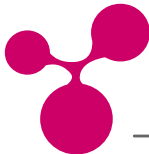


Technische Universität Dresden – Fakultät Informatik  
Professur für Multimedialechnik, Privat-Dozentur für Angewandte Informatik

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



# GENEME '08

GEMEINSCHAFTEN IN NEUEN MEDIEN

an der  
Fakultät Informatik der Technischen Universität Dresden

mit Unterstützung der

GI-Regionalgruppe Dresden  
Initiative D21 e.V.  
Kontext E GmbH, Dresden  
Medienzentrum der TU Dresden  
SALT Solutions GmbH, Dresden  
SAP Research CEC Dresden  
Saxonia Systems AG, Dresden  
T-Systems Multimedia Solutions GmbH  
3m5. Media GmbH, Dresden

am 01. und 02. Oktober 2008 in Dresden  
<http://www-mmt.inf.tu-dresden.de/geneme/>  
[geneme@mail-mmt.inf.tu-dresden.de](mailto:geneme@mail-mmt.inf.tu-dresden.de)

---

## C.3 Anwendungsübergreifende Web-2.0-Kollaborationsmuster

*Stefan Pietschmann, Vincent Tietz*

*Lehrstuhl für Multimedialechnik, Technische Universität Dresden*

### 1 Einleitung

Mit dem Paradigmenwechsel zum „Web 2.0“ hat das World Wide Web (WWW) den bisher wichtigsten Schritt von einem statischen Präsentationsmedium zu einer universellen Softwareplattform gemacht. Es dient heute sowohl der Verbreitung von Informationen als auch der Kommunikation und Kollaboration, was sich beispielsweise an der stark wachsenden Nutzerbeteiligung in virtuellen Gemeinschaften und sozialen Netzwerken ablesen lässt [Eime07].

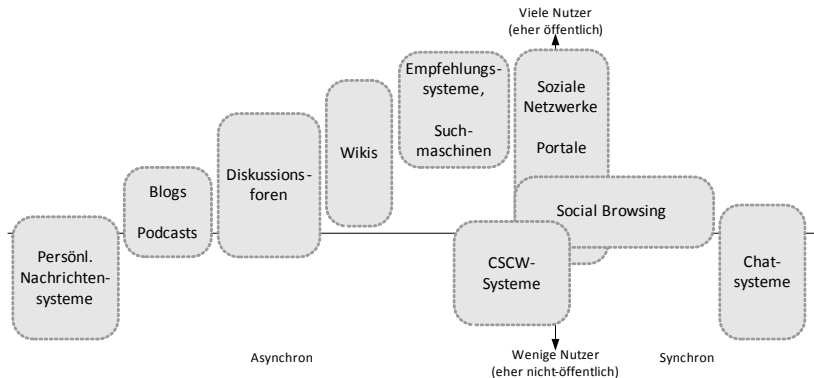
Charakteristisch für kollaborative Webanwendungen ist, dass sie häufig ähnliche Kollaborationstechniken bzw. -muster (z. B. Kommentieren, Bewerten) implementieren. Diese und das entstehende kollektive Wissen stehen nur registrierten Nutzern zu Verfügung und beziehen sich häufig nur auf Inhalte innerhalb der Anwendung. So entstehen eine Vielzahl voneinander isolierter Netzwerke und redundanter, unvollständiger Wissensbasen, die unabhängig voneinander aufgebaut und gepflegt werden. Ein Online-Artikel wird so mehrfach in diversen Foren, Blogs, und auf der Artikelseite selbst kommentiert und bewertet. Diskussionsstränge, Kommentare, Bewertungen und zusätzliche Verweise existieren in Netzwerken parallel und bauen nicht aufeinander auf. Es gibt keine gemeinsame Sicht (bzw. „Kollaborationsebene“), sodass die angestrebte, umfassende kollektive Intelligenz kaum erreicht wird. Neben kollaborativ erstellten und nutzbaren Inhalten existiert weiterhin eine große Menge an Webinhalten auf statischen Seiten, die (von der Referenzierung abgesehen) zur Einbeziehung in kollaborative Anwendungen unerreichbar bleiben. Hier besteht ein Bedarf, diese Inhalte nutzbar zu machen, ohne auf serverseitige Weiterentwicklungen o. ä. angewiesen zu sein.

Dieser Artikel stellt ein Konzept zur anwendungsunabhängigen, webbasierten Kollaboration vor, welches gängige, wiederkehrende Kollaborationsmuster von Webanwendungen trennt und in Form einer anwendungsübergreifenden Kollaborationsschicht, unabhängig vom reinen Web-Browsing, nutzbar macht. Zunächst werden analysierte Web-Anwendungen genannt und Kollaborationsmuster charakterisiert. Anschließend wird ein Konzept vorgestellt, welches die Beschreibung sowie dynamische Einbindung, Verwaltung und Nutzung solcher Muster unabhängig von konkreten Webinhalten ermöglicht. Dessen Umsetzung wird anschließend anhand eines Prototyps vorgestellt und diskutiert. Der letzte Abschnitt fasst die Ergebnisse zusammen und gibt einen Ausblick auf zukünftige Arbeiten.

## 2 Kollaborative Anwendungen im „Web 2.0“

Die Möglichkeiten zur Partizipation von Nutzern im Web sind sehr unterschiedlich und reichen vom Selektieren von Objekten über das Kommentieren und Bewerten bis hin zum Betreiben einer eigenen Webseite, z. B. in Form eines Weblogs. Häufig entstehen dabei gemeinsame Artefakte, die auch als kollektives Wissen oder kollektive Intelligenz bezeichnet werden können und das wesentliche Motiv für die virtuelle Kollaboration darstellen [Heyl99].

Zur Analyse vorhandener Kollaborationstechniken wurden gängige „Web 2.0“-Anwendungen betrachtet – ein Begriff, den wir im Kontext der Untersuchung an den folgenden Kriterien festmachen: **(1)** Wenigstens ein Teil der Anwendung verwendet das Web als Plattform. **(2)** Die aktive sowie passive Nutzung der Anwendung erzeugt oder verändert mindestens ein Artefakt (z. B. kollektives Wissen und Gruppengedächtnis). **(3)** Die Benutzer können untereinander soziale Beziehungen pflegen.



**Abbildung 1: Klassen kollaborativer Web-Anwendungen**

Bei genauerer Betrachtung wird deutlich, dass sich verschiedene Anwendungsklassen identifizieren lassen (vgl. [Kolb06]), die sich im Wesentlichen durch ihren Grad an Öffentlichkeit und Synchronität sowie die Komplexität der eingesetzten Kollaborationsmuster unterscheiden (Abbildung 1).

Persönliche Nachrichtensysteme sind die ursprünglichste Form der computerbasierten Kommunikation (z. B. E-Mail-Anwendungen) und sind durch den Austausch von asynchronen Nachrichten hauptsächlich in einem begrenzten Personenkreis charakterisiert. Chatsysteme ermöglichen den synchronen Austausch textbasierter und audiovisueller Nachrichten häufig nur zwischen wenigen Teilnehmern. Weblogs sind regelmäßig aktualisierte Webseiten mit in chronologischer Reihenfolge sortierten Beiträgen und unterstützen i.d.R. das Kommentieren und Bewerten. Analog

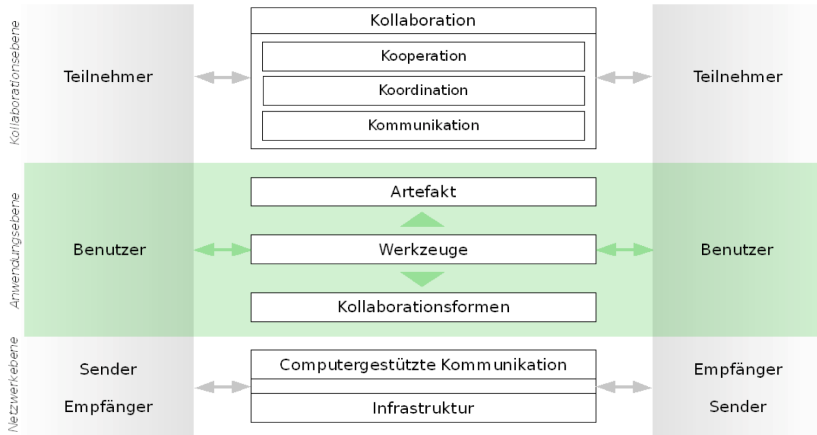
bezeichnen Podcasts regelmäßig unidirektional publizierte audiovisuelle Beiträge. **Diskussionsforen** erlauben den asynchronen Austausch meist öffentlicher und in der Regel langfristig abrufbarer Nachrichten unter sehr vielen Nutzern. **Wikis** ermöglichen es jedem Nutzer, selbst Hypertexte zu erstellen und zu ändern, ohne dass sich dieser registrieren muss. Aufgrund dieser Offenheit gibt es keinen Bezug zu einem Urheber und der Inhalt wird durch Versionierung und Reversibilität geschützt. **Empfehlungssysteme** und **Kollaborative Suchmaschinen** sind häufig in „Web-2.0“-Anwendungen integriert [Lind03] und erzeugen im Allgemeinen eine sortierte Auswahl von Items auf Basis spezieller Algorithmen und Kriterien, die einerseits vom Anwender explizit angegeben werden können (z. B. Suchanfrage oder Benutzerprofil) oder andererseits implizit aus einem vorhandenen Datenbestand stammen. In den meisten Fällen werden beide Möglichkeiten kombiniert [Niwa06,Resni94]. **Soziale Netzwerke** dienen vorrangig dazu, soziale Beziehungen von Menschen aufzubauen und darzustellen. Benutzer können Freunde einladen, eine sog. Freundesliste verwalten und mit anderen Nutzern über verschiedene Interaktionstechniken (z. B. Chat und Persönliche Nachrichten) kommunizieren. **Medien-Portale** implementieren auch Funktionen der Sozialen Netzwerke, setzen ihren Fokus allerdings auf das Teilen und Verteilen von Medienobjekten (z. B. Videos und Bilder). Auch hier können Inhalte häufig kommentiert, bewertet oder mit Schlagworten versehen werden. **Social-Browsing-Werkzeuge** sind charakterisiert durch die Integration verschiedener Kollaborations- und Awarenessstechniken in den Webbrowser, so dass das „gemeinsame“ Browsen auf einer Webseite möglich wird. **CSCW-Systeme** konzentrieren sich auf die Unterstützung der Zusammenarbeit in kleineren Teams. Sie integrieren ebenfalls häufig verschiedene Kollaborationstechniken (z. B. Chat, Persönliche Nachrichten) und sind nicht immer als Webanwendung realisiert.

### 3 Kollaborationsmuster

Ein Muster ist ein erkennbares, wiederkehrendes Schema, das durch Abstraktion in Form einer Beschreibung ähnliche Phänomene und Probleme verallgemeinert und verständlicher macht. Auch virtuelle Kollaboration kann in Mustern ausgedrückt werden, die jedoch auf gänzlich unterschiedlichen Ebenen betrachtet werden können, wie Abbildung 2 verdeutlicht.

Auf der untersten Ebene, der *Netzwerkebene*, ist Kollaboration durch den Austausch von Informationseinheiten sowie Aktionen zur Veränderung von Bits charakterisiert. Auf der *Kollaborationsebene* steht hingegen der zwischenmenschliche Aspekt im Mittelpunkt und Kollaboration kann ganze Arbeitsabläufe umfassen. Um für unser Konzept wiederverwendbare, anwendungsunabhängige Muster identifizieren zu können, muss Kollaboration deshalb auf der mittleren *Anwendungsebene* betrachtet werden. Hier interagiert der Benutzer indirekt unter Zuhilfenahme eines Werkzeugs mit anderen Benutzern, um so gemeinsam ein Artefakt zu bearbeiten. Das

Werkzeug implementiert dazu verschiedene Formen der Kollaboration, also virtuelle Kollaborationsmuster, wie z. B. das Kommentieren und Bewerten.



**Abbildung 2: Klassifikationsschema virtueller Kollaboration**

Ein Muster der virtuellen Kollaboration besteht aus den Entitäten Rolle (*Role*), Artefakt (*Artifact*), Aktion (*Action*), Kommunikationskanal (*Communication Channel*) und Kollaborationsraum (*Collaboration Space*) [Biuk03]. Organisatorische Rollen können von einem oder mehreren Subjekten (den Nutzern) eingenommen werden und besitzen einen eindeutigen Namen (z. B. Autor und Leser). Artefakte sind passive Objekte, die bestimmte Informationen (bzw. Daten) enthalten. Rollen verändern die Artefakte (z. B. Lesen und Schreiben) und kommunizieren mit anderen Rollen mit Hilfe von Aktionen (z. B. Senden und Empfangen) über einen Kommunikationskanal. Kollaborationsräume gruppieren die bisher genannten Elemente. Zum Beispiel existieren in einem text-basierten Chat die Rollen der aktiven (Autoren) und passiven Nutzer (Leser), Nachrichten und der gesamte Nachrichtenverlauf repräsentieren Artefakte, die über bestimmte Aktionen erstellt oder verändert werden, und der Chatraum stellt den Kollaborationsraum bzw. eine Instanz eines Kollaborationsmusters dar.

Bei der Analyse vorherrschender kollaborativer Web-Anwendungen (siehe Abschnitt 2) zeigt sich, dass eine Reihe von Kollaborationstechniken sehr häufig auftreten und Formen bzw. Spezialisierungen ganz grundlegender Muster darstellen. Dazu zählen insbesondere das *Verteilte Objekt*, das einer unidirektionalen Distribution eines Objektes entspricht und das *Gemeinsame Objekt*, das die Bearbeitung eines gemeinsamen Objekts ermöglicht. Darauf aufbauend können die Formen *Kommen-*

tieren, Bewerten und Persönliche Nachrichten identifiziert werden. Abbildung 3 zeigt beispielhaft die Muster *Diskussion* (a) und *Bewerten* (b) mit Hilfe der in [Biuk03] vorgestellten EMOO-Notation.

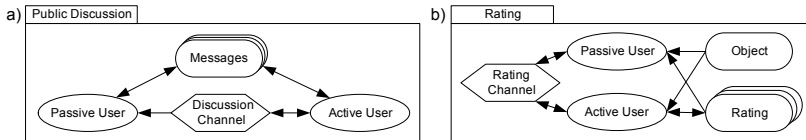


Abbildung 3: EMOO-Diagramm der Muster „Diskussion“ und „Bewerten“

## 4 PaBaCo: Ein System zur musterbasierten Kollaboration

Auf der Basis virtueller Kollaborationsmuster wird im Folgenden ein Konzept für ein Softwaresystem PaBaCo (Pattern-based Collaboration) vorgestellt, das die anwendungsübergreifende Kollaboration im Web ermöglicht. Die Grundidee besteht darin, wiederkehrende Kollaborationsmuster in Webanwendungen (vgl. Abschnitt 3) zu identifizieren und sie basierend auf einem abstrakten Datenmodell sowie einer passenden Beschreibungssprache zu definieren. Dabei wird auf die zuvor beschriebenen Entitäten, wie Rollen, Artefakte und Aktionen zurückgegriffen. Die konkrete Umsetzung der Muster (d. h., der UI und Interaktion) erfolgt durch Plug-Ins bzw. die kollaborative Anwendung selbst. Im Hintergrund wird das PaBaCo-System für Aufgaben wie die Nutzerverwaltung, das Artefaktmanagement und die Persistenz genutzt, und somit die anwendungsunabhängige Verwaltung kollaborativer Muster (und deren Artefakte) gewährleistet.

Zunächst wird das Gesamtsystem, welches sich in Front- und Backend unterteilen lässt, vorgestellt. Nach einer genaueren Betrachtung der jeweiligen Aufgaben und des abstrakten Modells zur Musterbeschreibung geben wir einen Einblick in den groben Ablauf im System zur Laufzeit.

### 4.1 Überblick über die Architektur

Das Gesamtsystem für die musterbasierte Kollaboration teilt sich in zwei Bereiche: Front- und Backend (vgl. Abbildung 4). Der erstere enthält anwendungsspezifische Logik zur Umsetzung von Kollaborationsmustern, während der letztere das Management der Kollaborationsdaten und Nutzer sowie die Synchronisation übernimmt. Im entwickelten Prototyp entspricht sie der Trennung zwischen Client und Server, jedoch ist auch ein Peer-to-Peer-Szenario denkbar, bei welchem das Gesamtsystem auf den Peers läuft. PaBaCo kann sowohl zur browser-basierten Kollaboration genutzt werden (z. B. über Browser-Erweiterungen wie in Abschnitt 5 beschrieben), als auch an beliebige andere kollaborative Anwendungen angebunden werden. Als wesentliche Grundanforderungen für die Nutzung des

Backends besteht lediglich die Unterstützung des PaBaCo-Datenmodells bzw. der Musterbeschreibung.

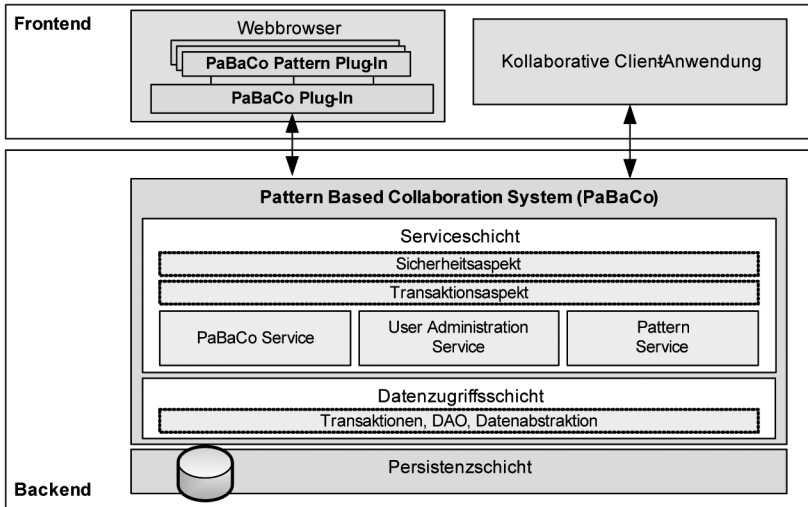


Abbildung 4: Softwarearchitektur für die musterbasierte Kollaboration

Das **Frontend** kapselt den anwendungsspezifischen Teil des Konzeptes und ermöglicht die Verwendung von Kollaborationsmustern im Anwendungskontext. Die hier umgesetzten Komponenten stellen die Präsentationsschicht der Muster dar, die es dem Nutzer bzw. der Anwendung erlauben, Artefakte hinzuzufügen, zu verändern, etc. Im Fall der Integration in den Browser lässt sich das Frontend in eine funktionale Basiskomponente sowie Muster-Plug-Ins unterteilen (Abbildung 4, *PaBaCo (Pattern) Plug-Ins*). Erstere bildet die Präsentationsschicht für die grundlegenden Aufgaben des Backends (Nutzer- und Musterverwaltung), während letztere jeweils ein konkretes Kollaborationsmuster umsetzen, indem sie Möglichkeiten (z. B. Dialoge) zur Manipulation von Artefakten anbieten. Generell können im Frontend unbestimmt viele Muster registriert sein, die jeweils neben ihrem Namen und Programmcode über eine deklarative Musterbeschreibung verfügen. Diese basiert auf einem abstrakten Datenmodell, auf welches näher in Abschnitt 4.2 eingegangen wird, und ermöglicht die Verwaltung auch nicht-antizipierter Muster durch das Backend.

Das **Backend** stellt eine generische, anwendungsunabhängige Serviceschnittstelle zur Verfügung, über welche die Speicherung und Verwaltung kollaborativer Muster und Artefakte (*Pattern Service*), grundlegende Community-Funktionalität, wie z. B. die Verwaltung von Benutzerprofil und Freundesliste (*User Administration Service*),

sowie allgemeine Funktionen, wie die Registrierung von Nutzern und die Auskunft über den Status des Systems (*PaBaCo Service*) ermöglicht wird. Natürlich müssen in einem solchen verteilten, kollaborativen System insbesondere Aspekte der Sicherheit (wie Schutz vor unautorisiertem Zugriff und Verschlüsselung von Daten) und Integrität der Daten Beachtung finden. Deshalb verfügt die Serviceschicht zusätzlich über Mechanismen zum Sicherheits- und Transaktionsmanagement.

## 4.2 Datenmodell

Die Verwaltung von Kollaborationsinformationen in PaBaCo basiert auf einem abstrakten Datenmodell, welches die Beschreibung und somit Verwaltung beliebiger Muster ermöglicht und in Abbildung 5 grau dargestellt wird. Um PaBaCo nutzen zu können, müssen Anwendungen ihre Kollaborationsmuster also zunächst auf Basis dieses Modells beschreiben. Eine solche Beschreibung besteht im Wesentlichen aus Artefakten (*Artifact Classes*), deren Attributen (*Artifact Class Attributes*) und Aktionen (*Actions*), die auf Artefakten durchgeführt werden können und an gewisse Rollen bzw. Berechtigungen (*Role Capabilities*) geknüpft sind.

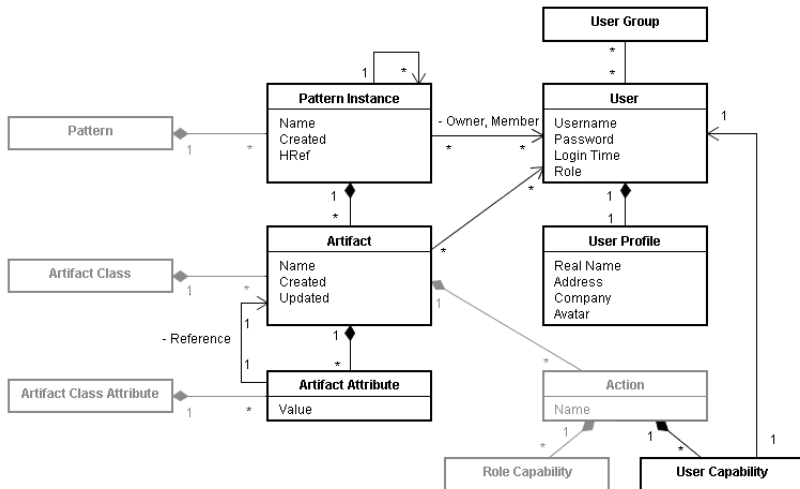


Abbildung 5: PaBaCo Datenmodell für Kollaborationsnutzer

Zur Nutzung von Kollaborationsmustern müssen diese zunächst instanziiert werden, wodurch sich ein etwas komplexeres Model ergibt (Abbildung 5). Neben Instanzen von Mustern, Artefakten und deren Attributen werden Abhängigkeiten zu Nutzern eingeführt (z. B. Besitzer eines Artefakts), welche bestimmte Rollen verkörpern (Role



und User Capabilities). Diese bestimmen die Aktionen, die Nutzer auf Artefakten ausführen dürfen. Somit wird ein nutzer- bzw. rollenbasiertes Rechtemanagement umgesetzt, das eine feingranulare Spezifikation von Zugriffs- und Manipulationsrechten ermöglicht.

### 4.3 Funktionsweise zur Laufzeit

An dieser Stelle sei kurz der Ablauf skizziert, der für die Durchführung einer einfachen Kollaborationsform, wie z. B. dem *Kommentieren*, nötig ist. Angenommen, ein Nutzer möchte einen Artikel eines Nachrichtenanbieters kommentieren, dessen Webanwendung diese Funktion nicht bereitstellt. Ist die PaBaCo-Funktionalität bereits in den Browser integriert (*PaBaCo Plug-In*), so benötigt der Nutzer lediglich ein *Pattern-Plug-In*, welches ihm die Möglichkeit bietet, Kommentare zu erstellen. Dieses kann ggf. automatisch installiert werden. Nun kann er eine Instanz des Musters erstellen, also in diesem Fall einen neuen Kollaborationsbereich vom Typ *Kommentieren*, welcher anderen Lesern des Artikels automatisch angezeigt wird. Bei der erstmaligen Instanziierung eines Musters werden transparent im Backend die nötigen Datenstrukturen angelegt. Anschließend kann ein neuer Kommentar, d. h. ein neues Artefakt angelegt werden, wobei der Besitzer verschiedene Rechte (Lesen, Bearbeiten, Löschen, etc.) vergeben kann. Andere Besucher der Seite können den erstellten Kommentar nun sehen und ihrerseits kommentieren, insofern sie die nötigen Rechte besitzen.

Auf Basis dieses einfachen Prinzips können Anwender nun Kommentare, Bewertungen, und Anhänge miteinander teilen. Über den unabhängigen Kollaborationsraum können sie Nachrichten austauschen und soziale Netzwerke aufbauen – völlig unabhängig von einer bestimmen kollaborativen Webanwendung und ihrer spezifischen Funktionalität. Die offene Serviceschnittstelle erlaubt zudem die Integration bzw. Anbindung an bestehende Netzwerke und Anwendungen, sodass nicht ein weiteres, isoliertes Netzwerk entsteht, sondern eine unabhängige, musterbasierte Kollaborationsplattform.

## 5 Umsetzung

Zur Validierung des Ansatzes wurde ein Proof-of-Concept-Prototyp umgesetzt. Er basiert auf einer Client-Server-Architektur und unterstützt die zwei Kollaborationsmuster *Kommentieren* (*Annotation*) und *Bewerten* (*Rating*). Wir wollen zunächst kurz auf die Implementierung des Prototyps eingehen und im Anschluss dessen Funktionsweise bzw. Nutzung näher erläutern.

## 5.1 Implementierung

Das Frontend wurde als Erweiterung für den Mozilla Firefox<sup>1</sup> Browser umgesetzt. Sowohl die Basisfunktionalität (Nutzer- und Musterverwaltung) als auch die beiden Kollaborationsmuster (Kommentieren und Bewerten) wurden als Erweiterungen gekapselt, wie bereits in Abbildung 4 angedeutet. Der Erweiterungsmechanismus ermöglicht die dynamische clientseitige Installation und Integration neuer Muster, die keinerlei Veränderungen an bisherigen Komponenten oder dem Backend nötig machen. Der Nachrichtenaustausch mit dem PaBaCo-Backend geschieht asynchron über eine Ajax-Schnittstelle im Hintergrund.

Das Backend wurde als Java-basierte Webanwendung umgesetzt, die über eine Web-Service-Schnittstelle die größtmögliche Interoperabilität bei maximaler Entkopplung vom Client bietet. Die Implementierung basiert auf dem Spring Framework und Apache CXF. Sie bleibt unabhängig von spezifischen Web Containern und durch den Einsatz von JAP und DAOs auch unabhängig von der verwendeten Persistenzschicht. Für unseren Prototyp wurden Apache Tomcat 6 als Application Server und MySQL 5 als Datenbank gewählt. Um grundlegende Sicherheit zu gewährleisten, wurde ein Authentifizierungsmechanismus auf Basis von Acegi Security eingebunden.

## 5.2 UI und Funktionsweise

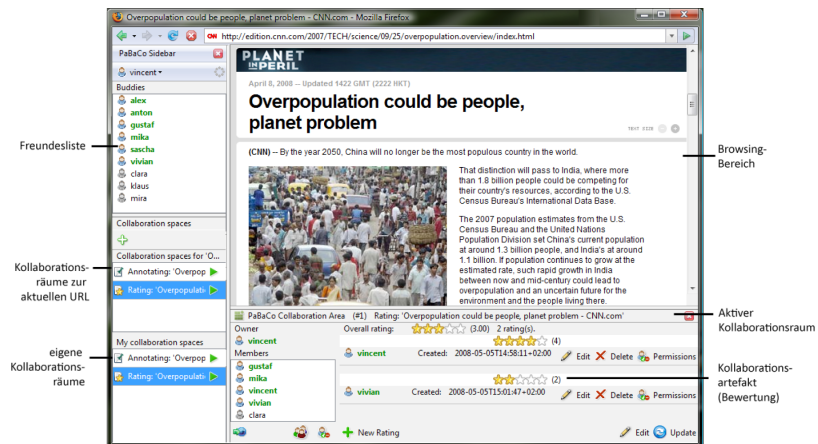


Abbildung 6: Oberfläche der PaBaCo Browser-Erweiterung

Abbildung 6 zeigt die Benutzeroberfläche des Prototyps. Auf der linken Seite, neben dem Darstellungsbereich des Browsers, ist eine **Seitenleiste** zu sehen, die von der

<sup>1</sup> <http://www.mozilla.com/products/firefox>

*PaBaCo Sidebar Extension* bereitgestellt wird. Sie entspricht dem konzeptionellen *PaBaCo Plugin* aus in Abbildung 4 und ermöglicht grundlegende Funktionalitäten wie die Benutzerverwaltung (Registrierung, Login und Logout von Nutzern), sowie die Verwaltung einer Freundesliste. Darunter werden die aktuell relevanten Kollaborationsräume angezeigt, d. h. jene, die in Relation zum angezeigten Webinhalt stehen. Sie stellen Instanzen konkreter Kollaborationsmuster dar, die in Form von Erweiterungen dynamisch (de)installiert werden können. Jeder Raum (d. h. jedes Muster) bezieht sich auf ein Web-Artefakt (u. U. mehrere), das durch eine URL bestimmt ist. Genau wie in der Liste darunter, die alle selbst erstellten Kollaborationsräume darstellt, erfolgt beim Klick auf einen Raum die Navigation zu der jeweiligen Web-Ressource und der **Kollaborationsbereich** (unterhalb der Webseite) mitsamt der enthaltenen (sichtbaren) Kollaborationsartefakte wird geöffnet. In Abbildung 6 sind dies Bewertungen verschiedener Nutzer zu einem Artikel. Andere Artefakte wie Kommentare, Notizen, Anhänge oder sogar Nutzer sind – je nach eingesetztem Muster – genauso denkbar.

### 5.3 Diskussion

Im Folgenden sollen einige Teilaspekte der vorgeschlagenen Lösung diskutiert werden, die im Kontext des PaBaCo-Konzeptes bzw. des Prototyps bei der Unterstützung kollaborativer „Web 2.0“-Anwendungen besonderer Betrachtung bedürfen.

#### 5.3.1. Lokalisierung von Artefakten

In modernen Webanwendungen ist die Lokalisierung bzw. Identifikation von Kollaborationsartefakten ein grundlegendes Problem, welches aus der zunehmenden Interaktivität und Dynamik der Inhalte resultiert. Während im Prototyp noch die Adressierung auf Basis von URLs geschieht, so bedarf es für die volle Unterstützung des Konzeptes mächtigerer Prinzipien. So erlauben URLs nicht die Einbeziehung von beliebigen Teilbereichen einer Seite. Hierfür wäre eine erweiterte Adressierung z. B. über XPath-Ausdrücke denkbar, jedoch ist auch dies nicht ohne Einschränkungen möglich. So können dynamische Variablen in den Pfaden, wie z. B. Session-IDs, zu Schwierigkeiten führen, da sie nach einer bestimmten Zeit ungültig werden. Ein weitaus größeres Problem stellen wechselnde Inhalte unter der gleichen URL dar, wie sie beispielsweise in Portalen auftreten. Ein möglicher Ansatzpunkt für die Lösung dieser Probleme ist die regelmäßige Überprüfung der Gültigkeit von Webinhalten durch das PaBaCo-System. Hierzu, und insbesondere zur Auflösung in Konfliktfällen, bedarf es jedoch weiterer Arbeiten.

#### 5.3.2. Performanz bzw. Verteilung

Das Konzept lässt prinzipiell offen, ob das System als Client/Server- oder Peer-to-Peer-Architektur umgesetzt wird. Bei unserem Prototyp wurde die Client/Server-

---

Variante gewählt, da durch die zentrale Datenspeicherung Sicherheit, Integrität und Synchronisation mit geringem Aufwand realisiert werden konnten. Dagegen gibt es Einschränkungen in der Skalierbarkeit, da die Anzahl der Clients durch die Kapazität des Servers begrenzt wird und beim dessen Ausfall das gesamte System nicht mehr genutzt werden kann. Für eine Peer-to-Peer-Architektur sprechen die deutlich höhere Robustheit und Skalierbarkeit, jedoch müssten Front- und Backend auf jedem Peer installiert, und aufwändige Synchronisations- und Sicherheitsmechanismen umgesetzt werden.

### **5.3.3. Anbindung und Integration**

Durch seine generische Service-Schnittstelle bietet das PaBaCo-Backend die Möglichkeit, Kollaborationsdaten mit beliebigen Anwendungen auszutauschen. Somit ist die Integration in bereits bestehende „Web-2.0-“ oder auch Desktop-Anwendungen möglich. Die Integration auf Datenebene wurde im Prototyp erfolgreich gezeigt und erlaubt sowohl die Verwendung (Extraktion) von PaBaCo-Daten in Anwendungen als auch den umgekehrten Weg der Bereitstellung kollaborativer Daten aus den Anwendungen für PaBaCo. Doch insbesondere die optische Integration stellt eine große und wichtige Herausforderung für die Nutzerakzeptanz des Systems dar. Hier gilt es Konzepte zu entwickeln, welche die Grenzen zwischen anwendungs- und PaBaCo-basierter Kollaboration verwischen und Kollaborationsmuster möglichst in die Benutzerschnittstelle der Webanwendungen „einweben“. Dadurch würde die künstliche Grenze, die sich im Prototyp durch die Trennung von Browser- und Kollaborationsbereich widerspiegelt, verschwinden und für Nutzer nicht mehr unterscheidbar sein, von welcher Seite Kollaborationsfunktionalität zur Verfügung gestellt wird.

## **6 Zusammenfassung**

In diesem Artikel wurde das Konzept für einen generischen, musterbasierten Ansatz zur anwendungsunabhängigen Web-Kollaboration vorgestellt. Da bei bestehenden („Web-2.0-“) Lösungen der Kollaborationsspielraum durch die Anwendung bzw. das Netzwerk begrenzt ist, lösen wir gängige, wiederkehrende Kollaborationstechniken in Form von Mustern aus den Anwendungen und stellen sie auf einer anwendungsunabhängigen Ebene zur Nutzung bereit. Auf Basis einer deklarativen Beschreibung und einem Client-Teil können Kollaborationsmuster dynamisch in das Kollaborationssystem integriert, Kollaborationsdaten bereitgestellt und auf die wirkliche „kollektive Intelligenz“ aller Teilnehmer zugegriffen werden. Das Konzept wurde auf Basis einer prototypischen Implementierung in Form von Browser-Erweiterungen und einer Serveranwendung erfolgreich erprobt und validiert. Der Prototyp unterstützt die dynamische Integration von Kollaborationsmustern und erlaubt deren Nutzung für beliebige Webinhalte. Somit können Kollaborationsformen wie Diskussion, Tagging,

Bewerten, Kommentieren, etc., auf allen Webseiten durchgeführt werden, die über den Browser erreichbar sind, auch wenn sie selbst diese Mittel nicht bereitstellen. Aufbauend auf den bisherigen, vielversprechenden Tests mit dem Prototyp ist eine Evaluation im Rahmen eines praxisnahen Szenarios geplant. Diese umfasst insbesondere die Betrachtung von Performanz, Sicherheit und Usability, um mögliche Schwachstellen identifizieren und verbessern zu können. In diesem Zusammenhang ist die Entwicklung weiterer Muster-Implementierungen angedacht. Wesentliche Herausforderungen für die Zukunft sind außerdem die in Abschnitt 5.3 diskutierten Probleme, wie die verbesserte Lokalisierung von Artefakten, die nahtlose Integration PaBaCo-basierter Kollaborationsmuster in Webanwendungen, sowie die Untersuchung der Realisierbarkeit einer Peer-to-Peer-basierten Lösung, um die Skalierbarkeit des Systems zu verbessern.

### **Literatur**

- [Biuk03] Biuk-Aghai, Robert P., 2003. Patterns of Virtual Collaboration. Dissertation an der University of Technology, Sydney, Australia
- [Eime07] Eimeren, B. und Frees. B., 2007: ARD/ZDF-Online-Studie 2007: Internetnutzung zwischen Pragmatismus und YouTube-Euphorie. In Media Perspektiven, Bd. 8, S. 362 - 378.
- [Heyl99] Heylighen, F., 1999. Collective Intelligence and its Implementation on the Web: Algorithms to Develop a Collective Mental Map. In Computational & Mathematical Organization Theory 5, S. 253 - 280.
- [Kolb06] Kolbitsch, J. und Maurer, H., 2006. The Transformation of the Web: How Emerging Communities Shape the Information we consume. In Journal of Universal Computer Science, Bd. 12, Nr. 2, S. 187 - 213
- [Lind03] Linden, Greg et al. 2003. Amazon.com Recommendations: Item-to-Item Collaborative Filtering. In IEEE Internet Computing, Bd. 7, Nr. 1, S. 76 - 80
- [Niwa06] Niwa, Satoshi et al. 2006. Web Page Recommender System based on Folksonomy Mining. In Proceeding of the 3rd International Conference on Information Technology: New Generations (ITNG'06). S. 388-393
- [Resni94] Resnik, Paul et al., 1994. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW'94). Chapel Hill, NC.