

Hallway: ein Erweiterbares Digitales Soziales Netzwerk

Leif Singer, Maximilian Peters

Fachgebiet Software Engineering
Leibniz Universität Hannover
Welfengarten 1
30167 Hannover
leif.singer@inf.uni-hannover.de
maximilian.peters@se.uni-hannover.de

Abstract: Social Software verwendet soziale Mechanismen, um beispielsweise Wissen besser in Organisationen verbreiten zu können, Kollaboration zu erleichtern oder das Engagement von Anwendern zu steigern. Durch die Wahl der Mechanismen können daher die Entwickler von Social Software entscheidenden Einfluss auf die Wirkungsweise der Gesamtanwendung nehmen. Bisher gibt es jedoch nur wenige wissenschaftliche Arbeiten, die soziale Mechanismen ihren Effekten zuordnen. Durch diesen Mangel an fundierten Richtlinien zum Entwurf von Social Software fließen Erfahrungen einzelner Entwickler, Spekulationen und Annahmen in die Entwicklung ein. Dadurch ist die die Erfüllung der Anforderungen an die Anwendung schlecht kontrollierbar.

Dieser Beitrag stellt ein digitales soziales Netzwerk (DSN) vor, das mit einem besonderen Fokus auf Erweiterbarkeit und Austauschbarkeit seiner Mechanismen entwickelt wurde. Ebenso wurden bereits einige Mechanismen in alternativen Ausführungen implementiert, so dass diese nun leicht ausgetauscht und Unterschiede im Nutzerverhalten sowie in der Ausbreitung von Informationen im DSN untersucht werden können. Diese Untersuchungen sollen zu Richtlinien für den Entwurf von Social Software führen, die Entwickler auf konkrete Softwareprojekte anwenden können sollen.

1 Einführung

Social Software verknüpft die Mitglieder ihrer Nutzergemeinde, wobei oft Effekte, die aus der impliziten oder expliziten Zusammenarbeit entstehen, nutzbar werden. Sie ist im Rahmen des „Web 2.0“ in den letzten Jahren sehr populär geworden. Die Idee selbst ist jedoch bereits seit Jahrzehnten in Gebrauch: die aktuellen Entwicklungen können als Weiterführung der Arbeiten aus der CSCW-Gemeinde (Computer Supported Cooperative Work) gesehen werden. Ein Beispiel für die implizite Zusammenarbeit ist das kollaborative Bewerten von Inhalten, das die Qualität von Empfehlungen verbessern kann. Explizite Zusammenarbeit bietet beispielsweise Google Docs – verschiedene Nutzer können gleichzeitig ein Dokument bearbeiten.

Ein digitales soziales Netzwerk (DSN) ist eine spezielle Form von Social Software, deren Fokus auf den Verbindungen zwischen den Benutzern liegt [BE08]. Dadurch können Informationen und Wissen durch das soziale Netz der Nutzer als Infrastruktur zur Verbreitung nutzen.

DSN werden beispielsweise zur Awareness-Steigerung in gemeinsamen Projekten, zur Expertensuche oder zum Wissensmanagement verwendet. Zum Software Engineering haben DSN zwei wichtige Bezüge.

1.1 Unterstützung der Softwareentwicklung durch Social Software

Einerseits kann Social Software eingesetzt werden, um Tätigkeiten im Software Engineering zu unterstützen. Dies ist besonders in geographisch oder auch organisatorisch verteilten Projekten interessant, da das wirkliche soziale Netzwerk zwischen den Projektteilnehmern schwächer ausgeprägt ist.

So wurden bereits spezielle anforderungsbezogene DSN eingesetzt, um die Kommunikation zwischen Entwicklern und Testern an verschiedenen Standorten zu verbessern [DM07]. Andere Arbeiten nutzen die in Versionsverwaltungssystemen implizit enthaltenen Informationen über das soziale Netzwerk der Entwickler, um durch Aufbereitung einen Überblick über Projektteilnehmer und -fortschritt zu generieren [BZ10a]. Abschnitt 3 nennt weitere Ansätze in dieser Richtung.

1.2 Ingenieursmäßige Verwendung sozialer Mechanismen

Für den vorliegenden Beitrag interessanter ist die Frage, wie man Mechanismen aus der Social Software systematisch und verlässlich in zu entwickelnde Systeme integrieren kann. Von Fortschritten in dieser Frage würden auch die oben genannten Ansätze profitieren. Populäre Beispiele wie Socialwok¹ zeigen, dass selbst klassische Anwendungen wie eine Textverarbeitung oder Tabellenkalkulation von der Integration sozialer Mechanismen profitieren können.

Um diese Mechanismen gezielt anwenden zu können ist es notwendig, ihre Voraussetzungen und Effekte zu kennen. So mag ein bestimmter Mechanismus nur in Organisationen mit besonders flachen Hierarchien den Wissensfluss zwischen den Mitarbeitern verbessern, starke Hierarchien könnten den Effekt ins Gegenteil umkehren. Wissen um diese Eigenschaften der Mechanismen wäre eine wertvolle Bereicherung des Software Engineering, da es ihren systematischen Einsatz erleichtern würde. Daher stellt dieser Beitrag ein DSN vor, das auf die Evaluierung solcher Mechanismen ausgelegt ist.

¹ <http://socialwok.com/>

Dieser Beitrag ist wie folgt aufgebaut: Im Anschluss an diese Einführung stellt Abschnitt 2 verschiedene soziale Mechanismen vor. Abschnitt 3 fasst verwandte Arbeiten aus den beiden erwähnten Kombinationsmöglichkeiten von Social Software und Software Engineering zusammen. Abschnitt 4 stellt schließlich „Hallway“ vor, das von uns entwickelte erweiterbare DSN. Der Beitrag schließt mit einer Zusammenfassung und einem Ausblick auf unsere weiteren Vorhaben.

2 Mechanismen in Digitalen Sozialen Netzwerken

Als charakteristische Funktionen eines DSN haben Boyd und Ellison Benutzerprofile, Verbindungen zwischen Benutzern sowie die Möglichkeit zur Anzeige der Verbindungen eines Benutzers identifiziert [BE08]. Meist können Benutzer auch kurze öffentliche Nachrichten und andere Inhalte wie Dokumente veröffentlichen. Auf einem *Activity Stream* erhält jeder Benutzer die Veröffentlichungen seiner Kontakte angezeigt und kann so dem aktuellen Geschehen in seinem sozialen Netzwerk folgen.

Das DSN wirkt dabei als Empfehlungssystem und Filter: Inhalte sind für einige Teilnehmer interessant, die sie dann an ihre eigenen Kontakte weiterreichen können. Auf diese Weise fließt Wissen durch das DSN bzw. die Organisation, bei der es eingesetzt wird. Um Struktur und Wirkungsweise des sozialen Netzwerk eines Benutzers zu beeinflussen, können einige Mechanismen ins DSN eingebracht oder vorhandene angepasst werden.

2.1 Kollaborative Empfehlungssysteme

Einige soziale Mechanismen zielen darauf ab, Informationen auch über die Kontakte eines einzelnen Benutzers hinaus zu transportieren. Ein populäres Beispiel hierfür ist die „Like“-Funktion von Facebook²: findet ein Benutzer einen für ihn interessanten Inhalt, so kann er dies durch einen Klick auf die „Like“-Schaltfläche ausdrücken. Dies erzeugt für diesen Nutzer eine neue, implizite Aktivität, die seinen Kontakten angezeigt wird. Auf diese Weise erreicht der eigentlich Inhalt auch Personen, die nicht zu den direkten Kontakten des Autors zählen. Mit den sog. „Retweets“ implementiert Twitter³ einen Mechanismus mit einem sehr ähnlichen Effekt. Interessante Inhalte werden weitergeleitet, uninteressante werden herausgefiltert.

Zu ähnlichen Zwecken gibt es einige unter anderem aus Onlineshops bekannte Empfehlungsmechanismen. So werden bspw. die Produktbewertungen aller Käufer genutzt, um einzelnen Käufern interessante Produkte zu empfehlen.

Der folgende Abschnitt skizziert, wie nicht nur Inhalte, sondern auch Verbindungen selbst durch Empfehlungen und andere Mechanismen beeinflusst werden können.

² <http://facebook.com>

³ <http://twitter.com>

2.2 Verbindungsmodelle

Die beiden klassischen Möglichkeiten, Nutzer eines DSN miteinander zu verbinden, sind das synchrone und das asynchrone Verbindungsmodell. Beim synchronen Modell beantragt ein Nutzer den Kontakt beim Anderen, der diesen dann bestätigen muss. Erst dann sind beide Nutzer miteinander verbunden – als populäres Beispiel verwendet Facebook dieses Modell.

Das asynchrone Modell wird beispielsweise durch Twitter genutzt. Verbindungen bestehen zunächst nur einseitig und müssen per Voreinstellung nicht bestätigt werden. Durch einen Kontakt abonniert ein Nutzer die Inhalte des anderen Nutzers. Oftmals wird dieser von diesem neuen Abonnement informiert, so dass er bei Interesse einen Kontakt in die Gegenrichtung herstellen kann.

Über diese beiden einfachsten Verbindungsmodelle sind noch viele weitere denkbar. Einige Beispiele skizziert die folgende Liste.

- Ein Kontakt zwischen Nutzern wird automatisch hergestellt, falls diese in ihrem Profil ähnliche Interessen angegeben haben oder ähnliche Inhalte als „interessant“ markieren.
- Kontakte werden automatisch hergestellt oder empfohlen, wenn Nutzer einen Schwellwert für räumliche oder organisatorische Nähe unterschreiten.
- Die Frequenz von ausgetauschten privaten Nachrichten zwischen Nutzern fließt in Kontaktempfehlungen oder die automatische Herstellung von Verbindungen ein.
- Verbindungen „verblasen“, falls diese nicht regelmäßig durch Informationsaustausch oder thematische Nähe verstärkt werden.

Diese verschiedenen Methoden, eine Verbindung zwischen Nutzern herzustellen, beeinflussen die Struktur des im DSN abgebildeten sozialen Netzwerks. Die Struktur hingegen beeinflusst, wie Wissen sich verbreiten kann – schnell oder langsam, in der Breite oder in der Tiefe, von Ort oder Thema beeinflusst.

Die Wahl des Verbindungsmodells ist so betrachtet ein Parameter, der die kollaborativen Filterungen und Empfehlungen beeinflusst. Synchrone Verbindungen findet man bspw. eher in identitätsbasierten Gemeinschaften wie Facebook oder XING⁴. Asynchrone Verbindungen sind hingegen sinnvoller, um thematische Gruppierungen zu fördern.

⁴ <http://xing.com>

2.3 Evaluierung alternativer Mechanismen und ihrer Effekte

Wie in den vorangegangenen beiden Abschnitten angedeutet, beeinflusst die Wahl des Verbindungsmodells und der Empfehlungsmechanismen auch die Funktionsweise oder gar den Erfolg der zu entwickelnden Software. Daher wäre es nützlich zu wissen, welche Effekte aus welchen Mechanismen entstehen.

Interessante Fragen, die durch Evaluierungen beantwortet werden könnten, sind beispielsweise die folgenden.

- Wissen weiterzugeben kann für manche Personen bedeuten, dass sie etwas von ihrem Wert aufgeben und deshalb ihren Stand in der Organisation – bspw. bei ihrem Arbeitgeber – verschlechtern. Kann man diesen Effekt abschwächen, indem man in allen Inhalten ihre Autoren als Urheber nennt? Hilft es, Ranglisten von Mitwirkenden zu erstellen oder zur Belohnung Auszeichnungen zu verteilen, die dann im Profil des Nutzers angezeigt werden?
- Die verschiedenen Verbindungsmodelle sind interessant, weil sie für Organisationen verschiedener Größe und mit abweichenden Kulturen womöglich unterschiedliche Effekte haben. Welche Eigenschaften von Organisationen führen zu welchen Effekten? Durch welche Verbindungsmodelle kann man diese ausnutzen oder ihnen entgegenwirken?
- Sind Gruppen mit synchronen Verbindungen „näher“, weil ja alle Verbindungen aus dem Willen beider jeweiligen Parteien entstanden sind? Welche Rolle spielt das Gefühl, eine Kontaktanfrage annehmen zu müssen? Welche Rolle spielt die organisatorische Stellung der Beteiligten hierbei?
- Welches Verbindungsmodell verbreitet Informationen schneller im Netzwerk: das synchrone oder das asynchrone? Da das asynchrone Modell eher thematische als persönliche Beziehungen fördert wirkt es intuitiv wahrscheinlich, dass es Inhalte schneller verteilt. Trifft das wirklich zu?

Das in diesem Beitrag beschriebene erweiterbare DSN soll bei Experimenten unterstützen, die diese und ähnliche Fragen beantworten könnten.

3 Verwandte Arbeiten

Dieser Abschnitt stellt Arbeiten vor, deren Ziel es ist, die Auswirkungen sozialer Mechanismen besser zu verstehen und diese dadurch gezielter und kontrollierter einsetzen zu können.

Nach Cetina bestehen Verbindungen zwischen Menschen vor allem über die Objekte, an denen sie gemeinsames Interesse haben [Ce97]. Dies wurde von der Forschergemeinschaft um DSN aufgegriffen und stellt eine wichtige Entwurfsrichtlinie für DSN dar (siehe bspw. Breslin et. al. [BPD09]).

Ogata et. al. stellen ein System zur Expertensuche vor, das das soziale Netzwerk der Anwender ausnutzt [OYF01]. Ein Vergleich mit alternativen Strategien zur Expertensuche findet nicht statt, es erfolgt lediglich eine Evaluierung des einen, eigenen Ansatzes. Die Arbeit schließt mit der Feststellung, dass die CSCW-Literatur zwar immer mehr soziologische Studien über Arbeit und Zusammenarbeit enthält. Technische Unterstützung, die auf sozialen Aspekten basiert, habe jedoch bisher nur wenig Aufmerksamkeit erhalten.

McDonald untersucht zwei verschiedene Strategien zur Expertensuche [Mc03]. Die eher grundlagenorientierte Arbeit stellt allgemeine Probleme bei der Konstruktion von Social Software dar. Konkrete Handlungs- oder Entwurfsempfehlungen finden sich jedoch leider nicht.

Ähnlich wie Ogata et. al. kommt auch McDonald zu dem Schluss, dass soziale Netzwerke als Entwurfselement von Groupware bisher zu wenig untersucht wurden und weitere Evaluationen und Studien notwendig sind.

Im Projekt „Community Lab“⁵ sind einige relevante Arbeiten entstanden. In [RLT06] präsentieren Rashid et. al. eine Studie, in der Nutzern der Wert ihres Beitrags angezeigt wurde. Dabei vergleichen sie verschiedene Methoden, um diesen Wert zu berechnen. Ähnliche Studien sind [BLW04] von Beenen et. al. sowie [LCF04] von Ludford et. al.. In [RKK07] untersuchen Ren et. al. die Unterschiede zwischen Gemeinschaften, die auf persönlichen Verbindungen bzw. einer gemeinsamen Gruppenidentität beruhen.

Kraut vergleicht gemeinsam mit Farzan et. al. in [FDK] verschiedene Strategien, um das Engagement der Nutzer von DSN positiv zu beeinflussen. In [RK] entwickeln Ren und Kraut ein Modell, das beim Entwurf von DSN die Auswahl des geeignetsten Moderationslevels erleichtern soll. In [KR] stellen Kraut und Resnick eine Sammlung von Richtlinien zum Entwurf von DSN zusammen. Diese drei Arbeiten befinden sich jedoch noch in der Begutachtung bzw. sind noch nicht fertiggestellt.

Im Gegensatz zu den wenigen Arbeiten zur Nutzung sozialer Mechanismen bei der Konstruktion von Software gibt es verhältnismäßig viele Arbeiten dazu, wie Social Software zur Verbesserung des Software Engineering *eingesetzt* werden könnte.

Treudes Arbeiten zeigen Ansätze, wie die Awareness von Softwareentwicklern mit Hilfe von Social Software verbessert werden kann [Tr10], [TS10]. Black präsentiert einige Fallstudien, in denen „Web 2.0“-Technologien verwendet wurden, um den Softwareentwicklungsprozess zu verbessern [BJ10]. Begel zeigt einen Activity Stream, der auf der Codebook-Plattform zur Untersuchung von Versionsverwaltungssystemen aufsetzt. [BZ10a]. Damian et. al. nutzen DSN zur Unterstützung der Anforderungserhebung [DM07]. Einen Überblick über Entwicklungswerkzeuge mit sozialen Mechanismen bieten Storey et. al. in [STD10] und leiten offene Forschungsfragen ab.

⁵ <http://communitylab.org>

Die vorgestellten Arbeiten zeigen, dass durchaus Interesse und Bedarf an der Nutzung sozialer Mechanismen besteht, ihre Wirkungen aber noch besser untersucht werden müssen. Der folgende Abschnitt stellt nun ein erweiterbares DSN vor, das für solche Untersuchungen konzipiert wurde. So soll nicht nur den Einsatz sozialer Mechanismen verlässlicher gestaltet, sondern auch das Software Engineering selbst bereichert werden.

4 Hallway: Konzeption und Entwurf

Um die Effekte von Social Software bei der Softwareentwicklung kontrollierter nutzen zu können, müssen diese empirisch überprüft werden können. Zu diesem Zweck wurde ein DSN entwickelt, in welches zu überprüfende Mechanismen leicht integriert werden können. Das DSN hat den Arbeitstitel „Hallway“.

4.1 Architektur

Nach Boyd und Ellison kennzeichnet ein Digitales Soziales Netzwerk, das Benutzer in einem solchen webbasierten Dienst Profile besitzen, Verbindungen zueinander herstellen können und sich Verbindungen anderer Benutzer ansehen können [BE07]. In der Praxis bieten sie zudem häufig eine Ansicht, in der Benutzer über die Neuigkeiten ihrer Kontakte informiert werden.

Um ein solches DSN zu konstruieren, wurde als Basisarchitektur das Play!-Framework gewählt. Es forciert eine Web-MVC-Architektur, ist gleichzeitig einfach gehalten und erlaubt schnelle Iterationen, was unserem Entwicklungsprozess entgegenkam.

Das DSN sollte zudem die Möglichkeit bieten, leicht Module mit unterschiedlichen Mechanismen austauschen zu können und diese Mechanismen im Parallelbetrieb evaluieren zu können. Hierzu wurde der von Koziolk vorgestellte Mehrmandanten-Architekturstil SPOSAD (Shared, Polymorphic, Scalable Application and Data) gewählt. SPOSAD ist eine Erweiterung des n-Schichten-Modells und folgt dem Ziel, die Kosten für den Betrieb zusätzlicher Mandanten – Kunden des Softwareanbieters – zu reduzieren.

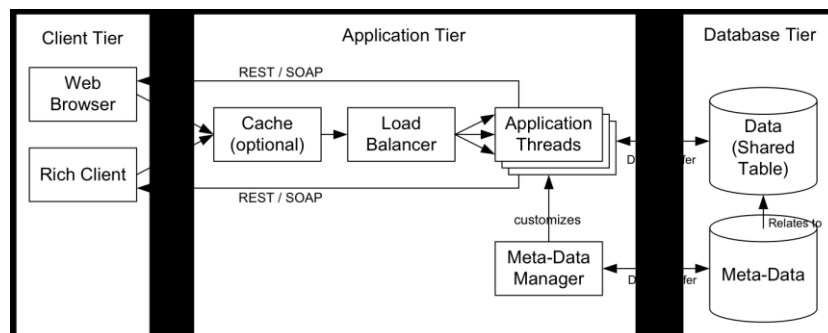


Abbildung 1: Mehrmandanten-Architekturstil SPOSAD, aus Koziolk [Ko10]

Eine SPOSAD-Anwendung wird dadurch gekennzeichnet, dass sie eine einzige Code-Basis für alle Mandanten bereitstellt und durch Ressourcenteilung in der Datenbankschicht sowohl mandantenspezifische Daten, als auch mandantenspezifische Anpassungen (bspw. durch Metadaten) ermöglicht (vgl. Abbildung 1). Bei SPOSAD benötigt die Anwendung lediglich eine Hardware, ein Betriebssystem und eine Datenbank, wobei hiervon beliebig viele Mandanten bedient werden können – die Anwendungen sind hoch skalierbar und günstig zu warten [Ko10].

Der SPOSAD Architekturstil wurde bei Hallway umgesetzt (vgl. Abbildung 2). Auch nach Bezemer und Zaidman, die Multi-Tenant-, Multi-Instance- sowie Multi-User-Anwendungen gegeneinander abgrenzen [BZ10], ist Hallway eine Multi-Tenant- bzw. Mehrmandanten-Anwendung.

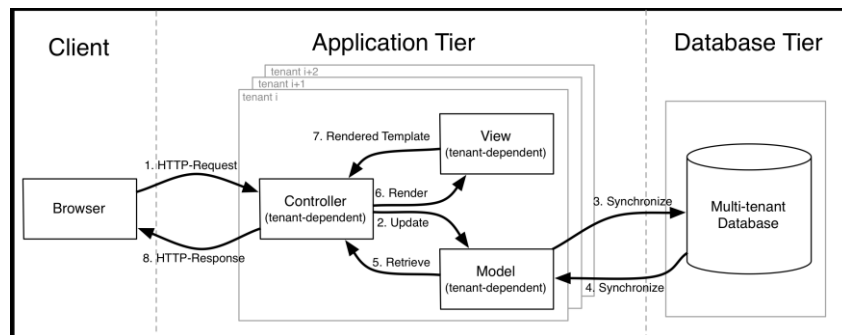


Abbildung 2: Mehrmandanten-Architektur von Hallway im Play!-Framework

Die Mehrmandantenarchitektur von „Hallway“ wurde inzwischen soweit verfeinert, dass mit einer einzigen HTTP-Anfrage ein neuer Mandant angelegt werden kann, sofern dieser noch nicht existiert. Bspw. legt eine Anfrage auf <http://se2011.example.org> den Mandanten „se2011“ in der unter *example.org* laufenden Hallway-Instanz an.

4.2 Module

Mit der umgesetzten Mehrmandantenarchitektur können nun aufgrund des speziell auf Erweiterbarkeit ausgerichteten Entwurfs leicht bspw. Verbindungsmodelle ausgetauscht werden oder zwei unterschiedliche Verbindungsmodelle im Parallelbetrieb von zwei Mandanten verglichen werden. Diese Austauschbarkeit wurde über ein Modulsystem (vgl. Abbildung 3) umgesetzt.

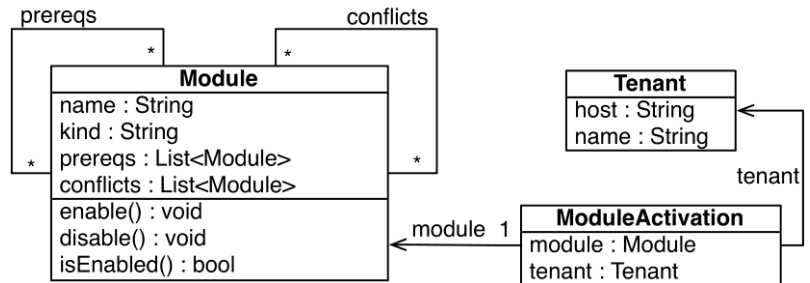


Abbildung 3: Modul-System zum Austausch unterschiedlicher zu untersuchender Mechanismen

Beim Starten der Anwendung werden alle Module registriert – hierfür gibt es für jedes Modul eine Klasse, welche das Interface *RegisterableModule* implementiert. Sobald diese Klasse beim Starten der Anwendung geladen wird, registriert sie ihr entsprechendes Modul.

Diese *RegisterableModule*-Klassen enthalten zudem *setup()*- und *destroy()*-Methoden, die bei Aktivierung bzw. Deaktivierung des Moduls ausgeführt werden und dann benötigte Daten in die Datenbank laden oder bspw. ein Mapping für ein Alternativmodul vornehmen können. Dies kann etwa für die Umstellung von „Likes“ auf „Retweets“ notwendig sein. Jedes Modul kann zwei Listen anderer Module referenzieren: Voraussetzungen (*prereqs*) und Konflikte (*conflicts*). So können bei Aktivierung des Moduls andere benötigte Module aktiviert oder die Aktivierung verweigert werden, falls bereits ein Modul aktiv ist, welches ähnliche Funktionalität bereitstellt.

Durch das beschriebene Modul-System können nun für jeden Mandanten spezielle Module ein- bzw. ausgeschaltet werden. Dabei wurde bewusst auf zwar sehr leistungsstarke, aber auch komplizierte Komponentenmodelle wie OSGI verzichtet. Die Module sollen in zeitlich begrenzten Experimenten genutzt werden, so dass eine einfache und schnelle Erweiterbarkeit beim Entwurf Vorrang hatte.

4.3 Mechanismen

Hallway soll auch an verschiedene Domänen anpassbar sein, um Experimente bei unterschiedlichen Partnern mit verschiedenen Geschäftsmodellen durchführen zu können.

Hierzu wurden im Anschluss an die Implementierung verschiedene Arten von Dokumententypen entwickelt, um die Erweiterbarkeit auch in dieser Hinsicht zu demonstrieren. Als Beispieltypen können nun Fotoalben mit Fotos angelegt, Spreadsheets von Google Docs importiert und versionierte Dateien angelegt werden. Diese Typen von Dokumenten besitzen alle jeweils einen Activity Stream und Benutzer können sich zu ihnen verbinden (vgl. Abbildung 4).

Beispielsweise könnten durch die Integration von SVN-Commit-Nachrichten und Meldungen vom Bug-Tracker in den Activity Stream die Arbeitsabläufe eines Software Entwicklers unterstützt werden.



Abbildung 4: ActivityStream eines Benutzers, der zu verschiedenen Gruppen, Dokumenten und Personen verbunden ist

Einige der eingangs erwähnten Mechanismen wurden bereits in Hallway umgesetzt und lassen sich leicht austauschen. Insbesondere für Verbreitungsmechanismen, die den Wissensfluss begünstigen, liegen schon Implementierungen vor. So ist bspw. ein Mechanismus an „Likes“ von Facebook angelehnt, während ein anderer wie die „Retweets“ von Twitter funktioniert.

Auch das verwendete Verbindungsmodell lässt sich austauschen. Implementiert sind das synchrone sowie das asynchrone Modell. Die Verbindungen lassen sich hierbei jeweils mit einem Mapping übertragen.

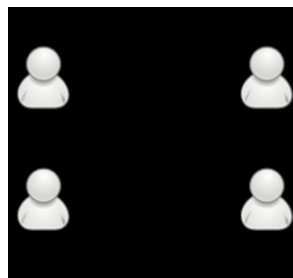


Abbildung 5: Synchrone und asynchrone Verbindungen zwischen Benutzern

Abbildung 5 illustriert den bereits o.g. Unterschied zwischen den beiden Modellen: im synchronen Modell beruht eine Verbindung immer auf Gegenseitigkeit (zwei Möglichkeiten der Verbindung: verbunden oder nicht verbunden), im asynchronen Modell ist dies nicht der Fall (vier Möglichkeiten).

Diese sowie die bereits o.g. möglichen Konzepte für die Verbindungen zwischen Benutzern lassen sich in Hallway leicht über das beschriebene Modul-System austauschen. Durch die Mehrmandanten-Architektur können sie schnell parallel aufgesetzt werden und stehen dann für Evaluationen bereit.

Weiterhin sind die Empfehlungsmechanismen austauschbar. Derzeit ist eine einfache Methode für die Empfehlung zuständig, die hierfür Tags der Benutzer mit denen anderer vergleicht. `hybreed`⁶ ist ein Framework, das die Komposition von Empfehlungs-Workflows unterstützt, die auf erprobten Algorithmen basieren und diese neu kombinieren. Durch eine Integration in Hallway könnten so leicht alternative Empfehlungsstrategien eingebunden werden, um diese miteinander zu vergleichen.

5 Zusammenfassung und Ausblick

Dieser Beitrag hat den im Software Engineering bestehenden Bedarf nach Evaluierungen sozialer Mechanismen motiviert und ein erweiterbares DSN vorgestellt, das durch ein leichtgewichtiges Modulsystem zur Durchführung dieser Evaluierungen geeignet ist. Das DSN unterstützt die in Literatur und Praxis gängigen Grundfunktionen und beinhaltet unterschiedliche soziale Mechanismen, die teilweise auch in alternativen Ausführungen vorliegen. Mittels einer Mehrmandantenarchitektur können nun parallele Experimente mit wenig Aufwand aufgesetzt werden.

Das DSN inklusive seiner Module ist für Evaluierungen, nicht für den längerfristigen, produktiven Einsatz ausgelegt. Daher wurde beim Entwurf des Modul-Systems ein sehr leichtgewichtiger Ansatz mit einigen, für Produktivsysteme unerwünschten Abhängigkeiten gewählt. So sind die Module nicht logisch voneinander getrennt und ziehen sich durch Modell-, Controller- und Präsentationsschicht. Dieser Kompromiss steht jedoch in Einklang mit den Zielen der Plattform: die Umsetzung neuer Module kann mit wenig Aufwand und in kurzer Zeit geschehen.

Eine erste Möglichkeit zur Evaluierung unseres Ansatzes in der Industrie befindet sich zur Zeit in der Anbahnung. Eines der nächsten Ziele ist eine klarere und vollständigere Kategorisierung der sozialen Mechanismen. Aktuelle Arbeiten am Fachgebiet beschäftigen sich außerdem mit der Herleitung angemessener Metriken und Visualisierungen, um die Effekte der Mechanismen erkennen und bewerten zu können.

Literaturverzeichnis

- [BE08] Boyd, D.M.; Ellison, N.B.: Social network sites: Definition, history, and scholarship. In: Journal of Computer-Mediated Communication, Wiley Online Library, 2008; Vol. 13, Nr. 1, S. 210-230.
- [BJ10] Black, S.; Jacobs, J.: Using Web 2.0 to Improve Software Quality. In: Proceedings of the 1st Workshop on Web 2.0 for Software Engineering, ACM, 2010, S. 6-11.

⁶ <http://hybreed.org>

- [BLW04] Beenen, G.; Ling, K.; Wang, X.; Chang, K.; Frankowski, D.; Resnick, P.; Kraut, R.E.: Using Social Psychology to Motivate Contributions to Online Communities. In: Proceedings of the 2004 ACM conference on Computer supported cooperative work, ACM, 2004, S. 212-221.
- [BPD09] Breslin, J.G.; Passant, A.; Decker, S.: The Social Semantic Web. Springer Verlag, Heidelberg, 2009.
- [BZ10] Bezemer, C.; Zaidman, A.: Multi-Tenant SaaS Applications: Maintenance Dream or Nightmare?. In: ERCIM Workshop on Software Evolution (EVOL) and International Workshop on Principles of Software Evolution (IWPSE), ACM, Antwerp, 2010
- [BZ10a] Begel, A.; Zimmermann, T.: Keeping Up With Your Friends: Function Foo, Library Bar.DLL, and Work Item 24. In: Proceedings of the 1st Workshop on Web 2.0 for Software Engineering, ACM, 2010, S. 20-23.
- [Ce97] Cetina, K.K.: Sociality with Objects : Social Relations in Postsocial Knowledge Societies. In: Theory, Culture & Society, SAGE, Thousand Oaks, CA, USA, 1997, Vol. 14, Nr. 4, S. 1-30.
- [DM07] Damian, D.; Marczak, S.; Kwan, I.: Collaboration patterns and the impact of distance on awareness in requirements-centred social networks. In: 15th IEEE International Requirements Engineering Conference, 2007. RE'07. IEEE, 2007; S. 59-68.
- [FDK] Farzan, R., Dabbish, L., Kraut, R. E., & Postmes, T. (eingereicht). Increasing Commitment in Online Communities via Building Social Attachment.
- [Ko10] Koziolok, H.: Towards an Architectural Style for Multi-tenant Software Applications. In (Engels, G.; Luckey, M.; Schäfer, W.): Software Engineering 2010, Bonner Köllen Verlag, Paderborn, 2010.
- [KR] Kraut, R. E. & Resnick, P. (unter Vertrag). Evidence-based social design: Mining the social sciences to build online communities. MIT Press, Cambridge, MA:
- [LCF04] Ludford, P.J.; Cosley, D.; Frankowski, D.; Terveen, L.: Think Different: Increasing Online Community Participation Using Uniqueness and Group Dissimilarity. In: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM, 2004, S. 631-638.
- [Mc03] McDonald, D.W.: Recommending Collaboration with Social Networks: A Comparative Evaluation. In: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM, 2003, S. 593-600.
- [OYF01] Ogata, H.; Yano, Y.; Furugori, N.; Jin, Q.: Computer supported social networking for augmenting cooperation. In: Computer Supported Cooperative Work (CSCW), Springer Verlag, Heidelberg, 2001, Vol. 10, Nr. 2, S. 189-209.
- [RK] Ren, Y., & Kraut, R. E. (eingereicht). A simulation for designing online community: Member motivation, contribution, and discussion moderation.
- [RKK07] Ren, Y.; Kraut, R.; Kiesler, S.: Applying Common Identity and Bond Theory to Design of Online Communities. In: Organization Studies, SAGE, Thousand Oaks, CA, USA, 2007, Vol. 28, Nr. 3, S. 377-408.
- [RLT06] Rashid, A.M.; Ling, K.; Tassone, R.D.; Resnick, P.; Kraut, R.; Riedl, J.: Motivating Participation by Displaying the Value of Contribution. In: Proceedings of the SIGCHI conference on Human Factors in computing systems, ACM, 2006, S. 955-958.
- [STD10] Storey, M.-A.; Treude, C.; van Deursen, Arie; Cheng, L.-T.: The Impact of Social Media on Software Engineering Practices and Tools. Wird erscheinen in: Proceedings of the FSE/SDP Workshop on the Future of Software Engineering Research. ACM, 2010.
- [Tr10] Treude, C.: The Role of Emergent Knowledge Structures in Collaborative Software Development. In: Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 2, ACM, 2010, S. 389-392.
- [TS10] Treude, C.; Storey, M.A.: Awareness 2.0: Staying Aware of Projects, Developers and Tasks using Dashboards and Feeds. In: Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1, ACM, 2010, 365-374.