Nunmehr zum achten Male liegt ein Sammelband zum Workshop „GeNeMe – Gemeinschaften in Neuen Medien“ vor, der Beiträge zu folgenden Themenfeldern enthält:

- Konzepte für GeNeMe (Geschäfts-, Betriebs- und Architektur-Modelle),
- IT-Unterstützung (Portale, Plattformen, Engines) von GeNeMe,
- E-Learning in GeNeMe,
- Wissensmanagement in GeNeMe,
- Anwendungen und Praxisbeispiele von GeNeMe und
- Soziologische, psychologische, personalwirtschaftliche, didaktische und rechtliche Aspekte von GeNeMe.

Sie wurden aus einem breiten Angebot interessanter und qualitativ hochwertiger Beiträge zu dieser Tagung ausgewählt.

Das Interesse am Thema GeNeMe (Virtuelle Unternehmen, Virtuelle Gemeinschaften etc.) und das Diskussionsangebot von Ergebnissen zu diesem Thema sind im Lichte dieser Tagung also ungebrochen und weiterhin sehr groß.

Die thematischen Schwerpunkte entsprechen aktuellen Arbeiten und Fragestellungen in der Forschung wie auch der Praxis. Dabei ist die explizite Diskussion von Geschäfts- und Betreibermodellen für GeNeMe, insbesondere bei der aktuellen gesamtwirtschaftlichen Lage, zeitgemäß und essentiell für ein Bestehen der Konzepte und Anwendungen für und in GeNeMe.

In zunehmendem Maße rücken weiterhin auch Fragen nach den Erfolgsfaktoren und deren Wechselbeziehungen zu soziologischen, psychologischen, personalwirtschaftlichen, didaktischen und rechtlichen Aspekten in den Mittelpunkt. Deshalb wurde hierzu ein entsprechender Schwerpunkt in der Tagung beibehalten.

Konzepte und Anwendungen für GeNeMe bilden entsprechend der Intention der Tagung auch weiterhin den traditionellen Kern und werden dem Anspruch auch in diesem Jahr gerecht.

Die Tagung richtet sich in gleichem Maße an Wissenschaftler wie auch Praktiker, die sich über den aktuellen Stand der Arbeiten auf dem Gebiet der GeNeMe informieren möchten.

www-mmt.inf.tu-dresden.de/geneme

ISBN: 3-86005-491-0

Meißner / Engelien • Virtuelle Organisation und Neue Medien 2005

TECHNISCHE UNIVERSITÄT DRESDEN

Klaus Meißner / Martin Engelien (Hrsg.)

Virtuelle Organisation und Neue Medien 2005

Workshop GeNeMe2005
Gemeinschaften in Neuen Medien

TU Dresden, 6./7.10.2005

TECHNISCHE UNIVERSITÄT DRESDEN

B.3 Vernetzung virtueller Gemeinschaften mit P2P-Technologien

Hans Friedrich Witschel[1], *Herwig Unger*[2]
[1]*Universität Leipzig*
[2]*Universität Rostock*

1. Einleitung

Mittelständische und große Unternehmen sehen sich durch die rasant fortschreitende Digitalisierung von Medien und Kommunikation mit dem Problem konfrontiert, ihre Daten konsistent und logisch strukturiert zu verwalten. Häufig wird dieses Problem durch verteilte Strukturen (Filialen, Zweigstellen, Zulieferer, mobile Mitarbeiter) noch verstärkt. Die häufig verwendeten zentralisierten Strukturen (Server, Datenbanken) sind zudem ein wesentlicher Angriffspunkt für die Systemsicherheit und erfordern großen Aufwand zu ihrer Wartung und Aktualisierung.

In diesem Papier soll ein neuartiges Recherchesystem für virtuelle Gemeinschaften vorgestellt werden, welches – aufbauend auf einer Peer-to-Peer-Technologie – einige wesentliche Mängel der eben genannten Ansätze behebt[1]:

- Das Einbringen von Inhalten in ein Content-Management-System kann langwierig und aufwendig sein: oft gelangen neue Dokumente nur nach einem Redaktionsprozess ins System. Das Einfügen eines Dokumentes in ein P2P-Netzwerk ist sehr viel unkomplizierter und erleichtert so das Einbringen „halboffizieller“ Information.
- Gemeinschaften sollten nicht statisch vernetzt sein: die im übernächsten Abschnitt vorgestellte Selbstorganisation des P2P-Netzes führt zu einer automatischen Bildung von Gemeinschaften: Menschen, die an ähnlichen Problemen arbeiten bzw. interessiert sind, werden automatisch auf Systemebene vernetzt. Dies kann zur Aufdeckung unbekannter Gemeinsamkeiten und somit zu neuen Synergien führen.
- Nebeneffekte der verteilten Architektur sind schließlich erhöhte Ausfallsicherheit und Kosteneinsparung bei der Wartung, da die Pflege zentraler Server entfällt.

Das vorgestellte System soll mehr sein als ein verteiltes Recherchetool: die automatische Selbstorganisation virtueller Gemeinschaften bildet einen wichtigen Teil des Vorhabens. Das System wird anhand von Simulationen getestet und die dabei erhaltenen Ergebnisse – unter anderem die Feststellung, dass sich die angestrebten

[1] Die Autoren wurden durch die DFG im Projekt Nr. 255712 gefördert

Strukturen tatsächlich ergeben und für die Suche unterstützend wirken – werden dargestellt.

2. Verwandte Ansätze

P2P-Architekturen wie z.B. Gnutella oder Freenet [3], [1] sind die Basis für unsere neue Systemarchitektur: Alle – für sich autonom arbeitenden *Peers* (oder auch *Servents* = Server *und* Client) halten Listen von Adressen anderer Peers – ihrer *Nachbarn* – vor und kontaktieren diese, wenn eine Suchanfrage an sie gerichtet wird. Die Systeme unterscheiden sich darin, wie die Nachbarschaften organisiert sind – sie können dynamisch (wie in [1]) oder fest (wie in [2]) sein. Inhalte können algorithmisch an bestimmte Knoten des Netzes gebunden sein, wie dies bei sogenannten Distributed Hash Tables (DHTs) [13],[9] der Fall ist oder – wie in allen anderen Fällen – an beliebigen Orten bzw. am Ort ihrer Erstellung gespeichert sein. Ein letzter wichtiger Unterschied äußert sich in der Weiterleitung von Nachrichten: In Systemen wie Gnutella [3] leitet ein Peer Suchanfragen jeweils an *alle* seine Nachbarn weiter (Broadcast); in Systemen mit informierter Suche kann hingegen eine inhaltliche Auswahl unter den Nachbarn getroffen werden. Dies reduziert die Anzahl verwendeter Nachrichten erheblich.

Die von uns angestrebte Architektur fällt in die Kategorie der *dynamischen* Netze mit Datenspeicherung am *Erstellungsort* und *informierter* Suche: um virtuelle Gemeinschaften zu erkennen und zu vernetzen, müssen Nachbarschaften flexibel sein und Peers über ihre Inhalte identifizierbar sein.

In der Literatur finden sich etliche Ansätze, die eine dynamische Strukturierung von P2P-Netzen aufgrund inhaltlicher Kriterien vorschlagen: Systeme wie Bibster [4] oder [12] verwenden Ontologien zur Erstellung von Peer-Profilen (Expertisen). In [4] finden sich Peers mit ähnlichen Expertisen in Clustern zusammen, in [12] wird hingegen eine echte Small-World-Struktur (s.u.) angestrebt.

In [5] wird – aufbauend auf Gnutella – ebenfalls ein Peer-Clustering nach Inhalten (*attractive links*) vorgenommen, zusätzlich zu den normalen Gnutella-Nachbarn (*random links*). Die Suche basiert weiterhin auf Broadcasting über random links, welches allerdings abbricht, wenn ein Peer-Profil einer Anfrage ähnlich genug ist; dann werden nur noch attractive links betrachtet.

Die in [14] und [7] beschriebenen Verfahren stammen aus dem Bereich der strukturierten Ansätze (DHTs). Beiden gemeinsam ist die Annahme eines k-dimensionalen Vektorraumes, in welchem Datenobjekte und Peers repräsentiert werden. Bezogen auf Dokumente bedeutet dies, dass Latent Semantic Indexing (LSI) verwendet werden muss, um die Dimensionalität handhabbar zu halten. In [14] werden die k

semantischen Koordinaten für den Aufbau eines DHT verwendet, die in [7] eingeführte *Semantic Small World* (SSW) erweitert dies um Intergroup-Nachbarn, erstellt inhaltliche Peer-Cluster und ordnet diese dann statt im k-dimensionalen Raum in einer eindimensionalen Liste an, um die Anzahl der Nachbarn pro Peer zu reduzieren. Die Position des Peers im Netz bleibt daraufhin starr, auch wenn sich dessen Bibliothek stark verändert.

Im Gegensatz zu allen erwähnten Ansätzen zielt unser Verfahren auf ein Maximum an Flexibilität: weder wird von der Existenz von Ontologien ausgegangen (welche oft schwer zu beschaffen sind), noch wird ein starrer k-dimensionaler Raum oder starre Positionierung von Peers in einem solchen angenommen. Auch das Fluten des Netzes mit Anfragen wird durch unseren Ansatz komplett vermieden.

Stattdessen versucht unser Ansatz, auf der Basis des lokalen Nutzerverhaltens ein *adaptives* Profil zu erstellen, durch das eine Optimierung von Nachbarschaften nach dem Small-World-Kriterium (s.u.) erreicht werden soll. Die vorliegende Publikation untersucht die Praktikabilität eines solchen Ansatzes.

3. Small Worlds

Die von uns angestrebte Selbstorganisation der virtuellen Peer-Gemeinschaft stützt sich auf Strukturen, wie sie in vielen realen selbstorganisierenden Systemen, insbesondere in der Gesellschaft, entdeckt wurden.

Bereits in den 60er Jahren führte Stanley Milgram [8] Experimente zur Untersuchung der sozialen Vernetzung der Gesellschaft durch, indem er Versuchspersonen Briefe gab, mit dem Auftrag, sie an eine Zielperson weiterzuleiten. Name, Beruf und Wohnort des Adressaten waren dabei bekannt, nicht jedoch seine genaue Adresse. Die Versuchspersonen sollten nun unter ihren persönlichen Bekannten denjenigen auswählen, von dem sie annahmen, dass er den Brief am weitesten in Richtung der Zielperson bringen könnte. Die Ergebnisse der Studie – seitdem bekannt als *six degrees of separation* – ergaben, dass diejenigen Briefe, die ihr Ziel erreichten, im Mittel nur über 5 Mittelsmänner weitergeleitet wurden, d.h. nach 6 Stationen ihr Ziel erreichten.

Die Ideen von Milgram und die von ihm aufgedeckten Strukturen wurden erst sehr viel später (in den 90er Jahren) wieder aufgegriffen und graphentheoretisch untersucht. Watts und Strogatz [15] gaben als erste eine Definition für den Begriff *Small World* an: eine Small World nach Watts/Strogatz ist ein Graph G=(V,E) mit hohem Clustering-Koeffizienten und kurzer mittlerer Weglänge.

- Die Weglänge für ein beliebiges Paar (u,v) von Knoten des Graphen ist dabei definiert als die Länge eines kürzesten Weges zwischen u und v, die mittlere

Weglänge L(G) des Graphen als arithmetisches Mittel über die Weglängen zwischen allen Knotenpaaren (u,v).

- Der (lokale) Clustering-Koeffizient C_v eines Knotens v in einem gerichteten Graphen G ist definiert als die Anzahl vorhandener Kanten zwischen Knoten aus der Nachbarschaft N_v des Knotens v, geteilt durch die Anzahl der möglichen Kanten innerhalb dieser Nachbarschaft. Er kann als die Wahrscheinlichkeit dafür gedeutet werden, dass zwei Nachbarn eines Knotens v selbst wieder durch eine Kante verbunden sind. Der Clustering-Koeffizient C(G) des gesamten Graphen ist das arithmetische Mittel aller Werte C_v.

Watts und Strogatz wiesen die Small-World-Eigenschaft für eine Reihe realer Graphen nach. Sie gaben daraufhin ein Modell, also eine Vorschrift zur Erzeugung solcher Graphen, an. Dieses Modell beschreibt Small Worlds als Graphen, in welchen lokale Cluster von Knoten existieren, welche wiederum lose – über sogenannte *random shortcuts* – miteinander verbunden sind.

In unserem Ansatz sollen Peers anhand ähnlicher Inhalte geclustert werden und diese Cluster sollen wiederum lose vernetzt sein, um kurze Wege zu garantieren.

Kleinberg weist in seiner Arbeit [6] darauf hin, dass die Existenz kurzer Wege allein nicht ausreicht, um den Erfolg eines Suchalgorithmus zu garantieren: die Knoten müssen auch in der Lage sein, aufgrund sogenannter „latent navigational clues“ die richtigen Nachbarn auszuwählen, d.h. die kurzen Wege auch zu finden.

Die von uns gewählten Hinweise sind weniger versteckt: jeder Peer des Netzes wird mit einem sogenannten „Profil“ ausgestattet, d.h. einer kompakten Zusammenfassung der Inhalte, die er anbietet. Ein Peer kennt nun nicht nur die Adresse seiner Nachbarknoten, sondern auch deren Profil.

Aufbauend auf einem Small-World-Netzgraphen und den durch die Profile gegebenen „navigational clues“ lässt sich ein Suchalgorithmus definieren, welcher direkt auf Milgrams Experiment aufbaut: wenn ein Peer P eine Anfrage nach einer bestimmten Information erhält, prüft er zunächst, ob er die Anfrage selbst beantworten kann. Falls nicht, leitet er die Nachricht an denjenigen seiner Nachbarn weiter, dessen Profil am besten zur Anfrage passt. Dies wird fortgesetzt, bis die time-to-live (TTL) der Nachricht erschöpft ist. Aufgrund der oben angedeuteten Struktur der Small World kann diese Suche als eine Art Hill Climbing interpretiert werden: die Nachricht bewegt sich hin zu Peers, die immer besser zur Anfrage passen; ist der richtige Cluster gefunden, so kann dieser aufgrund der starken Vernetzung innerhalb von Clustern schnell abgesucht werden. Im Folgenden sollen die Algorithmen genauer beschrieben werden, welche zur Selbstorganisation des Netzes und dann zur Suche darauf eingesetzt werden.

4. Selbstorganisation und Suche

4.1 Definitionen

Von nun an werden wir davon ausgehen, dass sich der Zustand eines Peers vollständig beschreiben lässt durch folgende drei Komponenten:

- Eine Menge von Dokumenten, die er für den Rest des Netzwerkes freigibt. Wir nennen diese Menge seine *Bibliothek*.
- Ein *Profil*, welches die Inhalte seiner Bibliothek knapp zusammenfasst.
- Eine Menge von *Nachbarn*, d.h. Adressen und Profile einiger anderer Peers. Der so entstehende Netzwerkgraph ist gerichtet.

Die Darstellung der Inhalte von Dokumenten, Profilen und Anfragen erfolgt mit Hilfe des Vektorraum-Modells des Information Retrieval, d.h. die Objekte werden mittels Vektoren dargestellt, welche aus Gewichten für die enthaltenen Schlüsselbegriffe bestehen.

Auf die genaue Implementierung kann hier nicht eingegangen werden, wichtig ist jedoch zu erwähnen, dass Profile niemals durch alle Schlüsselwörter charakterisiert werden können, die in den Dokumenten der Bibliothek auftreten: eine solche Darstellung wäre viel zu umfangreich, um sie in Nachrichten zu verschicken. Algorithmen zur Auswahl nur der signifikantesten Schlüsselwörter werden ebenfalls innerhalb unseres Projektes entwickelt und sind z.B. in [16] beschrieben.

Der prinzipielle Ablauf zur Erstellung von Profilen ist jedoch das Summieren der Dokumentvektoren $\mathbf{d} = (w_1,\ldots,w_n)$. Einträge mit kleinen Gewichten können dann nachträglich abgeschnitten werden, um zur gewünschten Profilgröße zu gelangen.

Wir nehmen an, dass Dokumentvektoren summennormiert sind. Profile hingegen werden nicht normiert, was dazu führt, dass Peers mit sehr großer Bibliothek generell größere Gewichte im Profil aufweisen als Peers mit wenigen Dokumenten. Die Ähnlichkeit sim(Q,D) zweier Vektoren Q und D wird durch das einfache Skalarprodukt beider Vektoren berechnet.

Im Folgenden sollen die zwei Algorithmen beschrieben werden, aus denen unser Recherchesystem besteht: die Selbstorganisation bzw. *Strukturbildung*, die eine Small World Struktur im P2P-Netz erzeugt und aufrechterhält und sie *Suche*, welche diese Struktur zum Auffinden von Daten nutzt. Da die Strukturbildung das Verfahren der Suche benutzt, soll letztere zuerst beschrieben werden.

4.2 Suche

Erhält ein Peer P eine Anfrage Q, so durchsucht er zunächst seine eigene Bibliothek nach Dokumenten, welche zu Q passen. Falls dies erfolgreich ist, werden die Vektoren (also Schlüsselwortbeschreibungen) der Dokumente an Q angehängt. P wählt sodann

denjenigen seiner Nachbarn aus, dessen Profil Q am ähnlichsten ist und leitet die Anfrage an ihn weiter. Dies geschieht solange, bis die TTL der Nachricht abgelaufen ist, woraufhin die Anfrage direkt zu P zurückgeleitet wird. Um Kreise zu vermeiden, trägt außerdem jeder Peer seine Adresse in das *Log* der Nachricht ein: Q wird im folgenden nicht an Peers weitergeleitet, welche bereits im *Log* enthalten sind.

4.3 Strukturbildung

Die Small-World-Struktur wird mittels eines sogenannten *Gossiping*-Verfahrens erzeugt. Dabei stellt jeder Peer P periodisch Anfragen nach seinem eigenen Profil, welche mit Hilfe des soeben beschriebenen Suchverfahrens verarbeitet werden. Der einzige Unterschied besteht darin, dass im Falle des Gossipings Peers zusätzlich zu ihrer Adresse auch ihr Profil in das *Log* der Nachricht eintragen. Erhält P nun die Antwort auf seine Anfrage, so kann er die Einträge des *Logs* inspizieren und sich evtl. neue Nachbarn wählen. Damit auch andere Peers von P's Existenz erfahren können, kann jeder Peer, welcher die Gossiping-Nachricht weiterleitet, ebenfalls deren *Log* inspizieren. Zur Auswahl neuer Nachbarn werden zwei Strategien verwendet:

- ***Cluster-Strategie:*** ein Peer wählt Nachbarn, deren Profile seinem eigenen möglichst ähnlich sind. Diese Strategie trägt zur Bildung semantischer Ähnlichkeitscluster bei.
- ***Intergroup-Strategie:*** ein Peer wählt Nachbarn, deren Profile seinem eigenen möglichst unähnlich sind. Dies schafft die *random shortcuts* zwischen den Clustern.

Die mittels beider Strategien gefundenen Nachbarn werden getrennt verwaltet, d.h. es gibt eine Menge von „Cluster-Nachbarn" und eine Menge von „Intergroup-Nachbarn". Die Größe der beiden Mengen kann nun so eingestellt werden, dass sich die gewünschte Struktur ergibt.

Die Cluster-Nachbarn eines Peers können für den menschlichen Betrachter von großem Interesse sein, da es sich bei den Betreibern der Nachbar-Peers oft um Menschen handelt, welche an Fragestellungen arbeiten, die dem eigenen Arbeitsgebiet ähnlich sind. Es kann sich also lohnen, diese offenzulegen, sodass sich virtuelle Gemeinschaften ähnlicher Peers nicht nur auf Systemebene, sondern auch für Menschen sichtbar bilden – evtl. sogar zu deren Überraschung.

4.4 Caching

Wie bei der normalen Suche wird auch beim Gossiping auf jedem Peer nach passenden Dokumenten gesucht und deren Vektoren werden an die Anfrage angehängt. Das bedeutet, dass der Peer, welcher nach seinem eigenen Profil gefragt hat, eine Liste von Dokumenten als Antwort erhält, welche gut zu seinem Profil passen. Ist auf einem Peer genug Speicherplatz vorhanden, so kann der Benutzer die Speicherung dieser Vektoren

(oder evtl. auch der Vollversionen) zulassen. Dies ist einerseits für den Benutzer interessant – als eine Art automatischer Literaturrecherche, welche seinen Dokumentenbestand semantisch homogen erweitert. Andererseits wird der Peer durch die Einbeziehung des neuen Wissens mehr Anfragen zu seinem Spezialgebiet beantworten können.

5. Simulationsergebnisse

5.1 Vorbereitungen

Um die prinzipielle Arbeitsweise der Algorithmen evaluieren zu können, implementierten wir eine Simulationsumgebung mit Hilfe des Netzwerksimulators OMNeT++[2]. Die Parameter der Simulation und ihre Ergebnisse sollen im Folgenden detailliert dargestellt werden.

Vereinfachende Annahmen

Um die Komplexität des Problems beherrschbar zu machen, mussten zunächst einige vereinfachende Annahmen gemacht werden:

- Statt reale Dokumente zu verwenden, gingen wir von der Existenz künstlicher semantischer Kategorien aus. Diese dienten dazu, (wiederum künstliche) Dokumentobjekte zu beschreiben, sodass ein Dokument nun nicht mehr durch einen Vektor von Termen, sondern durch einen Vektor von Kategorien $\mathbf{D} = (c_1,\ldots,c_n)$ dargestellt wird.
- Weiterhin vernachlässigten wir die Tatsache, dass sich in allen realen P2P-Systemen laufend Peers an- bzw. abmelden. Obwohl dies sehr unrealistisch ist, kann man doch davon ausgehen, dass Ergebnisse, die unter statischen Bedingungen zu beobachten sind, sich auch in dynamischen Peer-Populationen reproduzieren (vgl. [10])
- Schließlich nahmen wir an, dass jeder Peer mindestens ein Dokument in seiner Bibliothek hat. Messungen in [11] haben zwar ergeben, dass in realen P2P-Netzen 25% der Teilnehmer sogenannte *Free Riders* sind, also keine Dokumente anbieten. Diese Peers sind jedoch für unsere Strukturbildung ungeeignet, werden also zunächst ausgeblendet.

Simulationsmodi

Die Simulation wurde in zwei getrennten Phasen durchgeführt: zunächst wurde im *Strukturierungsmodus* solange Gossiping durchgeführt, bis sich die Netzwerktopologie stabilisiert hatte. Diese wurde dann im *Suchmodus* benutzt: 100 zufällig gewählte Peers

[2] http://www.omnetpp.org

sandten jeweils Anfragen nach allen semantischen Kategorien c_i aus und die gefundenen Dokumente wurden mit vorberechneten Ergebnismengen verglichen.

Simulationsparameter

In Tabelle 1 sind die wichtigsten Parameter der Simulationen dargestellt. Jeder Peer in unserer Simulation verfügt über begrenzten Speicherplatz für Nachbarn und Dokumente: in allen Durchläufen konnte Information über nur 20 Nachbarn vorgehalten werden. Im Lauf 1 waren diese nur nach der Cluster-Strategie gewählt, in den anderen Läufen gemischt nach beiden Strategien.
Der Parameter „storage factor“ gibt an, wie viele fremde bzw. neue Dokumente ein Peer speichern kann (s. Abschnitt 4.4): die erlaubte Größe des Dokumentencaches ist proportional zur Anfangsgröße |B| der Bibliothek; man erhält sie, indem man |B| mit dem *storage factor* multipliziert. Das Erweitern des Caches war nur in Lauf 3 erlaubt: hier konnte ein Peer bis zum Fünffachen der initialen Menge an Dokumenten speichern.

Parameter	Lauf 1	Lauf 2	Lauf 2	Zufallsgraph
# Peers	8000	8000	8000	8000
# Cluster-Nachbarn	20	14	14	-
# Intergroup-Nachbarn	0	6	6	-
Storage factor	1	1	5	-
Anzahl Dokumente	10.000	10.000	10.000	10.000
Anzahl sem. Kategorien	50	50	50	50
TTL f. Gossiping-Nachrichten	25	25	25	-

Tabelle 1: Parameter der Simulation

Initialisierung

Um das Netzwerk zu initialisieren, erhielt jeder der 8000 Peers in den Simulationsläufen 1 bis 3 zunächst drei initiale, zufällig gewählte Nachbarn, sodass sich anfangs ein Zufallsgraph ergab. Dieser war nicht stark zusammenhängend, sondern hatte eine große starke Komponente aus 7556 Peers; die restlichen 444 Peers waren zunächst isoliert.
In einem nächsten Schritt wurden künstliche Dokumentvektoren erzeugt und auf die Peers verteilt. Dabei gingen wir von zwei Annahmen aus:

- Der Benutzer eines Peers hat normalerweise bestimmte Interessen, die sich in seinen Dokumenten widerspiegeln. Jeder Peer wählte sich also zunächst ein bis drei Kategorien c_i, welche seine Interessen repräsentierten.
- Messungen in [11] haben gezeigt, dass die Anzahl von Dokumenten pro Peer in realen P2P-Systemen keineswegs gleichmäßig verteilt ist, sondern ungefähr einer

Zipf-Verteilung folgt. Daher wurde für jeden Peer die Größe seiner Bibliothek |B| so festgelegt, dass sich eine Zipf-Verteilung über die Peers ergab. Nun erhielt der jeweilige Peer |B| Dokumente, welche aus den vorher für diesen Peer gewählten Kategorien stammten.

Schließlich erzeugten wir einen Zufallsgraphen, in welchem jeder Peer dieselben Dokumente besaß wie in den Läufen 1 bis 3. Peers hatten hier jedoch 20 zufällig gewählte Nachbarn. Der Zufallsgraph wird später dazu dienen, herauszufinden, ob die Small-World-Strukturen, die sich in den Läufen 1 bis 3 einstellen sollen, tatsächlich eine Erleichterung für die Suche darstellen.

5.2 Ergebnisse

5.2.1 Graphanalyse

Die Kenngrößen der sich durch Gossiping ergebenden Graphen sind in Tabelle 2 dargestellt. Es muss hierbei noch erwähnt werden, dass für die Berechnung der mittleren Weglängen jeweils nur Knotenpaare (A,B) berücksichtigt wurden, für die ein Weg zwischen A und B *existiert*.

	Cluster-Koeffizient	Mittlere Weglänge	# Komponenten	Größe der größten Komponente
Lauf 1	0,56	3,7	6829	1168
Lauf 2	0,34	4,3	135	7865
Lauf 3	0,31	4,2	135	7865
Zufallsgraph	0,0024	3,3	1	8000

Tabelle 2: Eigenschaften der Netzwerkgraphen

Folgende Beobachtungen ergeben sich aus diesen Daten:

- Im Lauf 1 „zerbricht“ der Graph, d.h. die meisten Knoten sind am Ende der Strukturbildung isoliert. In den Läufen 2 und 3 hingegen kann die Anzahl der Komponenten von anfangs 445 auf 135 reduziert werden. Man sieht also, dass die Einführung von Intergroup-Nachbarn wichtig für den Zusammenhalt des Netzes ist.
- Die Clusterkoeffizienten sind in allen Durchläufen wesentlich höher als im Zufallsgraphen, die mittlere Weglänge hingegen nur unwesentlich. Es stellen sich also tatsächlich Small-World-Strukturen ein.

Abbildung 1 zeigt die Verteilung der Knoteneingangsgrade für die einzelnen Graphen. Für die Läufe 1 bis 3 stellt sich annähernd eine Zipf-Verteilung ein, d.h. es gibt einzelne

Knoten, die Nachbarn sehr vieler anderer sind, während die meisten Knoten nur relativ wenige eingehende Kanten haben.

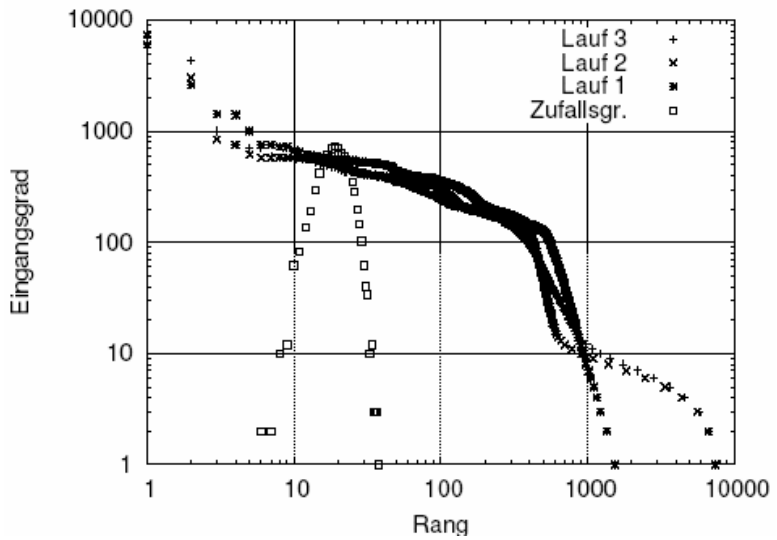

Abbildung 1: Verteilung der Knotengrade

Eine Korrelationsanalyse für Bibliotheksgröße |B| und Eingangsgrad k_{in} zeigte, dass zwar keine direkte Korrelation besteht, immerhin aber eine deutliche Tendenz erkennbar ist: Peers mit sehr vielen Dokumenten haben hohen Eingangsgrad. Die Verteilung der Eingangsgrade ist also unter anderem eine Konsequenz der Zipf-Verteilung von Dokumenten auf Peers und der in Abschnitt 4.1 beschriebenen Berechnung von Profilen und Ähnlichkeiten: Peers mit sehr vielen Dokumenten werden als Nachbarn bevorzugt, da ihre Profile große Gewichte für viele Kategorien enthalten.
Dies erscheint zunächst ungewollt, da Peers gleichberechtigt sein sollen; Messungen an P2P-Netzen (vgl. [11]) haben jedoch ergeben, dass Zipf-Verteilungen für Knotengrade Teil der Realität sind.

5.2.2 Recall

In der zweiten Phase der Simulation wurden 100 Peers zufällig ausgewählt, welche jeweils nacheinander Anfragen nach allen 50 Kategorien generierten. Die Ergebnisse dieser Suchen wurden mittels des Recalls

$$R = \frac{\#\,im\,P2P-Netz\,gefundene\,Dokumente}{\#\,im\,zentralen\,Index\,gefundene\,Dokumente}$$

evaluiert. Ein Dokument D wurde dabei bezüglich einer Anfrage Q als relevant eingestuft gdw. sim(Q,D)>0,5 galt. In allen vier Netzwerkgraphen – also auch im

Zufallsgraphen – wurde zur Suche der in Abschnitt 4.2 beschriebene Algorithmus verwendet. Abbildung 2 zeigt den Recall als Funktion der TTL.

Drei interessante Ergebnisse lassen sich ablesen:

- In den Läufen 1-3 konvergiert der Recall recht schnell (nach ca. 20 Hops) und nimmt dann kaum noch zu. Im Zufallsgraphen wächst er zwar linear, liegt aber auch nach 50 Hops noch weit unter dem Niveau der anderen Durchläufe. Small-World-Strukturen helfen also offensichtlich wirklich bei der Suche.
- Der Effekt des Dokumentencachings ist überraschend groß: die Ergebnisse in Lauf 3 sind bis zu 30% besser als in den anderen Läufen.
- Ebenfalls überraschend ist der geringe Unterschied zwischen Lauf 1 und 2: obwohl der Netzwerkgraph in Lauf 1 stark zerfällt, findet der Suchalgorithmus in etwa genauso viele Dokumente wie im Falle des durch Intergroup-Nachbarn zusammengehaltenen Graphen. Dies liegt vermutlich daran, dass die in Lauf 1 verbleibende Komponente aus 1168 Peers aus „Autoritäten“ besteht, d.h. aus Peers mit großen Bibliotheken und hohem Eingangsgrad. Diese zuerst zu besuchen, deckt offenbar bereits einen großen Teil des Erreichbaren ab.

Obwohl der Recall insgesamt nicht perfekt ist, kann doch – zumindest im Falle des Cachings von Dokumentvektoren – ein wesentlicher Teil der relevanten Dokumente nach nicht mehr als 20 Hops gefunden werden.

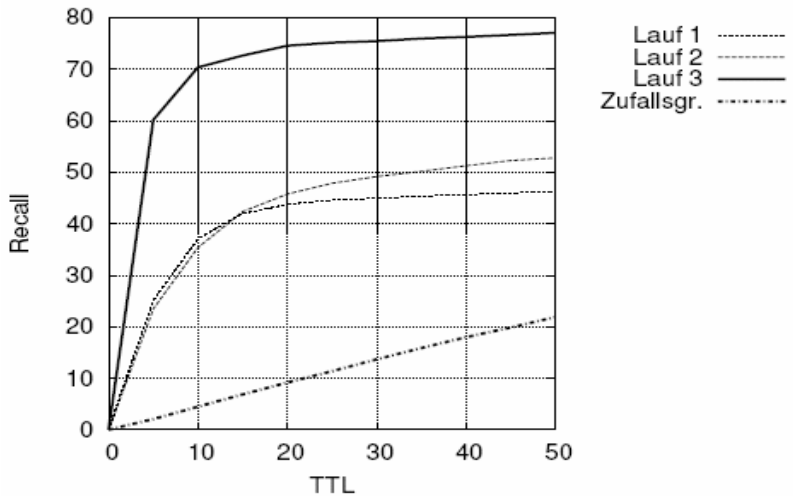

Abbildung 2: Recall als Funktion der Anzahl besuchter Peers (TTL)

6. Ausblick

Innerhalb unseres Projektes versuchen wir momentan, die Simulation Stück für Stück realistischer zu gestalten durch

- die Verwendung realer Daten (Dokumente)

- die Einbeziehung sich an- und abmeldender Peers
- die Untersuchung der Skalierbarkeit der oben vorgestellten Ergebnisse.

Was den zweiten dieser Punkte betrifft, so ist vorgesehen, die „Qualität“ von Nachbarn mit der Zeit abnehmen zu lassen, so dass Peers, die häufig online sind, mit der Zeit als Nachbarn bevorzugt werden.

Insgesamt stimmen uns die erhaltenen Ergebnisse jedoch bereits so optimistisch, dass wir hoffen, die Technologien bald auch in der Praxis einsetzen und testen zu können.

Literatur

[1] I. Clarke et al. (2001): Freenet: A Distributed Anonymous Information Storage and Retrieval System. *Lecture Notes in Computer Science*, 2009:46+, 2001.

[2] A. Crespo und H. Garcia-Molina (2002): Routing indices for peer-to-peer systems. *Proc. of the 28 th Conference on Distributed Computing Systems*.

[3] Gnutella. www.gnutella.com

[4] J. Broekstra et al. (2004): Bibster – A Semantics-Based Bibliographic Peer-to-Peer System. *Proc. of SemPGRID'04*, S. 3–22.

[5] I. King, C. H. Ng, K. C. Sia (2004): Distributed content-based visual information retrieval system on peer-to-peer networks. *ACM Transactions on Information Systems*, 22(3), S. 477–501.

[6] J. Kleinberg (2000): The Small-World Phenomenon: An Algorithmic Perspective. *Proc. of the 32nd ACM Symposium on Theory of Computing*.

[7] M. Li, W.-C. Lee, A. Sivasubramaniam (2004): Semantic Small World: An Overlay Network for Peer-to-Peer Search. *Proc. of the International Conference on Network Protocols (ICNP)*, 228-238.

[8] S. Milgram (1967): The small world problem. *Psychology Today*, 1(1):60–67, 1967.

[9] S. Ratnasamy et al. (2001). A Scalable Content Addressable Network. *Proc. of the ACM SIGCOMM.*

[10] G. Sakaryan (2004): A Content-Oriented Approach to Topology Evolution and Search in Peer-to-Peer Systems. PhD thesis, University of Rostock.

[11] S. Saroiu, P. Gummadi, und S. Gribble (2002): A Measurement Study of Peer-to-Peer File Sharing Systems. *Proc. of Multimedia Computing and Networking*.

[12] C. Schmitz (2005): Self-Organization of a Small World by Topic. *Proc. of 1st International Workshop on Peer-to-Peer Knowledge Management.*

[13] I. Stoica et al. (2001):. Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications. *Proc. Of ACM SIGCOMM.*

[14] C. Tang, Z. Xu, S. Dwarkadas (2003): Peer-to-peer information retrieval using self-organizing semantic overlay networks. *Proc. of ACM SIGCOMM*, S. 175–186.

[15] D. Watts und S. Strogatz (1998): Collective Dynamics of 'Small-World' Networks. *Nature*, 393(6):440–442.

[16] H.F. Witschel (2005): Terminology Extraction and Automatic Indexing – Comparison and Qualitative Evaluation of Methods. *Proc. of TKE*. [to appear]