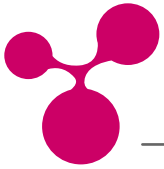
Technische Universität Dresden – Fakultät Informatik
Professur für Multimediatechnik, Privat-Dozentur für Angewandte Informatik

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



GeNeMe '09

Gemeinschaften in Neuen Medien

an der
Fakultät Informatik der Technischen Universität Dresden

mit Unterstützung der

3m5. Media GmbH, Dresden
GI-Regionalgruppe, Dresden
Communardo Software GmbH, Dresden
Kontext E GmbH, Dresden
Medienzentrum der TU Dresden
nubix Software-Design GmbH, Dresden
objectFab GmbH, Dresden
SALT Solutions GmbH, Dresden
Saxonia Systems AG, Dresden
T-Systems Multimedia Solutions GmbH

am 01. und 02. Oktober 2009 in Dresden

http://www-mmt.inf.tu-dresden.de/geneme/
geneme@mail-mmt.inf.tu-dresden.de

B.10 Aggregation, Filterung und Visualisierung von Nachrichten aus heterogenen Quellen - Ein System für den unternehmensinternen Einsatz

Torsten Lunze[1], Marius Feldmann[2], Thomas Eixner[2], Serkan Canbolat[2], Alexander Schill[2]
[1] Communardo Software GmbH
[2] TU Dresden, Institut für Systemarchitektur, Lehrstuhl Rechnernetze

1 Einleitung

Das *Web 2.0* lebt von der vereinfachten und schnellen Informationspublikation und -distribution. Die meisten in diesem Zusammenhang eingesetzten Technologien wie Web Feeds, Instant Messaging oder Weblogs dienen genau diesem Zweck. Unter dem Schlagwort *Enterprise 2.0* werden verschiedene Entwicklungen subsumiert, die dieses Grundparadigma auf die Unternehmenskommunikation - einschließlich der unternehmensinternen Kommunikation - ausdehnen.

Das dabei entstehende Problem ist evident: Die vorhandenen Informationsflut aus unterschiedlichsten Web-basierten Informationsquellen führt zu einem hohen Aufwand für den Informationsempfänger, um aus seiner Perspektive relevante Informationen von irrelevanten zu unterscheiden. Um diesen Prozess möglichst effizient zu gestalten, ist eine Systemunterstützung erforderlich. In diesem Beitrag soll ein solches Gesamtsystem in seinen zentralen Charakteristika vorgestellt werden. Einerseits zielt das Systembackend darauf ab, externe Informationsquellen wie RSS- bzw. Atom-Feeds und unternehmensinterne Kommunikationsquellen wie Enterprise-Microblog-Lösungen (wie z.B. [Com09]) zu aggregieren und eintreffende Nachrichten in ihrer Relevanz für einzelne Endanwender einzustufen. Andererseits ermöglicht das erstellte Frontend aufgrund der Verwendung verschiedener Visualisierungsformen eine effiziente Informationsaufnahme durch den Endnutzer.

Im vorliegenden Beitrag werden zunächst die Anforderungen an das entwickelte System vorgestellt, um auf deren Grundlage eine Abgrenzung von bestehenden Systemlösungen zu beschreiben (Kapitel 2). Nach einem allgemeinen Systemüberblick (Kapitel 3.1) wird auf zentrale Aspekte des Backends (Kapitel 3.2) und des Frontends (Kapitel 3.3) eingegangen. Bevor der Beitrag mit einer Zusammenfassung und einem Ausblick schließt (Kapitel 5), werden wichtige Entwurfsentscheidungen für das Backend bzw. Frontend bewertet (Kapitel 4).

2 Anforderungen und Stand der Technik

Nach einer Analyse verschiedener unternehmensinterner Szenarien (beschrieben in [Can09] bzw. [Eix09]) wurden die zentralen Anforderungen an ein System für die effiziente Verwaltung verschiedener Web-basierter Informationsquellen identifiziert. Als Basisfunktionalitäten muss ein solches System die Aggregation unterschiedlicher semistrukturierter Informationsquellen wie RSS- bzw. Atom-Feeds oder Microbloggingformate zu einem einheitlichen Format unterstützen (R1), um diese bei der Visualisierung auf homogene Art und Weise aufbereiten zu können. Die Infrastruktur muss darüber hinaus eine einfache Erweiterbarkeit hinsichtlich weiterer Nachrichtenformate offerieren (R2), um die Vielzahl an verschiedenen Informationsquellen in einem Unternehmen abdecken zu können. Aufgrund der unterschiedlichen Interessen und Präferenzen von Nutzern muss eine personalisierte Filterung der Nachrichten erfolgen (R3). Diese soll sowohl auf inhaltsbasierten als auch auf nutzerübergreifenden, kollaborativen Filtermechanismen basieren. Nutzer sollen die Möglichkeit haben, besonders relevante bzw. interessante Nachrichten explizit anderen Nutzern zu empfehlen (R4). Eintreffende Nachrichten sollen auf Grundlage automatisiert erstellter Metainformationen (Zuordnung von Schlüsselwörtern zu Inhalten) verschiedenen Nachrichtenkategorien der Nutzer zugeordnet werden können (R5). Das Frontend soll eine effiziente Informationsvermittlung ermöglichen (R6) und die Informationen unabhängig von der Quelle auf homogene Weise visualisieren. Während der Szenarienanalyse wurde festgestellt, dass zwei unterschiedliche Nutzungsformen des Informationssystems existieren. Einerseits erwartet ein Nutzer die Möglichkeit eines schnellen Überblicks über den aktuellen Informationszustand, andererseits eine nach Relevanz geordnete Darstellung der vorliegenden Nachrichten. Die Anforderung R6 umfasst damit die Notwendigkeit für unterschiedliche Sichten auf das Informationssystem.

Im Zusammenhang mit den gestellten Anforderungen wurde eine Analyse bestehender Systeme durchgeführt, die Consumer-Aggregatoren (bspw. AideRSS und Yahoo Pipes), Microblogging-Aggregatoren (bspw. Tweetdeck und Twhirl) als auch Enterprise-Aggregatoren (bspw. Attensa Feed Server [Att09] und Newsgator Social Sites 2.0 [NSS09]) umfasste. Obwohl die betrachteten Enterprise-Aggregatoren die Anforderungen zu einem Teil erfüllen und über fortgeschrittene kollaborative Verfahren der Filterung verfügen, wurde keine Lösung gefunden, die eine breite Unterstützung technologisch unterschiedlicher Nachrichtenquellen - insbesondere von Enterprise-Microbloggingslösungen - ermöglicht. Darüber hinaus bieten die meisten Systeme keine Möglichkeit, die Nachrichten aus unterschiedlichen Quellen auf homogene Weise im Frontend zu visualisieren. Gerade diese Anforderung spielt für das erstellte System eine zentrale Rolle, da dies eine effizientere Informationsaufnahme durch den Nutzer verspricht.

3 Systembeschreibung

3.1 Systemübersicht

Das System wurde als Webapplikation realisiert und ist, wie in Abbildung 1 dargestellt, als typische Drei-Schichten-Architektur aufgebaut.

Die Persistenzschicht kapselt den Zugriff auf die Datenbank, in welcher alle Anwendungs- und Nutzerdaten gespeichert werden. Die Serviceschicht baut auf der Datenbankschicht auf und beinhaltet die Anwendungslogik, die in verschiedene Services untergliedert ist. Durch diesen Dienst-basierten Ansatz wird eine Erweiterung bzw. Anpassung der Systemfunktionalitäten deutlich vereinfacht. Mittels einer wohldefinierten API bietet diese Schicht einen Zugriffspunkt für die Frontendschicht an. Die Frontendschicht generiert ein Ajax-unterstützendes HTML-Frontend, welches in Widgets aufgeteilt ist.

Im Zusammenhang mit der in den Anforderungen erwähnte personalisierten Filterung resultieren die zentralen Fragestellungen, wie die Interessen des Nutzers formalisiert werden können und wie auf dieser Grundlage die Filterung realisierbar ist. Der hier beschriebene Ansatz verwendet dafür eine einfache Lösung: Tags. Die Interessen eines Nutzers werden mittels gewichteter Tags modelliert und zudem werden jeder Nachricht automatisiert Tags aus dem gleichen Namensraum zugeordnet. Die Gewichte der Tags drücken dabei seine Relevanz in Bezug auf die Nutzerinteressen aus. Die Relevanzwerte werden durch drei verschiedene Nutzerinteraktionen beeinflusst: Manuelle Anpassung, Nutzung von persönlichen Informationsquellen sowie Lernen durch Bewertung.

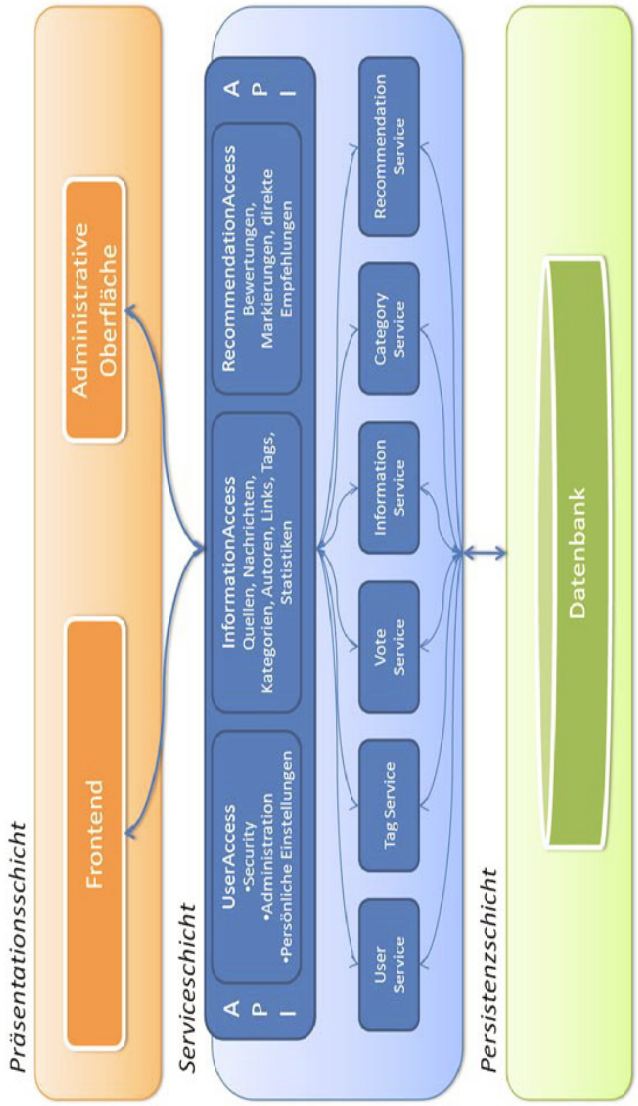


Abbildung 1: 3-Schichten-Architektur des Gesamtsystems

Lernen von Interesse

Ein initaler Import von persönlichen Informationsquellen (wie bspw. von Delicious [Del09]) oder manuelles Festlegen der Relevanzwerte der Tags ist nicht ausreichend für eine effektive Filterung der Nachrichten. So wird ein initaler Import nur in seltenen Fällen genau die Interessen des Nutzers widerspiegeln. Genauso wenig kann dem Nutzer zugemutet werden, seine Relevanzbewertungen kontinuierlich manuell zu editieren oder anzupassen - mitunter kann er seine eigenen Interessen nicht eindeutig formulieren.

Eine einfache Art und Weise, Feedback für einen impliziten, fortlaufenden Lernprozess der Nutzerinteressen zu gewinnen, ist es, ihn beurteilen zu lassen, ob eine Nachricht für ihn interessant oder uninteressant ist. Eine solche Bewertung kann mehrstufig sein (analog zu z. B. Amazon Bewertungen), wobei eine mehrstufige Bewertung einen tiefergehenden Entscheidungsprozess beim Nutzer erfordert: „Finde ich diese Nachricht sehr interessant oder nur interessant?“ Um diesen Entscheidungsprozess so einfach wie möglich zu halten, kann der Nutzer ausschließlich eine der folgenden Aktionen auf einer Nachricht ausführen, die implizit zur Bewertung führen: die Nachricht als interessant, als gelesen oder als uninteressant markieren, die Nachricht schließen oder die Nachricht in eine „To Read“-Liste verschieben. Eine als interessant sowie uninteressant bewertete Nachricht führt zu einer Aktualisierung des Nutzermodells. Eine einmal bewertete Nachricht erscheint nicht mehr in dem Informationsstrom für den Nutzer. Nachrichten können auf eine „To Read“-Liste verschoben werden: Der Nutzer kann somit eine Nachricht aus dem Informationsstrom entfernen und sie für eine spätere Bewertung vormerken.

3.2 Backend

Im Folgenden wird der Informationsfluss (Abbildung 2) beschrieben und es wird darauf eingegangen, wie die Nachrichten in der Datenbank verwaltet werden, wie darüber eine nutzerspezifische Filterung stattfindet und welche Rolle Tags bei diesem Prozess einnehmen.

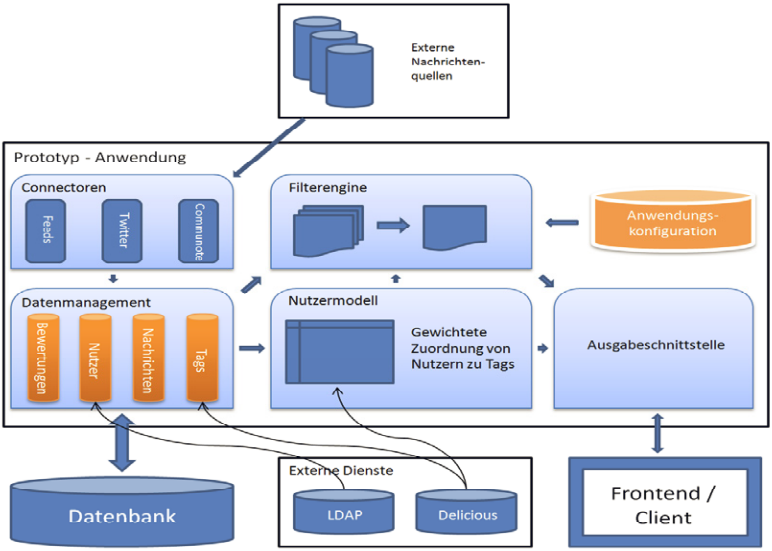


Abbildung 2: Informationsfluss durch das System

Nachrichtentransformation

Mit Hilfe von Connectoren können externe Nachrichtenquellen einfach angebunden werden. Ein Connector greift dabei periodisch auf eine Quelle zu, liest die Daten aus, transformiert diese in ein für das System einheitliches Format und legt diese in der Datenbank ab. Dieses homogene Format erlaubt es, die Filterung der Nachrichten unabhängig von der Art der Informationsquellen durchzuführen. In der Implementierung existieren Connectoren für RSS-Feeds sowie für die Microblogging Systeme Communote [Com09] und Twitter [Twi09]. Jeder dieser Connectoren kann dabei auf die Eigenheiten der Nachrichtenquellen eingehen.

Da Tags der zentrale Bestandteil für eine Zuordnung von Nutzern zu Nachrichten sind, bedingt dies, dass den Nachrichten Tags zugeordnet werden können. Es kann aber nicht davon ausgegangen werden, dass jede eingehende Nachricht mit Tags versehen ist. Daher wird versucht für diese Nachrichten Tags zu ermitteln, indem der Inhalt der Nachricht mit in der Datenbank vorhandenen Tags abgeglichen wird. Taucht ein Tag innerhalb einer Nachricht als Begriff auf, wird sie mit dem Tag assoziiert und mit ihm in der Datenbank abgelegt. Die Zuordnung der Tags zu den Nachrichten ist für beliebige Textinhalte möglich und kann für heterogene Quellen durchgeführt werden. Die Architektur erlaubt es dabei, dass unbekannte Typen von Quellen durch Entwickeln eines neuen Connectors im System einfach eingebunden werden können.

Nutzermodell

Um das Interesse von Nutzern an Themen zu repräsentieren, werden Tags verwendet, die zusammen mit einem Relevanzwert für jeden Tag des Nutzers das Nutzermodell bilden. Durch das Nutzermodell werden die Nutzerinteressen in eine entsprechende Datenstruktur abgebildet. Dabei werden die den Nutzern zugeordneten Tags mit einer Relevanz $t_i = [0,1]$ versehen, wobei 0 die kleinstmögliche und 1 die höchstmögliche Relevanz ausdrückt. Ein Beispiel eines derartigen Nutzermodells gibt [Eix09] in Kapitel 3.2.3.

Unabhängig davon wie das System die Relevanz berechnet, kann der Nutzer selbst Tags als interessant oder unwichtig markieren und damit die Filterung direkt beeinflussen.

Filterengine

Die Anwendung unterstützt neben der Filterung von Nachrichten ebenfalls die manuelle Gruppierung von Nachrichtenströmen durch frei definierbare Kategorien. Somit wird dem Nutzer die Möglichkeit gegeben, selbst Einfluss darauf zu nehmen, wie Nachrichten auf oberster Ebene eingeordnet werden. Für jede Kategorie kann der Nutzer mehrere Nachrichtenquellen angeben, um deren Inhalte der Kategorie zuzuweisen. Darüber hinaus können diesen Kategorien Tags zugeordnet werden. Dies hat zur Folge, dass jede eingehende Nachricht der abonnierten Quellen, die mindestens einem dieser Tags zugeordnet werden kann, in diese Kategorie einfließt. Unabhängig von den Kategorien wird für jeden Nutzer und für jede Nachricht eine Relevanz $R = [0,1]$ berechnet. Diese Relevanzberechnung untergliedert sich in eine inhaltsbasierende Phase und eine anschließende kollaborative Phase.

In der inhaltsbasierenden Phase werden für jede Nachricht und jeden Nutzer die Schnittmenge der Tags ermittelt - also die Tags, die Nutzer und Nachricht gemeinsam zugeordnet sind. Über diese Tags kann nun der durchschnittliche Relevanzwert R anhand der einzelnen Relevanzwerte t_k des Nutzermodells berechnet werden:

$$R_{Nutzer,Nachricht} = \frac{\sum_{k=1}^{n} t_k}{n}$$

Wenn der zu bewertenden Nachricht keine Tags zugeordnet sind, so wird der Relevanzwert anhand eines globalen in der Anwendung definierten Wertes bestimmt.

In der anschließenden kollaborativen Phase wird das Ergebnis der inhaltsbasierenden Phase aufgegriffen und verfeinert. Ziel dieses Vorgehens ist es, die subjektiven Bewertungen ähnlicher Nutzer in die Filterung einzubeziehen. Um dies zu erreichen, wird ein Boni-Strafen-Algorithmus verwendet, der die inhaltsbasierenden

Relevanzwerte aufwertet oder verringert. Im Einzelnen werden hierbei zunächst ähnliche Nutzer auf Basis ihrer Profile im Nutzermodell gesucht. Für die Ähnlichkeitsberechnung wurden drei unterschiedliche Algorithmen ausgewählt und implementiert: ein angepasstes Vektorraummodell, die Euklidische Distanz sowie die Pearson Korrelation (vgl. [BRE98]). Das klassische Vektorraummodell vergleicht, wie zwei Vektoren im Raum zueinander liegen, beachtet allerdings nicht die Länge der Vektoren. Dies ist entscheidend bei der Berechnung der Ähnlichkeit zwischen zwei Nutzern. Wenn zwei Nutzer gleichgelagerte Interessen mit unterschiedlichen Relevanzwerten haben, führt das klassische Vektorraummodell zu einer ungewollten Übereinstimmung. Das Vektorraummodell wurde daher angepasst, womit die Länge der Vektoren und damit die Relevanzwerte berücksichtigt werden.
Wenn ähnliche Nutzer gefunden wurden, wird geprüft, ob diese positive oder negative Bewertungen für die betreffende Nachricht abgegeben haben. Im Fall von positiven Bewertungen wird zu dem ursprünglichen inhaltsbasierenden Relevanzwert R ein Bonus B addiert und im Fall von negativen Bewertungen wird eine Strafe S subtrahiert. Dabei kann die Stärke dieser Boni und Strafen durch global definierte Faktoren (F_B, F_S) eingestellt werden.

$$\text{Bonus: } B = F_B * (1 - R) \qquad \text{Strafe: } S = F_S * R$$

Die ermittelten Boni und Strafen werden, wie bereits beschrieben, mit der inhaltsbasierenden Relevanz verrechnet und bilden somit das Interesse des Nutzers für das entsprechende Nachrichtenelement ab. Abhängig von der vom Nutzer gewählten Filterstärke werden alle Nachrichten unter diesem Schwellenwert verworfen, so dass nur noch die relevantesten Inhalte zur Anzeige gebracht werden. Eine detaillierte Beschreibung und Evaluation des Filteralgorithmus sowie Beispiele finden sich in [Eix09].

3.3 Frontend

Um die Aufnahme der Information beim Nutzer so effizient wie möglich zu gestalten, ist es notwendig, die aufbereitete Information auf geeignete Weise zu präsentieren. Zudem werden dem Nutzer möglichst kurze Informationswege zu relevanten Informationen geboten. Die Informationsstruktur wurde in einer 3-stufigen Hierarchie umgesetzt, bestehend aus den Übersichtsseiten, in denen die Kategorien, die der Nutzer über das Frontend erstellen kann, gekapselt werden. Diese wiederum beinhalten die eigentlichen Nachrichtenelemente. Die Kategorien können als Einteilung für Projekte genutzt werden, um im Unternehmenseinsatz die Information nach Projekten zu strukturieren und zu sortieren.
In verschiedenen Übersichtsdarstellungen erlangt der Nutzer Einblick auf neue und relevante Informationsquellen und Nachrichten, die er zuvor innerhalb seiner

Kategorien angelegt hat. Der Nutzer hat hier die Möglichkeit, die Informationen zu filtern, zu sortieren und mit den Nachrichten zu interagieren. Dabei werden dem Nutzer bei der Darstellung durch Positionierung der Nachrichten innerhalb der Sichten und Anpassung der Farbgebung Informationen über Relevanz und Herkunft der Nachricht gegeben.

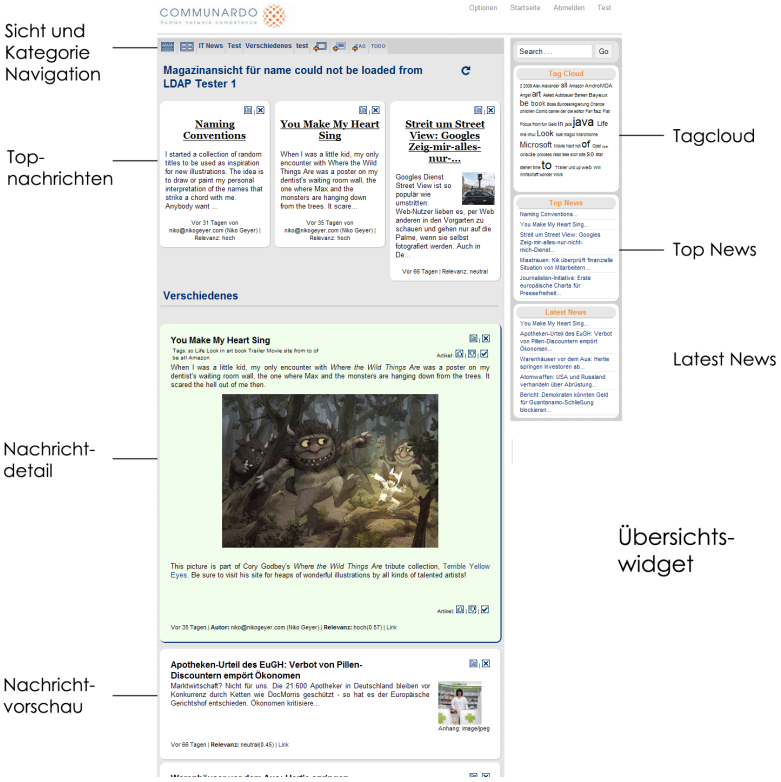


Abbildung 3: Prototyp Zeitungsansicht

Bei der Erstellung der unterschiedlichen Sichten wurden bestimmte Metaphern eingesetzt, um dem Nutzer beim Umgang mit der Information die Art der Darstellung intuitiver zugänglich zu machen.

Die *Zeitungsmetapher*, welche in Abbildung 3 dargestellt ist, wurde eingesetzt, um dem Nutzer in einer Übersichtsseite die neuesten Nachrichten in Form von Schlagzeilen zu präsentieren. Es wurden Paradigmen und Eigenschaften, die bei

der Darstellung von Informationen in Zeitungen genutzt werden, im Kontext der Web-basierten Visualisierung wiederverwendet. Dies bedeutet, dass Nachrichten, die thematisch miteinander verwandt sind, in Kategorien zusammengefasst visualisiert werden. Wichtige Nachrichten werden in den oberen Bereich der Seite integriert und erhalten dort eine größere Darstellungsfläche.
In einer weiteren Übersichtsdarstellung wurde die *Dashboardmetapher* genutzt, um dem Nutzer einen Überblick über den aktuellen Zustand des Systems zu geben. In der Dashboardsicht werden dem Nutzer keine konkreten Informationselemente präsentiert, vielmehr werden sogenannte „High-Level Summaries“ angeboten, in denen Auskunft über die Teilbereiche des Systems gegeben werden. Der Nutzer erkennt auf Anhieb in welcher Kategorie wie viele neue Nachrichten vorliegen, von welchem Typ oder welcher Quelle diese stammen und auch wie viele Nachrichten er in die „To Read“ Liste verschoben hat. Allgemein folgen viele Bereiche der Anwendung der von aktuellen Betriebssystemen oder Browsern bekannten *Desktopmetapher*. So erfolgt die Navigation zwischen den Kategorien und Sichten durch Reiter im oberen Bereich der Seite. Da sich die einzelnen Sichten und Teilbereiche in technischen und funktionalen Anforderungen bei der Darstellung unterscheiden, wurden diese in Widgets gekapselt. Diese Widgetarchitektur erlaubt es, nur bestimmte Widgets - abhängig von Nutzerinteraktionen - asynchron neu zu laden. Damit minimiert sich der Zeitaufwand, Informationen im Frontend zu aktualisieren.

4 Evaluation

Das in den vorangegangenen Abschnitten beschriebene System wurde als Prototyp implementiert und hinsichtlich seiner Funktionalität überprüft. Der Nutzer ist in der Lage eigene Nachrichtenquellen anzugeben und diese in Kategorieren zu sortieren sowie über Bewertungen von Nachrichten Einfluss auf sein Nutzermodell zu nehmen. Es wurden innerhalb funktionaler Testläufe verschiedene Nachrichtenquellen wie RSS-Feeds, Communote und Twitter angebunden und Kategorien zugeordnet. In der Frontendsicht erhielt man eine Übersicht über die Nachrichten dieser Quellen absteigend sortiert nach ihrer Relevanz. Dabei waren die Nachrichten, deren Tags einen hohen Relevanzwert im Nutzermodell besaßen, höher gewertet als Nachrichten mit Tags von geringeren Relevanzwerten. Damit wurde festgestellt, dass die Kernfunktionen Aggregation von heterogenen Nachrichtenquellen, Filterung und Empfehlung von Nachrichten und die Verfahren zur Interessenabbildung und Aktualisierung den eingangs gestellten Anforderungen (R1-R6) gerecht werden.
Darüber hinaus wurden die von der Anwendung unterstützten Algorithmen zur Ähnlichkeitsberechnung zwischen Nutzern evaluiert. Dabei konnte festgestellt werden, dass der Algorithmus des angepassten Vektorraummodells, gemessen an den Erwartungswerten, die besten Ergebnisse lieferte. Der Algorithmus der Pearson Korrelation führte in einigen Fällen zu starken Abweichungen von den Erwartungswerten.

Weiterhin wurden Präsentation und Darstellung der Informationen sowie die Interaktions- und Navigationsmechanismen in einer zweistufigen Nutzerstudie in einem Unternehmenskontext evaluiert. Von den angebotenen Sichten wurde die Zeitungsansicht von den meisten Nutzern favorisiert. Gerade von Projektleitern wurde der Ansatz der Dashboardansicht positiv bewertet, da sich hier eine Übersicht über alle Kategorien und damit Projekte offenbart. Die Nutzung dieser Ansicht muss durch eine Erweiterung der Funktionalität weiter forciert werden, so dass dem Nutzer auch hier nicht nur Informationen über den Status der Kategorien und Nachrichten, sondern auch Navigations- und Filtermöglichkeiten sowie direkte Wege zur Information oder zu Ausschnitten des Informationsraumes geboten werden. Die Grundfunktionalität bezüglich Navigation und Funktionalität wurde als überwiegend intuitiv erachtet. Nachdem ein Überblick über das System gewonnen wurde, erkannten diese Nutzer die vorgesehene Funktion der Verknüpfung von Schlagworten mit Kategorien. Nutzer, die bereits Erfahrungen mit ähnlichen Systemen hatten, fanden sich ohne Probleme im System zurecht. Gerade der kollaborative Umgang der Filterung wurde von den erfahrenen Nutzern als besonders wichtig bewertet.

Die Evaluierung des Prototypen sowie die Nutzerstudie haben gezeigt, dass das System dazu beiträgt, die Informationsflut beherrschbar und relevante Informationen schneller auffindbar zu machen. Es wurde deutlich, dass die Art der Anwendung vor allem auch im Bereich des Einsatzes im Unternehmen positiv akzeptiert wurde.

5 Zusammenfassung und Ausblick

In diesem Beitrag wurde ein für den unternehmensinternen Einsatz vorgesehenes System zur Aggregation und personalisierten Filterung von Nachrichten aus heterogenen Web-basierten Quellen in seinen Grundzügen vorgestellt. Nach der Beschreibung der notwendigen Anforderungen wurde ein Gesamtüberblick über das als Drei-Schichten-Architektur konzipierte und implementierte System gegeben. Darüber hinaus wurde detailliert auf die unterstützten inhaltsbasierten und kollaborativen Filtermechanismen eingegangen. Nach der Behandlung der Backend-Funktionalitäten wurde das Frontend vorgestellt, das die Informationsdarstellung auf Basis einer Zeitungs- und Dashboardmetapher realisiert. In einer Evaluation aus technischer Perspektive und Perspektive von Endnutzern wurde die Eignung des Systems beschrieben.

Für die weitere Entwicklung ist eine Verbesserung und Erweiterung des Tagging-Mechanismus vorgesehen. Dabei sollen vor allem eine Erkennung von Synonymen und eine Abbildung von Relationen zwischen Tags umgesetzt werden. Darüber hinaus sind fortgeschrittene Lernverfahren für Nutzerinteressen vorgesehen, wie beispielsweise die Detektion von Lesegewohnheiten oder –dauer.

Literatur und Verweise

[Can09] Serkan Canbolat, Web-basierte Visualisierung personalisierter Informationen aus semi-strukturierten Informationsquellen - Vergleichende Analyse. Konzeption und protoypische Realisierung, TU Dresden, 2009 (Verfügbar ab Juli 2009 im Onlineportal der SLUB Dresden)

[Com09] Communote Website: http://www.communote.com

[Del09] Delicious Website: http://delicious.com/

[Eix09] Thomas Eixner, Personalisierte Filterung von Nachrichten aus semistrukturierten Quellen, TU Dresden, 2009

[Twi09] Twitter Website: http://www.twitter.com

[Att09] Attensa Feed Reader Website: http://www.attensa.com/products/readers/

[NSS09] Newsgator Social Sites Website: http://www.newsgator.com/business/socialsites

[BRE98] Heckerman, David et al., Empirical analysis of redictive algorithms for collaborative filtering, Fourteenth Annual Conference on Uncertainty in Artificial Intelligence, pp. 43-52., 1998