätzliche Mausklicks, die in der Schnelligkeit der Informationsaufnahme vermieden wird.

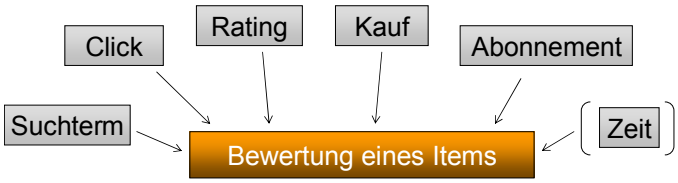

Abbildung 35 - Verschiedene Datenquellen zur Ressourcenbewertung

Anzeigehäufigkeiten oder Clicks auf Detailseiten zu Ressourcen können hingegen ohne das Zutun des Anwenders ausgewertet werden. Eine Einblendung kann als Zustimmung betrachtet werden, jedoch auch aus einem Bedienungsfehler resultieren. Häufigere Einblendungen der gleichen Ressourcen deuten allerdings auf eine positive Präferenz hin. Ähnlich kann die Suche gewertet werden. Tippt ein Benutzer eine Suchanfrage ein, wünscht er gezielte Informationen. Die gelieferten Suchergebnisse können so implizit in die Bewertung einfließen. Problematisch ist hierbei, dass die Relevanz der Suchergebnisse je nach Suchalgorithmus niedrig sein kann und die gefundenen Ressourcen für den Benutzer wenig nützlich sind. Häufigeres Suchen nach den gleichen oder ähnlichen Begriffen deutet auf eine hohe Benutzerpräferenz hin.

Abonnements von Newslettern zu bestimmten Themen oder mit Checkbox-Personalisierung gesetzte Filter können zusätzlich herangezogen werden. Schwierig ist je nach Informationssystem die Verknüpfung der gewählten Optionen mit einzelnen Ressourcen, beispielsweise durch Kategorien oder Inhaltsbasierte Filter. Hier können Ähnlichkeitsmaße zwischen Ressourcen herangezogen werden und als abgeleitete Bewertungen in das Benutzerprofil einfließen. Hat ein Benutzer z. B. eine Ressource explizit bewertet, können alle ähnlichen Ressourcen die gleiche abgeleitete und nach der Ressourcenähnlichkeit gewichtete Bewertung erhalten.

In Online-Shops und bei kostenpflichtigen Inhalten sind auch Käufe und Rücksendungen ein starker Gradmesser für eine Präferenz: Geld wird nur ausgegeben, wenn der Kunde von der Ware überzeugt ist. Eine gewisse Unsicherheit, ob das gekaufte Produkt tatsächlich gefällt, bleibt natürlich. Hier können Rücksendungen helfen, negative Präferenzen für Artikel auszudrücken. Ebenso ist eine Nachgeschaltete Kundenbefragung möglich, aber aufwändig.

Eine weitere, in Abbildung 35 noch eingeklammerte Quelle ist die Zeit, die im Folgekapitel 4.2 einbezogen wird. Zunächst geht es ohne zeitliche Komponente weiter.

4.1.2 Erweiterung des Datenmodells

Das in Kapitel 3.2 eingeführte Datenmodell für Bewertungen wird für die Berücksichtigung der verschiedenen Bewertungsquellen erweitert. Hinzu kommt ein Attribut *creationdate* für den Zeitpunkt der Bewertung und den Typ *type* der Bewertung.

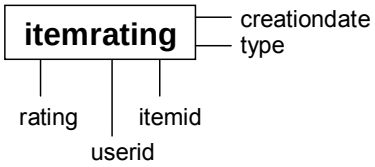

Abbildung 36 - Entität der erweiterten Ressourcenbewertung

Formal wird im Folgenden für eine Bewertung nicht mehr nur der numerische Wert sondern ein 3-Tupel gesetzt.

$$rating_{uid,iid}^{type} = [type; creationdate; value] \quad (4.1)$$

Als Typ werden die Bezeichner *explicit*, *click*, *search*, *similaritem*, *purchase*, *return*, *subscription* und *similaritem* verwendet. Idee ist, dass die Daten getrennt gesammelt und aufbewahrt werden und anschließend eine aggregierte Bewertung berechnet wird, wofür der Typ *aggregated* steht. Die in Kapitel 5 beschriebene Implementierung lehnt sich an diese Bezeichnungen an.

Ferner ist zu beachten, dass der Wert der Bewertung eine reelle Zahl zwischen 0 und 1 einschließlich darstellt. Die Darstellung in einem Informationssystem muss dem natürlich nicht entsprechen. Zur Verrechnung werden aber alle gesammelten Bewertungen auf dieses Intervall normiert. In der Mitte des Intervalls liegt die Standardbewertung 0,5, die – wie schon in vorangegangenen Kapiteln erwähnt – eingesetzt werden kann, wenn keine Bewertung vorliegt oder wenn eine neutrale Bewertung vorgenommen werden soll.

4.1.3 Herkunft der Datenquellen

Vorteilhaft beim Einsatz von impliziten Bewertungen ist, dass sie im Informationssystem gewissermaßen als Abfallprodukt anfallen. So wird ohnehin schon die Navigation des Benutzers mit Protokollkomponenten verfolgt, die zur allgemeinen Erfolgsmessung und zu technischen Zwecken dienen. Für implizite Bewertungen sind dabei aber nicht der verwendete Browser oder das Herkunftsland des Benutzers von Interesse, sondern wie oft und wann er eine Ressource angesehen hat. Diese Daten werden mit entsprechenden Analysewerkzeugen gewonnen und so gespeichert, dass ein leichter Zugriff darauf möglich ist, um die Bewertungen für eine Ressource und einen Benutzer abzufragen (siehe auch 2.3.4 Gewinnen von dynamischen Profildaten).

Eine Personalisierung mit Kollaborativen Filtern kann daher auch leicht in bestehende Systeme integriert werden, wenn man sich die ohnehin erzeugten Daten zu Nutze macht. Bei einem Neuaufbau eines Informationssystems kann man die Daten allerdings auch direkt gewinnen, wenn beispielsweise die Anzeige einer Ressource gerendert wird. Dann kann eine Speicherung der impliziten Bewertung erfolgen. Gleiches gilt auch für Suchmöglichkeiten in Informationssystemen. Die Suchausdrücke und gelieferten Ressourcen werden protokolliert und später ausgewertet. Oder sie werden direkt als implizite Bewertung vermerkt. Bei der direkten Berücksichtigung bietet sich zudem die Möglichkeit einer temporären oder sitzungsbezogenen Personalisierung, wenn man aus dem Benutzerverhalten heraus Empfehlungen ausgibt oder personalisiert Gewichtungen vornimmt. Eine Speicherung erfolgt jedoch nicht und daher fällt ein auf Protokollanalyse arbeitendes System hierfür aus.

Weiterhin werden Käufe und Rücksendungen in Online-Shops in Transaktionssystemen wie der Verkaufsabwicklung oder der Warenwirtschaft protokolliert und können so leicht als Bewertungen dienen. Die Ähnlichkeit von Ressourcen ist hingegen schwieriger zu ermitteln, da prinzipiell eine inhaltliche Analyse der Ressourcen nötig ist. Beispielsweise ist das mit Inhaltsbasierten Filtern möglich, wie in Kapitel 2.4.3 gezeigt wurde, wobei dann alle Vor- und Nachteile Inhaltsbasierter Filter zum Tragen kommen. Bei Grafikdaten ist das beispielsweise nur sehr schwer möglich. Hier können dann andere Verfahren ohne Berücksichtigung des Inhaltes wie die Assoziationsanalyse auf den Zugriffsprotokollen zur Bestimmung der Ähnlichkeit (siehe Kapitel 2.4.2) oder menschliche Experten herangezogen werden, wobei das die aufwändigste Variante ist.

Für explizite Bewertungen müssen geeignete Steuerelemente bereitstehen. Die Daten sind primär für die Personalisierung gedacht – in der Zweitverwertung können sie allerdings für Gesamtdurchschnitte genutzt werden und so nicht angemeldeten Besuchern Hinweise geben.

4.1.4 Zusammenführen der Datenquellen

Um dem Ziel gerecht zu werden, bestehende kollaborative Filteralgorithmen als Blackbox zu nutzen, müssen die verschiedenen Daten zusammengeführt werden. Die Aggregation dient dazu, aus den verschiedenen Bewertungen für eine Ressource und einen Benutzer eine einzige zu machen, die gespeichert werden kann. Technisch arbeiten die Filteralgorithmen dann auf diesen aggregierten Bewertungen. Die Einzelbewertungen bleiben aber erhalten, um Aktualisierungen anhand der Originaldaten zu ermöglichen.

Eine Aufgabe hierbei ist die Ausgestaltung der Gewichtung der verschiedenen Datenquellen, um der Aussagekraft einzelner Typen gerecht zu werden. Explizite Bewertungen sind beispielsweise wertvoll und sollten daher bevorzugt werden, während Bewertungen aufgrund der Ressourcenähnlichkeiten eine geringere Aussagekraft besitzen und als Ergänzung genutzt werden können.

Eine andere Herangehensweise ist, die Daten nicht zu aggregieren, sondern die Filteralgorithmen auf jedem Bewertungstyp einzeln laufen zu lassen und die Ergebnisse am Ende zusammenzuführen. Neben der längeren Laufzeit durch mehrfache Rechnungen ist aber die Frage zu klären, nach welchen Kriterien die Ergebnisse zusammengeführt werden sollen. Letztlich handelt es sich bei den berechneten Vorhersagen nur um Zahlenwerte, die ohne Metadaten wenig Aussagekraft haben. Daher wird im Folgenden der erste Ansatz mit aggregierten Bewertungen verfolgt und zunächst eine Rangfolge der Bewertungstypen definiert.

Explizite Bewertungen

Eine explizit vorgenommene Bewertung durch den Benutzer hat einen sehr hohen Informationsgehalt, da der Benutzer bewusst gehandelt hat. Eine solche Bewertung kann wie ein Machtwort betrachtet werden. Einmal ausgesprochen, gilt es und verhält sich wie ein Veto gegenüber allen anderen Bewertungen[45].

$$rating_{uid,iid}^{\text{explicit}} = [\text{explizit};date;value] \tag{4.2}$$

Käufe und Rücksendungen

Bei Artikeln, die gewöhnlich nur einmal oder selten gekauft werden wie Gebrauchsgütern, beispielsweise Büchern, Filmen oder Autos, wird ein Kauf mit etwas unterhalb der vollen Bewertung von eins bewertet. Kleiner eins deswegen, weil die volle Kundenzufriedenheit nicht mit Sicherheit feststeht. Die Annahme ist gerechtfertigt, da ein Kauf aufgrund der Artikel- und Markenvielfalt per se eine positive Aussage ist[46]. Würde der Kunde ein anderes Produkt präferieren, würde er entsprechend jenes wählen.

$$rating_{uid,iid}^{\text{purchase}} = [\text{purchase};date;0,9] \tag{4.3}$$

Bei Rücksendungen wird mit einem Wert etwas über null eine negative Präferenz gesetzt, da der Kunde mit der Ware unzufrieden ist. Hier wird nicht die null verwendet, weil der Kunde mit der Ware vermutlich nicht vollkommen unzufrieden ist. Andere Gründe wie temporär mangelnde Liquidität sind ebenfalls denkbar.

$$rating_{uid,iid}^{\text{return}} = [\text{return};date;0,1] \tag{4.4}$$

[45] Bei Einbeziehung von Zeit wie im nächsten Kapitel wird von dieser Festlegung abgegangen.
[46] Die Wahl von 0,9 bzw. 0,1 ist willkürlich und könnte durch empirische Prüfungen verbessert werden.

Es würde sich anbieten, die Kunden nach einem Kauf oder einer Rücksendung zu befragen, um eine Bewertung des Produktes zu erfahren. Die Datengrundlage wäre dann eine explizite Bewertung wie oben beschrieben, der Aufwand ist jedoch hoch.

Bei Verbrauchsgütern wie Nahrung, Reinigungsmitteln oder Büromaterial, die wiederholt gekauft werden, kann prinzipiell die gleiche Bewertung erfolgen. Es kann jedoch zusätzlich die Anzahl der Wiederholungskäufe einbezogen werden. Wird ein Artikel nur einmal gekauft, ist die Präferenz des Kunden noch unpräzise. Bei wiederholtem Kauf zeichnet sich aber eine positive Präferenz ab.

Clicks

Ein einzelner Click und die folgende Einblendung einer Ressource sagen nicht viel über die positive oder negative Präferenz des Benutzers aus. Ein Benutzer kann zwar eine Information gezielt abrufen, er kann aber auch zufällig darauf gestoßen sein. Ein falscher Click ist zudem möglich. Ob die Information aber als interessant eingestuft wird, kann nicht ermittelt werden. Der einmalige Click wird daher mit der Standardbewertung registriert.

Wiederholte Clicks auf die gleiche Ressource deuten hingegen auf Gefallen hin. Eine unübersichtliche Navigationsstruktur, die das mehrfache Anklicken eigentlich nicht gewünschter Ressourcen mit dem Ziel der Ansteuerung gewünschter Informationen erfordert, sei hier ausgeklammert. Daher wird die Anzahl der Clicks oder Einblendungen einer Ressource gezählt. Je mehr Einblendungen die Ressource vorzuweisen hat, desto höher wird sie bewertet. Wiederholungsklicks erhöhen die Aussagekraft aber weniger, da schon ab dem zweiten Besuch einer Ressource von gewolltem Abruf auszugehen ist. Vor allem bei großen Informationssystemen mit vielen Ressourcen und angenommener Gleichverteilung ist die Wahrscheinlichkeit zum Abruf einer Ressource sehr gering. Bei *N* Ressourcen im System liegt sie bei

$$\mathrm{Wk}\left(\mathit{Resourcenabruf}\right) = \frac{1}{N} \approx 0 \tag{4.5}$$

Die Wahrscheinlichkeit, die gleiche Ressource zweimal anzusteuern, ist unter Annahme von Unabhängigkeit noch wesentlich geringer (Quadrat von (4.5)), so dass bei Wiederholungsklicks nicht von Zufall sondern von bewusstem Vorgehen auszugehen ist. Für die implizite Bewertung ergibt sich so folgender Ausdruck, wobei *n* für die Anzahl der Clicks des Benutzers auf die Ressource steht:

$$\mathit{rating}_{\mathit{uid},\mathit{iid}}^{\mathrm{click}} = \left[\,\mathrm{click};\, \mathit{date};\, 0{,}9 \cdot \left(1 - \frac{1}{2^n}\right)\right] \text{ mit } n = \#\,\text{Clicks auf iid} \tag{4.6}$$

Der Zeitpunkt *date* der Bewertung wird auf das Datum des letzten Abrufes der Ressource gesetzt. Der Skalierungsfaktor von 0,9 wurde gewählt, um in Relation zu den Bewertungen bei Käufen und Rücksendungen zu stehen. Der Wertebereich impliziter Click-Bewertungen liegt dann im Intervall von [0,45...0,9]. Der einmalige Ressourcenabruf wird dadurch leicht negativ mit 0,45 bewertet, um die Unsicherheit auszudrücken, dass keine echte positive oder negative Präferenz des Benutzers bekannt ist. Am oberen Ende des Intervalls wird die 0,9 erreicht, da man auch hiermit die Unsicherheit von Fehlern in der Navigation berücksichtigt. Eine volle Zustimmung von eins sollte nur durch die direkte Willensbekundung in Form expliziter Bewertungen möglich sein.

In [HF01] beispielsweise wird eine solche implizite Bewertungstechnik basierend auf Abrufhäufigkeiten beschrieben, allerdings mit enger Fokussierung auf Online-Radios.

Suchterme

Bei Suchanfragen können implizite Bewertungen ähnlich wie bei Clicks gewonnen werden. Der Benutzer gibt hier sogar explizit einen Suchausdruck an, zu dem er Informationen wünscht. Der tatsächliche Nutzen und die Bewertung der gelieferten Suchergebnisse sind aber auch hier nicht bekannt.

Bei bewerteten, sortierten Suchergebnissen – beispielsweise absteigende prozentuale Angaben für die Übereinstimmung des Fundstückes mit der Suchanfrage – kann der Rang einer Ressource in die implizite Bewertung mit einfließen. So sind Ressourcen mit höherem Rang, also höherer Ähnlichkeit mit der Suchanfrage, nützlicher für den Benutzer als solche mit niedrigerem Rang. Auf dieser Annahme beruhen alle Suchsysteme wie beispielsweise die Internetsuchmaschinen. Die Qualität steht und fällt aber auch mit den dort bekannten Einschränkungen. Vor allem führen die Suchanfragen nicht automatisch zu guten Suchergebnissen, also hoher *Precision* und hohem *Recall* (siehe Kapitel 3.3 Erfolgsmessung).

Die Lieferung einer Ressource durch eine Suchanfrage wird unter diesen Annahmen wie folgt bewertet, wobei 0,5 wieder für die Standardbewertung steht.

$$0,5 \cdot rank_{iid} \text{ mit } rank_{iid} \in [0,1] \tag{4.7}$$

Wie bei den Clicks soll hier auch die Häufigkeit der Ressourcenlieferung berücksichtigt werden. Die Formel dazu weicht vom Ergebnis her nur leicht von (4.6) ab.

$$rating_{uid,iid}^{\text{search}} = \left[\text{search}; date; 0,95 \cdot \sum_{i=1}^{n} \frac{1}{2^{i-1}} \cdot 0,5 \cdot rank_{iid} \right] \tag{4.8}$$

Der Skalierungsfaktor ist mit 0,95 etwas höher als bei den impliziten Bewertungen der Clicks. Hintergedanke ist, dass die explizite Eingabe eines Suchausdruckes belohnt werden soll, da sie gewissermaßen einen höheren Grad an Bewusstheit seitens des Benutzers ausdrückt. *n* steht für die Anzahl der Suchergebnisse, in denen die Ressource *iid* enthalten war.

Zu überlegen ist, ob alle Suchergebnisse oder nur ein Teil berücksichtigt werden sollen. Mit einem Schwellwert könnte man z. B. alle Fundstücke mit niedrigem Suchrang aussortieren. Dadurch würde man je nach Wahl des Schwellwertes nur die besten Übereinstimmungen von Suchanfrage und Ressourcenbestand erhalten. Problematisch an niedrigen numerischen Werten der Ränge ist, dass dadurch die implizite Bewertung auch niedrig ausfällt. Eine Verfeinerung der Suchanfrage durch den Benutzer mit besseren Suchergebnissen wird in dem beschriebenen Schema nur unzureichend unterstützt. Da die Verfeinerung zu zeitlich späteren Suchergebnissen führt, wird sie nach der Bewertungsformel in (4.8) abgewertet. Hier wären alternative Konzepte wie Durchschnittsbildung über alle Einzelbewertungen denkbar, die aber wiederum die wiederholte Suche vernachlässigen. Daher wird in dieser Arbeit beim vorgestellten Schema geblieben.

Ressourcenähnlichkeit

Mit Hilfe der Ähnlichkeit zwischen den Ressourcen des Informationssystems können abgeleitete, implizite Bewertungen bestimmt werden. Die Idee ist, dass ähnliche Dokumente vom Benutzer ähnlich präferiert werden. Hat der Benutzer für eine Ressource eine explizite Bewertung abgegeben, dann hat das für sich genommen eine hohe Aussagekraft. Diese Bewertung kann auf ähnliche, aber nicht bewertete Ressourcen in der Umgebung der bewerteten Ressource abfärben.

Mit *sim* sei die Ähnlichkeit zweier Dokumente bezeichnet und I_{rated} die Menge der Ressourcen, die vom Benutzer bewertet wurden. *sim* soll dabei nach Normierung in einem Intervall von [0,1] liegen mit eins für hoher Ähnlichkeit und null für niedriger. Ermittelt werden kann sie z. B. aus einer

inhaltlichen Analyse oder der Assoziationsanalyse über den Zugriffshäufigkeiten. Für die Bewertung nach Ressourcenähnlichkeit steht dann

$$value_{uid,iid}^{\text{similaritem}} = 0{,}95 \cdot \sum_{j \in I_{rated}} sim(iid, j) \cdot value_{uid,j}^{\text{explicit}} \,|\, sim(iid, j) \geq 0{,}5 \tag{4.9}$$

value steht jeweils für den Wert der abgeleiteten und der expliziten Bewertung. In der 3-Tupel-Schreibweise eingesetzt steht so

$$rating_{uid,iid}^{\text{similaritem}} = \left[\text{similaritem}; date; value_{uid,iid}^{\text{similaritem}}\right] \tag{4.10}$$

Der Schwellwert von 0,5 wurde gesetzt, um nur die ähnlichsten Ressourcen zu berücksichtigen. Der Skalierungsfaktor ist hier wie bei der Suche ebenfalls auf 0,95 gesetzt, um auszudrücken, dass eine abgeleitete Bewertung nicht die gleiche Aussagekraft haben soll wie eine tatsächliche explizite Bewertung.

Rangordnung der Datenquellen

In der vorangegangen Beschreibung der einzelnen Bewertungsdatenquellen wurde die unterschiedliche Aussagekraft jeweils erläutert und beim Aufbau der Formeln berücksichtigt. Geordnet nach Aussagekraft ergibt sich die folgende Rangordnung:

$$\text{explizit} > \text{purchase}, \text{return} > \text{similaritem} > \text{search} > \text{click} \tag{4.11}$$

Das Größerzeichen > drückt aus, dass die linke Seite mehr Aussagekraft über die Präferenz des Benutzers für eine Ressource hat als die rechte Seite. In umgekehrter Rangordnung steht aber die Verfügbarkeit der Datenquellen. So sind Click-Bewertungen sehr einfach zu gewinnen und in hoher Zahl vorhanden, explizite Bewertungen hingegen werden nur selten abgegeben. Daten aus Käufen und Rücksendungen sind zudem nur bei Online-Shops und Bezahlinhalten gegeben.

Aggregation

Stehen die Bewertungen aus den einzelnen Quellen bereit, müssen sie zusammengeführt werden. Die Aggregation kann sowohl beim Eintreffen neuer Bewertungen der jeweiligen Datenquelle als auch in regelmäßigen zeitlichen Intervallen in Form eines Batchjobs erfolgen. Der Weg kann je nach Implementierung unterschiedlich ausfallen. Unabhängig davon ist das Zusammenführen nicht zeitintensiv, da für eine Ressource und einen Benutzer nur wenige Werte miteinander verrechnet werden. Die Verrechnung erfolgt anhand eines Regelsatzes, der problemlos um weitere Regeln erweitert werden kann.

- Regel 1: Aggregierte Bewertung als gewichtete Summe der Bewertungen
- Regel 2: Nur die verfügbaren Datenquellen werden herangezogen
- Regel 3: Rangordnung der Datenquellen bestimmt Gewichtung
- Regel 4: Explizite Bewertungen haben ein Veto

Die erste Regel sieht vor, dass sich die aggregierte Bewertung als gewichtete Summe aus den verschiedenen Bewertungen errechnet. Dazu ist für jede Datenquelle eine Gewichtung w_{type} nötig. Mit der zweiten Regel wird gewährleistet, dass nicht vorhandene Datenquellen in der Summe unberücksichtigt bleiben. *T* bezeichnet die Menge der Bewertungstypen mit vorhandenen Daten für den aktiven Benutzer *uid* und die Ressource *iid*.

$$rating_{uid,iid}^{\text{aggregated}} = \frac{\sum_{type \in T} w_{uid,iid}^{type} \cdot rating_{uid,iid}^{type}}{\sum_{type \in T} w_{uid,iid}^{type}} \tag{4.12}$$

Die Bestimmung der Gewichte erfolgt gemäß Regel 3 dynamisch. Sind beispielsweise viele Clicks auf eine Ressource gemessen worden, so ist die Aussagekraft der impliziten Click-Bewertung hoch und die Gewichtung muss ebenfalls hoch ausfallen. Tauchte die gleiche Ressource zusätzlich in einem Suchergebnis auf, jedoch nur einmal, dann ist die Aussagekraft dieser Such-Bewertung niedrig und ihre Gewichtung in Folge ebenfalls klein. Die Gewichtung für Click- und Suchbewertungen wird daher von der Anzahl der Clicks und Suchanfragen abhängig gemacht.

$$\begin{aligned} w_{uid,iid}^{\text{search}} &= \#\,\text{Suchanfragen mit gefundener Resource } iid \\ w_{uid,iid}^{\text{click}} &= \#\,\text{Clicks auf Resource } iid \end{aligned} \tag{4.13}$$

Die Ressourcenähnlichkeit hängt von der Verfügbarkeit einer Ähnlichkeitsfunktion zwischen Ressourcen ab. Ist sie gegeben, wird die Bewertung nach Ähnlichkeit mit einer konstanten Gewichtung einbezogen. Sie wird hier auf einen Wert von zwei gesetzt, da sie dadurch einerseits stärker als ein einzelner Click oder eine einzelne Suche herangezogen wird. Andererseits schwindet der Nutzen der Ressourcenähnlichkeit, wenn der Benutzer eine deutlichere und eindeutigere Willensbekundung durch wiederholte Clicks oder wiederholtes Suchen vornimmt.

$$w_{uid,iid}^{\text{similaritem}} = 2 \tag{4.14}$$

Regel 4 sieht schließlich vor, dass explizite Bewertungen – seien es manuelle auf einer Bewertungsskala oder Käufe und Rücksendungen – ein Veto haben. Diese Bewertungen sollen die impliziten Bewertungen dominieren, da sie wie geschildert eine höhere Aussagekraft haben. Das Veto kann dabei formal mit einer Gewichtung von unendlich oder einer großen Zahl g ausgedrückt werden.

$$w_{uid,iid}^{\text{explizit}} = w_{uid,iid}^{\text{purchase}} = w_{uid,iid}^{\text{return}} = g \quad \text{mit } g = \text{gross} \vee g = \infty \tag{4.15}$$

Wird eine große Zahl gewählt, stellt sich bei vielen impliziten Bewertung eine Verschiebung der aggregierten Bewertung hin zu den Werten der impliziten Bewertungen ein. Das kann erwünscht sein, um Abweichungen von expliziten Angaben und dem tatsächlichen Benutzerverhalten zu erkennen und zu berücksichtigen. Sollen die expliziten Bewertungen aber uneingeschränkt dominieren, wird die Gewichtung auf unendlich gesetzt. Bei Online-Shops, die sowohl explizite manuelle Bewertungen wie solche durch Käufe und Rücksendungen zulassen, können die expliziten, manuellen Bewertungen zudem die Käufe und Rücksendungen dominieren. Das wird entsprechend durch unterschiedliche Gewichtungen realisiert.

Zwei Beispiele zur Verrechnung seien abschließend betrachtet. Im ersten Fall sind drei Clicks und eine Bewertung durch Suche gegeben. Die Übereinstimmung bei der Suche beträgt 90%. Die aggregierte Bewertung für einen Benutzer uid = 5 und eine Ressource iid = 12 berechnet sich wie folgt:

$$
\begin{aligned}
rating_{5,12}^{\text{aggregated}} &= \frac{w_{5,12}^{\text{click}} \cdot 0,9 \cdot \left(1 - \frac{1}{2^n}\right) + w_{5,12}^{\text{search}} \cdot 0,95 \cdot 0,5 \cdot rank_{12}}{w_{5,12}^{\text{click}} + w_{5,12}^{\text{search}}} \\
&= \frac{3 \cdot 0,79 + 1 \cdot 0,43}{3 + 1} \\
&\approx 0,699
\end{aligned}
\tag{4.16}
$$

Die Bewertungen durch Clicks dominieren also und werden durch die Suchbewertung etwas nach unten korrigiert. Auf der Skala von null bis eins liegt die aggregierte Bewertung im oberen Mittelfeld.

Im zweiten Beispiel sind eine explizite Bewertung und sieben Clicks gegeben. Die Ressource wurde also gut besucht. Die Gewichtung der expliziten Bewertung ist hier nicht auf unendlich, sondern auf zehn gesetzt, um hohe Anzeigehäufigkeiten als Ausdruck des Benutzerverhaltens zu berücksichtigen. Der Wert der expliziten Bewertung liegt bei 0,4. Die aggregierte Bewertung für einen Benutzer uid = 5 und eine Ressource iid = 14 berechnet sich dann wie folgt:

$$
\begin{aligned}
value_{5,14}^{\text{aggregated}} &= \frac{w_{5,12}^{\text{explicit}} \cdot value_{5,14}^{\text{explicit}} + w_{5,14}^{\text{click}} \cdot 0,9 \cdot \left(1 - \frac{1}{2^n}\right)}{w_{5,14}^{\text{explicit}} + w_{5,14}^{\text{search}}} \\
&= \frac{10 \cdot 0,4 + 7 \cdot 0,89}{10 + 7} \\
&\approx 0,603
\end{aligned}
\tag{4.17}
$$

Die hohe Anzeigehäufigkeit der Ressource wird also berücksichtigt und die unterdurchschnittliche explizite Bewertung aufgewertet.

4.1.5 Schlussbetrachtung

Die hier vorgestellte Methode hilft, zwei grundlegende Probleme des Kollaborativen Filterns zu mildern: Erstens das Kaltstartproblem beim Aufbau neuer Informationssysteme und zweitens das Problem des dünnen Datenbestandes. Durch das Heranziehen alle verfügbaren Datenquellen und ihrer Aggregation auf einen Wert, stehen mehr Bewertungen für Ressourcen zur Verfügung. Der Datenbestand wird also vergrößert und beim Neuaufbau eines Systems kommt es zu einem schnelleren Anstieg der Bewertungszahl.

Völlig gelöst werden beide Probleme natürlich nicht, da der Datenbestand insbesondere bei großen Systemen mit sehr vielen Ressourcen immer noch vergleichsweise klein ist und bei neuen Systemen am Anfang auch keine Bewertungen geraten werden können. Erst die Benutzung des Systems durch Benutzer hinterlässt Datenspuren, die ausgewertet werden können. Zudem werden die expliziten Bewertungen entlastet, da sie nur noch eine und nicht mehr die ausschließliche Komponente für Personalisierungsalgorithmen repräsentieren.

Kritik bleibt bestehen, da die gemachten Annahmen zur Aussagekraft und Ableitung impliziter Bewertungen anfechtbar sind. Insbesondere die Gewichtungen eines einzelnen Clicks oder einer Suchanfrage können auch anders justiert werden. Mit empirischen Prüfungen könnte man sicherlich bessere Ergebnisse erzielen, aber das ist nicht mehr Gegenstand dieser Arbeit. Auch sind die Parameter abhängig vom jeweiligen Informationssystem und der tatsächlichen Verfügbarkeit der

Datenquellen. Beispielsweise stehen Transaktionsdaten nur bei Online-Shops zur Verfügung, die Bestimmung der Ressourcenähnlichkeit aber möglicherweise nicht.

4.2 Einbeziehung von Zeit

Ziel des zweiten Konzeptes ist, Änderungen im langfristigen Benutzerinteresse zu erkennen und so Benutzerprofile und Personalisierungsmechanismen dynamischer zu gestalten. Beweggründe dafür gibt es zahlreiche. So kann sich bei Informationsportalen der Interessensschwerpunkt des Benutzers verlagern, bei Online-Shops können sich Konsumgewohnheiten ändern oder bei Musikanbietern kann der Geschmack des Hörers im Laufe der Zeit wechseln. Ziel ist daher, Methoden zu entwickeln, mit denen das Informationssystem diese Interessensverlagerung beobachten und Benutzerprofile entsprechend anpassen kann.

Neue Interessen werden durch neue Bewertungen – egal ob impliziter oder expliziter Natur – erfasst. Das ist systemimmanent für Bewertungssysteme und dem darauf aufbauenden Kollaborativen Filtern. Je mehr Einzelbewertungen zur Verfügung stehen, desto besser ist sogar das Bild des Benutzers aus Systemsicht.

Bei schwindendem Interesse an Themen liegt aber keine so gute Unterstützung vor. Benutzer bewerten explizit zwar neue Ressourcen und das leider nur sporadisch, aber Korrekturen über die Zeit werden nicht vorgenommen. Dazu müsste ein Informationssystem zudem die Möglichkeit bieten, Bewertungsänderungen vorzunehmen, was viele umgesetzte Systeme in der Praxis nicht vorsehen. Implizite Bewertungen durch Anzeigehäufigkeit oder Suche nach Ressourcen werden hingegen automatisch erzeugt und können nachträglich mit der Zeit korrigiert werden. Die Aussagekraft ist aber wie im vorangegangen Kapitel gezeigt geringer als bei expliziten Bewertungen.

Die Idee für dieses zweite Konzept ist daher, dass durch die kontinuierliche Beobachtung des Benutzerverhaltens geprüft wird, ob die ehemals vorgenommenen Bewertungen wirklich noch interessant für den Benutzer sind. An die Verbreiterung der Datenbasis durch das erste Konzept wird hier angeknüpft, so dass zwei neue Aspekte betrachtet werden:

- Veränderung der Werte jedes Bewertungstypes mit der Zeit
- Gewichtsanpassung zur Verrechnung der verschiedenen Bewertungen mit der Zeit

Die zu klärenden Fragen sind vor allem, wie sich das wiederholte Bewerten von Ressourcen gestaltet, wie das Desinteresse an Ressourcen modelliert wird und wie die Verrechnung der Bewertungen aus den verschiedenen Quellen erfolgen soll.

4.2.1 Genereller Ansatz

Die Bewertung einer Ressource unter- oder oberhalb der Standardbewertung drückt eine negative oder positive Präferenz des Benutzers aus. Hat ein Benutzer noch keine Bewertung für eine Ressource abgegeben, so kann man entweder fehlende Werte berücksichtigen, oder die neutrale Standardbewertung annehmen. Wurde zu einem oder mehreren Zeitpunkten in der Vergangenheit ein Interesse bekundet, dann weicht die Bewertung positiv oder negativ von der Standardbewertung ab. Hier wird angenommen, dass eine Ressource im Laufe der Zeit uninteressant wird und die Bewertung in Richtung der Standardbewertung wandert, die ja Neutralität ausdrückt. Der Zeitraum, in dem das geschieht, ist dynamisch an das allgemeine Benutzerverhalten anzupassen, worauf weiter unten noch eingegangen wird.

Das Modell aus dem ersten Konzept wird so erweitert, dass die Rücknahme von Bewertungen möglich wird. Bei expliziten Bewertungen nähert sich die Bewertung von unten oder oben der Standardbewertung und bei impliziten Bewertungen werden die Aktionen, die zur Errechnung geführt haben, sukzessive zurückgenommen. Erfolgen jedoch neue Bewertungen oder Aktivitäten wie Clicks auf einer Ressource, wird der Absinkprozess gestoppt. Abgebildet werden kann dieses Absinken oder Neutralisieren mit einer Funktion, die die Bewertung je nach Typ mit der Zeit zurücknimmt und in Richtung Zukunft auf die Standardbewertung führt. Eingabeparameter für eine solche Funktion *sinkfunc* sind

- Zeitpunkt der letzten Bewertung
- Sinkgeschwindigkeit (z. B. 0,01 Wertverlust in 2 Tagen oder 10% in 3 Wochen)
- Bewertungswert selbst (wie z. B. 0,4 oder 0,8)
- Dauer einer stabilen Verzögerung nach der Bewertung, ab der das Absinken beginnt
- Standardbewertung (wie hier als in der Funktion fest kodierte 0,5)

Das Prinzip wird durch die exemplarische Visualisierung in Abbildung 37 betrachtet. Hier wird eine einzelne Bewertung – beispielsweise eine aggregierte – herangezogen, die im Laufe der Zeit durch Benutzeraktionen ansteigt und bei Nachlassen der Benutzeraktionen wieder absinkt.

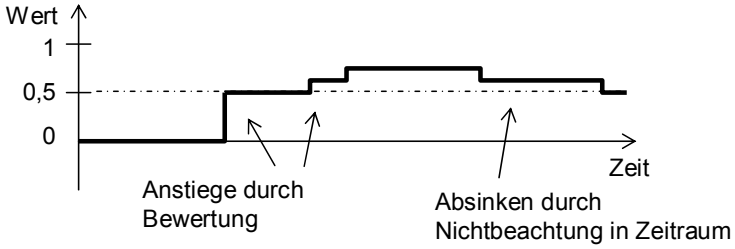

Abbildung 37 - Anpassung von Bewertungen mit der Zeit

Im statischen Modell des vorangegangen Kapitels gab es für die Click- und Suchbewertungen nur monoton steigende Bewertungen. Explizite konnten nur einmal vorgenommen und nicht mehr angepasst werden. Im hier beschriebenen dynamischen Modell sind dagegen wiederholte Bewertungen zugelassen. Die Bewertungsänderung *value'* berechnet sich dabei mit der oben skizzierten Funktion und stellt sich wie in (4.18) dar, wobei das Vorzeichen gekippt werden muss, je nachdem, ob zur Standardbewertung nach unten oder nach oben modifiziert werden soll. Die Minimierung dient als Begrenzung, um im Wertebereich zu bleiben.

$$\begin{aligned} value' &= value \pm \text{sinkfunc}\left(date, v^{\text{sink}}, value, delay\right) \\ &= value \pm \min\left(v^{\text{sink}} \cdot \left(now - date\right); \left|value - 0,5\right|\right) \end{aligned} \tag{4.18}$$

Eine wichtige Funktion hat hierbei die Absinkgeschwindigkeit v^{sink}, um sinkendes Benutzerinteresse auszudrücken[47]. Für den Benutzer *uid* und den Bewertungstyp *type* ist sie

$$v^{\text{sink}}_{type,uid} = \frac{Wertabnahme}{Zeitabschnitt} \tag{4.19}$$

[47] Zwar gibt es auch einen Anstieg hin zur Standardbewertung, wenn explizite Bewertungen neutralisiert werden, aber bei Click- und Suchbewertungen tritt ausschließlich ein Absinken ein. Daher wurde dieser Begriff gewählt.

Eine konstante Geschwindigkeit – beispielsweise um numerische 0,01 oder prozentuale 10% pro Woche – ist problematisch, da die Benutzer ein Informationssystem unterschiedlich intensiv nutzen. Manche arbeiten täglich mit dem System, andere wiederum nur einmal pro Woche. Daher wurde statt einer konstanten Geschwindigkeit eine benutzerbasierte gewählt. Zudem ist die Absinkgeschwindigkeit je nach Bewertungstyp unterschiedlich. Ziel soll dabei sein, dass die Werte bei aktiven Benutzern schneller absinken und bei seltenen Benutzern langsamer. Um die Nutzungshäufigkeit auszudrücken, wird die Zahl der Anmeldungen im System innerhalb eines festen Zeitraumes wie einer Woche herangezogen. Zusätzliche Komponenten könnten auch die Dauer der Systemnutzung, die Anzahl der Clicks oder die Zahl vorgenommener expliziter Bewertungen sein. Das wird aber hier nicht weiter verfolgt.

Zur Berechnung wird die Nutzungshäufigkeit ferner skaliert, indem sie durch eine feste Zahl *k* dividiert wird. Hier bieten sich neben einer festen systemspezifischen Konstante einerseits die durchschnittliche Anzahl von Anmeldungen aller Benutzer und die größte Zahl von Anmeldungen an.

$$v_{type,uid}^{\text{sink}} = \frac{\#\,Anmeldungen}{k} \cdot \frac{1}{Zeitabschnitt} \tag{4.20}$$

Zu beachten ist, dass die Absinkgeschwindigkeit nicht zu groß wird und dadurch die Bewertungen in zu schneller Zeit neutralisiert werden. *k* ist folglich ausreichend dimensioniert zu wählen, was aber je nach angebotenem Inhalt im Informationssystem unterschiedlich ausfallen kann. Sich schnell ändernde Geschmäcker wie bei Musik erfordern so höhere Wertabnahmen und Themen mit längerfristigen Präferenzen entsprechend niedrigere. Für den Zeitabschnitt muss ebenfalls ein passender Wert gewählt werden. Beispielsweise ein Tag oder eine Woche. Bei einem Benutzer mit zehn Anmeldungen, einem definierten Zeitabschnitt von sieben Tagen und einer Konstante *k* von 100 würde die Absinkgeschwindigkeit 0,014 Punkte pro Tag betragen. Eine Bewertung von 0,8 würde dann in 21 Tagen auf 0,5 absinken.

Für die mathematische Betrachtung ist dieser Formalismus angebracht. Aus technischer Sicht stellt sich aber die Frage, wann die Bewertungen zu aktualisieren sind. Ein denkbarer Zeitpunkt wäre die Anmeldung des Benutzers im Informationssystem. Diesen Vorschlag kann man sich direkt zu Nutze machen und die Absinkgeschwindigkeit und Nutzungshäufigkeit erheblich vereinfachen. Statt eines kontinuierlichen Werteabfalls, kann man die Werte in diskreten Schritten Richtung Standardbewertung anpassen. Dazu können die Änderungen bei allen Benutzern und zu jeder Anmeldung um einen konstanten Wert vorgenommen werden. Aktive Benutzer erhalten durch häufige Anmeldungen so eine zeitlich schnellere Neutralisierung nicht mehr interessanter Ressourcen und eher passive Benutzer eine zeitliche langsamere. Das entspricht der Vorgabe der an jeden Benutzer individuell angepassten Absinkgeschwindigkeit. Die neu verwendete Maßeinheit ist daher „1 Anmeldung“. Die konstanten Werte für die Bewertungsänderung müssen allerdings auch hier definiert werden und sind vom Bewertungstyp abhängig.

Ähnlich wie die Absinkgeschwindigkeit kann die in der Parameterkonfiguration aufgeführte Verzögerungszeit *delay* behandelt werden. Sie beschreibt die Zeitdauer, die eine Bewertung nach der Aktion (eine manuelle Bewertung, ein Click, eine Suche, usw.) zunächst stabil bleibt, bevor der Absinkprozess beginnt. Sie muss ebenfalls an die Nutzungshäufigkeit eines jeden Benutzers angepasst werden. Statt einer Angabe in einer zeitlichen Notation kann aber die Angabe in der Einheit Anmeldungen erfolgen und so die technische Behandlung vereinfachen. Bei allen Benutzern kann sie dadurch einen einheitlichen Wert annehmen. Beispielsweise könnten alle Bewertungen über fünf Anmeldungen hinweg stabil bleiben, bevor sie abgewertet werden.

4.2.2 Auswirkungen auf die Datenquellen

Die eingangs angeführten Aspekte, sowohl die Werte als auch die Gewichte der Bewertungen zu verändern, sind für die verschiedenen Bewertungstypen unterschiedlich zu behandeln. Die im vorherigen Abschnitt gelegten Grundlagen werden dazu angewendet und in Details angepasst.

Explizite Bewertungen

Bei explizit vorgenommenen Bewertungen durch manuelle Bewertung auf einer Bewertungsskala oder durch Kauf und Rücksendung wird kein dauerhaftes Veto mehr gesetzt. Stattdessen wird das Veto nur für die Verzögerungszeit *delay* aufrechterhalten. Anschließend wird der Wert *value* der Bewertung bei jeder Anmeldung in kleinen Schritten an die Standardbewertung 0,5 angeglichen. Die Schritte sollten tatsächlich klein sein, um die Bewertung nicht zu schnell zu neutralisieren. Technisch erfolgt das durch Setzen einer neuen expliziten Bewertung mit dem modifizierten Wert *value'*. Zur Unterscheidung wird der modifizierten Bewertung der neue Typ *explicit_undo* zugewiesen.

Das Absinken soll unterbrochen und rückgängig gemacht werden, wenn der Benutzer eine neue Bewertung für die Ressource abgibt. Bildhaft könnte man von „Aufwecken" – oder technischer Reaktivieren – sprechen. Der Auslöser kann eine neue explizite Bewertung sein, die dann fortan gilt und alle bisherigen expliziten Bewertungen überdeckt. Die alten Bewertungen werden nicht mehr benötigt und können aus dem System gelöscht werden. Genauso können impliziten Bewertungen als Auslöser dienen. In beiden Fällen wird der alte, präzise Wert der expliziten Bewertung wieder eingesetzt und so die hohe Aussagekraft dieses Bewertungstyps ausgenutzt.

Die Gewichtung der expliziten Bewertung bei der Aggregation der verschiedenen Bewertungen erfolgt analog zur Wertänderung. Für jede neue Anmeldung des Benutzers (und damit vergangene Zeiteinheit) wird das ursprüngliche Gewicht g um eins reduziert (siehe Kapitel 4.1.4). Technisch kann das so erfolgen, dass für die Verzögerungszeit g auf unendlich steht und anschließend einen kleineren, endlichen Wert annimmt. Beispielsweise kann g auf zehn gesetzt werden. Unter Einbeziehung aller Datenquellen als Auslöser ergibt sich dann folgende Formel für die Gewichtung:

$$w_{uid,iid}^{\text{explizit}} = \max\left(g - \#\text{ Anmeldungen seit Zeitpunkt der Bewertung}, 0\right) \qquad (4.21)$$

So wird gewährleistet, dass die Gewichtung abnimmt, wenn der Benutzer das Interesse an der Ressource verliert. Ein sinnvoll gewählter Wert von g führt so zu einer langsamen Neutralisierung der Bewertung, wenn keine wiederholte Bewertung der Ressource stattfindet.

Implizite Bewertungen

Die Benutzerhandlungen wie Anklicken einer Ressourcenansicht und Eingabe einer Suchanfrage wurden im ersten Konzept in Abschnitt 4.1 auf Bewertungen abgebildet, die positive und negative Präferenzen des Benutzers gegenüber den betroffenen Ressourcen ausgedrückt haben. Die Höhe der Präferenz entsprach dabei Annahmen und nicht einem direkten numerischen Ausdruck des Benutzers. Entsprechend können diese Bewertungen auch wieder rückgängig gemacht werden, wenn der Benutzer die Ressource nicht mehr anklickt oder nach ihr sucht. Bei expliziten Bewertungen ist die Rücknahme oder Neutralisierung aus Benutzersicht schwerer nachvollziehbar, da seine getätigten Angaben ohne sein Zutun vom System modifiziert werden. Manche Benutzer werden sogar Systemfehler vermuten.

Bei impliziten Bewertungen ist das jedoch nicht der Fall, da sie sowieso im Hintergrund ermittelt werden. Die Vorgehensweise kann daher sein, dass sich die Bewertung für eine Ressource wie im ersten Konzept durch jeden Click und jede Suchanfrage zunächst aufbaut. Besucht der Benutzer

nach der Verzögerungszeit die Ressource jedoch nicht mehr oder sucht nicht nach ihr, dann setzt auch hier der Absinkprozess ein. Die Verzögerungszeit kann analog wie bei expliziten Bewertungen eine feste Zahl von Anmeldungen betragen – beispielsweise fünf.

Das Absinken des Wertes erfolgt hier aber nicht durch einen kleinen, festen numerischen Wert, sondern durch die Rücknahme der jeweils letzten impliziten Bewertung. Bei jeder Anmeldung und Nichtbeachtung könnte so ein Undo-Click bzw. eine Undo-Suche für die Ressource erfolgen, mit der die letzte Bewertung neutralisiert wird. Für Click-Bewertungen ergibt sich dann der modifizierte Ausdruck:

$$value_{uid,iid}^{\text{click}'} = \left(1 - \frac{1}{2^n}\right) \cdot 0{,}9 \text{ mit } n = \#Clicks - \#Undo\_Clicks \tag{4.22}$$

Bei den Such-Bewertungen muss tatsächlich für jede Suche geprüft werden, ob es eine Undo-Suche gibt, da die Ränge berücksichtigt werden. Das ist technisch nicht weiter schwierig, da die pro Ressource und Benutzer in geringer Zahl vorhandenen Daten schnell durchlaufen werden. Beispielsweise können wie in (4.23) dargestellt vier Suchanfragen erfolgt sein, wobei die letzten drei durch Undo-Suchen rückgängig gemacht wurden. *S* steht für eine Suche und *U* für eine Undo-Suche, die beide in der Reihenfolge der Indizes erfolgt sind. Hierdurch wird ein Stack beschrieben.

$$\left(S_1;-\right), \left(S_2;U_1\right), \left(S_4;U_3\right), \left(S_4;U_2\right), ... \tag{4.23}$$

Die Berechnungsformel für die Such-Bewertungen bleibt ebenso erhalten, lediglich die Auswahl der Suchvorgänge wird auf diejenigen reduziert, für die es keine Undo-Suche gibt.

$$value_{uid,iid}^{\text{search}'} = 0{,}95 \cdot \sum_{i=1}^{n} \frac{1}{2^{i-1}} \cdot 0{,}5 \cdot rank_{iid} \text{ mit } n = \left|\left\{Suche \middle| \exists \neg Undo\_Suche\right\}\right| \tag{4.24}$$

Hintergedanke dieses Vorgehens ist, dass das Berechnungsmodell für Clicks und Suche beibehalten werden kann und der Wertebereich nicht verlassen wird. Schließlich werden keine unpassenden Werte abgezogen oder addiert, sondern lediglich vorangegangene Bewertungen rückgängig gemacht. Zu prüfen ist, ob bei jeder Anmeldung Undo-Clicks oder Undo-Suchen erfolgen sollen oder ob die Absinkgeschwindigkeit niedriger eingestellt wird. Beispielsweise könnten sie erst bei jeder fünften Anmeldung vorgenommen werden. Ähnlich wie die Justierung des konstanten Wertes, der bei der expliziten Bewertung addiert oder subtrahiert wird, ist diese Einstellung je nach Informationssystem und angebotenen Inhalten unterschiedlich zu setzen.

Die Gewichtung für die Aggregation berücksichtigt ebenso die Undo-Clicks und Undo-Suchen.

$$\begin{aligned} w_{uid,iid}^{\text{search}'} &= \#Clicks - \#Undo\_Clicks \\ w_{uid,iid}^{\text{click}'} &= \left|\left\{Suche \middle| \exists \neg Undo\_Suche\right\}\right| \end{aligned} \tag{4.25}$$

Wird eine Ressource so für längere Zeit nicht beachtet – also nicht abgerufen oder nach ihr gesucht – dann sinkt das Gewicht ab. Ist nur eine einzelne Datenquelle vorhanden, hat die Größe des Gewichtes keine Auswirkungen auf den aggregierten Wert. Fließen aber mehreren Datenquellen ein, dann kann durch die Anpassung der Gewichtung eine Verschiebung erfolgen.

Ein Nebeneffekt der dynamischen Bewertungsanpassung ist, dass kleine Ausreißer im Benutzerverhalten, wie falsches Anklicken durch missverständliche Navigation, nach kurzer Zeit von selbst wieder korrigiert werden.

Abgeleitete Bewertungen und Aggregation

Im ersten Konzept wurden als weitere Datenquelle noch abgeleitete Bewertungen aufgeführt. Sie sind genauso wie dort beschrieben auch im dynamischen Modell zu behandeln. Liegen explizite Bewertungen für Ressourcen vor, können diese auf ähnliche Ressourcen abfärben. Eine Berücksichtigung der Zeit ist hierbei nicht gesondert vorgesehen, da die expliziten Bewertungen bereits mit der Zeit angepasst werden und die ähnlichen Ressourcen diese modifizierten Bewertungen einfach übernehmen.

Auch beim Zusammenführen der verschiedenen Werte ist keine abgewandelte Vorgehensweise nötig, da sowohl die Änderungen der Werte als auch die Gewichtsveränderungen nur innerhalb des Bewertungstypes erfolgen. Die aggregierte Bewertung errechnet sich dann wie beschrieben als gewichtete Summe der verschiedenen Teilbewertungen.

4.2.3 Schlussbetrachtung

Zur Berücksichtigung von Veränderungen im langfristigen Benutzerinteresse wurde mit diesem Konzept ein Verfahren vorgestellt, das die Bewertungen von Ressourcen im Verlauf der Zeit manipuliert. Gleichzeitig wurde berücksichtigt, dass auch das Zusammenführen der Datenquellen beachtet werden muss, um auf nachlassende Aktivität in einem Bewertungstyp reagieren zu können. Das Verfahren arbeitet auf den Bewertungen und vertraut wie im ersten Konzept auf einen kollaborativen Filteralgorithmus als Blackbox, um Empfehlungen und Bewertungsvorhersagen zu berechnen. Gegenüber dem ersten Konzept wurde die Datenkomponente Zeit hinzugefügt.

Unterschiedliche Nutzungshäufigkeiten der einzelnen Benutzer werden registriert, indem eine individuelle Absinkgeschwindigkeit verwendet wird. Zur technischen Vereinfachung werden statt der Absinkgeschwindigkeit Anmeldungen oder Sitzungen im Informationssystem gezählt. Durch das Prinzip des Aufweckens wird sichergestellt, dass einmal vorgenommene explizite Bewertungen absinken, wenn der Benutzer kein aktives Interesse mehr daran bekundet. Sobald eine Aktion eintritt, die auf neues Interesse schließen lässt, wird die alte Bewertung wiederhergestellt. Dadurch gehen wertvolle explizite Bewertungen nicht verloren.

Kritisch anzumerken ist, dass das Absinken zu einem Informationsverlust führt, da aussagekräftige Bewertungen hin zur wenig aussagekräftigen Standardbewertung bewegt werden. Das ist insbesondere dann der Fall, wenn die Absinkgeschwindigkeit zu hoch ist und sich die Bewertungen zu schnell an die Standardbewertung annähern. Hier kann allerdings auch eingewendet werden, dass der Informationsverlust bewusst kalkuliert ist – eben um die Änderungen im Benutzerinteresse zu registrieren und dem Benutzer stets personalisierte Angebote bieten zu können, die seinen Interessen in der jeweiligen Zeitperiode entsprechen.

Generell ist die Ableitung von Bewertungen in einem Intervall wie [0,1] anhand von beobachtetem Benutzerverhalten mit Unsicherheiten behaftet. Das Prinzip wurde hier sicherlich nur ansatzweise umgesetzt, aber daran könnte man anknüpfen und die Ergebnisse beispielsweise durch empirische Tests verbessern.

Ferner stellt sich ein technisches Problem, da die Aktualisierung der Bewertungen Rechenzeit kostet. Je nach Größe des Informationssystems, Anzahl und vor allem Aktivität der Benutzer kann es so zu einigen Rechenoperationen kommen, die das Informationssystem verlangsamen. Mit technischen Mitteln wie effizienten Datenstrukturen, günstiger Berechnung und Zwischenspeicherung (Caching) von Bewertungen kann dem jedoch entgegengetreten werden.

4.3 Verknüpfung mit Inhaltsbasierten Filtern

Mit dem dritten Konzept werden Kollaborative mit Inhaltsbasierten Filtern verknüpft. Damit wird ein weiteres Problem des Kollaborativen Filterns angegangen, da neu ins Informationssystem eingefügte Ressourcen noch keinerlei Bewertungen haben. Entsprechend werden sie von den Filteralgorithmen, die auf einer ausreichend großen Menge von Ressourcenbewertungen über alle Benutzer aufbauen, nicht erfasst.

Das ist auch beim im Kapitel 3.4.2 beschriebenen Kaltstartproblem relevant, da neue Informationssysteme noch über keine Benutzer verfügen und keine Bewertungen vorhanden sind. Ein ähnliches Problem besteht zudem für neu ins System gekommene Benutzer, die ebenfalls noch kein Benutzerprofil bestehend aus Ressourcenbewertungen besitzen.

Weiterhin wird durch den Einsatz von Inhaltsbasierten Filtern das Optimierungspotential von Kollaborativen Filtern in Informationssystemen gut unterstützt. Kollaborative Filter lassen sich zwar in vielen Informationssystemen leicht implementieren und bieten gute Personalisierungserfolge, da sie aber auf Bewertungen angewiesen sind, haben sie die genannten Einschränkungen.

In diesem Abschnitt wird daher versucht, durch eine inhaltliche Analyse der Ressourcen eine Grundlage für die Arbeit der kollaborativen Filteralgorithmen zu bilden. Es erfolgt eine Einschränkung auf textuelle Informationen, da sie vergleichsweise leicht zu analysieren sind. Bilder, Musikstücke und andere binäre Daten werden ausgeklammert.

Das erste Ziel ist, zu jeder Ressource ähnliche Ressourcen liefern zu können und die Ähnlichkeit durch numerische Werte quantifizieren zu können. Es wird zunächst kurz auf Volltextsuche und Kategorisierung als Methoden zur Informationsfindung eingegangen, um dann mit Stichwortindizes eine Datenstruktur vorzustellen, aus der ähnliche Dokumente geliefert werden können. Anschließend wird an die beiden ersten Konzepte angeknüpft und eine mögliche Verbindung mit Kollaborativen Filtern beschrieben, die hier aus zwei Punkten besteht

- implizite, abgeleitete Bewertungen durch Ressourcenähnlichkeit
- Sortierung von Listen inhaltlich ähnlicher Dokumente mit Kollaborativen Filtern

Durch die Ableitung von Bewertungen wird die Datenbasis vergrößert, wenn der Benutzer bereits wenige Bewertungen vorgenommen hat. Die Ableitung dient also als Multiplikator von Ressourcenbewertungen. Durch die Sortierung von Listen ähnlicher Dokumente kann zu jeder Ressource, jeder Kategorie oder auch bei Suchergebnissen eine Gewichtung nach den Präferenzen des Benutzers erfolgen. Da die Listenausgabe aber auch ohne Benutzerpräferenzen rein inhaltlich arbeiten kann, wird das Kaltstartproblem gemildert. Konzepte zur Verknüpfung von inhaltlichen und Kollaborativen Filtern sind an verschiedenen Stellen in der Literatur anzutreffen und werden als ein Mittel zur Lösung der Probleme Kollaborativer Filter angesehen (siehe z. B. [Bau99], [FEBS02] und [MMN01]).

Weitergehend wäre es möglich, die Ableitung von Bewertungen auch auf Metadaten und Attribute auszudehnen und so auch Grafiken und anderen Binärdaten zu erfassen. Beispielsweise könnte die Bewertung auf eine Klassifikation einer Ressource – wie ein Genre oder eine Kategorie – abgeleitet und von den Ressourcen abstrahiert werden. Dadurch würden die Präferenzen des Benutzers für bestimmte Kategorien ermittelt, ohne dass der Benutzer explizit eine Aussage über das Interesse an einer Kategorie machen muss, wie es bei der Checkbox-Personalisierung der Fall ist. Um den Rahmen dieser Arbeit nicht zu sprengen, wurde auf diesen Punkt jedoch verzichtet.

4.3.1 Volltextsuche

Die große Menge an Daten in einem Informationssystem kann auf verschiedene Arten für Anwender erschlossen werden. Ein Verfahren dazu ist die traditionelle Volltextsuche. Sie durchforstet textuelle Ressourcen in Informationssystemen und liefert dem Anwender die Fundstellen in Dokumenten, die den von ihm eingegebenen Suchbegriffen möglichst nahe kommen. Solche Suchmöglichkeiten kennt man aus digitalen Bibliotheken, Internet-Suchmaschinen und lokalen Suchsystemen von Websites.

Problematisch an ihr ist allerdings, dass man die zu suchenden Begriffe bereits kennen muss und die Suche primär vom Anwender aus durch konkrete Suchanfragen gesteuert wird. Die Suche wird also mit Interaktion angetrieben. Zwar können auch Algorithmen von sich aus suchen, allerdings müssen sie dazu die zu verwendenden Suchbegriffe kennen. Für die Verknüpfung von Dokumenten ist die Suche weniger geeignet. Hinzukommt, dass sich die Volltextsuche am besten für textuelle Dokumente eignet. Für andere Medien wie Ton und Bild ist sie nicht anwendbar.

Das Fachgebiet des Information Retrieval bietet zahlreiche Konzepte zum Durchsuchen von Dokumenten. [BR99] ist eine gute Quelle hierzu.

4.3.2 Gliederung und Kategorisierung

Eine weitere Variante zum Auffinden von Informationen ist die Kategorisierung und Gliederung von Ressourcen. Dazu können hierarchische Gliederungen aufgebaut werden, die vom Allgemeinen ins Spezielle gehen. Die zu verwaltenden Ressourcen werden dann einem oder mehreren Gliederungspunkten zugeordnet. Der Anwender kann durch die Gliederungsebenen navigieren und die in den einzelnen Ebenen gespeicherten Informationen einsehen. Man kennt solche Verfahren aus Internet-Katalogen wie dem ursprünglichen Yahoo oder schlicht aus den Dateisystemen von aktuellen Betriebssystemen. Hierzu hat sich auch der eingedeutschte Begriff des "Browsens" für das Durchstöbern in Abgrenzung zum "Suchen" eingeprägt.

Durch die Zuordnung einer Ressource zu einer oder mehreren Kategorien wird eine semantische Information erfasst. Alle Ressourcen einer Kategorie teilen gemeinsame Eigenschaften, aufgrund derer sie dieser bestimmten Kategorie zugeordnet werden. Beispielsweise sind in einem Bibliotheksgebäude alle Titel einer bestimmten Thematik an einem physikalischen Ort anzutreffen. So wird man in einer Informatikbibliothek Titel zu Datenbanken an einem Ort finden, während Werke zur Wahrscheinlichkeitsrechnung an einem anderen Ort stehen.

In Informationssystemen kann die Kategorisierung aber wesentlich ausgefeilter und auch hierarchisch sein. So können einige wenige Wurzelkategorien eine grobe Einordnung von Inhalten vornehmen, während untergeordnete Kategorien speziellere Gemeinsamkeiten der Ressourcen beschreiben. Je tiefer man in der Hierarchie geht, desto spezialisierter werden die Kategorien und desto weniger Ressourcen sind einer einzelnen Kategorie zugeordnet. Praktisch an einer Hierarchie ist, dass man im umgekehrten Schluss, wenn man nach oben geht, wieder mehr ähnliche Ressourcen findet.

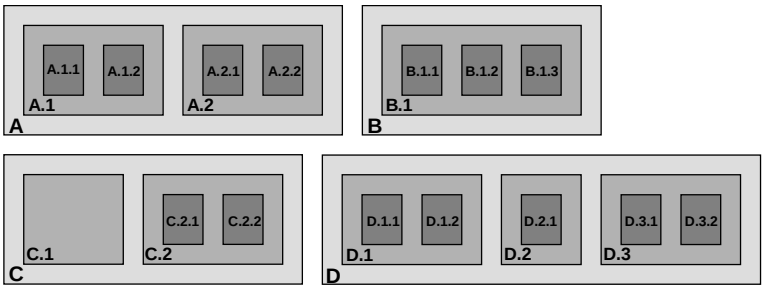

Abbildung 38 - Ressourcengliederung in hierarchischen Kategorien

Die Kategorisierung liefert also eine grobe semantische Beschreibung von Ressourcen und bietet die Möglichkeit, ähnliche Ressourcen zurückzuliefern. Im Rahmen der Personalisierung fällt das insbesondere in den Bereich des Inhaltsbasierten Filterns, da man zu einer Ressource weitere ähnliche liefern kann. Zudem ist man nicht auf eine einzelne Kategorisierung der Ressourcen beschränkt, sondern kann mehrere anwenden, die auch orthogonal zueinander stehen können. Denkbar wäre beispielsweise einerseits eine thematische Kategorisierung eines Ressourcenpools, andererseits auch eine Kategorisierung nach Dokumenttypen und -formaten wie Bild, Text, Ton und Dokumenttypen wie Artikel, Buch, Zeichnung oder Vortrag.

Wie gerade angedeutet, eignet sich Kategorisierung also nicht nur für textuelle Daten wie bei der Volltextsuche, sondern auch für andere Medien. Das wohl größte Manko an der Kategorisierung – insbesondere der inhaltlichen – ist ihr Erstellungsaufwand. Denn wie kann ein Informationspool in Kategorien aufgebrochen werden? Verfahren wie die Clusteranalyse stellen eine Grundlage für die Kategorisierung von Datensätzen dar (siehe Kapitel 3.2.2 Modellbasierte Algorithmen). Hier gibt es auch Varianten, die hierarchische Cluster ermitteln. Leider sind die gefundenen Cluster dabei so aufgebaut, dass für den Algorithmus zwar eine Einordnung der Ressourcen möglich wird, für den Benutzer aber nicht notwendigerweise zu erkennen ist, warum manche Ressourcen in diesem und manche Ressourcen in jenem Cluster landen.

Kategorisierung von Informationen wird also eher durch menschliche Bearbeiter vorgenommen, wobei das automatische Kategorisieren und Klassifizieren von Texten sicherlich in Zukunft verbessert werden wird – beispielsweise zur Spam-E-Mail-Bekämpfung oder zur automatischen Posteingangsbearbeitung. Gegenwärtig muss man sich aber damit anfreunden, dass präzise Kategorisierung von Menschen erledigt wird und daher im Vergleich zu maschinellen Verfahren aufwändig ist. Die Entwicklung der gespeicherten Datenbestände von Yahoo als Repräsentant eines Internetkataloges und Google als Vertreter einer Suchmaschine belegt das: Während der von Sachbearbeitern erfasste Datenbestand in Yahoo nur langsam angewachsen ist, explodierte der in Google geradezu, da hier so genannte Spider oder zu deutsch Spinnen – also Softwareprogramme – die Suche und Indizierung von Internetdokumenten automatisch vornehmen.

4.3.3 Schlüsselwortindizes

Die Erstellungsproblematik bei Kategorisierungen und die beschränkte Anwendbarkeit von Volltextsuche zur Ressourcenverknüpfung könnten durch ein Konzept, das zwischen beiden liegt, umgangen werden: Schlüsselwortindizes.

Schlüsselwortindizes finden sich in vielen Fachbüchern und Ausarbeitungen meist am Ende. Während das Inhaltsverzeichnis am Anfang eines Buches als Kategorisierung der Inhalte durch den Autor gesehen werden kann, ist der Index eine alphabetisch sortierte Liste von Wörtern mit besonderer Bedeutung, die dem Autor sinnvoll erschienen. Jedem Schlüsselwort *s* ist die Seitenzahl

zugeordnet, auf der es im Buch oder in der Ausarbeitung gefunden werden kann[48]. Dabei stellt der Index keine vollständige Erfassung aller Wörter des Buchtextes dar, also eine Art inverse Datei, sondern einen Extrakt von Begriffen mit besonderer Bedeutung, die im Kontext des behandelten Themas interessant sind.

Der Schlüsselwortindex reduziert also die Informationsmenge auf zentrale Begriffe. Der Umfang ist dabei variabel und vom Anwendungszweck abhängig. Je kleiner er aber ausfällt, desto eher sollten die wesentlichen Begriffe getroffen werden, die in der Ressource beschrieben werden. Einen solchen Index kann man zudem für eine gesamte Bibliothek von Büchern oder eben ein Informationssystem aufbauen. Jedes Element der vereinigten Menge von Schlüsselwörtern kann dann entweder nur auf die Ressource i_{iid} zeigen oder auf die Ressource plus der Fundstelle fs_j innerhalb der Ressource. In Formel (4.26) ist das exemplarisch dargestellt.

$$\mu_{index} : s \rightarrow \left\{ \left(i_9, fs_1\right), \left(i_4, fs_2\right), \left(i_{14}, fs_3\right), \ldots \right\} \tag{4.26}$$

Die Menge an Schlüsselwörtern pro einzelner Ressource sollte überschaubar sein, um den Inhalt grob aber dennoch präzise zu beschreiben und die zentralen Begriffe zu treffen, die den Inhalt der Ressource am besten wiedergeben. Bei einem einführenden Artikel in die Softwareentwicklung mit der Programmiersprache Java könnten die Schlüsselwörter so "Java, Softwareentwicklung, Anfänger" lauten.

Vorzugsweise sollten die Schlüsselwörter aus dem Textinhalt automatisch extrahiert werden können. Dennoch können auch manuell Verknüpfungen zwischen Ressourcen und Schlüsselwörtern gesetzt werden. So sind nicht nur textuell und automatisch erfassbare Ressourcen verwaltbar, sondern auch Bild-, Audio- und Videodaten genauso wie Softwareprogramme und andere binäre Daten, aus denen textuelle Informationen nicht oder nur mit hohem Aufwand extrahiert werden können.

Problematisch an der Vergabe der richtigen Schlüsselwörter zu einer konkreten Ressource ist, dass einerseits die zentralen Begriffe aus dem Text ermittelt werden müssen und andererseits nicht zu allgemeine Begriffe aufgegriffen werden sollten, da der Nutzen sonst gering ist. Bei einem Artikel über Softwareentwicklung ist „Software“ zwar mit hoher Wahrscheinlichkeit ein mögliches Schlüsselwort, jedoch nicht unbedingt ein sinnvolles, da es zu allgemein ist. Nicht jedes mögliche Schlüsselwort sollte also gewählt werden, wenn es ein konkreteres Schlüsselwort gibt. Hier muss ein Verfahren gewählt werden, bei dem beispielsweise statt "Software" das Schlüsselwort "Softwareentwicklung" verwendet wird.

Der Schlüsselwortindex kann gleichermaßen als Nachschlagewerk für Software und Anwender dienen. Er enthält die inhaltliche Beschreibung aller Ressourcen im Informationssystem und stellt darüber hinaus Verbindungen zwischen verschiedenen Ressourcen mit ähnlichen Inhalten her, die für die Ausgabe von Listen ähnlicher Dokumente und für abgeleitete Bewertungen nützlich sind. Daraus lässt sich eine einfache Regel entwickeln: je mehr übereinstimmende Schlüsselwörter zwei Dokumente besitzen, desto ähnlicher sind sie. Nicht übereinstimmende Schlüsselwörter können zudem als bestrafender Faktor eingesetzt werden.

Formal werden diese Anforderungen mit der bereits in Kapitel 2.4.3 und 3.2.1 vorgestellten Kosinusfunktion erfüllt und führen auf Formel (4.27) für die Ähnlichkeit zweier Ressourcen. Hier wird die Kosinusfunktion auf dem Schlüsselwortvektor i^s einer Ressource und nicht dem Vektor d aller Wörter definiert. Die Komponenten des Vektors stehen jeweils für ein Schlüsselwort. Sie sind numerische Werte, wobei eins für das Auftauchen des Schlüsselwortes in der Ressource steht und null, wenn es nicht auftaucht. Möchte man die Bedeutung eines Schlüsselwortes für die Ressource

[48] Siehe auch im Anhang dieser Diplomarbeit

feiner ausdrücken, kann man auch Zwischenwerte nutzen (weitergehende Hinweise hierzu wie die Inverse Dokumentfrequenz siehe in [BR99]). Technisch muss die Abfolge der Schlüsselwörter in beiden Vektoren gleich sein, um die richtigen Gewichte miteinander zu verrechnen. Oder beide Vektoren bieten gleich Platz für alle Schlüsselwörter des Indexes.

$$sim\left(\vec{i}_1^{\,s}, \vec{i}_2^{\,s}\right) = \frac{\vec{i}_1^{\,s} \cdot \vec{i}_2^{\,s}}{\left|\vec{i}_1^{\,s}\right| \cdot \left|\vec{i}_2^{\,s}\right|} \tag{4.27}$$

Diese Ähnlichkeitsbestimmung von Informationen ist sicherlich nicht perfekt und es wird Ressourcen geben, die zwar über ähnliche Schlüsselwortmengen verfügen, aber dennoch nur im weitesten Sinne ähnlich sind. Immerhin ist damit aber überhaupt eine Aussage über die Ähnlichkeit möglich.

Die Abgrenzung zur Volltextsuche ist dabei, dass nur ein kleiner Teil der Wörter im Dokument betrachtet wird und nicht die Gesamtheit – Stoppwörter wie „und“, „oder“, „der“, etc. ausgenommen – wie bei der Suche. Ziel muss dabei die Extraktion möglichst guter und repräsentativer Schlüsselwörter sein. Im Unterschied zur Kategorisierung kann hier jede Ressource mit jeder verglichen werden. Eine Ressource kann so auch zu ganz unterschiedlichen ähnlich sein, während die Zuordnung bei der Kategorisierung eher fest ist.

Aufbau von Schlüsselwortindizes

Das manuelle Zuordnen von Schlüsselwörtern zu Datensätzen wäre aber viel zu aufwändig, da ein Sachbearbeiter zu jedem neuen Datensatz eine Reihe von Schlüsselwörtern festlegen müsste. Sie wäre noch aufwändiger als eine einfachere Kategorisierung der Ressourcen in wenige Kategorien. Gegenüber dem Aufwand bei Kategorisierung und Gliederung wäre also nichts gewonnen. Wenn sehr viele neue Datensätze ins System gelangen, müsste entsprechend viel Aufwand betrieben werden, um die Datensätze zuzuordnen. Eine automatische Verknüpfung mittels Software ist also wünschenswert und würde damit die manuelle Kategorisierung vom Aufwand und den Kosten her schlagen.

Zunächst hat man die Menge an aufzunehmenden und mit Schlüsselwörtern zu verknüpfenden Ressourcen. Hier werden nur textuelle Informationen betrachtet, da sie maschinell leichter zu erfassen sind als Grafiken, Musik und andere Binärdaten. Mit Hilfe einer Textanalyse werden markante Wörter aufgespürt, diese in die Schlüsselwortdatenbank aufgenommen und die Verknüpfung zwischen Datensatz und Schlüsselwort hergestellt. Zwar sind entsprechende Verfahren zur Wortzerlegung und -erkennung in Texten schon weit fortgeschritten, sogar zur Erkennung von Substantiven (siehe z. B. [HS00] oder [NBBBB97]). Welche der extrahierten Wörter aber einen hohen inhaltlichen Informationsgehalt besitzen und in einem Schlagwortindex gut aufgehoben wären, ist eine ungleich schwierigere Frage, da hier die Semantik und nicht nur die Syntax der Sprache ins Spiel kommt. Einfache Konzepte wie die Inverse Dokumentfrequenz (siehe Kapitel 3.2.1) helfen nur bedingt weiter, da die Wissensdomäne der verfügbaren Informationen eine entscheidende Rolle spielt. Beispielsweise haben im Maschinenbau ganz andere Begriffe eine hohe Bedeutung als in den Rechtswissenschaften.

Die Inverse Dokumentfrequenz liefert die Häufigkeit eines Wortes innerhalb eines Textes und trifft der Annahme nach eine Aussage über die Bedeutung des jeweiligen Wortes für jenen Text. Wörter, die selten oder nur einmal vorkommen, haben demnach eine hohe Bedeutung für den Text. Bei den markanten Worten eines Textes kann dass so sein, muss es aber nicht! Bei einem Text über Personalisierung kann dieser Begriff sehr häufig auftauchen, gleichzeitig können wenig verwendete Wörter keine besondere Bedeutung haben. Die Semantik der Wörter spielt also eine Rolle. Da die Verfahren zur automatischen Erfassung der Bedeutung eines Textes noch in den Kinder-

schuhen stecken, muss man sich eines anderen Konzeptes bedienen. Von Verfahren zur Zusammenfassung von Texten, die bereits in modernen Textverarbeitungen integriert[49] und für E-Mail-Programme verfügbar sind, sei hier abgesehen, da einzelne Wörter und keine zusammengefassten Texte benötigt werden.

Verfahren zum Schlüsselwortindexaufbau

Der Vorschlag ist daher, dass man umgekehrt vorgeht: Nicht die Texte sind die primäre Quelle für Schlüsselwörter, sondern die Wissensdomäne selbst. Anders als bei der Indizierung von Dokumenten, wie es Google und andere Textsuchmaschinen machen, um einen schnellen Zugriff zu den Fundstellen zu ermöglichen, wird zunächst ein Schlüsselwortindex erstellt. Die Wahl der enthaltenen Wörter obliegt einem Experten dieser Wissensdomäne. Der initiale Index kann später auch erweitert werden, wenn nach Einschätzung des Experten neue Wörter nötig sind. Auch können dazu Textanalysewerkzeuge neue Dokumente vorab analysieren und dem Experten Empfehlungen für potentiell relevante Wörter liefern. Er entscheidet aber letztlich, welche Wörter in den Index aufgenommen werden. Durch den Einsatz eines Experten wird zudem gewährleistet, dass nur solche Wörter in den Index aufgenommen werden, die in der Domäne eine hohe Bedeutung haben. Gleichzeitig wird der Schlüsselwortindex auch mindestens eine kritische Masse an Wörtern besitzen, um das Gebiet ausreichend zu beschreiben, was je nach Domäne eine unterschiedliche Größe sein kann.

Steht der Schlagwortindex, ist die Verknüpfung mit dem Ressourcenbestand vorzunehmen. Dazu zerlegt das Programm jede Textressource *iid* mit einem Parser in einzelne Wörter. Bei Sprachen wie Deutsch, Englisch oder Französisch ist das einfach, da nur die Wortzwischenräume wie Leerzeichen, Interpunktionszeichen oder Tabulatoren zur Unterscheidung von Wörtern erkannt werden müssen. Verfahren des Information Retrieval wie Stoppwortelimination und „Stemming“ – die Reduktion der Wörter auf ihre Grundform – können zur Verbesserung der Wortzerlegung dienen. Anschließend liegt eine Menge W von Wörtern w_i vor, die Reihenfolge spielt keine Rolle. Eine Ressource *iid* hat *c(iid)* Wörter.

$$W_{iid} = \left\{ w_1, w_2, w_3, w_4, ..., w_{c(iid)} \right\} \tag{4.28}$$

Jedes Wort w_i des Textes wird im Schlüsselwortindex *Index* nachgeschlagen. Wurde es gefunden, wird eine Verknüpfung zwischen der aktuellen Ressource *iid* und dem passenden Schlüsselwort s_j vorgenommen. Andernfalls wird es verworfen. Im folgenden einfachen Algorithmus in Pseudocode bezeichnet I_{new} die Menge an Ressourcen, die neu zu indizieren sind.

```
forall i_iid in I_new do
    W := Zerlege Text von Ressource i_iid in einzelne Wörter;
    forall w_i in W do
        forall s_j in Index do
            if (s_j == w_i) then Index.AddRessource ( s_j, i_iid );
        endfor;
    endfor;
endfor;
```

[49] Beispielsweise in Microsoft Word XP verfügbar, siehe http://office.microsoft.com/home/default.aspx

Abbildung 39 - Algorithmus zur Verknüpfung von Ressourcen mit Schlüsselwörtern

Der Algorithmus ist prinzipiell sehr einfach und arbeitet entsprechend schnell. Die Laufzeit in Wortvergleichen liegt in

$$O(w \cdot n \cdot m) \tag{4.29}$$

w steht für die mittlere Anzahl der Worte pro Ressource, *n* für die Gesamtzahl der zu verarbeitenden Ressourcen und *m* die Anzahl der Schlüsselworte.

Probleme können technische Aspekte bereiten, beispielsweise das schnelle Nachschlagen der Schlüsselwörter wenn die Datenmengen groß sind und der Hauptspeicher begrenzt. Je nach Umfang der Textressourcen und des Schlüsselwortindexes kann es dann sinnvoll sein, umgekehrt vorzugehen. Dazu wird der Index durchlaufen und nach den Wörtern im aktuellen Text gesucht statt über die Wortmenge des aktuellen Textes zu iterieren.

Eine technische Ergänzung benötigt das Verfahren, wenn es möglich sein soll, alte Schlüsselwörter aus dem Index zu entfernen und neue hinzuzufügen. Dann muss der bereits indizierte Ressourcenbestand erneut durchlaufen werden und obsolete Verknüpfungen werden gelöscht und neue kommen hinzu.

4.3.4 Verknüpfung mit Kollaborativen Filtern

Mit den vorgestellten Schlüsselwortindizes steht eine Datenstruktur bereit, mit der auf einfache Weise die Ähnlichkeit zwischen Ressourcen ermittelt werden kann. Der Einsatz in Verbindung mit Kollaborativen Filtern wird in den folgenden zwei Abschnitten vorgestellt.

Abgeleitete Bewertung durch Ressourcenähnlichkeit

Durch das Ähnlichkeitsmaß zwischen Ressourcen können neue oder bislang durch einen Benutzer unbewertete Ressourcen abgeleitete Bewertungen erhalten. In Konzept 1 und 2 wurden das prinzipielle Vorgehen und die Einbeziehung in aggregierte Bewertungen bereits vorgestellt. Hier wird noch kurz darauf eingegangen, wie die Ableitung technisch erfolgt.

Um die Bewertungen abfärben zu lassen, werden in regelmäßigen Aktualisierungsläufen für jeden Benutzer alle von ihm bewerteten Ressourcen betrachtet. Zu jeder Ressource wird eine Umgebung von ähnlichen Ressourcen ermittelt. Hierzu dient die Ressourcenähnlichkeit, wobei alle Ressourcen mit einer Ähnlichkeit größer einem Schwellwert einbezogen werden. Dadurch wird sichergestellt, dass der Aktualisierungslauf nicht zu lange dauert und vor allem, dass nur tatsächlich inhaltlich nahe stehende Ressourcen einbezogen werden.

Zur Berechnung der abgeleiteten Bewertung wird Formel (4.9) aus Kapitel 4.1.4 herangezogen, die die gewichtete Summe aller bereits bewerteten ähnlichen Ressourcen I_{rated} für eine abgeleitete Bewertung *iid* bildet. Die Aktualisierung erfolgt für jeden Benutzer unabhängig, da die Verrechnung mit Bewertungen anderer Benutzer dem kollaborativen Filteralgorithmus vorbehalten bleibt. Der Schwellwert wird auf 0,5 gesetzt und im Ableitungsalgorithmus zunächst die Menge $I_{similar}$ der Ressourcen ermittelt, für die abzuleitende Bewertungen erstellt werden sollen, damit nicht alle Ressourcen des Informationssystems betrachtet werden müssen. Dabei wird davon ausgegangen, dass in der Implementierung eine Methode zur Verfügung steht, die über dem Schwellwert ähnliche Ressourcen zu einer gegebenen Ressource liefern kann. Das ist beispielsweise durch Caching effizient möglich. Anschließend wird für die gesammelten Ressourcen die Bewertung bestimmt.

```
I_similar := ∅;

forall i_iid in I_rated do

    I_similar := I_similar ∪ { Alle Ressourcen i_j mit sim(i_iid,i_j) ≥ 0,5 };

endfor;

forall i_iid in I_similar do

    rating_value_iid := Berechne abgeleitete Bewertung für i_iid nach (4.9)

    if (Abgeleitete Bewertung für i_iid vorhanden) then
        Aktualisierte abgeleitete Bewertung für i_iid mit rating_value_iid;
    else
        Erzeuge neue abgeleitete Bewertung für i_iid mit rating_value_iid;
    endif;

endfor;
```

Abbildung 40 - Algorithmus zur Aktualisierung abgeleiteter Bewertungen

Die Aktualisierungsläufe können prinzipiell bei jeder neu eingetroffenen Bewertung eines Benutzers vorgenommen werden. Ist das in einer Implementierung während des Betriebes zu aufwändig, kann aber auch ein Batchbetrieb mit regelmäßigen Aktualisierungen für alle Benutzer erfolgen – beispielsweise zu Zeiten niedrigerer Last oder wenn eine kritische Zahl neuer Bewertungen eingetroffen ist.

Sortierung von Listen mit Kollaborativen Filtern

Listen von Ressourcen sind typisch für Informationssysteme. Benutzer wählen beispielsweise eine Kategorie an und erhalten eine Übersicht der darin enthaltenen Informationen in Listenform als Hyperlinks mit einer Kurzbeschreibung zu jedem Element. Klicken sie auf einen dieser Hyperlinks, gelangen sie zu einer Detailansicht mit dem vollständigen Inhalt der Ressource. Die Generierung solcher Listen ist zwar mit Kollaborativen Filtern direkt möglich, die Algorithmen arbeiten allerdings an sich auf dem gesamten Datenbestand.

Die Listen werden jedoch eher anhand inhaltlicher Filter aufgebaut. Die Selektion kann anhand einer Suche, einer Kategoriezugehörigkeit oder mit der Bestimmung ähnlicher Ressourcen vorgenommen werden. So wird nur ein Ausschnitt des gesamten Datenbestandes betrachtet. Die gelieferten Ressourcen sind aber in keiner Weise personalisiert. Hier kann der Kollaborative Filter einspringen und die Sortierung anhand der Präferenzen des Benutzers vornehmen. Dazu wird zu jeder Ressource der Liste entweder eine vorhandene Bewertung herangezogen oder mit der *pred*-Operation des Filtermodells die Bewertungsvorhersage berechnet.

```
I_list := { i_iid | i_iid wurde durch Inhaltsbasierten Filter geliefert };

W := Gewichtsvektor mit |I_list| Komponenten initialisiert auf 0,5;

forall i_iid in I_list do

    if (Bewertung für i_iid vorhanden) then
        w_iid := Bewertungswert für i_iid;
    else
        w_iid := Berechne Vorhersage für aktiven Benutzer und i_iid;
    endif;
```

```
endfor;

I_list := Sortiere I_list nach Gewichtungen W neu;
```

Abbildung 41 - Algorithmus zur Sortierung von Listen mit Kollaborativen Filtern

So arbeiten Kollaborative Filter nur auf einem Teil des Datenbestandes, der durch inhaltsbasierte Technik bestimmt wird und bieten eine Sortierung des Datenausschnittes. Ressourcenlisten, die bereits sortiert waren und Gewichtungen haben, können durch Multiplikation der ursprünglichen Gewichtung mit dem Wert aus dem kollaborativen Filteralgorithmus und anschließender Neusortierung ebenfalls personalisiert werden. Beispielsweise kann das nach Anfrageübereinstimmung sortierte Ergebnis einer Suche dadurch personalisiert werden.

4.3.5 Schlussbetrachtung

Mit dem dritten Konzept wurde eine Verknüpfung von Kollaborativen und Inhaltsbasierten Filtern vorgestellt. Mit dem Mittel der inhaltsbasierten Ressourcenähnlichkeit kann das Informationssystem schon direkt nach dem Start alternative Ressourcen empfehlen, die einen ähnlichen Inhalt haben. Einen von einem menschlichen Experten aufgebauten Schlüsselwortindex vorausgesetzt. Nehmen Benutzer erste explizite oder implizite Bewertungen vor, dann werden diese durch die abgeleiteten Bewertungen multipliziert.

In beiden Fällen werden das Kaltstartproblem und das Problem des dünnen Datenbestandes abgeschwächt. Im ersten Fall handelt es sich jedoch nicht um personalisierte Empfehlungen. Hier könnte eine Checkbox-Personalisierung nachgerüstet werden, in der der Benutzer einige Kategorien auswählt, welche als Grundlage für eine inhaltsbasierte Personalisierung dienen können.

Problematisch an dem vorgestellten Verfahren ist, dass die Ressourcenähnlichkeit auf automatische Weise nicht optimal bestimmbar ist, da nur einzelne Wörter und nicht die Bedeutung des gesamten Textes erfasst werden. Menschliche Experten könnten bessere Werte für die Ähnlichkeit festlegen, aber unter deutlich höherem Aufwand. Hier treten also die Probleme des Inhaltsbasierten Filterns auf (siehe Kapitel 2.4.3 Inhaltsbasierte Personalisierung).

Die Qualität der Ressourcenähnlichkeitsbestimmung ist abgängig von den Schlüsselwörtern im Index. In der Implementierung hat sich gezeigt, dass eine Ressource mit einer markanten Zahl von Schlüsselwörtern verknüpft sein sollte, damit die Dokumentähnlichkeit sinnvoll bestimmbar ist. Besonders kritisch ist, wenn kein Wort des Ressourcentextes im Index vorhanden ist, da diese Ressourcen nie in Listen ähnlicher Dokumente auftauchen. Hierbei könnte man natürlich suggerieren, dass sie tatsächlich mit keiner anderen Ressource ähnlich sind. Fehlen jedoch die richtigen Wörter im Index, werden mögliche Zusammenhänge zwischen Ressourcen nicht berücksichtigt. Die richtige Wahl des Vokabulars ist entscheidend für den Erfolg.

Andererseits führen zu allgemein eingesetzte Wörter wie „Software“, die im Testdatenbestand der Implementierung häufig auftauchen, zu einer zu hohen Ressourcenähnlichkeit. Dieses Problem wird aber abgemildert, da im Testdatenbestand meist mehrere Wörter mit einer Ressource verknüpft sind. Die Kosinusfunktion hilft hier durch die Aufwertung in beiden Ressourcen vorkommender Wörter und ihre Normalisierung über alle Wörter der Ressourcen – nämlich auch über die nicht übereinstimmenden.

4.4 Umsetzung der Konzepte

Die drei vorgestellten Konzepte ergänzen kollaborative Filteralgorithmen um unterschiedliche Elemente, mit denen die Personalisierung verbessert werden soll. Zum Einsatz kommt in der im folgenden Kapitel beschriebenen Implementierung ein speicherbasierter Filteralgorithmus mit Pearson-Korrelationskoeffizient als Ähnlichkeitsmaß. Statt expliziter Bewertungen erhält er die beschriebenen aggregierten Bewertungen. Ziel war dabei vor allem die Überprüfung der Machbarkeit.

Da die vollständige Umsetzung aller drei Konzepte recht aufwändig gewesen wäre, zumal zunächst ein einfaches Informationssystem mit Kategorienverwaltung und Suche als Basis entwickelt werden musste, wurden einzelne Teilbereiche ausgewählt. So wurde die Aggregation aus verschiedenen Bewertungsquellen implementiert. Als Bewertungsquellen kommen explizite Bewertungen und implizite Click- und Suchbewertungen zum Einsatz.

Zudem wurde die Implementierung des Schlüsselwortindexes mit Bestimmung der Ressourcenähnlichkeit vorgenommen. Zu jeder Ressource können so ähnliche Inhalte geliefert werden. Weiterhin wurde die Suche so gestaltet, dass sie personalisiert und unpersonalisiert ausgeführt werden kann. Bei personalisierter Suche erfolgt eine wie oben beschriebene Gewichtung mit den Bewertungen oder Vorhersagen, je nach verfügbaren Daten.

Die zeitabhängige Anpassung der Bewertungen aus Konzept 2 und die Bestimmung abgeleiteter Bewertungen wurden allerdings ausgelassen. Nähere Angaben zur Implementierung und den verwendeten Techniken finden sich in größerer Breite im nächsten Kapitel.

5 Implementierung

Die Umsetzung eines Teils der in Kapitel 4 vorgestellten Optimierungsansätze wurde im Rahmen eines einfachen Informationssystems überprüft. Das *MiniPortal*[50] getaufte Informationssystem ist ein webbasiertes Informationsportal, das Inhalte verwalten und darstellen kann. Als Testdatenbestand wurden diverse technische Artikel gewählt, die jeweils durch eine Kurzbeschreibung und einen Link auf die Quelle des jeweiligen Urhebers repräsentiert werden. Der Benutzer navigiert durch hierarchisch verschachtelte Kategorien und kann einzelne Inhalte in einer Detailansicht abrufen. Die Inhalte können dabei von unterschiedlicher Art sein, wie Texte, Grafiken oder andere Typen. Die Ressourcen sind in Kategorien gruppiert und können einer oder mehreren Kategorien angehören.

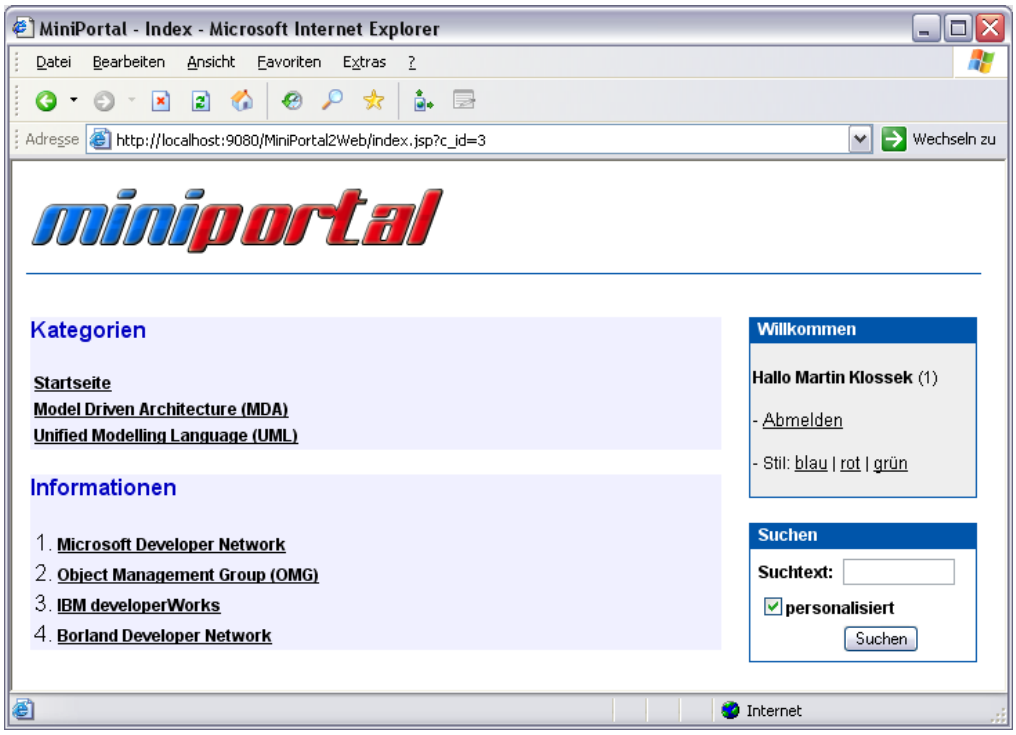

Abbildung 42 - MiniPortal mit Kategorieinhalt und angemeldetem Benutzer

Ein einfaches Benutzerkontensystem ermöglicht das Anmelden von Benutzern im System und das Speichern individueller Parameter in Benutzerprofilen. Der Benutzer gibt dazu in einem Formularfeld im Portal seinen Benutzernamen und sein Passwort ein und findet anschließend das personalisierte Portal vor. Der Schwerpunkt der Implementierung lag dabei nicht auf der perfekten Ausgestaltung einer Internetwebsite, da dies vom Ziel der Diplomarbeit abgewichen wäre. Vielmehr wurden verschiedene Möglichkeiten der Personalisierung untersucht. Dazu gehören

- Individuelle Begrüßung jedes Benutzers, Erkennung häufiger Nutzung
- Einfachste Personalisierung durch Layout- und Farbwahl

[50] Im Screenshot in Abbildung 42 ist das Logo des Informationssystems zu sehen, das kleine Buchstaben verwendet. Zur besseren Lesbarkeit wird im Text aber die Schreibweise „*MiniPortal*“ benutzt.

- Implementierung eines speicherbasierten kollaborativen Filteralgorithmus'
- Explizite und implizite Bewertung von Ressourcen (mit Visualisierung)
- Schlüsselwortindizierung und Ähnlichkeitsbestimmung von Ressourcen als Inhaltsbasierte Filterung
- Suche mit optionaler Gewichtung durch Kollaboratives Filtern

Der verwendete speicherbasierte kollaborative Filteralgorithmus bestimmt die Ähnlichkeit von Benutzern nach dem Pearson-Korrelationskoeffizient, der in Kapitel 3.2.1 vorgestellt wurde. Die Berechnung von Vorhersagen für unbewertete Ressourcen erfolgt anhand der Ähnlichkeit zu anderen Benutzern, die alle oder einen Teil der Ressourcen bewertet haben, die auch der aktive Benutzer bewertet hat. Der aktive Benutzer ist in der Implementierung derjenige, der gerade angemeldet ist.

Als Datenquelle bezieht der Algorithmus die Bewertungen ein, die von Benutzern auf einer Bewertungsskala von 1 bis 5 explizit vorgenommen wurden. Zudem werden implizite Bewertungen erzeugt, wenn der Benutzer eine Ressource in der Detailansicht abruft oder sie in einem Suchergebnis auftaucht. Abgeleitete Bewertungen anhand der inhaltlichen Dokumentähnlichkeit sind in den Datenstrukturen bereits vorgesehen, wurden aber nicht mehr umgesetzt. Die Aggregation der Bewertungen erfolgt so, dass für jede Ressource und jeden Benutzer eine Zahl aus dem Intervall [0..1] bereitsteht, wobei null eine niedrige und eins eine hohe Präferenz ausdrücken. Die aggregierten Werte werden dazu in einer eigenständigen Datenbanktabelle gespeichert, um sie schnell abrufen zu können. Ressourcen die weder implizit noch explizit bewertet wurden, haben jedoch auch hier keine gespeicherte Bewertung. Vorteil an dieser Aggregation ist, dass der kollaborative Filteralgorithmus ohne Änderung genutzt werden kann. Lediglich die Datengrundlage ist größer (dies spiegelt den in Kapitel 4.1 Verbreiterung der Datenbasis beschriebenen Ansatz wieder).

Die Schlüsselwortindizierung aller Ressourcen erfolgt über den Aufruf einer internen URL durch einen fiktiven Administrator. Hier werden nach dem Verfahren aus Kapitel 4.3.3 Schlüsselwortindizes alle textuellen Ressourcen in einzelne Wörter zerlegt und diese im Schlüsselwortindex nachgeschlagen. Sind sie dort vorhanden, wird die jeweilige Ressource mit dem Schlüsselwort verknüpft. Dadurch werden beim Abruf der Detailansicht einer Ressource neben den Nutzdaten und Metainformationen der angeforderten Ressource auch Hyperlinks auf inhaltlich ähnliche Informationen eingeblendet.

Abbildung 43 - Detailansicht einer Ressource mit eingeblendeten Empfehlungen

Die Suchfunktion ermöglicht das Nachschlagen von Ressourcen im gesamten Datenbestand. Dazu gibt der Benutzer einen Suchbegriff ein und erhält alle Artikel, die den Suchbegriff enthalten, in unsortierter Reihenfolge. Schaltet er die Suche auf den personalisierten Modus um, werden die gefundenen Ressourcen anhand seiner individuellen Bewertung neu sortiert. Die Bewertungen kommen hier – falls vorhanden – aus dem Bewertungsspeicher oder werden durch den kollaborativen Filteralgorithmus berechnet. Ist keine Berechnung möglich, weil der Benutzer keine Nachbarn hat – also mindestens einen Benutzer, der eine gleiche Ressource bewertet hat – dann wird die Standardbewertung von 0,5 angenommen.

Der recht beträchtliche Aufwand zur Konzeption und Implementierung des Portals wird dadurch gerechtfertigt, dass nützliche Informationen zum Aufbau und zum Verhalten der kollaborativen Filtertechnik im Speziellen und Personalisierungstechnik im Allgemeinen gewonnen werden konnten. Vor allem standen die erzielten Erfahrungen mit der Umsetzung der Algorithmen im Vordergrund. Ein wichtiger Punkt war daher auch das Bereitstellen von Beispielressourcen und der Test des Systems mit fiktiven Benutzern. Im Folgenden werden ein grober Überblick über den technischen Aufbau des Informationssystems gegeben und besonders kritische Punkte näher besprochen.

5.1 Motivation zur Wahl der eingesetzten Software

In einem relativ frühen Stadium der Diplomarbeit wurde die Festlegung auf die Implementierung mittels der Java 2 Enterprise Edition (J2EE)[51] und der darin enthaltenen Techniken vorgenommen. Beweggrund war, dass es sich hierbei um eine erprobte und ausgereifte Technik handelt, die beim Aufbau von Webanwendungen in der Praxis eingesetzt wird. Ferner sollten die Daten wie Ressourcen, Bewertungen und Benutzerdaten in einer relationalen Datenbank gespeichert werden. Die Anbindung solcher Datenbanken durch J2EE ist mit verschiedenen Mitteln wie Enterprise Java Beans (EJB) (siehe [SUN01]) und Java Database Connectivity (JDBC) gut gelöst.

Ein weiterer Beweggrund war, dass der Lehrstuhl für Datenbanken und Informationssysteme[52] des Instituts für Informatik am Fachbereich Biologie und Informatik (15) an der Johann-Wolfgang-Goethe-Universität in Frankfurt, an dem die Diplomarbeit erstellt wurde, mit der IBM Deutschland

[51] Umfangreiche Informationen und Spezifikationen unter http://java.sun.com/j2ee/
[52] Nähere Informationen zum DBIS-Lehrstuhl siehe http://www.dbis.informatik.uni-frankfurt.de/

Entwicklung GmbH[53] kooperiert. Im Rahmen dieser Kooperation nimmt der Lehrstuhl auch am IBM Scholars Program[54] Teil und hat durch dieses Programm Zugriff auf IBM-Software zu Forschungs- und Studienzwecken. Die Wahl fiel entsprechend leicht und so wurden die Windowsversionen von WebSphere Version 5.0 als Applikationsserver und DB2 UDB Version 8.1 als relationale Datenbank gewählt.

Andere Produkte oder Sprachen wie Microsofts ASP/ASP.NET oder das freie PHP schieden in Folge aus. Allerdings sei anzumerken, dass WebSphere das komplette Spektrum der J2EE-Möglichkeiten abdeckt und DB2 eine ebenfalls sehr mächtige Datenbank ist. Die Fähigkeiten beider Produkte wurden in der Implementierung nur zu einem kleinen Teil genutzt. Da der Aufwand für die Einarbeitung und grundlegende Beherrschung der Produkte erheblich gewesen ist, war ihr Einsatz möglicherweise überdimensioniert. Nichtsdestotrotz konnte der Autor dadurch einen guten Einblick in die Arbeit mit diesen industrieerprobten Produkten gewinnen.

Ferner wurde der WebSphere Studio Application Developer 5.0 (WSAD)[55] als voll integrierte Entwicklungsumgebung eingesetzt, die insbesondere mit dem WebSphere Applikationsserver und DB2 gut harmoniert. Zur Darstellung von Diagrammen in der Weboberfläche wurde die Open Source Cewolf-Komponente[56] benutzt, die ihrerseits JFreeChart[57] als Grafikkomponente kapselt.

5.2 Beschreibung des technischen Aufbaus

Im Folgenden wird zunächst ein grober Überblick über die Struktur des Systems gegeben. Anschließend werden die Datenstrukturen und Softwarebaugruppen vorgestellt.

5.3 Systemaufbau

Typischerweise wird bei größeren Informationssystemen eine mehrschichtige Architektur eingesetzt. Damit sollen die verschiedenen softwaretechnischen Aufgabenbereiche voneinander abgegrenzt werden und sowohl die Erstellung als auch die Wartung der Software verbessert werden. Anwendungslogik und Benutzungsinterface können so entkoppelt werden und beispielsweise den Austausch einer der Schichten ermöglichen. Für *MiniPortal* wurde eine dreischichtige Architektur gewählt.

Auf unterster Ebene ist mit DB2 die Datenbank angesiedelt, die für die Speicherung von Benutzerdaten, Inhalten und den Bewertungen verantwortlich ist. Zudem werden in dieser Schicht die Verknüpfungen der Datenblöcke gespeichert.

[53] Website von IBM Deutschland Entwicklung GmbH unter http://www-5.ibm.com/de/entwicklung/
[54] IBM Scholars Programm siehe http://www-3.ibm.com/software/info/university/
[55] Informationen zu WSAD unter http://www-106.ibm.com/developerworks/websphere/zones/studio
[56] Website des Open Source Cewolf-Projektes siehe http://cewolf.sourceforge.net
[57] Website des Open Source JFreeChart-Projektes siehe http://www.jfree.org/jfreechart/index.html

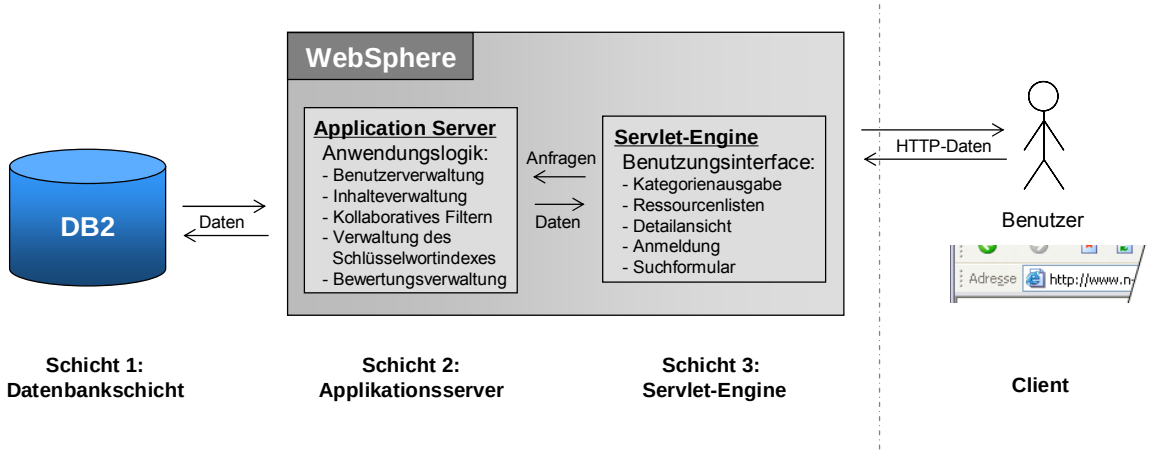

Abbildung 44 - Aufbau MiniPortal

Die zweite Schicht wird vom eigentlichen Applikationsserver in WebSphere gebildet. Hier sind einerseits mit Hilfe von Entity-Beans Kapselungen der Datenbank implementiert. Andererseits wurde die Anwendungslogik in Form von Inhalteverwaltung und Personalisierung in Session-Beans programmiert.

Entity-Beans stellen eine Abstraktionsschicht für den Zugriff auf Datenbanken dar, wobei zwischen container-verwalteter und bean-verwalteter Persistierung unterschieden wird. Bei der container-verwalteten Persistierung, die für die Benutzer- und Inhaltetabelle eingesetzt wurde, regelt der Applikationsserver selbstständig den Zugriff auf die Datenbank für das Speichern und Laden von Datensätzen. Der Entwickler instanziert nur die gewünschten Objekte, die die Daten aus der Datenbank kapseln, und ruft Methoden auf den erzeugten Objekten auf. Um die Speicherung und Zugriff auf die Datenbank braucht man sich nicht zu kümmern. Bei bean-verwalteter Persistierung muss der Datenbankzugriff selbst programmiert werden. Allerdings bietet sich hier eine weitaus höhere Flexibilität für die Interaktion mit der Datenbank (weitere Informationen zu Entity- und Session-Beans unter [SUN01] oder [FCF03], S. 237-309).

Nach den Vorgaben der Enterprise Java Beans Spezifikation ([SUN01]) sollen Anwendungen nicht direkt auf Entitiy-Beans zugreifen, sondern der Zugriff über Session-Beans erfolgen. In *MiniPortal* sind die Zugriffe auf die Benutzer- und Inhaltedaten daher in Session-Beans gekapselt. Ferner gibt es weitere Session-Beans für die Extraktion der Schlüsselwörter, für die Verwaltung der Bewertungen und zur Berechnung von Vorhersagen mit dem kollaborativen Filteralgorithmus.

In der dritten Schicht ist die Funktionalität zum Aufbau der Benutzungsoberfläche und zur Interaktion mit dem Anwender enthalten. Sie bedient sich der WebSphere Servlet Engine und arbeitet mit Java Servlets und Java Server Pages (JSP).

Java Server Pages sind HTML-Dateien, die mit Java-Programmcode angereichert werden können und von WebSphere in ausführbare Java-Klassen kompiliert werden. Bei *MiniPortal* wurde so wenig Programmcode wie möglich in den Java Server Pages-Dateien gehalten, damit eine leichte Änderung an der Oberfläche möglich wäre. Beispielsweise ist so die Funktion für den Layout- und Farbwechsel leicht zu programmieren gewesen. Jeder Benutzer kann dabei die gewünschte Farbkombination aus drei Alternativen auswählen, die eingestellt wird, sobald er sich im System anmeldet, und im Benutzerprofil gespeichert wird.

Der Programmcode für den Ausgabebereich der Weboberfläche – also die Ausgabe von Kategorien und Inhalten – wurde in so genannte Tagbibliotheken ausgelagert (siehe [FCF03], S. 187, ff.). Statt Programmcode kann man damit selbst definierte HTML-ähnliche Tags wie

```
<mptags:item iid="5">
```

in den JSPs platzieren und WebSphere führt den damit verknüpften Programmcode aus, wenn ein Benutzer die Seite abruft.

Benutzeraktionen wie die Anmeldung oder die Bewertung, die keine Ausgaben vornehmen, wurden in Servlets implementiert. Sowohl bei den Tagbibliotheken als auch den Servlets findet eine Interaktion mit der zweiten Schicht und den Enterprise Java Beans statt, um Daten zu speichern und zu laden, Berechnungen auszuführen oder andere Prozesse anzustoßen.

Der Benutzer schließlich greift mit einem Webbrowser auf diese dritte Schicht zu und kann die Funktionen des Portals nutzen. Die erste und zweite Schicht bleiben dem Benutzer verborgen.

5.3.1 Überblick Datenstrukturen

Die verschiedenen Daten werden in einzelnen Datenbanktabellen gespeichert. Aufgrund des Prototypcharakters von *MiniPortal* sind nur wenige Tabellen nötig:

- *item*: Speicherung der Ressourcen (Inhalte)
- *category*: Kategorien der Ressourcen (wie Softwareentwicklung, Security, Internet, ...)
- *itemcategoryrel*: Verknüpfung zwischen Ressourcen und Kategorien
- *user*: Benutzerdaten wie Anmeldename, Passwort, Kontaktdaten und Einstellungen
- *itemrating*: Bewertungen von Ressourcen durch Benutzer aus verschiedenen Quellen
- *itemrating_aggregated*: die aggregierten Werte der Bewertungen
- *keyword*: Schlüsselwörter zum Inhaltsbasierten Filtern
- *itemkeywordrel*: Verknüpfung zwischen Schlüsselwörtern und Ressourcen

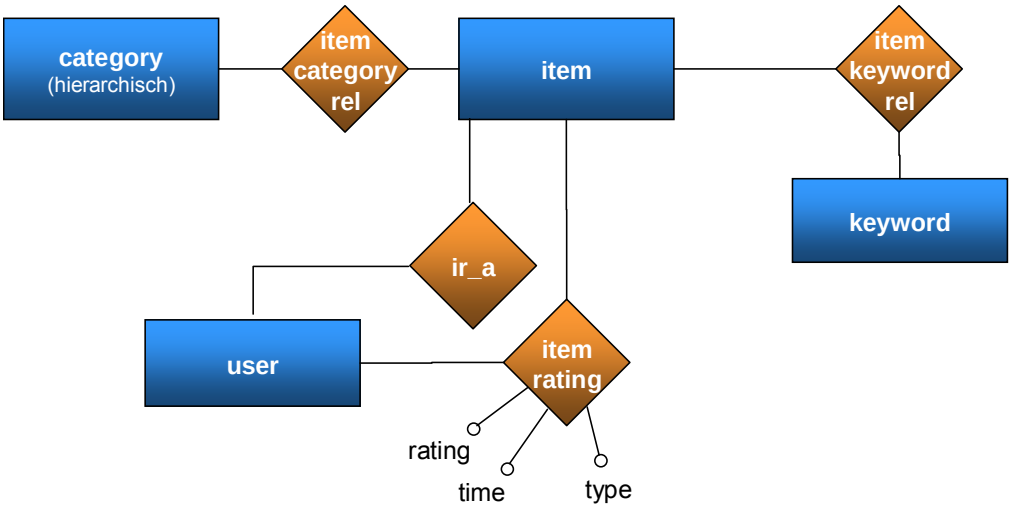

Abbildung 45 - Entity-Relationship-Diagramm von MiniPortal

Die Erläuterung der genauen Struktur der Tabellen mit ihren Feldern wäre an dieser Stelle zu umfangreich und kann daher den SQL-Erstellungsbefehlen im Anhang A Datenbankerstellungsbefehle entnommen werden. Die Tabellen *item*, *category*, *user* und *keyword* enthalten jeweils einen numerischen Primärschlüssel, mit dem ein Datensatz eindeutig identifiziert werden kann.

Abbildung 45 gibt einen Überblick, wie die aufgeführten Tabellen miteinander in Beziehung stehen.

5.3.2 Erläuterung Softwarebaugruppen

Die in *MiniPortal* zum Einsatz gekommenen Java-Klassen sind zur besseren Verwaltung in verschiedene Packages unterteilt worden. Abbildung 46 Zeigt die Pakethierarchie.

Im *data*-Package sind die Klassen für Entity- und Session-Beans enthalten, die für die Inhalteverwaltung verantwortlich sind. Dazu gehören

- *Item*: Entity-Bean für Ressourcen, die die Datenbanktabelle item kapselt
- *Category*: Entity-Bean für Kategorien, die die Datenbanktabelle category kapselt
- *ItemCategoryRel*: Entity-Bean zur Verknüpfung von Ressourcen und Kategorien
- *User*: Entity-Bean für Benutzerdatensätze, die die Datenbanktabelle user kapselt
- *Content*: Session-Bean, die einzelne Ressourcen oder Listen nach verschiedenen Kriterien für die Webanwendung bereitstellt und die die Schlüsselwortindizierung steuert

```
edu.uniffm.miniportal
|
|- data
|
|- personalization
|  |- cf
|
|- util
|  |- text
|
|- minportal
   |- web
      |- modules
      |- taglibs
```

Abbildung 46 - Package-Struktur von MiniPortal

Der Programmcode der container-verwalteten Entity-Beans, der von WebSphere Appication Developer generiert wurde, musste nur leicht erweitert und modifiziert werden. Schwierig war hier, das Zusammenspiel mit der Datenbank richtig zu konfigurieren. Die Session-Bean *Content* enthält jedoch umfangreiche Codepassagen, die zur Inhalteverwaltung dienen. Dabei sind folgende Methoden interessant:

- *getItemsByCategoryId*: liefert die Ressourcen einer angegebenen Kategorie
- *getItemById*: liefert die Ressource mit dem angegebenen Schlüssel
- *getItemsByContent*: liefert die Ressourcen, in der das angegebene Suchwort enthalten ist
- *getSimilarItems*: liefert eine Ressourcenfolge, die der übergebenen inhaltlich ähnlich sind
- *getItemSimilarity*: bestimmt die inhaltliche Ähnlichkeit zweier übergebener Ressourcen
- *upateItemKeywordAssociations*: aktualisiert die Schlüsselwortverknüpfungen

Im *personalization*-Package ist die Session-Bean *Rating* nebst zwei Hilfsklassen *RatingAtom* und *RatingAtomList* untergebracht, die den Zugriff auf die in der Datenbank gespeicherten Ressour-

cenbewertungen vornimmt und aggregierte Bewertungen berechnet. In Kapitel 5.4.2 Berechnung von aggregierten Bewertungen werden diese Klassen noch näher betrachtet.

Die Klasse *CorrBasedRecommendations* im *personalization.cf*-Package implementiert als Session-Bean den auf dem Pearson-Korrelationskoeffizienten basierenden Filteralgorithmus. Zur Berechnung von Vorhersagen werden aus Geschwindigkeitsgründen direkte Datenbankzugriffe gemacht, da sich viele Berechnungen – vor allem über alle Benutzer – bereits in der Datenbank tätigen lassen. Bei der ersten abgefragten Vorhersage einer Session-Bean-Instanz wird dazu das Bewertungsniveau aller Benutzer als jeweils durchschnittliche Bewertung in der Datenbank berechnet. Der nötige SQL-Befehl für DB2 lautet:

```
SELECT uid, AVG(rating) AS avg_rating
FROM itemrating_aggregated
GROUP BY uid
```

Die Daten werden anschließend im Speicher gehalten und bei der Berechnung der gewichteten Summe für die Vorhersage abgerufen. Die Abfrage der relevanten Bewertungsdaten und die Berechnung der Benutzerähnlichkeit wurden ebenfalls mit SQL-Befehlen umgesetzt und sind bei entsprechender Wahl von Datenbankindizes sehr effizient.

Im *util*-package schließlich sind noch einige Hilfsklassen für den generischen Datenbankzugriff, die Zerlegung von Texten in einzelne Wörter und für die Benutzeranmeldung enthalten. Für alle Session-Beans gibt es zudem so genannte Access-Beans, die einen Stub oder Proxy zum Zugriff auf die zugehörigen Session-Beans bereitstellen. Damit lässt sich der Zugriff auf die Beans aus der Webanwendung heraus sehr viel leichter realisieren, da die Entity Java Beans spezifischen Methodenaufrufe gekapselt sind. Eine an sich entfernte Session-Bean kann dadurch wie eine gewöhnliche, lokale Java-Klasse behandelt und instanziert werden.

Die Klassen und Java Server Pages für die Benutzungsoberfläche sind in den Paketen unterhalb von *miniportal* enthalten. Die Java Server Pages werden direkt vom Benutzer angesteuert, wenn er die entsprechende URL im Browser abruft. Zur Wahl stehen

- index.jsp: die Startseite mit Darstellung von Kategorien und Ressourcenlisten
- details.jsp: stellt Details zu einer Ressource mit Bewertung und ähnlichen Ressourcen dar
- search.jsp: gibt die gefundenen Ressourcen wieder, wenn eine Suche angestoßen wurde
- admin.jsp: hier kann die Schlüsselwortextraktion und -Verknüpfung neu gestartet werden

Mit den Parametern der Seiten wird festgelegt, welche Kategorien und Ressourcen abgerufen werden sollen oder wonach gesucht wird. Ein Beispiel ist der Aufruf, der zum Aufbau der in Abbildung 43 teilweise dargestellten Detailansicht geführt hat. Der *c_id*-Parameter steuert, welche Kategorie abgefragt wird und der *i_id*-Parameter, welche Ressource.

```
/details.jsp?c_id=4&i_id=50
```

Ferner sind im *miniportal*-Package noch weitere Dateien wie die Klassen der Tab-Bibliothek, die für den Aufbau der Weboberfläche verwendet werden, enthalten. Sie werden hier aber nicht näher erläutert, da sie in einen Formalismus gegossen lediglich für die Auslagerung von Code aus den HTML-lastigen JSP-Dateien sorgen und primär Daten aus den Session-Beans – und damit der zweiten Schicht des Portals – holen.

Weitere Informationen zur Implementierung können auch direkt aus dem Programmcode oder der zugehörigen JavaDoc-Dokumentation entnommen werden, die beide unter

http://www.klossek3000.de/cf/

oder auf der beigelegten CD verfügbar sind.

5.4 Neuralgische Punkte im entwickelten System

An dieser Stelle werden noch die Programmteile etwas näher vorgestellt, die die Auswahl von Optimierungsansätzen aus Kapitel 4 implementieren. Im Folgenden wird als erster Punkt beschrieben, wie die Bewertungen von Benutzern unter Berücksichtigung der Bewertungsquelle aus der Datenbank gelesen und in sie geschrieben werden. Als zweiter Punkt wird die Berechnung der aggregierten Bewertungen innerhalb der Session-Bean *Rating* näher erläutert. Mit dem dritten Punkt wird schließlich die Vorgehensweise bei der Schlüsselwortextraktion deutlich gemacht.

5.4.1 Speichern und Auslesen von Bewertungen

Immer wenn der Benutzer eine Bewertung veranlasst hat – sei es durch Abruf einer Ressource, das Auftreten einer Ressource im Suchergebnis oder eine explizit vorgenommene Bewertung – dann wird dazu die Session-Bean *Rating* verwendet. Dazu wird zunächst mit der Hilfsklasse *RatingAccessBean* eine Instanz der Session-Bean geholt und anschließend die gewünschte Bewertungsmethode aufgerufen. Zur Auswahl stehen

```
rateItemExplicit ( Long uid, Long iid, Double rating )

rateItemByClick ( Long uid, Long iid )

rateItemBySearchResult ( Long uid, Long iid, Integer position )
```

Die drei Methoden nehmen jeweils die Schlüssel des bewertenden Benutzers und der zu bewertenden Ressource entgegen. Der Methode zur Speicherung expliziter Bewertungen wird zusätzlich die Bewertungsangabe auf einer Skala von 0 bis 1 übergeben. Der *rateItemBySearchResult*-Methode wird die Position der betroffenen Ressource im Suchergebnis übergeben. Damit werden Ressourcen, die weiter oben im Suchergebnis auftreten, höher bewertet, als solche, die weiter unten stehen. In *MiniPortal* ist die Suche jedoch nicht nach Relevanz sortiert, da die verwendete EJB-QL-Abfragesprache (siehe [SUN01]) keine geeigneten Möglichkeiten hierfür anbietet, so dass die Reihenfolge mit der Übergabe von *null* als Position auch unberücksichtigt bleiben kann. Dann wird genauso wie bei der *rateItemByClick*-Methode immer mit der Standardbewertung von 0,5 bewertet.

Alle drei Methoden speichern die Bewertungen direkt via JDBC in der Datenbanktabelle *itemrating*. Der Weg über eine Entity-Bean wurde nicht gegangen, da der Zugriff auf die Bewertungsdaten sehr schnell erfolgen muss und zudem eine hohe Flexibilität benötigt wird. Das wäre mit Entity-Beans nur in der bean-verwalteten Variante möglich und dort müsste dann ebenfalls ein Zugriff mit JDBC programmiert werden. Nach der Speicherung erfolgt die Aggregation durch Aufruf der *calcAggregatedRating*-Methode (siehe 5.4.2). Die aggregierte Bewertung wird ebenfalls in der Datenbank gespeichert, um bei der Voraussage von Bewertungen schnell darauf zugreifen zu können (Caching).

Neben Methoden zur Speicherung der Bewertungen verfügt die Klasse über Funktionalität zum Auslesen, die beispielsweise zur Anzeige in der Detailansicht einer Ressource verwendet wird.

```
getLatestRating ( Long uid, Long iid, Integer type )

getRatingSequence ( Long uid, Long iid, int mode )
```

```
getAverageUserRating ( Long uid )

getAverageItemRating ( Long iid )
```

Mit *getLatestRating* kann die letzte Bewertung eines Benutzers und einer Ressource ermittelt werden. Wird *null* als Typ übergeben, dann liefert die Methode die letzte insgesamt vorgenommene Bewertung. Wird ein Typ angegeben, dann die letzte Click-, Such- oder explizite Bewertung. Mit *getRatingSequence* werden alle bisherigen Bewertungen zurückgegeben. Der *mode*-Parameter dient zur Unterscheidung von tatsächlichen und aggregierten Bewertungen. Mit *getAverageUserRating* kann schließlich die durchschnittliche Bewertung eines Benutzers und mit *getAverageItemRating* die durchschnittliche Bewertung einer Ressource über alle Benutzer hinweg abgefragt werden. Mit der letztgenannten Methode lässt sich beispielsweise nicht angemeldeten Benutzern ein Grad für die Interessantheit einer Ressource angeben.

5.4.2 Berechnung von aggregierten Bewertungen

Die Berechnung der aggregierten Bewertung wie sie in Kapitel 4.1 Verbreiterung der Datenbasis beschrieben wurde, erfolgt durch Aufruf der *calcAggregatedRating*-Methode der Session-Bean *Rating*.

```
calcAggregatedRating ( Long uid, Long iid )
```

In dieser Methode werden als erstes alle bisherigen Bewertungen des angegebenen Benutzers und der angegebenen Ressource aus der Datenbank geladen. Jede Bewertung wird in ein *RatingAtom*-Objekt gelegt und alle Objekte dann in eine Instanz von *RatingAtomList*. Die eigentliche Berechnung erfolgt durch einen Aufruf der *calcAggregatedRating*-Methode der *RatingAtomList*-Instanz. Durch diese Kapselung ist die eigentliche Berechnung übersichtlich in eine eigene Klasse verpackt. Die berechnete aggregierte Bewertung wird anschließend in der *itemrating_aggregated*-Tabelle der Datenbank gespeichert und steht zur weiteren Verwendung bereit.

Die Klasse *RatingAtom* stellt einen einfachen Datenspeicher für eine Bewertung dar. Sie enthält Attribute für den Bewertungstyp (Suche, Click, Explizit), den Zeitpunkt der Bewertung und den eigentlichem Wert. Neben dem Konstruktor

```
RatingAtom( int type, Timestamp creationdate, double rating )
```

enthält die Klasse noch Accessoren für die Felder.

Die Klasse *RatingAtomList* ist eine Ableitung von *ArrayList* und bietet so schon die Verwaltung von Objekten an. Spezialisierte Methoden erleichtern den typsicheren Zugriff auf die Listendaten.

```
getRatingsByType ( int type, java.sql.Timestamp checkpoint )

getAsAggregatedRatings ()
```

Mit *getRatingsByType* können alle in der Instanz enthaltenen Bewertungen eines Typs geliefert werden und mit *getAsAggregatedRatings* wird für jedes Datum einer enthaltenen Bewertung die aggregierte Bewertung bestimmt. Da beide Methoden die *RatingAtom*-Objekte nach Zeit aufsteigend sortiert zurückgeben, lässt sich damit der Verlauf der Bewertung einer Ressource durch einen Benutzer dokumentieren. Die Methoden dienen als Datenquelle bei der Ausgabe des Bewertungsdiagramms in der Weboberfläche, in Abbildung 47 dargestellt. Hier sind sowohl die aggregierte als auch die tatsächliche Bewertung nebeneinander aufgetragen. Der Benutzer hat die Ressource drei mal abgerufen und so eine aggregierte Bewertung von 4,375 bzw. normalisiert 0,875 erreicht.

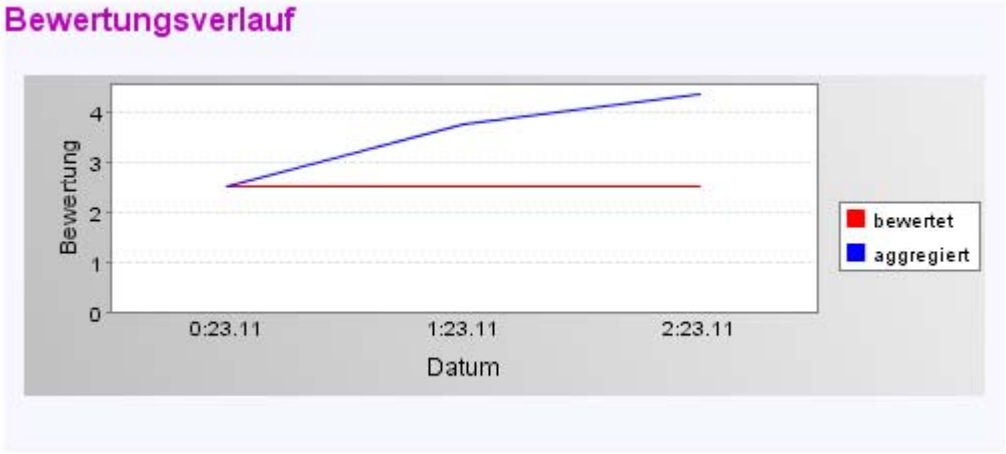

Abbildung 47 - Verlauf einer Ressourcenbewertung (tatsächlich und aggregiert)

Die eigentliche Berechnung der aggregierten Bewertung erfolgt in der Methode calcAggregatedRating. Hier werden die in Kapitel 4.1 Verbreiterung der Datenbasis beschriebenen Vorgaben umgesetzt. Beispielhaft sei hier der Programmcode zur Ableitung der Click-Bewertung aus der Anzahl der Ressourcenabrufe dargestellt (die in Form von Einzelbewertungen in der Datenbank vorliegen).

```
...
RatingAtom ratingClick = null;
if (ratingsClick.size() >= 1) {
    double clickRating = 1.0 - 1.0 /
        Math.pow(2.0, ratingsClick.size());

    ratingClick = new RatingAtom (
        RatingAtom.CLICK,
        new Timestamp( new java.util.Date().getTime() ),
        clickRating
    );
}
...
```

Abbildung 48 - Programmcode zur Berechnung der Click-Bewertung

5.4.3 Schlüsselwortextraktion

Die Verknüpfung der Ressourcen mit den Schlüsselwörtern aus dem Schlüsselwortindex erfolgt mittels der *upateItemKeywordAssociations*-Methode der Session-Bean *Content*. Zu verknüpften Ressourcen kann mit der *getSimilarItems*-Methode eine Menge von ähnlichen Ressourcen geliefert werden. Die Ähnlichkeit wird mit der Menge der Schlüsselwörter und der Kosinusfunktion berechnet.

Die Schlüsselwortverknüpfung beginnt zunächst mit dem Laden der Schlüsselwörter aus der Datenbanktabelle *keyword* in lexikalisch sortierter, aufsteigender Reihenfolge[58]. Die Schlüsselwörter stehen anschließend in einem einfachen Array zur Verfügung, wobei eine Klasse *KeywordData* zur Kapselung eines Schlüsselwortes eingesetzt wird.

[58] Es ist vertretbar, die Schlüsselwörter vollständig in den Hauptspeicher zu laden, da die Anzahl beschränkt ist und kein Speicherplatzproblem besteht.

Anschließend werden die neu zu indizierenden Ressourcen aus der Datenbank geladen und nacheinander indiziert. Dazu wird die statische Methode *process* der Klasse *TextTokenizer* benutzt. Sie extrahiert mit der Java-*StringTokenizer*-Klasse alle Wörter aus dem übergeben Text, filtert Dubletten aus und sortiert die Wörter in lexikalischer Reihenfolge aufsteigend.

```
StringTokenizer strTok = new StringTokenizer( text,
    " ,.?!-;\t\n\r()[]{}" );
```

Hier könnte noch weitere Vorverarbeitungsschritte zur Verbesserung der Datenqualität erfolgen wie das Entfernen von Stoppwörtern oder die Stammformenanalyse (Stemming, siehe [BR99], Kapitel 7.2.3).

Mit der Rückgabe der extrahierten Wörter erfolgt der Abgleich mit den Schlüsselwörtern des Indexes. Beide Listen wurden sortiert, damit der Abgleich schnell und im Reißverschlussverfahren ähnlich wie bei Merge-Sort erfolgen kann. Wörter der aktuellen Ressource, die im Schlüsselwortindex vorhanden sind, werden mit einem neuen Datensatz in der *itemkeywordrel*-Tabelle gespeichert, auf der dann die *getSimilarItems*-Methode arbeitet.

5.4.4 Logdateianalyse

Anfänglich war die Gewinnung impliziter Verhaltensdaten mit Hilfe der Logdateianalyse geplant. Hierzu wurde auch Code geschrieben, der Logdateien parst und die nützlichen Informationen extrahiert. Der Ablauf folgt der Darstellung in Abbildung 49. Der Ansatz wurde jedoch nicht mehr weiterverfolgt, da sich die direkte Erfassung von Ressourcenabrufen und Suchanfragen in der Datenbank als einfacher erwiesen hat.

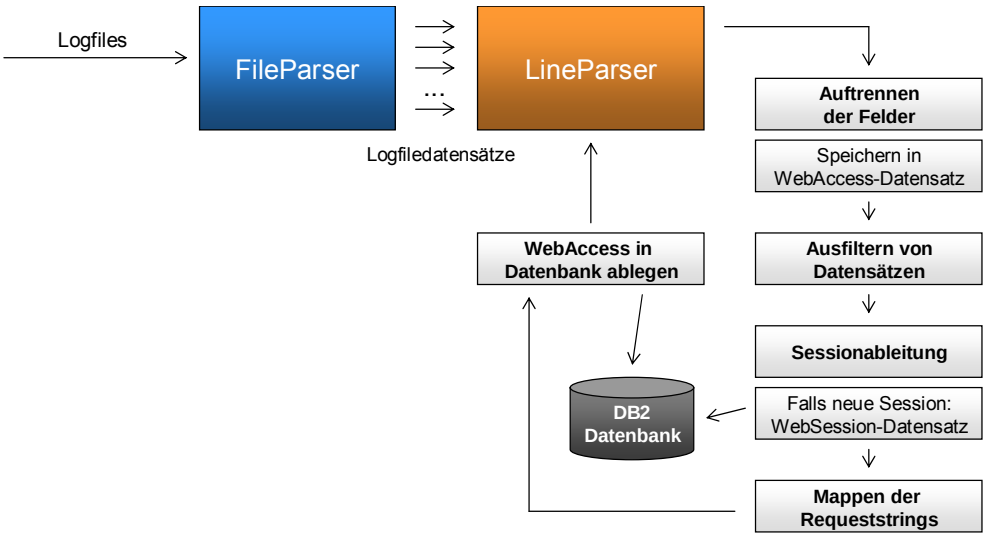

Abbildung 49 - Programmfluss Protokollanalyse

Zudem stellte sich ein grundsätzliches Problem: Innerhalb eines abgerufenen HTML-Dokumentes können verschiedene Informationen nebeneinander und gleichberechtigt enthalten sein. In der Logdatei steht jedoch nur ein einzelner Datensatz für den Zugriff auf das Dokument. Besonders bei den wie hier dynamisch erzeugten HTML-Dokumenten ist die Zuordnung der tatsächlich abgerufenen Informationen anhand des Protokolleintrages nicht einfach (siehe dazu auch 1.3.2 Repräsentation von Informationen im World Wide Web – Anatomie von Websites und 2.3.5 Logdateianalyse zur Ableitung impliziter, dynamischer Daten).

5.5 Testdaten

Zum Testen der *MiniPortal*-Implementierung wurden Artikel und Spezifikationen aus verschiedenen technischen Bereichen in englischer und deutscher Sprache zusammengetragen und in 17 Kategorien unterteilt, wobei ein Artikel mehreren Kategorien angehören kann. In der Datenbank sind jedoch nicht die Texte selbst gespeichert, sondern nur Kurzfassungen und URLs der Quellen, da damit mögliche rechtliche Probleme verhindert werden. Im Prinzip handelt es sich bei diesem Konzept um eine große Bookmarkverwaltung mit kurzen Inhaltsangaben. Insgesamt wurden 100 Texte zusammengetragen. Die Kategorien sind beispielsweise XML, Java oder Security.

Die Kurzfassung der Texte war wichtig, um die Verknüpfung mit den Schlüsselworten nach dem in Kapitel 4.3.3 und 5.4.3 beschriebenen Verfahren vorzunehmen. An Testdaten wurden 74 Schlüsselwörter zusammengetragen, wobei das Verfahren keinen Unterschied zwischen deutschen und englischen Begriffen macht. Subjektiv hat sich diese Zahl als zu wenig erwiesen, da zwar zu fast allen Ressourcen ähnliche Begriffe geliefert werden, aber manche inhaltlich zu weit entfernt sind. Manche Ressourcen enthalten auch Begriffe, die nicht im Schlüsselwortindex enthalten sind. Hier wäre also eine Aufstockung denkbar.

Die Zahl von Benutzern ist ein ebenfalls kritischer Punkt. Es wurden insgesamt 20 Testbenutzer angelegt und Bewertungen verschiedener Ressourcen vorgenommen, um ein unterschiedliches Benutzerinteresse zu simulieren. Zum Testen der Implementierung hat diese Zahl ausgereicht, für eine präzise numerische Evaluierung ist sie aber zu gering.

Qualitätsaspekte

Wichtig sind zwei zentrale Qualitätsaspekte. Zum einen muss die Menge der Ressourcen ausreichend groß sein, damit sich Personalisierungstechniken überhaupt lohnen. Der Benutzer muss das Gefühl haben, dass viel zu viele Informationen vorliegen und er Assistenz bei der Erschließung der Informationen braucht. Das ist bei 100 Ressourcen bereits der Fall, vor allem auch da eine Ressource häufig mehreren Kategorien zugeordnet wurde. In einigen Kategorien werden daher sehr viele Informationen ausgegeben und eine Personalisierung ist wünschenswert. Generell sind aber noch deutlich mehr Testdaten ratsam.

Zum anderen sollten die Themengebiete ausreichend breit gestreut sein, um verschiedenen Interessensschwerpunkten der Benutzer genügen zu können. Beispielsweise könnte es Ressourcen zu Java- und Delphi-Entwicklung geben und gleichzeitig auch zu XML-Themen. Ein Anwender der Interesse an Java hat, wird vermutlich weniger an Delphi interessiert sein. Beide könnten sich jedoch für XML interessieren, so dass hier eine Brücke geschlagen wird, die beiden Benutzern neue Ressourcen erschließt.

6 Zusammenfassung und Ausblick

In dieser Arbeit wurde eine breite Einführung in die Personalisierung im Internet gegeben und das Verfahren des Kollaborativen Filterns als besonders erfolgreiche Technik hervorgehoben. Zur Optimierung dieser Technik wurden drei Ansätze vorgestellt, die die Qualität von Bewertungsvorhersagen verbessern. Anschließend wurde die Umsetzung dieser Ansätze mit Hilfe eines prototypischen Informationssystems beschrieben.

In Kapitel 1 „Grundlagen“ wurde die Basis für die späteren Betrachtungen der Personalisierung mit der Erklärung vom Aufbau von Websites, dem Abruf von Informationen und der Protokollierung von Zugriffen gelegt. Dem Aspekt des Datenschutzes wurde an dieser Stelle ein eigener Bereich gewidmet, da es sich bei der Erfassung von individuellen Benutzerprofilen aus Datenschutzsicht um ein kritisches Thema handelt, die Verfügbarkeit von Profilen aber ein wesentlicher Bestandteil aller Personalisierungstechniken ist. Im Anhang findet sich dazu eine Datenschutz-Richtlinie für das implementierte Informationssystem, die den Vorgaben des P3P-Projektes entspricht (siehe Kapitel 1.4.2).

Kapitel 2 „Personalisierung“ erläuterte intensiv die Grundlagen der Personalisierung im Internet. Sowohl die zu personalisierenden Bestandteile einer Website als auch die dazu einsetzbaren Verfahren wie das Regel- und Inhaltbasierte Filtern wurden beschrieben. Die dargestellten Verfahren wurden zudem in einer Vergleichsmatrix gegenüber gestellt und einzelne Vor- und Nachteile bewertet. Es wurde darauf geachtet, keine Einschränkung auf einzelne Typen von Informationssystemen vorzunehmen, sondern Informations- und Unternehmensportale, Communities und Online-Shops gleichermaßen anzusprechen. Ferner behandelte dieses Kapitel mit der Darlegung von Benutzerprofilen, impliziten Verhaltensdaten und expliziten Formen der Ressourcenbewertung auch die Datengrundlage der Personalisierung.

Im folgenden Kapitel 3 „Kollaboratives Filtern“ ging es dann um die spezielle Personalisierungstechnik des Kollaborativen Filterns. Dort wurde zunächst die grundlegende Arbeitsweise beschrieben und an Fallbeispielen erläutert. Verschiedene Algorithmen zur Berechnung von Bewertungsvorhersagen wurden untersucht und Hinweise zur Erfolgsmessung gegeben. Es wurde gezeigt, dass sich das Kollaborative Filtern leicht einsetzen lässt und für die Betreiber von Informationssystemen nur ein kleiner Arbeitsaufwand anfällt, da die Bewertungen von Ressourcen zur Festlegung der Präferenzen durch die Benutzer selbst vorgenommen werden. Ferner wurde beschrieben, dass keine inhaltliche Analyse von Ressourcendaten nötig ist, und sich das Verfahren daher gut für schwer zu klassifizierende Medien wie Grafik, Video und Musik eignet, bei denen andere Personalisierungstechniken weniger erfolgreich sind. Abschließend wurden bestehenden Schwächen der Technik nebst Vorschlägen zur Verbesserung betrachtet.

Diese Schwächen des Kollaborativen Filterns – vor allem das Problem des dünnen Datenbestandes aufgrund zu selten vorgenommener Bewertungen und die mangelnde Unterstützung bei änderndem Benutzerverhalten – wurden in Kapitel 4 „Optimierung“ aufgegriffen. Als Verbesserungsmöglichkeiten wurden die drei folgenden Ansätze vorgeschlagen und konzipiert

- Verbreiterung der Datenbasis
- Einbeziehung von Zeit zur Berücksichtigung von Interessensänderungen
- Verknüpfung mit Inhaltsbasierten Filtern

Zur Verbreiterung der Datenbasis wurden im ersten Ansatz verschiedene explizite und implizite Bewertungsquellen zueinander in Bezug gesetzt und eine Verrechnung modelliert, so dass alle verfügbaren und geeigneten Daten in einem Informationssystem als Bewertungsgrundlage genutzt werden können. Darauf aufbauend ging es im zweiten Ansatz um die Anpassung der Bewertungen durch Erkennung von verändertem Benutzerverhalten unter Einsatz der neuen Datenkomponente Zeit.

Mit dem dritten Ansatz wurde die Verknüpfung von Kollaborativem und Inhaltsbasierten Filtern verfolgt. Beschrieben wurde ein Verfahren, das Schlüsselwörter aus textuellen Informationen extrahiert und damit eine inhaltliche Ähnlichkeit zwischen Ressourcen bestimmen lässt. Die gewonnene Ähnlichkeitsfunktion diente zur Ableitung von Ressourcenbewertungen, zur Verbreiterung der Datenbasis und für den Fall, dass der Kollaborative Filter keine Vorhersage berechnen kann, weil keine Bewertungen vorhanden sind.

Im prototypischen Informationssystem, das in Kapitel 5 „Implementierung“ vorgestellt wurde, sind die beschriebenen Ansätze teilweise umgesetzt worden. Verzichtet wurde auf die Einbeziehung der Zeit, da ein sinnvoller Test der Funktionstüchtigkeit aufgrund der zeitlichen Datenkomponente einen längeren Zeitraum erfordert hätte, der im Rahmen der Diplomarbeit nicht gegeben war. Dennoch wurden mit den anderen Ansätzen und der Implementierung eines Kollaborativen Filters basierend auf dem Pearson-Korrelationskoeffizienten gute Erfahrungen gesammelt, die im fünften Kapitel näher erläutert wurden. Nicht zuletzt hat dazu auch die zum Einsatz gekommene J2EE-Technik beigetragen.

Die Kombination von impliziten und expliziten Bewertungen führte auf den Testdaten zu einem erheblichen Anstieg der Ressourcenbewertungen und damit zu einer Verbreiterung der Datenbasis. Dadurch wurden die Vorhersagen für unbewertete Ressourcen qualitativ verbessert. Durch den Einsatz des Inhaltsbasierten Filterns war zudem die Ausgabe von Empfehlungen ähnlicher Ressourcen möglich, selbst wenn ein simulierter Testbenutzer keine Bewertungen vorgenommen hatte. Bei manchen Ressourcen konnten jedoch keine inhaltsbasierten Ähnlichkeiten zu anderen Ressourcen ermittelt werden, da die entsprechenden Schlüsselwörter im Index gefehlt haben. Ein einfacher Erweiterungsansatz wäre daher, den Schlüsselwortindex mit weiteren Wörtern auszustatten – beispielsweise durch eine automatisierte Analyse der textuellen Inhalte.

Die Implementierung des Informationssystems mit einfacher Inhalte- und Benutzerverwaltung war aufgrund der eingesetzten J2EE-Technologie und der nötigen Einarbeitung des Autors zeitintensiv. Möglicherweise hätte man hier Abstriche machen und stattdessen die im vierten Kapitel beschriebenen Optimierungsansätze vollständig umsetzen können. Problematisch an dieser Vorgehensweise wäre aber gewesen, dass die nötige Einordnung und Beurteilung des Kollaborativen Filterns ohne ein solches Informationssystem nicht gut möglich wäre, da das Benutzerverhalten simuliert und untersucht werden musste. Der Autor verfolgte daher den eingeschlagenen Weg.

Ausblick

Durch die aus der Beschäftigung mit der Personalisierung und dem Kollaborativen Filtern gewonnen Erkenntnisse erscheinen verschiedene Weiterentwicklungsmöglichkeiten in direkter Anknüpfung an diese Arbeit interessant.

So wäre es denkbar, die vorgeschlagene Erweiterung der Datenbasis um die Komponente Zeit in die Implementierung einzubeziehen, um damit die Veränderung des langfristigen Benutzerinteresses zu untersuchen. Zusätzlich wäre ein Vergleich der verschiedenen Ansätze in einer quantitativen Evaluierung möglich, wozu allerdings ein repräsentativer Testdatenbestand benötigt wird. Dieser Datenbestand könnte aus einem öffentlich zugänglichen Informationssystem gewonnen werden, das die beschriebenen Verfahren implementiert.

Ebenfalls wäre es in diesem Zusammenhang interessant, eine Formel für die Zahl kritischer Ressourcenbewertungen zu entwickeln, ab der ein Informationssystem durch Kollaboratives Filtern eine erfolgreiche Personalisierung anbieten kann. Die Formel müsste mindestens die Anzahl der verwalteten Ressourcen und der Benutzer als Eingabe entgegennehmen. Vermutlich sind noch weitere Parameter nötig wie die durchschnittliche Nutzungshäufigkeit und Angaben über die tendenzielle Bereitschaft der Benutzer, Bewertungen vorzunehmen. Nützlich wäre eine solche Formel, wenn der Einsatz des Kollaborativen Filterns in Internetinformationssystemen in Betracht gezogen wird und vorab ein möglicher Erfolg vorhergesagt werden soll.

In der Praxis setzt sich die Personalisierung in Internetportalen und Online-Shops nur langsam durch. Dennoch gibt es vor allem bei großen Systemen immer mehr Anwendungen, die Benutzern eine personalisierte Sicht des Datenbestandes bieten. Sehr nützlich wäre die Personalisierung auch in Internetanwendungen für Mobiltelefone und Personal Digital Assistants (PDAs), da sowohl die Übertragungsbandbreite als auch die Geräteeigenschaften der Menge der übermittelten Daten enge Grenzen setzen. Der Benutzer könnte nur die Informationen angezeigt bekommen, die für ihn in der jeweiligen Situation tatsächlich nützlich sind. Die Konfiguration der Präferenzen wäre mit einer Variante des gleichen Informationssystems möglich, das vom Arbeitsplatz aus komfortabel bedient werden kann.

Eine weitere zukünftige Einsatzmöglichkeit besteht für Internetsuchmaschinen wie Google, bei denen eine Personalisierung denkbar ist (siehe [Heise03]). Gegenwärtig werden dort nur einfache Angebote wie die Wahl der Sprache bereitgestellt. Durch die Nutzung der Personalisierung wäre es vorstellbar, dass die Menge der gelieferten Suchergebnisse durch die Präferenzen der einzelnen Benutzer modifiziert und eingeschränkt wird. Mit Kollaborativem Filtern könnte man die Selektion der Inhalte basierend auf den Ähnlichkeiten zu anderen Benutzern vornehmen.

Generell wird die Durchdringung von Informationssystemen mit Personalisierungsverfahren davon abhängen, inwieweit standardisierte und preiswerte Standardsoftware für diese Zwecke verfügbar ist. So wäre beispielsweise eine Anwendung von Kollaborativen Filtern in Online-Fernsehprogrammzeitschriften möglich, um über den Geschmack der unterschiedlichen Zuschauer individuelle Empfehlungen abgeben zu können. Dazu geeignete, preiswerte und leicht zu installierende Softwareprodukte würden Websitebetreiber in Zukunft sicherlich stärker dazu animieren, Personalisierungsmöglichkeiten anzubieten.

Zudem ist die Personalisierung auch in der Offline-Welt auf dem Vormarsch: Mit Kundenkarten erfassen große Kaufhausketten und andere Anbieter schon seit längerem das Kaufverhalten ihrer Kunden. Bisher werden die Daten eher für Marktforschungszwecke genutzt (siehe [BS99]). Zukünftig wäre es aber denkbar, an der Kasse individuelle Empfehlungen von interessanten Produkten auszusprechen, wozu sich das Kollaborative Filtern hervorragend eignet.

Anhang

A Datenbankerstellungsbefehle in DB2

Tabelle item

```
CREATE TABLE KLOSSEK.ITEM (
    "IID" BIGINT  NOT NULL ,
    "TYPE" INTEGER  NOT NULL  WITH DEFAULT 0,
    "URI" VARCHAR (255) ,
    "TITLE" VARCHAR (255) NOT NULL WITH DEFAULT '',
    "SHORTDESC" CLOB (16384)  NOT  LOGGED  NOT  COMPACT ,
    "CREATIONDATE" TIMESTAMP  NOT NULL,
    PRIMARY KEY (IID)
)  DATA CAPTURE NONE   IN USERSPACE1
```

Tabelle category

```
CREATE TABLE KLOSSEK.CATEGORY (
    "CID" BIGINT  NOT NULL,
    "PARENTID" BIGINT  NOT NULL  WITH DEFAULT -1,
    "TITLE" VARCHAR (255)  NOT NULL WITH DEFAULT '',
    "TYPE" INTEGER  NOT NULL  WITH DEFAULT 0,
    "SHORTDESC" VARCHAR (255),
    "CREATIONDATE" TIMESTAMP  NOT NULL,
    PRIMARY KEY (CID)
)  DATA CAPTURE NONE   IN USERSPACE1
```

Tabelle itemcategoryrel

```
CREATE TABLE KLOSSEK.ITEMCATEGORYREL (
    "IID" BIGINT  NOT NULL ,
    "CID" BIGINT  NOT NULL ,
    PRIMARY KEY (IID, CID)
)  DATA CAPTURE NONE   IN USERSPACE1
```

Tabelle user

```
CREATE TABLE KLOSSEK.USER (
    "UID" BIGINT  NOT NULL ,
    "LOGINNAME" VARCHAR (64)  NOT NULL WITH DEFAULT '',
    "PASSWORD" VARCHAR (32)  NOT NULL WITH DEFAULT '',
    "NUMBEROFLOGINS" INT   NOT NULL WITH DEFAULT 0,
    "FIRSTNAME" VARCHAR (64)  NOT NULL WITH DEFAULT '',
    "LASTNAME" VARCHAR (64)  NOT NULL WITH DEFAULT '',
    "GENDER" CHAR (1)  NOT NULL WITH DEFAULT 'M',
    "EMAIL" VARCHAR (64)  NOT NULL WITH DEFAULT '',
    "PHONE" VARCHAR (32) ,
```

```
    "MOBILEPHONE" VARCHAR (32) ,
    "FAX" VARCHAR (32) ,
    "DATEOFBIRTH" TIMESTAMP NOT NULL WITH DEFAULT '1980-01-01 00:00:00',
    "ZIPCODE" VARCHAR (10)  NOT NULL WITH DEFAULT '',
    "CITY" VARCHAR (64)  NOT NULL WITH DEFAULT '',
    "COUNTRY" CHARACTER (3)  NOT NULL WITH DEFAULT 'DE',
    "LANGUAGE" CHARACTER (5)  NOT NULL WITH DEFAULT 'de_DE',
    "JOB" VARCHAR (64) ,
    "COLORSTYLE" INT  NOT NULL WITH DEFAULT 1,
    PRIMARY KEY (UID)
)  DATA CAPTURE NONE   IN USERSPACE1
```

Tabelle itemrating

```
CREATE TABLE KLOSSEK.ITEMRATING (
    "IID" BIGINT  NOT NULL ,
    "UID" BIGINT  NOT NULL ,
    "TYPE" INTEGER  NOT NULL  WITH DEFAULT 0,
    "RATING" DOUBLE  NOT NULL ,
    "CREATIONDATE" TIMESTAMP  NOT NULL,
    PRIMARY KEY (IID, UID, CREATIONDATE)
)  DATA CAPTURE NONE   IN USERSPACE1
```

Tabelle itemrating_aggregated

```
CREATE TABLE KLOSSEK.ITEMRATING_AGGREGATED (
    "IID" BIGINT  NOT NULL ,
    "UID" BIGINT  NOT NULL ,
    "RATING" DOUBLE  NOT NULL ,
    "CREATIONDATE" TIMESTAMP  NOT NULL,
    PRIMARY KEY (IID, UID)
)  DATA CAPTURE NONE   IN USERSPACE1
```

Tabelle keyword

```
CREATE TABLE KLOSSEK.KEYWORD (
    "KID" BIGINT  NOT NULL ,
    "MID" BIGINT  NOT NULL ,
    "TOKEN" VARCHAR (64)  NOT NULL ,
    "LANG" VARCHAR (5) NOT NULL ,
    "CREATIONDATE" TIMESTAMP  NOT NULL ,
    PRIMARY KEY (KID)
) DATA CAPTURE NONE   IN USERSPACE1
```

Tabelle itemkeywordrel

```
CREATE TABLE KLOSSEK.ITEMKEYWORDREL (
    "IID" BIGINT  NOT NULL ,
    "MID" BIGINT  NOT NULL ,
    PRIMARY KEY (IID, MID)
) DATA CAPTURE NONE   IN USERSPACE1
```

B P3P-Datenschutzrichtline für MiniPortal

```
<?xml version="1.0" encoding="UTF-8"?>
<POLICIES xmlns="http://www.w3.org/2002/01/P3Pv1">
  <EXPIRY max-age="86400" />

  <!-- Fehlerfrei geprüft mit W3C P3P Validator http://www.w3.org/P3P/validator.html -->

  <!-- Die Datenschutzrichtlinie für das MiniPortal -->
  <POLICY name="Datenschutz"
    discuri="http://www.klossek3000.de/cf/miniportal/Datenschutzinfo.html"
    opturi="http://www.klossek3000.de/cf/miniportal/Datenschutzinfo.html" xml:lang="de">

    <!-- Die Entität, die hinter MiniPortal steht. Das ist der Autor der Arbeit -->
    <ENTITY>
      <DATA-GROUP>
        <DATA ref="#business.contact-info.telecom.telephone.number">
          +49-172-6104537</DATA>
        <DATA ref="#business.contact-info.online.email">privacy@klossek3000.de</DATA>
        <DATA ref="#business.contact-info.online.uri">
          http://www.klossek3000.de/cf/miniportal</DATA>
        <DATA ref="#business.contact-info.postal.organization">Martin Klossek</DATA>
        <DATA ref="#business.contact-info.postal.street">Jügelstraße 1/230</DATA>
        <DATA ref="#business.contact-info.postal.city">Frankfurt am Main</DATA>
        <DATA ref="#business.contact-info.postal.postalcode">60325</DATA>
        <DATA ref="#business.contact-info.postal.country">Germany</DATA>
        <DATA ref="#business.name">Martin Klossek</DATA>
      </DATA-GROUP>
    </ENTITY>

    <!-- Definiert, ob und welcher Zugriff auf die gesammelten Daten durch den
    Benutzer möglich ist -->
    <ACCESS>
      <!-- für MiniPortal wird keine Zugriffsmöglichkeit angeboten. Möglicherweise wird
       das manche Benutzer veranlassen, dass Portal nicht zu benutzen, da sie Ihre
       gesammelten Daten nicht einsehen können. Würde man vollen Zugriff integrieren,
       müßte hier stattdessen <all/> stehen. Dazwischen gibt es Abstufungen. -->
      <none/>
    </ACCESS>

    <!-- Für Streit- oder Schadensfälle wird eine Kontaktadresse definiert -->
    <DISPUTES-GROUP>
      <DISPUTES resolution-type="service"
        service="http://www.klossek3000.de/cf/miniportal/Support.html"
        short-description="Dispute/Assurence">
        <LONG-DESCRIPTION>Supportkontakt: support@klossek3000.de</LONG-DESCRIPTION>
        <REMEDIES>
          <!-- Vorfälle in Bezug auf Datenschutz werden korrigiert, wenn reklamiert -->
          <correct />
        </REMEDIES>
      </DISPUTES>
    </DISPUTES-GROUP>

    <!-- Statement für die Overall Information Gruppe: Hier wird definiert, welche Daten
     verarbeitet und wie lange sie gespeichert werden sowie was mit ihnen gemacht wird.
     Eine Aufspaltung in verschiedene Statements nach der Art der Daten und ihrer
     Verarbeitung wäre möglich gewesen.Da aber alle Daten auf die eine oder andere Art zur
     Personalisierung verwendet werden, wurde nur ein Statement gewählt. -->
    <STATEMENT>
      <EXTENSION optional="yes">
        <GROUP-INFO name="Overall Information" />
      </EXTENSION>

      <!-- Was passiert mit den Daten? Beschreibung im Klartext -->
      <CONSEQUENCE>
        Massive Verwendung aller Daten zur Personalisierung und Anpassung der
        Website. Daten werden aber in keiner Weise weitergegeben.
      </CONSEQUENCE>

      <!-- Der Grund für die Datensammlung -->
      <PURPOSE>
```

```
        <!-- Da MiniPortal die Daten auf verschiedene Weise analysiert, direkte Aktionen
         ausgeführt werden, der Kontakt mit dem Benutzer möglich sein soll und die Daten
         auch nicht anonym verarbeitet werden, werden eine Reihe von Zwecken gesetzt.
         Keiner der Zwecke ist per opt-in oder opt-out möglich, da MiniPortal hierfür
         keinerlei Mechanismen vorsieht. Fordert das ein Benutzer jedoch, kann ihn
         MiniPortal in dieser Frage nicht unterstützen. -->

        <current/> <!-- Daten für Zwecke der Anpassung in Sitzung verarbeitet -->
        <admin/> <!-- die Daten werden zur Optimierung der Technik benutzt -->
        <develop/> <!-- die Daten werden zur Weiterentwicklung und Forschung benutzt -->
        <tailoring/> <!-- es erfolgt eine Anpassung innerhalb der Sitzung -->
        <individual-analysis/> <!-- es erfolgt eine Analyse der Benutzerdaten -->
        <individual-decision/> <!-- Entscheidungen erfolgen anhand der Benutzerdaten -->
        <contact/> <!-- auch später soll Kontakt mit den Benutzern möglich sein -->
        <telemarketing/> <!-- auch später soll telefonischer Kontakt möglich sein -->
      </PURPOSE>

      <!-- Der Empfänger der Daten -->
      <RECIPIENT>
        <!-- Bei MiniPortal verwenden nur wir die Daten und geben sie nicht weiter. Würde
         man die Daten weitergeben, könnten hier andere Elemente stehen. <same/> bsp.,
         wenn man die Daten an Personen weitergibt, die die gleichen Datenschutzrichtlin-
         ien wie wir haben. Oder <public/> wenn es sich um ein öffentliches Diskussions-
         forum halten würde und Daten an die Öffentlichkeit wandern -->
        <ours/>
      </RECIPIENT>

      <!-- Zeitdauer, für die die gesammelten Daten gehalten werden -->
      <RETENTION>
        <!-- Bei MiniPortal sind alle Daten auf unbestimmte Zeit gespeichert, um auch
         zukünftig Personalisierung zu ermöglichen. Würden die Daten nach einiger Zeit
         gelöscht oder gar nicht gespeichert, könnte hier ein anderes Element plaziert
         werden. Das schließt z. B. auch ein Element für eine gesetzliche Aufbewahrungs-
         pflicht ein. -->
        <indefinitely/>
      </RETENTION>

      <!-- Beschreibung der betroffenen und verwendeten Daten -->
      <DATA-GROUP>
        <DATA ref="#dynamic.clickstream"/>
        <DATA ref="#dynamic.http"/>
        <DATA ref="#dynamic.clientevents"/>
        <DATA ref="#dynamic.searchtext"/>
        <DATA ref="#dynamic.interactionrecord"/>
        <DATA ref="#dynamic.cookies">
          <CATEGORIES> <computer /> <demographic /> <online /> <physical />
            <preference /> <purchase /> <state /> <uniqueid /> </CATEGORIES>
        </DATA>
        <DATA ref="#user.name.given" /> <!-- der Vorname des Benutzers -->
        <DATA ref="#user.name.family" /> <!-- der Nachname des Benutzers -->
        <DATA ref="#user.jobtitle" /> <!-- der Beruf des Benutzers -->
        <DATA ref="#user.home-info" /> <!-- umfasst Adress-, E-Mail- und Telefondaten -->
        <DATA ref="#user.login.id" /> <!-- Benutzername -->
        <DATA ref="#user.login.password" /> <!-- Passwort -->
        <DATA ref="#user.gender" /> <!-- Geschlecht -->
        <DATA ref="#user.bdate" /> <!-- Geburtsdatum -->
        <DATA ref="#dynamic.miscdata">
          <CATEGORIES><navigation /></CATEGORIES> <!-- Anzahl der Anmeldungen -->
        </DATA>
        <DATA ref="#dynamic.miscdata">
          <CATEGORIES><preference /></CATEGORIES> <!-- Farbstil -->
        </DATA>
        <DATA ref="#dynamic.miscdata">
          <CATEGORIES><preference /></CATEGORIES> <!-- Sprache -->
        </DATA>
        <DATA ref="#dynamic.miscdata">
          <CATEGORIES><interactive /></CATEGORIES> <!--explizite Ressourcenbewertungen-->
        </DATA>
        <DATA ref="#dynamic.miscdata">
          <CATEGORIES><navigation /></CATEGORIES> <!-- Zählung von Ressourcenabrufen -->
        </DATA>
      </DATA-GROUP>
    </STATEMENT>

  </POLICY>
</POLICIES>
```

C Quellenverzeichnis

[Aar03] **Aaronson, Jack: "Personalization, Meet Mass Customization"**
Online-Artikel auf www.clickz.com, 16. Oktober 2003, abgerufen am 22. November 2003
http://www.clickz.com/crm/crm_strat/article.php/3091931

[Apa1] **Apache Foundation: "Apache Tutorial: Introduction to Server Side Includes" aus Dokumentation der Apache 1.3 Webserversoftware**
http://httpd.apache.org/docs/howto/ssi.html

[Apa2] **Apache Foundation: Kapitel "Log Files" aus Dokumentation der Apache 2.0 Webserversoftware**
http://httpd.apache.org/docs-2.0/logs.html

[AZ03] **ARD und ZDF: "ARD/ZDF-Online-Studie 2002"**
Programmdirektion Erstes Deutsches Fernsehen, 2003, Arnulfstraße 42, 80335 München
http://www.daserste.de/service/ardonl02.pdf

[Bau99] **Baudisch, Patrick: "Joining Collaborative and Content-based Filtering"**
Integrated Publication and Information Systems Institute IPSI, Arbeitspapier wurde gezeigt auf CHI '99 Conference, CHI' 99 Workshop Interacting with Recommender Systems, Pittsburgh, Pennsylvania, USA, Mai 1999
http://www.darmstadt.gmd.de/rec99/WorkingNotes/PatrickBaudischJoiningCollaborativeAndContent-basedFiltering.pdf

[Ber70] **Bertelsmann: "Das Moderne Lexikon", Band "A-Art"**
Bertelsmann Lexikon-Verlag, Gütersloh, 1970

[BHK98] **Breese, J. S., Heckerman, D. und Kadie, C.: "Empirical Analysis of Predictive Algorithms for Collaborative Filtering"**
Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, pp. 43-52, Madison, WI, July 1998, Morgan Kaufmann Publisher

[BLI97] **Bacus Laboratories, Inc.: "Common Log Format", Lombard, USA, 1997**
http://www.bacuslabs.com/WsvlCLF.html

[Bul02] **Bulmahn, Edelgard: "Informationsgesellschaft Deutschland: Fortschrittsbericht zum Aktionsprogramm der Bundesregierung"**
Bundesminsisterium für Bildung und Forschung, Statement von Bundesforschungsministerin Edelgard Bulmahn anlässlich der Pressekonferenz, 06.03.2002, Bundespressekonferenz Berlin
http://www.bmbf.de/pub/mr-20020306.pdf

[BR99] **Baeza-Yates, Ricardo und Ribeiro-Neto, Berthier: "Modern Information Retrieval“**

Addison Wesley/Perason Education Limited, Harlow, England und ACM Press, New York, USA, 1999, ISBN 0-201-39829-X

[BS99] **Buse, Uwe und Schnibben, Cordt: "Der nackte Untertan"**
Artikel im Spiegel, Ausgabe Nr. 27, 5. Juli 1999, Der Spiegel, Hamburg

[BS02] **Bankier, Jean-Gabriel und Schatsky, David: "How to Roll Out a Successful Enterprise Information Portal"**
Jupitermedia Corporation, 2002, New York, Boston
http://enterprise.yahoo.com/pdf/portal/jupiter_killer_app_honeywell.pdf

[CC03] **Carlson, Chistopher N.: "Data Smog, Precision und Recall: Retrievalstrategien zur Ballast-Reduzierung bei Internet-Recherchen"**
Paper zu Vortrag auf ODOK'03, 10. Österreichisches Online-Informationstreffen und 11. Österreichischer Dokumentartag, Salzburg, Paris-Lodron Universität, 2003
http://voeb.uibk.ac.at/odok2003/carlson.pdf

[CS96] **Cheeseman, P. und Stutz, J.: "Bayesian Classification (AutoClass): Theory and Results"**
Kapitel 6 im Buch "Advances in Knowledge Discovery and Data Mining" von Fayyad, Usama M., Piatetsky-Shapiro, Gregory, Smyth, Padhraic und Uthurusamy, Ramasamy (Herausgeber), AAAI Press/MIT Press, 1996
http://citeseer.nj.nec.com/cheeseman96bayesian.html

[DLR77] **Dempster, A., Laird, N. und Rubin, D.: "Maximum likelihood from incomplete data via the EM algorithm"**
Journal of the Royal Statistical Society, 1977, Series B 39:1-38
http://www.jstor.org/view/00359246/di993190/99p0278c/0

[DR02] **Delphi Research: "Customer Portals: the Value Framework"**
Delphi Group, 2002, Ten Post Office Square, Boston, MA 02109
http://enterprise.yahoo.com/pdf/portal/Delphi.pdf

[EK96] **Europäische Kommission: "Die Informationsgesellschaft"**
Generaldirektion Information, Kommunikation, Kultur, Audiovisuelle Medien, Redaktion: Abteilung "Veröffentlichungen", Rue de la Loi 200, B-1049 Bruxelles, 1996
http://europa.eu.int/en/comm/dg10/infcom/euromove/info-soc/de/cover.htm

[EPREU02] **Das Europäische Parlament und der Rat der Europäischen Union: "Richtlinie 2002/58/EG des Europäischen Parlaments und des Rates über die Verarbeitung personenbezogener Daten und den Schutz der Privatsphäre in der elektronischen Kommunikation (Datenschutzrichtlinie für elektronische Kommunikation)"**
Brüssel, vom 12. Juni 2002, in Kraft seit 1. November 2003
http://europa.eu.int/eur-lex/pri/de/oj/dat/2002/l_201/l_20120020731de00370047.pdf

[FCF03] **Farley, Jim, Crawford, William und Flanagan, David: "Java Enterprise in a Nutshell"**

Deutsche Übersetzung der gleichnamigen englischen Originalausgabe, O'Reilly, Köln, 2003, ISBN 3-89721-334-6

[FEBS02] **Ferman, A. Mufit, Errico, James H., van Beek, Peter, Sezan, M. Ibrahim: "Content-based filtering and personalization using structured metadata"**
Proceedings of the second ACM/IEEE-CS joint conference on Digital libraries, 2002, ACM Press, New York, USA
http://doi.acm.org/10.1145/544220.544341

[GNOT92] **Goldberg, David, Nichols, David, Oki, Brian M. und Terry, Douglas: "Using Collaborative Filtering to Weave an Information Tapestry"**
In Communications of the ACM, December 1992, No. 12, S. 61-70, ACM Press, New York, USA
http://doi.acm.org/10.1145/138859.138867

[Gut1] **Gutenberg.de: "Datenleiste zum 15. Jahrhundert – Die Zeit Johannes Gutenbergs"**
Stadt Mainz, Amt für Öffentlichkeitsarbeit und Gutenberg-Museum, Institut für Mediengestaltung der Fachhochschule Mainz
http://www.gutenberg.de/zeitgutb.htm

[Gut2] **Gutenberg.de: "Die Erfindung Gutenbergs – Eine bahnbrechende Erfindung"**
Stadt Mainz, Amt für Öffentlichkeitsarbeit und Gutenberg-Museum, Institut für Mediengestaltung der Fachhochschule Mainz
http://www.gutenberg.de/erfindun.htm

[HB96] **Hallam-Baker, Phillip M. und Behlendorf, Brian: "Extended Log File Format"**
W3C Working Draft WD-logfile-960323, 1996
http://www.w3.org/TR/WD-logfile.html

[Heck95] **Heckerman, David: "A Tutorial on Learning With Bayesian Networks"**
Technical Report MSR-TR-95-07, Microsoft Research, Redmond, USA
http://www.research.microsoft.com/research/pubs/view.aspx?msr_tr_id=MSR-TR-95-06

[Heise99] **heise.de-Newsticker: "Amazon.com: Big Brother is watching you"**
Meldung vom 26.08.1999, Heise Zeitschriften Verlag, Hannover
http://www.heise.de/newsticker/data/jk-26.08.99-002/

[Heise99b] **heise.de-Newsticker: "Amazon verspricht nun doch mehr Datenschutz"**
Meldung vom 27.08.1999, Heise Zeitschriften Verlag, Hannover
http://www.heise.de/newsticker/data/jk-27.08.99-000/

[Heise00] **heise.de-Newsticker: "Datenschützer drehen Amazon den Rücken zu"**
Meldung vom 14.09.2000, Heise Zeitschriften Verlag, Hannover
http://www.heise.de/newsticker/data/mbb-14.09.00-000/

[Heise03] **heise.de-Newsticker: "Neue Algorithmen: Google-Turbo aus Stanford"**

Meldung vom 18.05.2003, Heise Zeitschriften Verlag, Hannover
http://www.heise.de/newsticker/data/atr-18.05.03-000/

[HF01] **Hauver, David B. und French, James C.: "Flycasting: Using Collaborative Filtering to Generate a Playlist for Online Radio"**
Proceedings of the First International Conference on WEB Delivering Music (WEDELMUSIC'01), IEEE Computer Society, 2001

[Hjelm01] **Hjelm, Johan: "Creating the Semantic Web with RDF"**
John Wiley & Sons, Inc., New York, USA, 2001, ISBN 0-471-40259-1

[HKBR99] **Herlocker, J., Konstan, J., Borchers, A., Riedl, J.: "An Algorithmic Framework for Performing Collaborative Filtering"**
Proceedings of the 1999 Conference on Research and Development in Information Retrieval, August 1999, ACM Press, New York, USA
http://www.cs.umn.edu/Research/GroupLens/papers/pdf/algs.pdf

[HKR00] **Herlocker, J., Konstan, J., und Riedl, J.: "Explaining Collaborative Filtering Recommendations."**
Proceedings of ACM 2000 Conference on Computer Supported Cooperative Work, S. 241-250, Dezember 2000
http://www.cs.umn.edu/Research/GroupLens/papers/pdf/explain-CSCW.pdf

[HS00] **Schmid, Helmut und Schulte im Walde, Sabine: "Robust German Noun Chunking With a Probabilistic Context-Free Grammar"**
Erschienen in COLING 2000 Volume 2: The 18th International Conference on Computational Linguistics, Universität des Saarlandes, Saarbrücken, Juli/August 2000
http://acl.ldc.upenn.edu/C/C00/C00-2105.pdf

[HZ97] **Haas, Rolf und Ziegelbauer, Holger: "Personenbezug"**
Artikel in der Zeitschrift iX, Ausgabe 9/97, S. 44, Heise Zeitschriften Verlag, Hannover
http://www.heise.de/ix/artikel/1997/09/044/

[HMSCM97] **Hensley, P., Metral, M., Shardanand, U., Converse, D. und Myers, M.: "Proposal for an Open Profiling Standard"**
Diskussionsnotiz des W3C, eingereicht am 2. Juni 1997
http://www.w3.org/TR/NOTE-OPS-FrameWork.html

[IETF95] **IETF: "Lightweight Directory Access Protocol", RFC 1777**
The Internet Engineering Taskforce (IETF), Reston, USA, 1995
http://www.ietf.org/rfc/rfc1777.txt

[IETF99] **IETF: "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616**
The Internet Engineering Taskforce (IETF), Reston, USA, 1999
ftp://ftp.isi.edu/in-notes/rfc2616.txt

[IO03] **Ideal Observer: "KnowHow zum Thema Website-Optimierung", 2001-2003, Frank Reese, Boitze Nr.8, 21368 Boitze**
http://www.idealobserver.de/htdocs/knowhow.html

[Jen00] **Jenkins, Jon: "Personalization for Web Applications"**
jjenkins@netperceptions.com, Net Perceptions, Inc., Vortrag auf Wrox 2000 Conference, Developer to Developer, Las Vegas
http://www.topxml.com/conference/wrox/2000_vegas/Powerpoints/jon_personal.pdf

[Kim01] **Kim, Amy Jo: "Community Building"**
Galileo Press GmbH, Bonn, 2001, 1. Auflage 2001, ISBN 3-934358-115-5

[KM97] **Kristol, D. und Montulli, L.: "HTTP State Management Mechanism", RFC 2109, Network Working Group, 1997**
http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc2109.html

[Kloss01] **Klossek, Martin: "Seminarvortrag Web Log Mining"**
Seminar WWW und Datenbanken, Sommersemester 2001, Lehrstuhl für Datenbanken und Informationssysteme, Uni Frankfurt
http://www.stormzone.de/uni/Hauptstudium/seminare/wwwdb/MK/list.php3

[Koe02] **Köhler, Susanne: "WebControlling WebMining WebSmiling – Mehr Umsatz durch zufriedene Nutzer"**
Mindlab GmbH, 73728 Esslingen, Vortrag bei "Data Mining Cup 2002"
http://www.data-mining-cup.de/2002

[LB02] **Lafon, Yves und Bos, Bert: "Describing and retrieving photos using RDF and HTTP"**
W3C Note 19 April 2002, Copyright © 2002 W3C® (MIT, INRIA, Keio)
http://www.w3.org/TR/photo-rdf/

[Lut95] **Luotonen, A.: "The Common Logfile Format", 1995**
http://www.w3.org/pub/WWW/Daemon/User/Config/Logging.html

[LWH03] **Liu, Jiming, Wong, Chi Kuen und Hui, Ka Keung: "An Adaptive User Interface Based on Personalized Learning"**
IEEE Intelligent Systems Journal, März/April 2003, S. 52-57

[MAB00] **Mulvenna, Maurice D., Anand, Sarabjot S. und Büchner, Alex G.: "Personalization on the Net using Web Mining"**
Communications of the ACM, Vol. 43(8), Seite 123-125, August 2000
http://www.infj.ulst.ac.uk/~cbgv24/PDF/CACM00.pdf

[Mena00] **Mena, Jesus: "Data Mining und E-Commerce"**
Symposium Publishing GmbH, Düsseldorf, 2000, ISBN 3-933814-12-X

[MMN01] **Melville, Prem, Mooney, Raymond J. und Nagarajan, Ramadass: "Content-Boosted Collaborative Filtering"**
Proceedings of the SIGIR-2001 Workshop on Recommender Systems, New Orleans, LA, September 2001
http://www.cs.utexas.edu/users/ml/publication/recommender.html

[MRK97] **Miller, B., Riedl, J., und Konstan, J.: "Experiences with GroupLens: Making Usenet useful again."**
Proceedings of the 1997 Usenix Winter Technical Conference, Januar 1997
http://www.cs.umn.edu/Research/GroupLens/papers/pdf/usenix97.pdf

[Mue99] **Mueller, Scott Hazen: "What is spam?"**
http://spam.abuse.net/overview/whatisspam.shtml, 1996-1999

[NBBBB97] **Neumann, Gunter, Backofen, Rolf, Baur, Judith, Becker, Markus und Braun, Christian: "An Information Extraction Core System for Real World German Text Processing"**
Fifth Conference on Applied Natural Language Processing, 1997, Washington DC, USA
Die Autoren sind Mitglieder des ParaDime-Projektes der Deutsche Forschungszentrum für Künstliche Intelligenz GmbH (DFKI) in Saarbrücken, siehe unter http://www.dfki.de
http://acl.ldc.upenn.edu/A/A97/A97-1031.pdf

[NCC99] **Netscape Communications Corp: "Persistent Client State HTTP Cookies – Preliminary Specification", 1999**
http://wp.netscape.com/newsref/std/cookie_spec.html

[NCL03] **Netcraft LTD: "April 2003 Web Server Survey", Bradford on Avon, UK**
http://news.netcraft.com/archives/2003/04/13/april_2003_web_server_survey.html

[Pine93] **Pine, B. Joseph II: "Mass Customization: The New Frontier in Business Competition"**
Harvard Business School Press, USA, 1993, ISBN 0-87584-372-7

[Ric00] **Dean, Richard: "What is Personalization?"**
http://builder.cnet.com/webbuilding/pages/Business/Personal/ss01.html, 2000

[RISBR94] **Resnick, P., Iacovou, N., Sushak, M., Bergstrom, P., and Riedl, J.: "GroupLens: An open architecture for collaborative filtering of netnews"**
Proceedings of the 1994 Computer Supported Collaborative Work Conference, S. 175-186, ACM Press, New York, USA
http://doi.acm.org/10.1145/192844.192905

[RK02] **Riedl, John und Konstan, Joseph: "Word of Mouse"**
Warner Books, Inc., New York, 2002, ISBN 0-446-53003-4

[Run00] **Runkler, Thomas A.: "Information Mining - Methoden, Algorithmen und Anwendungen intelligenter Datenanalyse"**
Friedrich Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig/Wiesbaden, 2000, ISBN 3-528-05741-6

[SAP02] **SAP: "mySAP Enterprise Portal"**
SAP Deutschland AG & Co. KG, 2002
http://www.sap.com/germany/media/50053495.pdf

[SF98] **Southwick, Scott und Falk, J. D.: "The Net Abuse FAQ – What is Spam?"**

The Net Abuse FAQ, 1994-1998
http://www.cybernothing.org/faqs/net-abuse-faq.html#2.1

[Shn02] **Shneiderman, Ben: "User Interface Design - Deutsche Ausgabe"**
mitp-Verlag, Bonn, 3. Auflage 2002, ISBN 3-8266-0753-8

[SUN01] **Sun Microsystems: "Enterprise JavaBeansTM Specification, Version 2.0"**
Sun Microsystems, Inc., Palo Alto, California, USA
http://java.sun.com/products/ejb/docs.html

[SKKR00] **Sarwar, B. M., Karypis, G., Konstan, J. A., and Riedl, J.: "Application of Dimensionality Reduction in Recommender System – A Case Study"**
ACM WebKDD 2000 Web Mining for E-Commerce Workshop
http://www.cs.umn.edu/Research/GroupLens/papers/pdf/webKDD00.pdf

[SKKR01] **Sarwar, B. M., Karypis, G., Konstan, J. A. und Riedl, J.: "Item-based collaborative filtering recommendation algorithms"**
Proceedings of the 10th International World Wide Web Conference (WWW10), Hong Kong, May 2001
http://www.cs.umn.edu/Research/GroupLens/papers/pdf/www10_sarwar.pdf

[SKKR02] **Sarwar, B. M., Karypis, G., Konstan, J. und Riedl, J.: "Recommender Systems for Large-Scale E-Commerce: Scalable Neighborhood Formation Using Clustering."**
Proceedings of the Fifth International Conference on Computer and Information Technology (ICCIT 2002), 2002
http://www.cs.umn.edu/Research/GroupLens/papers/pdf/sarwar_cluster.pdf

[StJ85] **StJohns, Mike: "Authentication Server"**
RFC 931, Network Working Group, 1985
http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc0931.html

[TO01] **Theisen, H. und Ott, M.: "Personalisierung oder Wie binde ich meinen Besucher?"**
contentmanager.de, 2001, F&P GmbH - FEiG & PARTNER, Leipzig
http://www.contentmanager.de/magazin/artikel_60_personalisierung_oder_wie_binde_ich_meinen.html

[W3C02] **World Wide Web Consortium (W3C): "Platform for Privacy Preferences Project (P3P) Specification"**
W3C Recommendation, herausgegeben am 16. April 2002, World Wide Web Consortium W3C (http://www.w3.org)
http://www.w3.org/TR/P3P/

[WCM01] **WCM Online: "Interview mit André Klahold, CEO InterRed (CMS-Anbieter)"**
Market Flash, WCM Online, Düsseldorf, Oktober 2001, www.newmediasales.com
http://www3.newmediasales.com/dl/1224/CMS_InterRed_Interview.pdf

[Wer00] **Werner, Peter: "Cookies Test Page"**
Website mit Informationen und Demos zum Thema Cookies, Universität Frankfurt, Fachbereich Biologie und Informatik, Lehrstuhl für Datenbanken und Informationssysteme (DBIS), 2000
http://www.dbis.informatik.uni-frankfurt.de/~werner/project_x/cookies/limitation.html

[WF01] **Witten, Ian H. und Frank, Eibe: "Data Mining – Praktische Werkzeuge und Techniken für das maschinelle Lernen"**
Carl Hanser Verlag, Münchenm, Wien, 2001, ISBN 3-446-21533-6

[Wong00] **Wong, Clinton: "http – kurz & gut" O'Reilly Taschenbuch**
O'Reilly Verlag GmbH und Co. KG, Köln, 2000, ISBN 3-89721-230-7

[Zie01] **Ziedek, Gisela: "MyContent im Trend: Wann Personalisierungen Sinn machen", 11/2001, erschienen bei Contentmanager.de**
contentmanager.de, 2001, F&P GmbH - FEiG & PARTNER, Leipzig
http://www.contentmanager.de/magazin/artikel_100_mycontent_im_trend_wann_personalisierungen_sinn.html

D Abbildungsverzeichnis

E Tabellenverzeichnis

F Stichwortindex

O

P

Q

R

S

T

U

V

W

Z