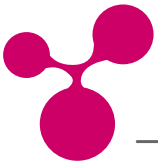
Technische Universität Dresden – Fakultät Informatik
Professur für Multimediatechnik, Privat-Dozentur für Angewandte Informatik

Prof. Dr.-Ing. Klaus Meißner
PD Dr.-Ing. habil. Martin Engelien
(Hrsg.)



GeNeMe '10

Gemeinschaften in Neuen Medien

an der
Fakultät Informatik der Technischen Universität Dresden

mit Unterstützung der

3m5. Media GmbH, Dresden
ANECON Software Design und Beratung GmbH, Dresden
Communardo Software GmbH, Dresden
GI-Regionalgruppe, Dresden
itsax.de | pludoni GmbH, Dresden
Kontext E GmbH, Dresden
Medienzentrum der TU Dresden
objectFab GmbH, Dresden
SALT Solutions GmbH, Dresden
SAP AG, Resarch Center Dresden
Saxonia Systems AG, Dresden
T-Systems Multimedia Solutions GmbH, Dresden

am 07. und 08. Oktober 2010 in Dresden

www.geneme.de
info@geneme.de

B.3 Einsatz von Empfehlungssystemen bei „Business on Demand“

Eva-Maria Schwartz
SALT Solutions GmbH

1 Motivation

IT und Software sind aus der heutigen Welt nicht mehr wegzudenken. In den vergangenen Jahrzehnten wurden sie zu einem wichtigen Bestandteil der Gesellschaft und in den hochentwickelten Ländern ist ein Leben ohne IT nicht mehr vorstellbar. Dabei hat sich die Entwicklung der Software in den Jahren stark verändert. Während in den 50er Jahren Entwickler, Anwender und Technikverantwortlicher noch die gleiche Person waren, werden heutzutage Softwareanwendungen von und vor allem für eine Vielzahl von Menschen entwickelt. Des Weiteren veränderte sich auch die Kostenaufteilung in der Software-Entwicklung. 1972 prägte Edsger W. Dijkstras in seiner Dankesrede zum Turing Award „The Humble Programmer“ den Begriff der Softwarekrise, da die Kosten für die Software die Hardwarekosten überschritten. Diesem Fakt wird unter anderem das Scheitern der ersten großen Softwareprojekte zugesprochen.

Diese stetig steigenden Kosten für die reine Software-Entwicklung und die oft nachfolgende „Kostenlawine“ durch Wartung, Anpassung und Weiterentwicklung, machen es gerade kleinen und mittelständischen Unternehmen (KMU) schwierig, geeignete Software schnell und bedarfsgerecht zu bekommen.

Dabei bestimmen gerade KMUs in Deutschland die wirtschaftliche Struktur. Sie leisten einen großen Beitrag zu wirtschaftlicher und gesellschaftlicher Stabilität und bilden ein starkes Gegengewicht zu den multinationalen Konzernen mit ihren globalen wirtschaftlichen Verflechtungen und Einflüssen.

Ein großer Teil des Mittelstandes wird sich zukünftig stärker als bisher auf internationale Märkte orientieren. Das bedeutet, dass KMUs sich immer mehr den sich ständig ändernden Anforderungen, die das Geschäftsumfeld an Unternehmen stellt, stellen müssen. Um diesen Anforderungen gerecht zu werden, wurden in den letzten Jahren neue Geschäftsmodelle zum Erwerb von Software entwickelt, unter anderem Software as a Service.

1.1 Software as a Service und Software on Demand 2.0

Software as a Service bzw. Software on Demand sind Geschäftsmodelle, welche das Prinzip verfolgen, Systeme als Dienstleistung im Internet bereit zustellen.

Der Nutzer erhält über öffentliche Netze Zugriff auf zentrale, serverbasierende und vorkonfigurierte Anwendungen. Der Softwareanbieter bzw. Dienstleister betreibt die Software zentral und organisiert die gesamte Administration wie Updates, Patches

oder auch Backups. Der Zugriff auf die Software erfolgt webbasiert, womit lokal zu installierende Programmkomponenten, Clients ausgenommen, entfallen. Für den Kunden bestehen dabei aber kaum Konfigurations- und Anpassungsmöglichkeiten z.B. für Benutzeroberflächen oder Funktionsumfang. An dieser konzeptioneller Lücke setzt das Forschungsprojekt Software on Demand 2.0 (SWoD 2.0) der TU Dresden, Privat-Dozentur für Angewandte Informatik, und dem mittelständischen Industriepartner Salt Solution GmbH an. Es wurde ein Portal erstellt, welches die Integration der „on Demand“ bezogenen Software-Module in bestehende Software-Infrastruktur und ihre Adaptivität an unternehmensspezifische Bedürfnisse stark vereinfacht bzw. erst ermöglicht. Um diesen Zusammenschluss zwischen bestehenden und neuen Modulen zu ermöglichen, wurde ein Analyseframework geschaffen, welches es dem Nutzer ermöglicht, sein Unternehmen in einer geeigneten Art und Weise abzubilden. Für diese Abbildung wurden sog. SWoD-Maps entwickelt, anhand derer der Nutzer sein Unternehmen, Strukturen und Prozesse abbilden kann (weitere Informationen in [Tei09]). Mit Hilfe dieses Frameworks können KMUs auf wechselnde Anforderungen schnell reagieren.

Um den Nutzern dieser Software eine bestmögliche Unterstützung bei der Auswahl ihrer bedarfsgerechten Komponenten zu geben, sollen ihnen anhand von Entscheidungen bereits bestehender Kunden Vorschläge für Objekte unterbreitet werden. Diese Objekte können je nach System zum Beispiel Konfigurationseigenschaften, Inhaltsmodule oder Layoutdarstellungen sein. Dieses ist notwendig, da bei der Inhaltsentwicklung zwar auf Erfahrungen aus anderen Projekten zugegriffen werden kann, aber ein allgemeingültiges Expertensystem, welches alle Branchen und Kooperationen darstellt, daraus nicht abgebildet werden kann. Durch die Integration der Komponentenveränderungen entwickelt sich das System kontinuierlich weiter und hat damit eine kollektiv entstandene Wissensbasis.

Für die Unterbreitung von Vorschlägen wird davon ausgegangen, dass ähnliche Nutzer auch ähnliche Objekte benötigen. Aus diesem Grund sollen die Nutzer bzw. Unternehmen miteinander verglichen werden und ihnen Empfehlungen anhand der Entscheidungen ähnlicher Nutzer gegeben werden.

1.2 Einsatzmöglichkeiten von Empfehlungssystemen

Nach Klahold [Kla09] kann ein Empfehlungssystem folgendermaßen definiert werden: „Ein Empfehlungssystem (oft auch „Recommender System“ genannt) ist ein System, das einen Benutzer in einem gegebenen Kontext aus einer gegebenen Entitätsmenge aktiv eine Teilmenge „nützlicher“ Elemente empfiehlt. Der Kontext konstituiert sich dabei aus dem Benutzerprofil P, der Entitätsmenge M und der Situation S. (…) Die empfohlenen Elemente T (Teilmenge von M) sollten den Nutzen

des Benutzers B im gegebenen Kontext K maximieren. Formal besteht die Aufgabe eines Empfehlungssystems daher in folgender Optimierung: max(Nutzerwert(B.K.T)) mit K=(P,M,S)“
Die Verfahren zur Ermittlung von Empfehlungen werden in drei wesentliche Vorgehen unterteilt:

1. **Content Based Filtering** basiert auf den Eigenschaften der Empfehlungselemente (Entitäten der Menge M). Das Verhalten der Nutzer wird nicht mit einbezogen.
2. **Collaborative Filtering** nutzt das Verhalten der Benutzer und damit deren Ähnlichkeiten im Nutzerprofil.
3. **Hybrid-Verfahren** verbinden das Collaborative und Content Based Filtering und versucht die Vorteile beider Ansätze zu kombinieren.

In diesem Artikel wird hauptsächlich auf das Verfahren des Collaborative Filterings eingegangen, da im Bereich von SaaS die Eigenschaften der Nutzer bzw. Unternehmen ausschlaggebend sind.
Im folgenden Kapitel wird ein kurzer Überblick über das Prinzip des Collaborative Filterings gegeben, um im darauf folgenden den Einsatz in Business on Demand Systemen zu diskutieren. In Kapitel 3 wird ein Algorithmus vorgestellt, welcher auf die Anforderungen und speziellen Bedürfnisse von Komponentenempfehlungen eingeht. Der sogenannte Komparative Ähnlichkeitsalgorithmus wird danach an einem Beispiel erläutert.

2 Collaborative Filtering

Collaborative Filtering verwendet die Ähnlichkeit von Benutzerprofilen, welche das Benutzerverhalten in Form von Empfehlungselementen repräsentieren. Als Grundlage wird eine Benutzer-Empfehlungsmatrix benutzt. Folgendes kurzes Beispiel soll das Prinzip des Collaborative Filterings erläutern:
In Tabelle 2 befinden sich Bewertungen der einzelnen Nutzer für Filme, welche sie gesehen haben. Leere Spalten bedeuten, dass der Nutzer den Film noch nicht gesehen hat.

Tabelle 1: Benutzer-Bewertungsmatrix

	Film 1	Film 2	Film 3	Film 4
Nutzer 1	1	4	4	5
Nutzer 2		3	3	
Nutzer 3	5	5	2	
Nutzer 4	5		5	5

In diesem Beispiel soll dem Nutzer 2 ein Film empfohlen werden. Grundsätzlich könnten Filme gefunden werden, welche von Nutzern ähnlich bewertet werden (Element-Basierter-Nähester-Nachbar-Algorithmus), oder ein Nutzer, welcher Filme ähnlich bewertet und abhängig davon einen Film empfehlen (Nutzer-Basierter-Nähester-Nachbar-Algorithmus). Benutzt man den Nutzer-Basierten Algorithmus erkennt man, dass Nutzer 1 und Nutzer 2 gemeinsame Filme ähnlich bewerten. Damit kann davon ausgegangen werden, dass diese Nutzer einen gemeinsamen Filmgeschmack haben und Filme, welche Nutzer 1 als sehr gut bewertet hat, Nutzer 2 empfohlen werden. In diesem Beispiel kann Film 4 für Nutzer 2 empfohlen werden.

2.1 Algorithmen und ihr Einsatz

In diesem Abschnitt werden die verschiedenen Konzepte der Empfehlungsberechnung vorgestellt. Grundsätzlich kann in folgende drei Kategorien unterteilt werden:

Benutzerbezogener Algorithmus

Benutzerbezogene Algorithmen beziehen sich auf Bewertungen oder Verhalten von Nutzern. Dabei wird anhand von n Benutzern und m Empfehlungselementen eine Matrix $R=(r_{ij})$ mit $i=1..n$ und $j=1..m$ erzeugt. Ziel ist es, den ähnlichsten Nutzer zu einem Benutzer U zu finden, um daraus das erwartete Interesse von U zu berechnen. Dabei wird die Ähnlichkeit anhand von Distanz- oder Ähnlichkeitsmaßen ermittelt.

Elementbasierter Algorithmus

Beim elementbasierten Algorithmus wird wie beim benutzerbezogenen Algorithmus die Matrix erstellt, um auf ihrer Grundlage Empfehlungen zu generieren. Dazu werden Ähnlichkeiten der Bewertungen von anderen Benutzern, die jeweils beide Empfehlungselemente bewertet haben, ein Vektor gebildet, der dann für die Ähnlichkeitsbestimmung verwendet wird. Es werden basierend auf gut bewerteten Empfehlungselementen des Benutzers alle Paare von Empfehlungselementen, in denen eines dieser Empfehlungselemente vorkommt, selektiert.

Modell- und speicherbasiertes Verfahren

Beim speicherbasierten Verfahren wird die komplette Basis der Benutzerprofile mit deren Verhalten zur Berechnung der Empfehlungen verwendet. Dabei werden statistische Verfahren zur Bestimmung der Nachbarn verwendet.
Das modellbasierte Verfahren basiert auf eine „Benutzer-Cluster"-Matrix. Bekannteste Beispiele dieses Verfahrens sind Bayes´sche Netze, Clusterbildung und neuronale Netze. Ziel ist es, Cluster von Nutzern zu bilden, welche ähnliche Präferenzen haben.

2.2 Anwendbarkeit des Collaborative Filtering

Ausgehend von der These, dass Nutzer mit ähnlichen Vorliegen, Aufgaben und Umgebungsvariablen auch ähnliche Objekte benutzen, wurde sich für das Prinzip des Collaborative Filterings entschieden. Eine Herausforderung liegt dabei in der Beschreibung eines Nutzers und dessen Unternehmen. Diese können durch eine Vielzahl von Merkmalen gekennzeichnet werden, welche je nach Objekt eine unterschiedliche Wichtigkeit bei der Entscheidung haben. Eine einzelne Abweichung der Beschreibung des Nutzers kann entscheidende Auswirkungen auf die zu empfehlende Komponente haben. Aus diesem Grund müssen die bereits bestehenden Algorithmen erweitert werden, so dass für die jeweilige Empfehlungskomponente die ausschlaggebenden Nutzermerkmale erkannt und priorisiert werden. Bei der Bewertung eines Objektes müssen also entsprechende Wichtungen für die jeweiligen Nutzermerkmale integriert werden.
Im kommenden Abschnitt wird ein Algorithmus vorgestellt, welcher Nutzer anhand ihrer Merkmale vergleicht, Priorisierung der Merkmale festlegt und daraus folgend Empfehlungen für Objekte ausgibt.

3 Entwicklung eines angepassten Algorithmus

3.1 Algorithmenbasis

Zur Lösung des Problems der polyoptimalen Ablaufsteuerung in der Fertigungssteuerung wurde in [Bal76] ein Prioritätsalgorithmus vorgestellt, welcher die priorisierte Bewertung einzelner Merkmale mit heuristischen Prinzipien einbezieht. Auf Grund des beschriebenen Verfahrens des komparativen Bewertens ist die Idee entstanden, die Ansätze des Verfahrens bei der Empfehlung für Objekte für Nutzer anhand von Nutzermerkmalen zu integrieren. Das Ziel dabei ist, Nutzern Objekte zu empfehlen, welche andere Nutzer mit ähnlichen Merkmalen bereits einsetzen. Durch diese Empfehlungen kann die kollektive Intelligenz der Systemnutzer angewendet werden. Grundsätzlich sollen die Merkmale der Nutzer miteinander verglichen werden, wobei eine geeignete Priorisierung erfolgt, und die Objekte, welche die Nutzer mit der höchsten Ähnlichkeit verwenden, angezeigt werden. Dabei wird davon ausgegangen, dass die einzelnen Merkmale des Nutzers nicht immer gleich priorisiert werden, sondern in Abhängigkeit vom Objekt gewichtet werden müssen. Diese Wichtungsfaktoren müssen aus der Masse der bestehenden Daten (Nutzer und deren angewendete Objekte) analysiert und berechnet werden. Die periodisch zu berechnenden Wichtungen können als kollaborative Priorisierungsfaktoren angesehen werden.

3.2 Komparativer Ähnlichkeitsalgorithmus (CSA)

Voraussetzungen

Der Nutzer N wird durch eine endliche Anzahl von Merkmalen M1 bis MN beschrieben. Jedes dieser Merkmale kann unterschiedlich erfassbare Ausprägungen besitzen. Damit kann der Nutzer durch den Merkmalsvektor (m1, ..., mN) beschrieben werden.

Ein Objekt O hat die Eigenschaften E1 bis ES. Jede dieser Eigenschaften kann unterschiedliche Ausprägungen besitzen. Die Eigenschaften des Objekts sind voneinander unabhängig und die einzelnen Ausprägungen der Eigenschaften können unabhängig ausgewählt werden. Jede beliebige Auswahl (e1, ..., eS) der Eigenschaften E1 bis ES wird als Objektkonfiguration bezeichnet.

Jeder Nutzer des Objekts O bevorzugt eine für ihn individuelle Konfiguration. Die für den Nutzer geeignete Einstellung hängt von seinen individuellen Merkmalen ab. Vorausgesetzt wird, dass Nutzer mit einem identischen Merkmalsvektor die gleiche Objektkonfiguration bevorzugen.

Beschreibung des Algorithmus

Der Algorithmus gliedert sich in zwei Teile:

1. Bestimmung eines „ähnlichen“ Nutzers und der Entscheidung, worauf aufbauend eine Empfehlung bzw. Empfehlungen getroffen werden.
2. Algorithmus zur sukzessiven Verbesserung der Wichtungsfaktoren während der gesamten Laufzeit des Systems.

Berechnung der Ähnlichkeitsmatrix

Der neue Nutzer wird mit den bereits bestehenden Nutzern des Systems verglichen. Zum Vergleich kann eine Abstands- (zum Beispiel Euklidischer Abstand) oder Vergleichsfunktion verwendet werden. Die Auswahl der geeigneten Funktionen ist abhängig von den verwendeten Daten und deren Definition.

Die Ergebnisse der einzelnen Vergleiche werden in einer Ähnlichkeitsmatrix gespeichert. In den Spalten stehen dabei die Ähnlichkeit zwischen dem neuen Nutzer und dem bestehenden Nutzer X. In den Zeilen befindet sich die Ähnlichkeit bei den einzelnen Eigenschaften der Nutzer. Diese Ähnlichkeitsmatrix wird dann für jedes zu empfehlendes Objekt mit der entsprechenden Priorisierungsmatrix multipliziert und das Objekt, welches der bestbewerteste bestehende Nutzer benutzt, wird dem neuen Nutzer empfohlen.

Periodische Berechnung der Wichtungsfaktoren (Priorisierungsmatrix)
Es wird davon ausgegangen, dass die Auswahl der Objekte von unterschiedlichen Merkmalen des Nutzers abhängig ist. Aus diesem Grund muss für jedes Objekt ein Wichtungsfaktor berechnet werden. Dazu werden alle Nutzer, welche das gleiche Objekt verwenden, miteinander verglichen. Wenn es ein entscheidendes Nutzermerkmal für ein Objekt gibt, muss die Abweichung beim Vergleich zwischen den Nutzermerkmalen sehr gering sein. Um diese Abweichung zu berechnen, ist die Varianz geeignet. Nutzer eines gleichen Objektes, welche bei einem Merkmal die gleiche Ausprägung haben, haben in diesem Fall die Varianz von Null. Zur Berechnung des Wichtungsfaktors des jeweiligen Merkmals wird folgende Berechnung verwendet:

$$w_{ij} = \frac{1}{1 + Varianz}$$

Damit liegen alle Wichtungsfaktoren zwischen 0 und 1. Merkmale, die eine geringere Streuung bezüglich der Objekteigenschaft haben, sind größer als die mit einer großen Streuung. Die Neuberechnung der Wichtungsfaktoren sollte periodisch wiederholt werden, damit das System Veränderung erkennt.
Zu Beginn einer Laufzeit können die Wichtungsfaktoren noch nicht berechnet werden. In diesem Fall sollte eine a-priori Schätzung durchgeführt oder alle Wichtungsfaktoren auf 1 gesetzt werden.

4 Experimentelle Evaluation anhand eines Beispiels

In diesem Kapitel soll ein kleines Beispiel vorgestellt werden, welche die Empfehlung von Layouts erläutert. Dabei werden zufällige Daten erzeugt, welche durch eine Regel manipuliert werden. Das Beispiel zeigt auf, dass die Regel vom Algorithmus erkannt und das richtige Element empfohlen wird.
Die Besonderheit des Komparativen Ähnlichkeitsalgorithmus ist die Empfehlung von einzelnen Eigenschaften. Das bedeutet, dass für jedes Layout die besten Komponenten gefunden werden, um so den Ansprüchen von Arbeit und Umgebung gerecht zu werden.

4.1 Ausgangsinformationen

Für das Beispiel gehen wir davon aus, dass ein Layout durch den Aufbau einer Seite (Abb.1), die verschiedenen Navigationsarten (Abb.2) und die unterschiedlichen Anzeigearten eines Diagramms (Abb.3) beschrieben werden kann. Den Objekten werden die nominalen Daten 1 bis 3 der Reihenfolge nach zugeordnet.

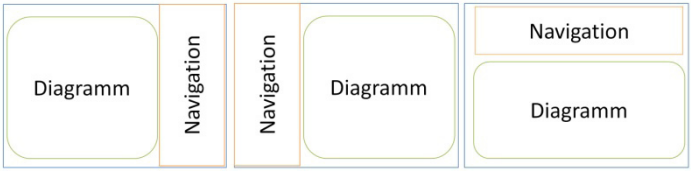


Abbildung 1: Aufbau der Seite

Abbildung 2: Navigationsarten

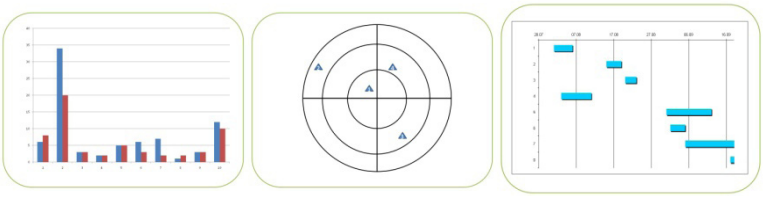


Abbildung 3: Diagrammtypen (Ist-Soll-Vergleich, Radar und GANTT)

Die Zuordnung der nominalen Daten geschieht durch die direkte Abbildung auf die Variantenauswahl. Zum Beispiel hat das Radar-Diagramm die Zuordnung 2. Dabei ist zu erwähnen, dass die Zuordnungen nicht ordinal sind und damit eine Ordnung nicht gegeben ist. Die Nutzer des Systems werden durch die Merkmale Alter, Aufgabe und Zugangsgerät beschrieben. In Tabelle 2 ist eine Beschreibung der Merkmalszuordnung aufgezeichnet.

Tabelle 2: Nutzerbeschreibung mit nominaler Zuordnung

Alter	< 40 Jahre	1
	40 – 70 Jahre	2
	> 70 Jahre	3
Aufgabe	Arbeiter	1
	Geschäftsführer	2
	Projektleiter	3
Zugangsgerät	Desktop-Rechner	1
	Laptop	2
	Smartphone	3

Für das Beispiel wurde per Zufallsprinzip eine Nutzerbasis geschaffen (Tab.3).

Tabelle 3: Bestehende Nutzerdaten

	Nutzer 1	Nutzer 2	Nutzer 3	Nutzer 4	Nutzer 5
Alter	1	3	2	1	2
Aufgabe	2	3	2	1	3
Zugangsgerät	3	3	1	2	2

Im nächsten Schritt wird eine Ergebnismatrix erstellt, die anzeigt, welche Nutzer welche Komponenten des Layouts verwenden. Dazu wurden wieder rum Zufallszahlen erzeugt, welche durch folgende Regeln manipuliert wurden:
Ein Arbeiter verwendet immer das Diagramm „Pilot“. Ein Projektleiter verwendet immer das Diagramm „GANTT“. Ein Geschäftsführer verwendet immer das Diagramm „Ist-Soll“. Damit entsteht folgende Ergebnismatrix:

Tabelle 4: Ergebnisdaten

	Nutzer 1	Nutzer 2	Nutzer 3	Nutzer 4	Nutzer 5
Aufbau	2	3	1	1	2
Navigation	1	3	2	1	1
Diagramm	1	3	1	2	3

Zur Überprüfung des Algorithmus wird nun ein zufälliger neuer Nutzer erstellt, dem Empfehlungen gegeben werden sollen.

4.2 Layout-Empfehlungen anhand des CSA

Der neue Nutzer des Systems ist älter als 70 Jahre. Seine Tätigkeit im Unternehmen ist die Projektleitung, wobei er ein Smartphone benutzt.
Bei herkömmlichen Algorithmen werden die Nutzer miteinander verglichen. In unserem Beispiel zeigt es sich, dass der neue Nutzer jeweils mit jedem vorhandenen Nutzer in einem Merkmal gleich ist (siehe Abbildung 4 links). Bei diesem Algorithmus ist eine Empfehlung für ein geeignetes Layout nicht möglich.

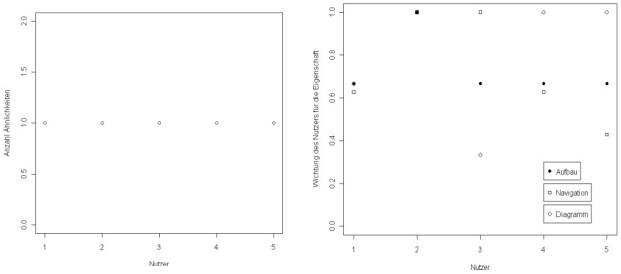


Abbildung 4: Ergebnis der Ähnlichkeitsanalyse (links Summe der Ähnlichkeiten, rechts Wichtung der Ähnlichkeiten)

Mit Hilfe des komparativen Ähnlichkeitsalgorithmus kann jedoch eine Empfehlung erstellt werden (siehe Abbildung 5 rechts). Dabei ist zu beachten, dass sowohl die Aufbau- als auch Navigationsergebnisse aus Zufallszahlen erzeugt wurden. Das in Abbildung 6 zu sehende Layout ist ein Beispiel für die Erläuterung des Algorithmus.

Abbildung 5: Empfehlung nach dem komparativen Ähnlichkeitsalgorithmus

Zusammenfassung und Ausblick

Diese Arbeit beschreibt einen Algorithmus zur Empfehlung von Objekten anhand Ähnlichkeiten zwischen Nutzern, wobei die verschiedenen Merkmale der Nutzer verglichen werden. Die Priorisierung der Merkmale erfolgt mit kollaborativen Wichtungsfaktoren. Diese werden auf Grundlage der bereits bestehenden Daten von Nutzern berechnet. Das Ziel des Algorithmus und des darauf aufbauenden Recommender-Systems ist es, die Probleme, der mittels „non-user-centered"-Design entstandenen Produkte zu lösen. Dabei bilden die Nutzerdaten, welche auf Grund der neuen Nutzungsmodelle vorhanden sind, die neue Wissensgrundlage. In diesen Daten sollen Gesetzmäßigkeiten zur Anwendung von Objekten im Zusammenhang mit speziellen Nutzermerkmalen erkannt werden.

In den folgenden Arbeiten wird der beschriebene Algorithmus getestet. Dabei ist es im Speziellen notwendig festzustellen, ab wann die berechneten Gewichtsfaktoren zuverlässig in ihrer Aussage sind. Dies impliziert eine Anzahl von vorhandenen Nutzerdaten für das jeweilige Objekt. Die genaue Anzahl kann nur durch direkte Tests an speziellen Objekten bestimmt werden. Dabei wird die Auswahl der Merkmale von Nutzern und Objekten eine wichtige Rolle spielen.

Literatur

[Bal76] Baldeweg H.; Stahn, F.; Jungclaussen H.: Grundlagen der Kybernetik II/ Zentralinstitut für Kernforschung Rossendorf bei Dresden. 1976

[Boe87] Boehm, B.: Improving Software Productivity. IEEE 20, 1987

[Dij72] Dijkstra, Edsger W.: The Humble Programmer. In: ACM Turing Lecture, 1972

[Kla09] Klahold, Andre: Empfehlungssysteme - Recommender Systems – Grundlagen, Konzepte und Lösungen; 2009

[Sar01] Sarwar, B.; Karypis, G.; Konstant, J.; Riedl, J.: item-based collaborative filtering recommendation algorithms, Proceedings of the 10th international conference on World Wide Web, 2001, Springer

[Tei09] Teichmann, Gunter; Schulz, Alexandra: Kollaborative Problemanalyse in Business Communities mit SWoDMaps, Workshop GeNeMe 2009, TUDpress 2009