DIPLOMARBEIT

Semalan
Semantic Mailing List Analyzer

von

Markus Göbel

eingereicht am 10.01.2006 beim
Institut für Angewandte Informatik
und Formale Beschreibungsverfahren
der Universität Karlsruhe

Referent: Prof. Dr. Rudi Studer
Betreuer: Dipl.-Inform. Max Völkel

Heimatanschrift & Studienanschrift:
Markus Göbel
Bachstraße 49
76669 Bad Schönborn

Semalan

Semantic Mailing List Analyzer

Augmenting Email
by Discussion Model Extraction

Diplomarbeit

Institut für
Angewandte Informatik und Formale Beschreibungsverfahren
Lehrstuhl Studer
Fakultät für Wirtschaftswissenschaften
Universität Karlsruhe (TH)

Markus Göbel

Betreuer:
Dipl.-Inform. Max Völkel

verantwortlicher Betreuer:
Prof. Dr. Rudi Studer

Tag der Anmeldung: 08.07.2005
Tag der Abgabe: 10.01.2006

Abstract: Diese Arbeit stellt ein System namens Semalan vor, mit dem per E-Mail geführte Diskussionen auf ein strukturiertes Diskussionsmodell abgebildet werden können, das dem Benutzer über Schnittstellen zugänglich gemacht wird. Damit soll es möglich sein, sich schneller einen Überblick über relevante Teile einer Diskussion zu verschaffen, als dies bei heute üblichen Darstellungsweisen von E-Mail-Diskussionen der Fall ist. In der Arbeit wird zunächst analysiert, inwiefern sich E-Mails aus Mailinglisten zum Erstellen eines Diskussionsmodells eignen. Es wird ein Diskussionsmodell entworfen, das eine feingranulare Darstellung einer E-Mail-Diskussion ermöglicht und das semantische Annotationen von Nachrichten unterstützt. Eine Architektur wird erstellt und geeignete Algorithmen werden gewählt, um E-Mails auf das Diskussionsmodell abzubilden. Es wird eine Abfragesprache, eine REST-artige Schnittstelle und eine grafische Benutzungsoberfläche entworfen, die einen Zugriff auf das Diskussionsmodell gestatten. Ein Prototyp dieser Architektur wird im Rahmen der Arbeit in Java implementiert.

Inhaltsverzeichnis

Danksagung

An dieser Stelle möchte ich mich bei den Personen herzlich bedanken, die mich bei der Erstellung dieser Diplomarbeit unterstützt haben.

Prof. Dr. Rudi Studer und insbesondere Dipl.-Inform. Max Völkel danke ich für die Bereitstellung des interessanten Diplomarbeitsthemas und für die Betreuung meiner Diplomarbeit.

Allen Freunden und Komilitonen möchte ich danken, die durch Korrekturlesen oder Testen der entwickelten Software zum Gelingen dieser Arbeit beigetragen haben.

Nicht zuletzt möchte ich meinen Eltern und meiner Freundin für die entgegengebrachte Geduld und Unterstützung danken.

1 Einleitung

Motivation E-Mails stellen in Form von Mailinglisten und anderen strukturell verwandten Nachrichtenformaten wie Newsgroups ein wichtiges Medium für Diskussionen im Internet dar, das seit vielen Jahren in weitgehend unveränderter Form existiert. Obwohl sie in einigen Bereichen in den vergangenen Jahren teilweise durch neue Kommunikationsformen wie Webforen, Blogs, Wikis und Instant-Messaging ersetzt wurden, sind Mailinglisten und Newsgroups, insbesondere im technischen und wissenschaftlichen Bereich, immer noch weit verbreitet. Davon zeugen große Mailinglisten-Anbieter wie *Domeus* mit 11 Millionen registrierten Benutzern oder *Yahoo Groups* mit mehr als 40 Millionen Benutzern und *Sourceforge* mit Mailinglisten für über 100.000 Softwareprojekte. Allein für die derzeit populärste Software zur Verwaltung von Mailinglisten, Mailman, lassen sich per *Google* Suche über 2,5 Millionen damit betriebene Mailinglisten finden[1]. Ähnlich populär ist das Usenet mit über 100.000 Newsgroups und mehr als 1 Milliarde bei *Google Groups* archivierter Newsgroupbeiträge.

Im Gegensatz zu herkömmlichen, mündlich geführten Diskussionen wird der Diskussionsverlauf in diesen Medien dadurch geprägt, dass die Diskussion nicht in Echtzeit stattfindet und durch die Möglichkeit, in einer Nachricht Textabschnitte aus verschiedenen vorangehenden Nachrichten zu zitieren und zu kommentieren. Dies führt häufig dazu, dass sich Diskussionen in mehrere parallele Diskussionsstränge verzweigen, die verschiedene Aspekte des ursprünglichen Diskussionsthemas behandeln oder von diesem auch vollständig abweichen können. Das Lesen von Nachrichten in einer Diskussion, die Zitate aus verschiedenen vorhergehenden Nachrichten enthalten, ist mühsam, da der Leser selbst gedanklich die Zuordnung von jedem Zitat zu der Nachricht aus der es zitiert wurde vornehmen muss. Auch muss durch die fehlende Feinstrukturierung von E-Mails und Newsgroupbeiträgen immer die komplette Nachricht gelesen werden um festzustellen, ob sich eine Nachricht auf eine bestimmte andere Nachricht bezieht. Insbesondere für neue Diskussionsteilnehmer und bei Diskussionen mit vielen Teilnehmern ergibt sich das Problem, dass sie erst viel Zeit zum Lesen aller vorherigen Nachrichten und Diskussionsstränge aufwenden müssen, um sich ein Bild über den bisherigen Verlauf und den aktuellen Stand einer Diskussion zu verschaffen. Dabei müssen auch die Nachrichten gelesen werden, die wenig zur Diskussion beigetragen haben und die im Verlauf der Diskussion keine weitere Rolle spielen. Denn der Leser kann

[1] http://www.google.de/search?q=inurl:/mailman/listinfo/

erst dann beurteilen, welches die zentralen und wichtigen Nachrichten einer Diskussion sind, wenn er sich einen Überblick über den Diskussionsverlauf verschafft hat. Dass dies oft nicht leicht ist, zeigen Mailinglisten wie die *Linux-Kernel* Mailingliste mit mehreren hundert E-Mails pro Tag und oft Dutzenden E-Mails zu einem Diskussionsthema.

Wenn ein Teilnehmer einer Mailingliste aber nur unzureichend über den aktuellen Stand einer Diskussion informiert ist, reduziert dies den Nutzen der Mailingliste in zweifacher Hinsicht erheblich. Zum einen erfüllt die Mailingliste für denjenigen nicht mehr ihren Hauptzweck, der im Gedanken- und Informationsaustausch mit anderen Teilnehmern besteht. Zum anderen kann dies dazu führen, dass er redundante oder nicht zum Thema passende Nachrichten verfasst, die den Ablauf der Diskussion für die anderen Diskussionsteilnehmer stören. In Newsgroups und Mailinglisten häufig zu lesende Bitten, sich vor dem Schreiben eigener Nachrichten über den Stand der Diskussion zu informieren, sowie das Vorhandensein von entsprechenden Regeln in der sogenannten Netikette, einem unverbindlichen Verhaltenskodex für Mailinglisten und Newsgroups[2] zeigen, dass in der Praxis nicht über den Diskussionsverlauf informierte Diskussionsteilnehmer in der Tat ein Problem darstellen.

Ähnlich schwierig ist es, sich über den Verlauf einer bereits abgeschlossenen Diskussion zu informieren. Denn auch hierfür müssen zunächst sowohl alle relevanten, als auch alle nicht relevanten Nachrichten gesichtet werden, bevor der Kern einer Diskussion ersichtlich ist.

Derzeitige E-Mail-Programme und Newsreader unterstützen den Nutzer nur unzureichend darin, sich einen Überblick über eine Diskussion zu verschaffen und für die Diskussion wichtige Nachrichten zu identifizieren. Sie verknüpfen Nachrichten nur aufgrund äußerer Merkmale wie Betreffzeile, Datum, Absender oder direkter Antwort-Relationen und lassen den tatsächlichen Inhalt der Nachrichten dabei außer Acht. Daher wird auch oft die fehlende inhaltliche Feinstrukturierung als Nachteil von Mailinglisten genannt[3].

Insbesondere können die E-Mail-Programme keine Antworten auf folgende Fragen liefern, deren Beantwortung einen Einblick in inhaltliche Zusammenhänge einer Diskussion geben kann:

- Wie oft, und von welchen anderen Nachrichten wird eine Nachricht zitiert?
- Welche anderen Nachrichten zitiert eine Nachricht?
- Welche Textabschnitte werden in einer Gruppe von Nachrichten besonders häufig zitiert?

[2] http://www.netplanet.org/netiquette/netnews.shtml

[3] http://www.opentheory.org/mailinglisten/text.phtml?par=45
http://www.opentheory.org/mailinglisten/text.phtml?par=56
http://www.opentheory.org/mailinglisten/text.phtml?par=123
http://www.itservices.ubc.ca/services/internet/access/newsmail1/newsmail3.html

- Welche anderen Nachrichten haben denselben Textabschnitt zitiert?
- Wie wurde ein Textabschnitt oder eine Abstimmung von anderen Diskussionsteilnehmern kommentiert, beziehungsweise beantwortet?

Dass eine Möglichkeit benötigt wird, sich auf einfache Art und Weise einen Überblick über Diskussionen in Mailinglisten zu verschaffen, zeigt auch die Existenz von Diensten wie kerneltraffic.org[4]. Dort werden der Verlauf und das Ergebnis der wichtigsten Diskussionen in der *Linux-Kernel* Mailingliste manuell von mehreren Autoren zusammengefasst.

Bei der gegenwärtigen Nutzung von Mailinglisten und Newsgroups ebenfalls nur unzureichend vorhanden sind Möglichkeiten automatisch zu ermitteln, welche Meinungen in einer Diskussionsgruppe zu einem Thema vorherrschen, sowie Abstimmungen und Ideensammlungen durchzuführen. Diese Informationen sind zwar meist implizit in den einzelnen Nachrichten vorhanden, lassen sich aber aufgrund der noch begrenzten Möglichkeiten des maschinellen Textverständnisses nur schwer daraus extrahieren.

Idee Abbildung 1.1 illustriert die Kernidee dieser Arbeit die darin besteht, die E-Mail einer Mailingliste in inhaltlich zusammengehörige und aufeinander Bezug nehmende Textabschnitte zu unterteilen und daraus ein Modell der über die Mailingliste geführten Diskussionen zu erstellen. Dabei soll sich zur Identifikation dieser Textabschnitte das in Mailinglisten übliche Zitieren aus anderen E-Mails zunutze gemacht werden.

Das Diskussionsmodell soll den Nutzern über eine einfach zu bedienende Benutzungsschnittstelle zugänglich gemacht werden.

Mit Hilfe des Diskussionsmodells sollen zentrale, besonders intensiv diskutierte Nachrichten und Textabschnitte einer Diskussion schneller erkannt werden können. Zudem soll dadurch der Zugang zu Gruppen thematisch zusammengehöriger Nachrichten innerhalb einer Diskussion erleichtert werden.

Zusätzlich sollen die Diskussionsteilnehmer Textabschnitte zwecks Durchführung von Abstimmungen und Ideensammlungen um semantische Annotationen ergänzen können, die ebenfalls in das Diskussionsmodell integriert werden.

[4] http://www.kerneltraffic.org/

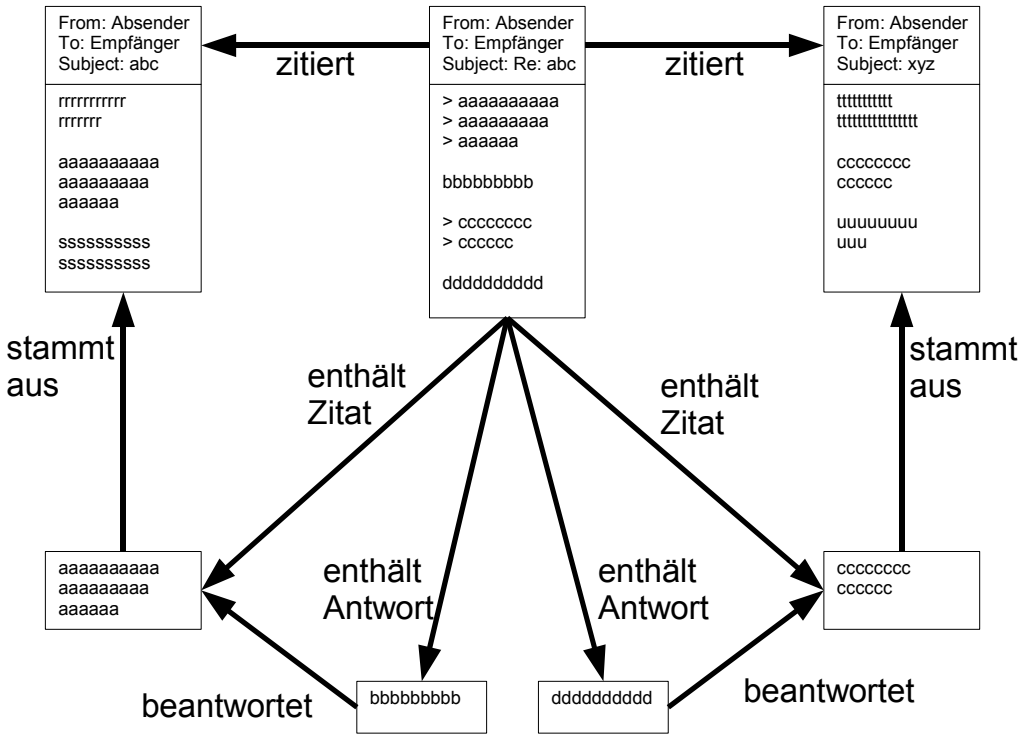

Abbildung 1.1: Aufteilen einer E-Mail in inhaltlich zusammengehörige Bereiche

Aufgabenstellung Diese Diplomarbeit umfasst die Modellierung eines Diskussionsmodells, den Entwurf einer Architektur zur Abbildung von Nachrichten aus Mailinglisten auf das Diskussionsmodell sowie den Entwurf geeigneter Benutzungsschnittstellen. Die Architektur wird in Java implementiert.

Es soll analysiert werden, welche der in den Nachrichten enthaltenen Informationen verlässlich für den Aufbau des Diskussionsmodells genutzt werden können. Anschließend sollen geeignete Methoden ausgewählt werden, um Textabschnitte innerhalb der Nachrichten zu identifizieren und miteinander in Verbindung zu setzen.

Der Zugriff auf die im Diskussionsmodell enthaltenen Informationen soll mit herkömmlichen, unmodifizierten E-Mail-Programmen und Webbrowsern möglich sein.

Gliederung Das zweite Kapitel beschäftigt sich mit sich den gegenwärtig von E-Mail-Programmen und Newsreadern genutzten Methoden zur Darstellung von Diskussionen und beschreibt deren Vorteile, sowie Unzulänglichkeiten. Es werden Arbeiten vorgestellt, die versuchen diese Unzulänglichkeiten durch neue Strukturierungs- und Darstellungsarten von Mailinglisten-Diskussionen zu beheben. Den Abschluss des zweiten Kapitels bildet eine Übersicht über verschiedene, im Rahmen dieser Arbeit verwendete Techniken.

Im dritten Kapitel wird zunächst ein Überblick über die Architektur des gesamten Systems und die zugrundeliegenden Ideen gegeben. Es wird analysiert, inwiefern die zur Verfügung stehenden Ausgangsdaten für die Übernahme in das Diskussionsmodell geeignet sind. Anschließend werden alle erforderlichen Schritte zum Erstellen des Diskussionsmodells erläutert. Abschließend wird erklärt, wie das Diskussionsmodell über verschiedene Schnittstellen den Nutzern zugänglich gemacht wird.

Das vierte Kapitel behandelt einige zentrale Details der Implementierung, wie die Datenspeicherung mittels RDF und die Kommunikation zwischen Web-Schnittstelle und Server.

Kapitel fünf zeigt die Leistungsfähigkeit und Grenzen der Implementierung auf und erörtert Erfahrungen von Nutzern aus einem Praxistest.

Im sechsten Kapitel wird eine Zusammenfassung über die ganze Arbeit gegeben.

Begriffserklärungen Da die in dieser Arbeit beschriebene Architektur zwar primär für E-Mails und Mailinglisten konzipiert ist, daneben jedoch auch verwandte Nachrichtenformate wie Newsgroupbeiträge verarbeiten kann, wird anstelle von *E-Mail* oder *Newsgroupbeitrag* weitgehend der allgemeine Oberbegriff *Nachricht* genutzt.

Wenn sich beschriebene Eigenschaften, durchgeführte Untersuchungen oder zitierte Arbeiten speziell auf eines dieser Nachrichtenformate beziehen, kommen jedoch auch die spezifischeren Bezeichnungen wie *E-Mail* oder *Newsgroupbeitrag* zur Verwendung.

Textabschnitt bezeichnet einen Teil des Nachrichtentextes, der aufgrund einer wie auch immer gearteten Unterteilung identifizierbar ist. Ein Spezialfall eines Textabschnittes ist ein *Zitat*, das aus Text besteht, der aus einer anderen Nachricht zitiert wurde.

2 Grundlagen

Aufgrund der verbreiteten Nutzung des Mediums E-Mail und von Newsgroups existieren verschiedene Methoden, die Darstellung von darüber geführten Diskussionen für den Benutzer übersichtlich zu gestalten. In diesem Kapitel werden sowohl Darstellungsmethoden aktuell genutzter Software beschrieben, als auch Arbeiten vorgestellt, in denen neue, verbesserte Darstellungsmethoden entwickelt wurden. Abschließend werden verschiedene Techniken erläutert, die den in dieser Arbeit entworfenen Benutzungsschnittstellen zugrundeliegen.

2.1 Möglichkeiten aktuell genutzter Software

Dieser Abschnitt beschreibt die beiden, in aktuellen E-Mail-Programmen und Newsreadern genutzten Darstellungsweisen von Diskussionen, die sequentielle Darstellung und die Baumdarstellung, und geht auf deren Vorteile und Nachteile ein. Auch wird erläutert, inwieweit bereits Abstimmungen in Mailinglisten unterstützt werden.

2.1.1 Darstellung inhaltlicher Zusammenhänge

Die Darstellung von Diskussionen in Mailinglisten und Newsgroups beschränkt sich, wie auch von Gina D. Venolia et al. erläutert [VN03], bei den meisten heute verwendeten Programmen und Web-Schnittstellen auf Variationen der nachfolgend beschriebenen und in Abbildung 2.1 illustrierten Darstellungsweisen:

- **Sequentielle Darstellung**. Dabei werden die Nachrichten unstrukturiert in einer Liste präsentiert, die nach verschiedenen Kriterien wie Absendedatum, oder alphabetisch nach Absender oder Betreffzeile sortiert werden kann. Ein Vorteil, der sequentiellen Darstellung ist, dass bei entsprechender Sortierung die chronologische Abfolge, in der die Nachrichten gesendet wurden leicht ersichtlich ist. Den größten Nachteil stellt dagegen das Fehlen jeglicher Informationen darüber dar, wie die Nachrichten inhaltlich zusammenhängen und auf welche andere Nachricht sich eine Antwort bezieht.

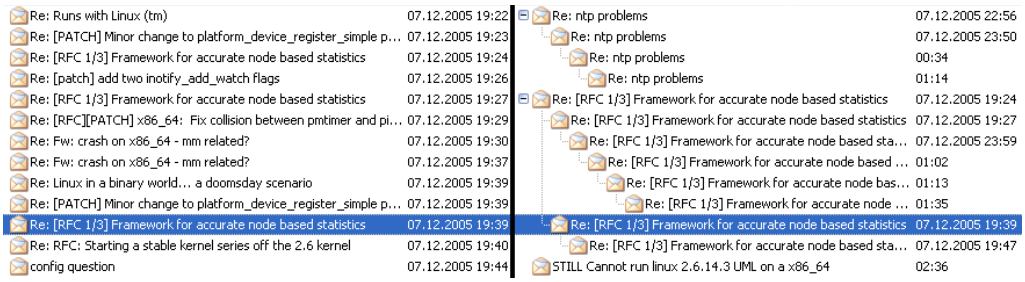

Abbildung 2.1: Sequentielle Darstellung und Baumdarstellung

- **Baumdarstellung**. Bei der Baumdarstellung werden die Nachrichten so angeordnet, dass eine Nachricht, die als direkte Antwort auf eine andere Nachricht erstellt wurde, in der Baumstruktur als Unterknoten der beantworteten Nachricht eingefügt wird. Die Informationen, auf welche Nachricht sich eine Antwort bezieht, sind in deren Kopfzeilen enthalten. Zusätzlich versuchen einige Programme Nachrichten, die sich nicht anhand der Kopfzeilen in die Baumstruktur einordnen lassen, Nachrichten mit ähnlicher Betreffzeile zuzuordnen. Während bei dieser Darstellung die unmittelbaren Antwortstrukturen in einer Diskussion ersichtlich sind, geht die chronologische Ordnung der Nachrichten weitgehend verloren. Zudem werden auch hier keine inhaltlichen Zusammenhänge zwischen Nachrichten dargestellt, die nicht in einer direkten Antwort-Relation zueinander stehen.

Es existieren verschiedene, in der Praxis eingesetzte Varianten der sequentiellen Darstellung und der Baumdarstellung die versuchen, die Vorteile beider Darstellungsweisen zu kombinieren.

Eine Variante ist die Gruppierung der Nachrichten nach Betreffzeilen. Innerhalb einer Gruppe von Nachrichten mit gleicher Betreffzeile werden die Nachrichten dann wiederum ohne weitere Strukturierung nach dem Absendedatum oder einem anderen Kriterium sortiert.

Eine andere Variante bietet *Netscan* [SF01], das insbesondere für die Darstellung von Diskussionen in Newsgroups konzipiert wurde. *Netscan* präsentiert die Newsgroupbeiträge in einer speziellen Baumstruktur die so unterteilt ist, dass die einzelnen Beiträge dennoch chronologisch nach Tagen gruppiert werden können.

Eine Abwandlung der Baumdarstellung, bei der jeweils ein Ast, also ein Diskussionsstrang, als eine Folge von Nachrichten präsentiert wird, kommt unter anderem bei *Google Mail*[1] zum Einsatz. Dabei ist sowohl die inhaltliche, als auch die zeitliche Abfolge der Nachrichten innerhalb dieses Diskussionsstrangs leichter erkennbar. Allerdings geht damit der Überblick über die gesamte Diskussion verloren, falls diese sich in mehrere Diskussionsstränge aufteilt.

[1] http://mail.google.com

Fast alle E-Mail-Programme und Newsreader wie *Microsoft Outlook*, *Eudora* oder *Thunderbird* und auch Web-Schnittstellen wie die von *Google Groups* bieten wahlweise sowohl eine Baumdarstellung als auch eine sequentielle Darstellung der Nachrichten an.

Alle zuvor beschriebenen Lösungen versuchen zeitliche oder inhaltliche Zusammenhänge zwischen Nachrichten aufgrund äußerer Merkmale wie Betreffzeile, Datum und Absender oder direkter Antwort-Relationen darzustellen. Jedoch kann eine Nachricht Aussagen zu verschiedenen Themen enthalten, sowie sich auf mehrere vorangehende Nachrichten beziehen und aus diesen zitieren. Auch entwickeln sich bei Diskussionen in Mailinglisten und Newsgroups häufig verschiedene Diskussionsstränge, die unterschiedliche Aspekte des ursprünglichen Themas behandeln, oder von diesem vollständig abweichen können. Daher ist eine Darstellung, die sich nur an äußeren Merkmalen der Nachrichten, nicht aber am tatsächlichen Nachrichteninhalt orientiert nur bedingt geeignet, den Verlauf einer Diskussion zu visualisieren. Derartige inhaltliche Aspekte werden aber von keinem der heute genutzten Programme und keiner der zuvor beschriebenen Darstellungsweisen berücksichtigt.

2.1.2 Semantische Annotation/Abstimmungen

Bei einer Diskussion ist es von Vorteil einen Überblick zu haben, welche Meinung jeder Diskussionsteilnehmer zum diskutierten Thema vertritt. Dies kann mittels einer Abstimmung, beziehungsweise dem Einholen kurzer Kommentare von allen Diskussionsteilnehmern ermittelt werden. Weder E-Mails, noch Newsgroupbeiträge bieten zur Auswertung solcher Abstimmungen eine ausreichende Unterstützung, da in den entsprechenden Standards keinerlei Strukturierung oder Annotation des Nachrichteninhaltes für derartige Zwecke vorgesehen ist. Die Meinungen der Diskussionsteilnehmer sind daher üblicherweise nur implizit im Nachrichtentext enthalten.

Einige E-Mail-Programme wie *Microsoft Outlook* bieten die Möglichkeit, Abstimmungen unter Mitgliedern einer Arbeitsgruppe durchzuführen und auszuwerten. Dies sind jedoch proprietäre Lösungen, die nur mit dem jeweiligen E-Mail-Programm nutzbar sind. Auch werden hierbei oft für die Übertragung der Abstimmungsergebnisse vom E-Mail-Programm zu einem Server nicht die E-Mails selbst, sondern andere Kommunikationswege benutzt.

2.2 Verwandte Arbeiten

In diesem Abschnitt werden verwandte Arbeiten vorgestellt, die sich mit alternativen, im Gegensatz zur sequentiellen Darstellung oder Baumdarstellung, verbesserten Methoden zur Darstellung von Diskussionsstrukturen in E-Mails und Newsgroups beschäftigen. Insbesondere sind dies Arbeiten, die dafür auch auf den Nachrichteninhalt und auf Zitatzusammenhänge zurückgreifen. Des Weiteren werden Arbeiten beschrieben, die mittels spezieller semantischer Annotationen die Nachrichteninhalte um maschinenlesbare Informationen ergänzen.

2.2.1 Inhalts- und Zitatanalyse von Nachrichten

Paula S. Newman stellt im Rahmen des Mail Content Projektes Weiterentwicklungen und Kombinationen herkömmlicher Baumdarstellungen von E-Mail-Diskussionen vor [New02] [New01]. Basierend auf Betreffzeilen sind dies *Enhanced Subject Listings* und *Leightweight Subject Listings*.

Enhanced Subject Listings stellen dabei sowohl die Betreffzeile, als auch den Anfang der ersten E-Mail in einer Diskussion dar, wobei die Diskussionen nach Datum oder Anzahl der E-Mails geordnet werden können.

Leightweight Subject Listings extrahieren Schlagwörter aus Betreffzeilen und bieten eine Übersicht, in der E-Mails, deren Betreffzeilen die gleichen Schlagwörter enthalten, zu Gruppen zusammengefasst werden.

Des weiteren beschreibt sie *Narrow Trees*, sowie *Tree Tables* die beide versuchen, eine Baumdarstellung mit einer linearen Darstellung der E-Mails zu kombinieren.

Im Gegensatz zu herkömmlichen Baumdarstellungen weisen *Narrow Trees* flachere Strukturen auf, da hierbei Antworten auf eine E-Mail in der gleichen Ebene dargestellt werden wie die beantwortete E-Mail, solange nicht mehr als eine Antwort vorliegt. Erst bei mehreren Antworten werden diese in einem neuen Unterast des Baumes platziert.

Tree Tables verwenden eine tabellenartige Darstellung zur besseren Ausnutzung des auf dem Bildschirm zur Verfügung stehenden Platzes. Dabei werden verschiedene Äste einer Diskussion durch Tabellenspalten unterschiedlicher Breite repräsentiert. An jedem Punkt, an dem sich eine Diskussion in mehrere Äste aufteilt, teilt sich auch die entsprechende Tabellenspalte in die gleiche Anzahl zusätzlicher Tabellenspalten auf.

Sowohl in *Narrow Trees*, als auch in *Tree Tables* wird jede E-Mail in Form einer kurzen Zusammenfassung des relevanten Nachrichteninhaltes repräsentiert, der durch entfernen von Elementen wie Zitaten, Grußformeln und Signaturen, sowie weitere Inhaltsanalyse ermittelt wird. Um den Vorgänger einer Nachricht zu identifizieren, wird sowohl auf Kopfzeilen, als auch auf inhaltliche Zusammenhänge zurückgegriffen. Allerdings werden hierbei die inhaltlichen Zusammenhän-

ge nur genutzt, um den besten Vorgänger einer E-Mail zu finden und nicht, um weitergehende Verbindungen zwischen allen inhaltlich verwandten, einander zitierenden E-Mails und Textabschnitten herzustellen. Somit weicht dieser Ansatz diesbezüglich nicht von der üblichen Baumdarstellung ab, die ebenfalls nur 1:n Relationen zwischen E-Mails und Antwortmails vorsieht.

Einen mit *Narrow Trees* vergleichbaren Entwurf beschreibt Gina D. Venolina et al. [VN03], bei dem auch eine sequentielle Darstellung und eine Baumstruktur mit reduzierter Tiefe kombiniert werden. Gegenüber *Narrow Trees* werden hier die einzelnen E-Mails noch zusätzlich durch Linien verschiedener Stärke verbunden, um die Zusammenhänge zwischen beantworteter und antwortender E-Mail zu verdeutlichen.

Noch stärker zitatorientiert ist der Ansatz von Jun Yabe et al. [YST00]. Hierbei stehen insbesondere Diskussionen in Newsgroups im Vordergrund. Yabe weist auf Unterschiede zwischen mündlich geführten Diskussionen, beziehungsweise Diskussionen in Echtzeit und Diskussionen in Newsgroups hin. Während bei mündlichen Diskussionen üblicherweise zusammengehörige Aussagen räumlich und zeitlich nah beieinander angesiedelt sind, ist bei schriftlichen Diskussionen diese Nähe meist nicht gegeben und Zusammenhänge können nur aufgrund formaler Zitat-Antwort-Relationen hergestellt werden. Unter diesem Gesichtspunkt betrachtet er die Abfolge von Zitaten und Antworten in Newsgroupbeiträgen. Der Schwerpunkt liegt dabei nicht auf der Visualisierung der Zusammenhänge zwischen den einzelnen Nachrichten, sondern darauf, die Zitate und Antworten so zu ordnen, dass sie als fortlaufendes Konversationsprotokoll dargestellt werden können. Ein zentrales Element stellt dabei auch die visuelle Unterscheidung zwischen einzelnen Diskussionsteilnehmern dar, die mittels animierter 3D Avatare realisiert wird.

Ebenfalls auf die Unterschiede zwischen mündlichen und per Mailinglisten oder in Newsgroups geführten Diskussionen geht Dimitri Popolov ein [PCL00]. Er kommt dabei zu dem Schluss, dass E-Mails für Textkommunikation typische Zitat-Antwort-Strukturen nutzen, um Elemente einer mündlich geführten Diskussion nachzubilden. Im Gegensatz zu Yabe versucht Popolov nicht, den Verlauf einer Diskussion auf ein lineares Konversationsprotokoll abzubilden. Er stellt *Converspace*, den Prototyp einer Anwendung vor, mit deren Hilfe die Diskussion auf einen zweidimensionalen Raum abgebildet wird. Dabei stellt eine Dimension die Zeitachse und die andere Dimension die Diskussionsstruktur dar. Die konventionelle Struktur der E-Mail-Kommunikation wird hier völlig aufgegeben und eine neue Form der Textkommunikation geschaffen, bei der einzelne Diskussionsfragmente und Zitate im Vordergrund stehen. Dies soll die Spontanität einer mündlichen Diskussion vermitteln. Die von Popolov vorgestellte Idee der Abbildung einer Diskussion auf einzelne Diskussionsfragmente und Zitate ist jedoch auch auf bereits existierende, im E-Mail-Format vorliegende Diskussionen anwendbar.

2.2.2 Semantische Annotation von Nachrichten

Einen Schritt über die reine Verknüpfung von Zitaten und Antworten hinaus geht die Auswertung, wie in einer Antwort der Inhalt eines zitierten Textabschnittes bewertet wird. Eines Ansatz hierfür könnte auf der IBIS Methode [KR70] basieren. IBIS dient zur gemeinsamen Problemanalyse und Problemlösung in Arbeitsgruppen. Dabei wird zunächst das Gebiet identifiziert, in dem das zu lösende generelle Problem liegt. Anschließend werden einzelne Problemteile als Fragen formuliert und mögliche Lösungen dafür zur Diskussion gestellt. Im Verlauf der Diskussion können Argumente, die für oder gegen eine Problemlösung sprechen vorgebracht werden, die dann entsprechend der IBIS Methodologie notiert und visualisiert werden. Programme zur Visualisierung und Steuerung von Gruppendiskussionen, die auf IBIS basieren sind unter anderem *Questmap*, sowie dessen Nachfolger *Compendium*[2].

Die Brücke von IBIS zu Mailinglisten schlägt Eugene E. Kim et al. [EH02], der einen Ansatz zur Verknüpfung von Nachrichten anhand unterschiedlicher Kriterien, einschließlich IBIS Annotationen, beschreibt. Als Grundlage dafür dient das *Open Hyperdocument System* (OHS) [Eng90] [Eng00], das eine feingranularere Verknüpfung von Dokumententeilen beschreibt, als die heute im Internet übliche Verknüpfung ganzer Dokumente. OHS führt auch typisierte Verweise, sowie inverse Verweise ein, mit denen andere Dokumente erreicht werden können, die wiederum auf das aktuelle Dokument verweisen. Zur konkreten Umsetzung von OHS schlägt Kim die Nutzung einer graphenorientierten Metasprache wie RDF vor, um Inhalt, sowie Verknüpfungsinformationen von Dokumenten zu beschreiben. Am Beispiel von Mailinglisten und Newsgroups zeigt er, dass auf diese Weise sowohl die schon zuvor beschriebene Einordnung von Nachrichten in Baumstrukturen, die Verknüpfung von beantworteten mit antwortenden Nachrichten, als auch eine inhaltliche Verknüpfung von Textabschnitten und deren Bewertungen analog der IBIS Methode im gleichen Graphen und Datenmodell erfolgen kann.

Eine mögliche Kombination von IBIS mit Mailinglisten und Newsgroups beschreibt auch Danny Ayers, der einen möglichen Anwendungsfall für sein auf RDF basiertes IBIS Vokabular[3] in der semantischen Annotation von E-Mails mit speziellen IBIS-Tags wie *ibis:Position* oder *ibis:con* sieht. Dies nutzt er zur inhaltlichen Klassifikation und Verknüpfung von E-Mails und Textabschnitten.

2.2.3 Unterschiede zu vorherigen Arbeiten

Im Gegensatz zu den zuvor vorgestellten Arbeiten beschränkt sich die in dieser Diplomarbeit entworfene Architektur nicht nur auf eine inhaltsbasierte, nicht weitergehend verwertbare Verknüpfung von Nachrichten, sondern bildet diese

[2] http://www.compendiuminstitute.org/tools/compendiumvsquestmap.htm
[3] http://dannyayers.com/xmlns/ibis/

auf ein zugrundeliegendes feingranulares Diskussionsmodell ab. Auch wird die Diskussionsstruktur im Gegensatz zu anderen Arbeiten mittels RDF in einer Form gespeichert, die es auch anderen Programmen erlaubt, die darin enthaltenen Informationen auf einfache Art und Weise zu nutzen. Die inhaltlichen Zusammenhänge werden nicht wie bei Newman [New02] [New01] dazu verwendet, existierende Darstellungsformen wie die Baumdarstellung zu modifizieren, sondern stellen eine eigene Repräsentation der Struktur einer Diskussion dar, die somit nicht den Beschränkungen der sequentiellen Darstellung, beziehungsweise der Baumdarstellung unterworfen ist. Auch wird nicht versucht, die für Mailinglisten und Newsgroups typische Unterteilung in einzelne Nachrichten vollständig aufzulösen wie bei Yabe et al. [YST00] oder die E-Mail-Kommunikation durch eine neue Kommunikationsform wie *Converspace* von Popolov [PCL00] zu ersetzen. Stattdessen soll der grundlegende Charakter und die Kompatibilität zum Medium E-Mail erhalten bleiben. Es sollen jedoch zusätzliche Informationen aufgrund der Einbeziehung inhaltlicher Aspekte und Zusammenhänge erschlossen und zugänglich gemacht werden.

2.3 Zugrundeliegende Techniken

Dieses Unterkapitel beschreibt Techniken, die in ähnlicher Form im Rahmen dieser Arbeit, wie in Abschnitt 3.6 beschrieben, zum Zugriff auf die im Diskussionsmodell enthaltenen Informationen, sowie zur Visualisierung des Modells eingesetzt werden.

2.3.1 Steuerung von Prozessen per Email

Die schon in Abschnitt 2.2.2 beschriebene semantische Annotation von Nachrichten kann auch dazu verwendet werden, diese um Befehle zu ergänzen, mit denen eine zentrale, die Nachrichten verarbeitende Instanz zur Durchführung bestimmter Aktionen veranlasst werden kann, wie die beiden nachfolgend beschriebenen Arbeiten zeigen.

Ein allgemeines Konzept für Semantic E-Mail, bei dem E-Mails um RDF-Aussagen ergänzt werden, beschreibt Luke McDowell et al. [MEHL04]. Der RDF-Teil einer Nachricht kann dabei von einem zentralen E-Mail-Manager automatisch erkannt und verarbeitet werden. Auf diese Weise ist es möglich, per E-Mail aus einem zugrundeliegenden RDF Datenmodell Informationen anzufordern, neue Daten zu diesem hinzuzufügen oder komplexere Prozesse in Gang zu setzen. McDowell entwirft dabei ein formales Modell semantischer E-Mail-Prozesse und geht auf die logischen und entscheidungstheoretischen Grundlagen ein, auf denen die Verarbeitung der RDF Daten basiert. Die Implementierung erfolgt mit Hilfe von Webformularen für verschiedene semantische Prozesse, um zu vermeiden, dass die Benutzer mit der RDF-Notation in Kontakt kommen.

Ähnliches wurde bei den HP Laboratories Bristol von Olu Ibidunni [Ibi02] in Form eines Mailinglisten-Assistenten implementiert, um die Koordination von Arbeitsgruppen zu vereinfachen. Der Mailinglisten-Assistent empfängt dabei alle an die Mailingliste gerichteten E-Mails und archiviert diese in einem RDF-Datenmodell. Die Nutzer der Mailingliste können ihre E-Mails um die Arbeitsgruppe betreffende Befehle ergänzen, die der Mailinglisten-Assistent automatisch erkennt und ausführt. Alle im RDF Modell gespeicherten Daten sind dabei zusätzlich über eine Web-Schnittstelle zugänglich.

2.3.2 REST

Die REST-Architektur wurde von Roy T. Fielding entworfen [Fie00] und beschreibt Methoden und Schnittstellen zum einfachen Zugriff auf dezentrale, miteinander verknüpfte Informationen.

Zentrale Elemente von REST sind die zustandslose Kommunikation zwischen Client und Server und eine einheitliche Schnittstelle zum Datenaustausch zwischen allen beteiligten Komponenten. Das grundlegende Datenelement in REST sind Ressourcen, die eine eigenständige Information oder eine Sammlung anderer Ressourcen repräsentieren können. Jede Ressource ist durch einen eindeutigen Identifikator (URI) adressierbar, der auch dann gültig bleibt, wenn sich die dadurch adressierte Ressource ändert. Ein weiteres Merkmal von REST ist, dass die Ressourcen auch untereinander über die URIs verknüpft sind.

Leigh Dodds [Dod05] beschreibt einen Ansatz, wie der Zugriff auf in einem RDF Vokabular wie FOAF[4] vorliegende Daten durch REST Webservices ermöglicht werden kann. Dabei kommt er zu dem Schluss, dass die Nutzung dieser Daten durch die einheitliche REST-Schnittstelle vereinfacht wird und dadurch neue Nutzungsmöglichkeiten eröffnet werden. Er zeigt auch am Beispiel einiger populärer Dienste wie flickr[5], dass REST-ähnliche Webservices schon heute erfolgreich eingesetzt werden.

Speziell mit der Erweiterung von E-Mail-Kommunikation durch REST-Dienste beschäftigt sich Paul Prescod[6]. Dabei werden viele Objekte wie E-Mail-Konten oder E-Mails durch, mittels URIs adressierbare Ressourcen repräsentiert. Sämtliche im Zusammenhang mit E-Mails durchführbaren Aktionen wie das Senden, Lesen oder Löschen von E-Mails werden mit Standardbefehlen des HTTP Protokolls wie *GET* und *PUT* realisiert. Als Vorteile einer auf REST basierten E-Mail-Architektur werden unter anderem die Nutzung von Standard-Webservern und die Verwendung des einfachen HTTP Protokolls mit seinen Sicherheitsfunktionen genannt. Zudem wird auf die auf diesem Weg mögliche Integration des E-Mail-

[4] http://www.foaf-project.org/
[5] http://flickr.com/services/api/
[6] http://www.prescod.net/rest/restmail/

Namensraums in den Web-Namensraum hingewiesen, was eine einheitliche Adressierung von Webseiten, E-Mail-Konten und E-Mails ermöglicht.

2.3.3 Visualisierung der Diskussionsstruktur

RDF Daten werden oft in Form von Graphen visualisiert, in denen Subjekt und Objekt einer RDF Aussage Knoten im Graphen und die Prädikate Verbindungen zwischen den Knoten darstellen. Ein Beispiel dafür ist Foafscape [Gie04]. Foafscape ist eine Software zur Visualisierung von Daten, die im FOAF Format, einem RDF-Vokabular zur Abbildung von sozialen Netzwerken, vorliegen. Im Gegensatz zu anderen, graphenorientierten Visualisierungen von RDF Daten, stellt Foafscape nicht alle Knoten eines Graphen dar, sondern aggregiert bestimmte Arten von Knoten, um eine kompaktere Darstellung zu erreichen.

2.3.4 Verwendung in dieser Arbeit

Die zuvor vorgestellten Arbeiten beschreiben einzelne Techniken wie die Übermittlung von Befehlen per E-Mail an einen Server, eine REST-Schnittstelle und aggregierte Graphendarstellungen zur Visualisierung von RDF Daten und implementieren diese auch zum Teil. In dieser Arbeit werden einige Aspekte dieser Techniken aufgegriffen und, wie in wie in Abschnitt 3.6 beschrieben, miteinander kombiniert, so dass diese sich ergänzen und helfen, den Benutzern Diskussionsstrukturen und Zusammenhänge zwischen Nachrichten leichter zugänglich zu machen.

2.4 Zusammenfassung der Unterschiede

Über die zuvor beschriebenen Arbeiten hinausgehend bietet Semalan:

- Die Abbildung einer Mailinglisten-Diskussion auf ein feingranulares Diskussionsmodell.
- Das Einbinden von Zusatzfunktionen wie Abstimmungen in das Diskussionsmodell.
- Eine einfachere Weiterverwendung der Informationen im Diskussionsmodell dank RDF.
- Neue Zugriffsmöglichkeiten auf das Diskussionsmodell mittels Techniken wie REST oder einer Graphendarstellung.

3 Design

Die Kernidee dieser Arbeit besteht darin, per E-Mail oder in Newsgroups geführte Diskussionen auf ein feingranulares Diskussionsmodell abzubilden und dadurch ein schnelleres und besseres Verständnis des Diskussionsverlaufs zu ermöglichen. Entsprechend bestehen die drei Hauptfunktionen der in diesem Kapitel beschriebenen Semalan-Architektur, wie in Abbildung 3.1 illustriert aus dem Einlesen von Nachrichten, deren Einordnung in das Diskussionsmodell und dem Bereitstellen der dadurch gewonnenen Informationen über eine Benutzungsschnittstelle.

Entscheidend ist dabei, dass im Diskussionsmodell die einzelnen Nachrichten nicht mehr als unstrukturierte Texte vorliegen, sondern eine Aufteilung jeder Nachricht in einzelne, inhaltlich zusammenhängende Textabschnitte erfolgt. Dies erlaubt eine wesentlich feingranularere Modellierung einer Diskussion, als bei einer Beschränkung auf die kompletten, nicht weiter strukturierten Nachrichtentexte. Darüber hinaus kann das Diskussionsmodell noch zusätzliche Informationen wie Annotationen zu einzelnen Textabschnitten enthalten.

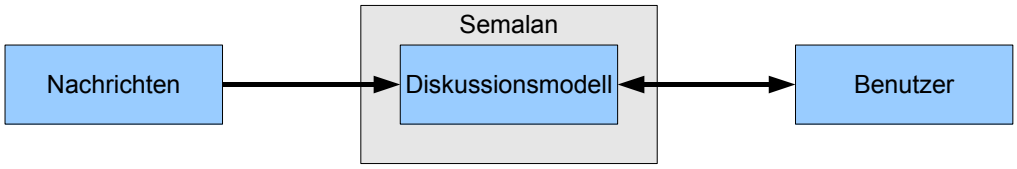

Abbildung 3.1: Semalan Grundstruktur

3.1 Das Diskussionsmodell

Abbildung 3.2 zeigt exemplarisch, wie verschiedene Teile einer Nachricht in das Diskussionsmodell übernommen werden.

Ein zentrales Element im Semalan Diskussionsmodell sind Nachrichten aus Diskussionen, die E-Mails oder Newsgroupbeiträge repräsentieren können. Diese Nachrichten enthalten neben Metadaten in Form verschiedener Kopfzeilen auch den vollständigen Nachrichtentext. Außerdem sind mit jeder Nachricht von Semalan erzeugte Verwaltungsinformationen verknüpft, wie eine eindeutige Nachrichten-Nummer, Informationen über die direkt vorhergehenden und nachfolgenden Nachrichten im Diskussionsbaum, und Verweise auf die anschließend erläuterten Textabschnitte.

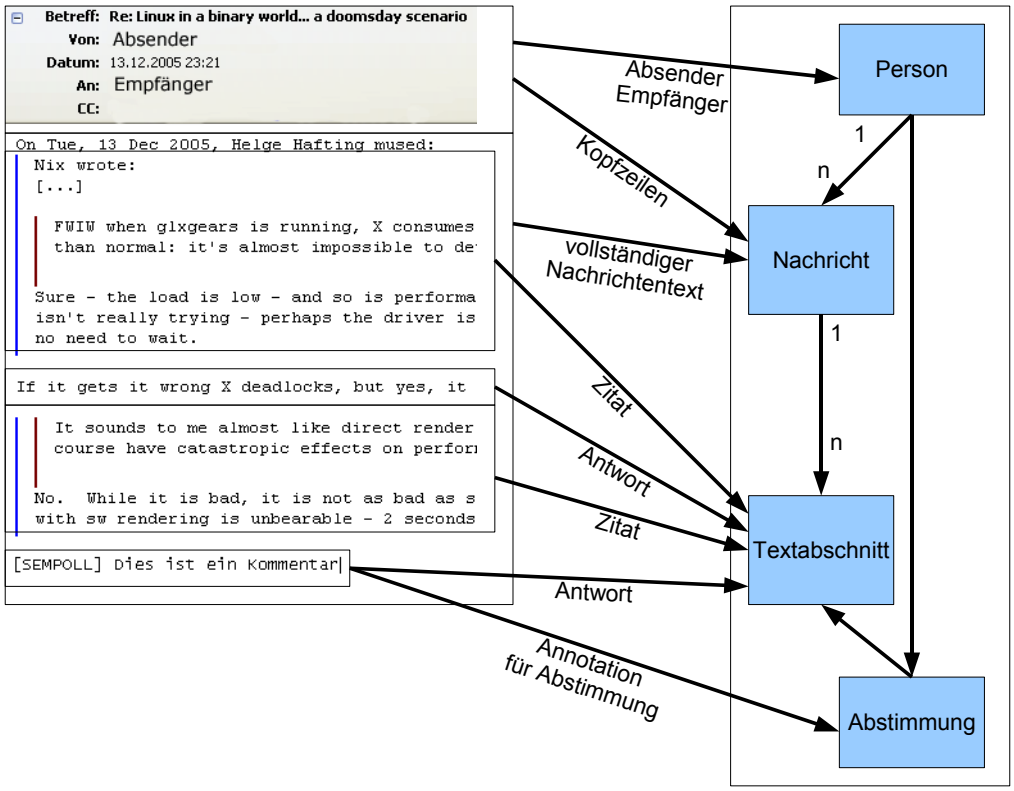

Abbildung 3.2: Abbildung einer Nachricht auf das Diskussionsmodell

Das zweite wichtige Element im Diskussionsmodell sind Textabschnitte, die aus Teilen von Nachrichten bestehen. Dies können direkt aus anderen Nachrichten zitierte Texte sein, aus anderen Nachrichten übernommene Zitate, oder eigene, nicht zitierte Texte. Jeder Textabschnitt im Diskussionsmodell enthält sowohl einen genauen Verweis auf die Stelle, an der er sich in der zitierenden Nachricht befindet, als auch auf die Stelle, an der er in der Nachricht enthalten ist, aus der er zitiert wurde. Stellt ein Textabschnitt eine Antwort auf einen anderen zitierten Textabschnitt dar, wird dies durch eine Verknüpfung der beiden Textabschnitte repräsentiert. Zudem ist eine Prüfsumme zur eindeutigen Identifikation jedes Textabschnittes hinterlegt. Durch diese Kombination von untereinander verknüpften Nachrichten und Textabschnitten sind in dem Diskussionsmodell sowohl Informationen darüber gespeichert, welche Nachrichten und Textabschnitte sich aufeinander beziehen und einander zitieren, als auch welcher Textabschnitt wo und wie oft im Laufe der Diskussion zu finden ist. Somit sind für den Verlauf einer Diskussion wichtige und häufig zitierte Textabschnitte leicht zugänglich, was es dem Leser erleichtert dem Verlauf einer Diskussion zu folgen. Auch lässt

sich einfach ersehen, was jeder Diskussionsteilnehmer auf einen bestimmten Textabschnitt geantwortet hat.

Ein weiteres Element im Diskussionsmodell sind Abstimmungen, die der Speicherung von in Nachrichten enthaltenen semantischen Annotationen dienen. Hier werden der annotierte Textabschnitt, das Datum an dem die Umfrage gestartet wurde, alle Teilnehmer einer Abstimmung und deren Antworten, sowie Verweise auf die Nachrichten, die Annotationen zur Umfrage enthalten, abgelegt.

Die Diskussionsteilnehmer sind im Diskussionsmodell ebenfalls als eigenständige, eindeutig identifizierbare Elemente repräsentiert. So kann im Modell bei jedem Vorkommen eines Diskussionsteilnehmers, sei dies als Autor oder Empfänger einer Nachricht oder als Teilnehmer einer Umfrage, auf diesen verwiesen werden.

3.2 Integration in bestehende Prozesse

Abbildung 3.3 zeigt die Integration von Semalan in bestehende Kommunikationsprozesse. Entsprechend der in Abbildung 3.1 illustrierten Funktionsweise von Semalan sind bei der externen Kommunikation drei Parteien involviert. Ein Server, der eine Mailingliste oder Newsgroup verwaltet, Semalan, das darauf aufsetzend seine Dienste anbietet und die Benutzer, die auf beides zugreifen. Um den Nutzern zu ermöglichen auch weiterhin auf gewohnte Art und Weise mit Mailinglisten und Newsgroups zu interagieren und ihre bevorzugten E-Mail-Programme und Newsreader einzusetzen, sitzt Semalan nicht als Mittler zwischen Mailingliste beziehungsweise Newsgroup und deren Teilnehmern, sondern kann als eigenständige Anwendung angesprochen werden.

- **Mailserver/Newsserver ↔ Benutzer** Die Kommunikation zwischen Mailinglisten/Newsgroups und deren Benutzern bleibt durch Semalan unberührt. Es können weiterhin Nachrichten direkt an den Mailserver oder Newsserver gesendet und von diesem empfangen werden.
- **Mailserver/Newsserver → Semalan** Semalan selbst bezieht, wie alle anderen Nutzer einer Mailingliste/Newsgroup neue Nachrichten direkt vom Mailserver oder Newsserver. Zudem kann Semalan zusätzlich auf alte E-Mails in Mailinglistenarchiven zugreifen, die von vielen Mailinglisten zur Verfügung gestellt werden.
- **Benutzer ↔ Semalan** Die Kommunikation zwischen Semalan und den Benutzern findet per E-Mail oder Webbrowser statt. Ein Benutzer kann Anfragen via E-Mail an Semalan schicken und bekommt auf gleichem Weg eine Antwort zugesandt. Er kann ebenfalls per Web-Schnittstelle auf das von Semalan verwaltete Diskussionsmodell zugreifen.

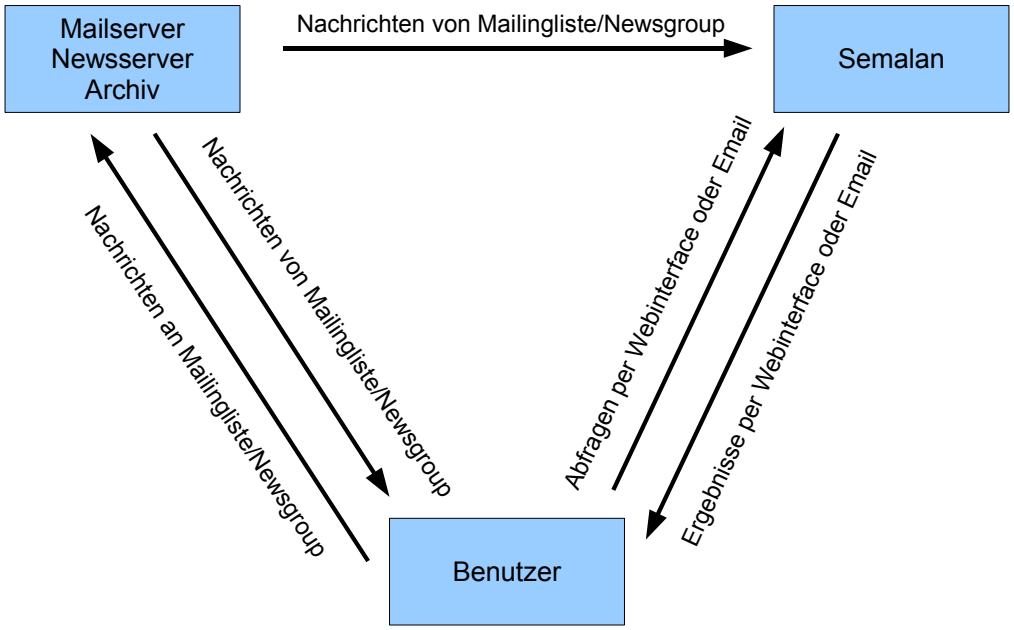

Abbildung 3.3: Integration in bestehende Prozesse

Durch dieses Kommunikationsmodell ist sichergestellt, dass an bestehenden Strukturen zur E-Mail-Kommunikation weder auf Seite des Servers, noch auf Seite der Benutzer Änderungen vorgenommen werden müssen, um Semalan einsetzen zu können.

3.3 Architektur

Abbildung 3.4 gibt einen Überblick über die Architektur und die einzelnen Verarbeitungsschritte in Semalan, die in den nachfolgenden Unterkapiteln näher erläutert werden. Die drei Hauptfunktionen, der Import von Nachrichten, die Abbildung selbiger auf das Diskussionsmodell, sowie der Zugriff über Benutzungsschnittstellen spiegeln sich auch in den internen Datenverarbeitungsprozessen wieder.

Datenimport Wie in Abschnitt 3.4 erklärt, werden aus verschiedenen Quellen wie POP3 E-Mail-Konten, Mailinglisten Archiven und Newsservern Nachrichten eingelesen. Des weiteren werden bestimmte Arten von Fehlern behoben, soweit dies für die weitere Verarbeitung der Daten notwendig ist und falls dies die in den Daten enthaltenen Informationen nicht verfälscht.

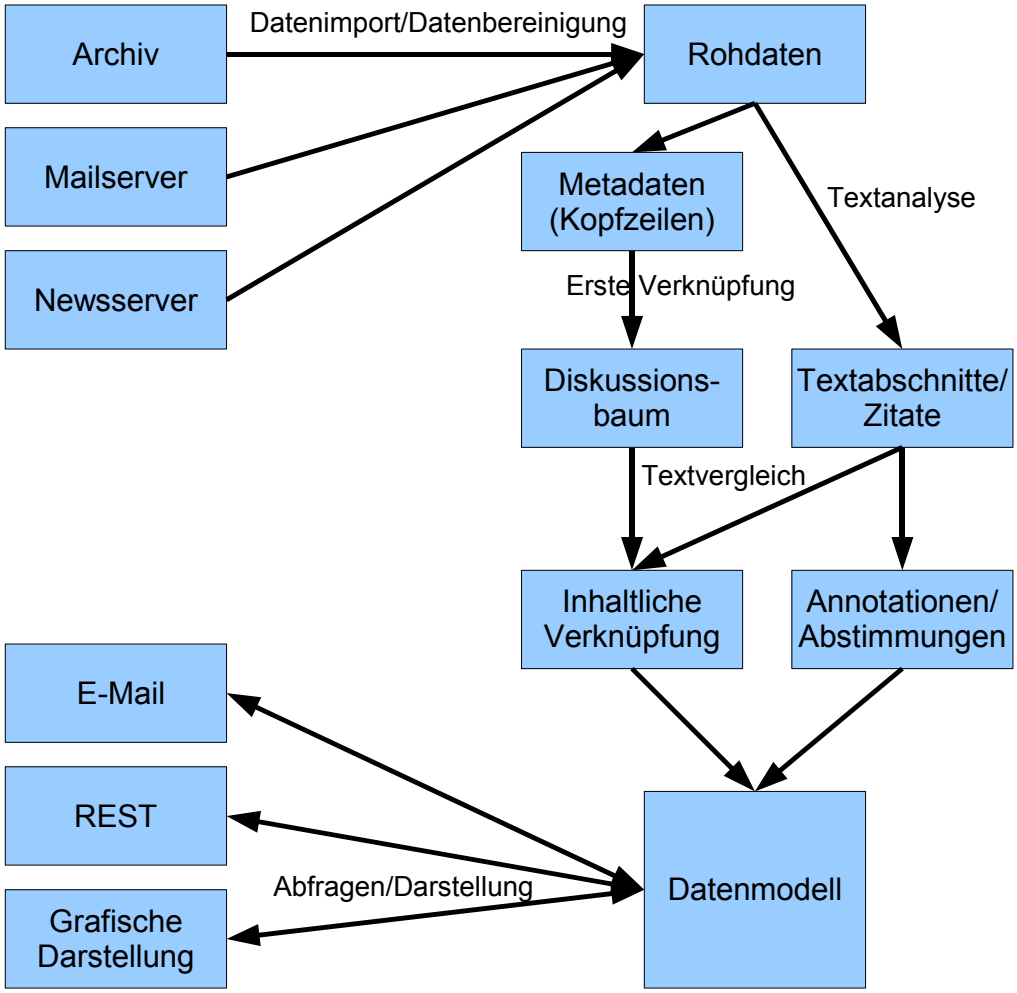

Abbildung 3.4: Verarbeitungsschritte in Semalan

Überführung in das Diskussionsmodell Mit diesem Verarbeitungsschritt beschäftigt sich Abschnitt 3.5 dieser Arbeit. Den Nachrichten werden in den Kopfzeilen enthaltene Metadaten entnommen. Basierend auf diesen Metadaten wird eine erste Verknüpfung der Nachrichten vorgenommen. Der Nachrichtentext wird analysiert, in einzelne Abschnitte unterteilt und darin enthaltene Zitate identifiziert. Aufgrund der Zitate wird eine zweite, inhaltliche Verknüpfung durchgeführt. Bei dieser inhaltlichen Verknüpfung werden zitierte Textabschnitte mit den Nachrichten verknüpft, aus denen sie zitiert wurden, sowie mit den Nachrichten, von denen sie zitiert werden. Zudem werden zitierte Textabschnitte mit den Textabschnitten verknüpft, von denen sie beantwortet werden. Ebenfalls werden in den Nachrichten vorkommende semantische Annotationen extrahiert. Alle gewon-

nenen Daten werden in dem in Abschnitt 3.1 beschriebenen Diskussionsmodell gespeichert.

Benutzungsschnittstellen Zum Zugriff auf die im Diskussionsmodell enthaltenen Informationen stellt Semalan drei in Abschnitt 3.6 beschriebene Benutzungsschnittstellen zur Verfügung. Eine Abfragemöglichkeit per E-Mail, eine REST-artige Schnittstelle, sowie eine grafische Benutzungsoberfläche.

3.4 Datenanalyse und Datenimport

In diesem Unterkapitel wird die Qualität der Nachrichten analysiert, die aus den Semalan zur Verfügung stehenden Datenquellen gewonnen werden können. Es werden Unterschiede und Gemeinsamkeiten der aus den drei Datenquellen bezogenen Nachrichten erklärt. Zudem wird untersucht, wie homogen die Struktur der Nachrichten innerhalb einer Datenquelle ist, welche Elemente der Nachrichten zur weiteren Verarbeitung in Semalan übernommen werden können und welche Korrekturen gegebenenfalls an den Nachrichten vorgenommen werden müssen.

Semalan ist in der Lage, Nachrichten aus drei verschiedenen Arten von Datenquellen zu importieren.

- **Mailinglistenarchive** Bestehende Archive von Mailinglisten können verarbeitet werden, sofern diese im MBOX Format vorliegen. MBOX[1] ist ein weit verbreitetes Format zum Speichern von E-Mails, das ursprünglich aus der UNIX Welt stammt und von allen bekannten Mailinglisten-Managern wie *Majordomo*[2] oder *Mailman*[3] genutzt wird. Viele Betreiber von Mailinglisten wie das W3C[4] bieten die Archive ihrer Mailinglisten zum Download an. Zudem unterstützen auch die meisten E-Mail-Programme den Export von E-Mails im MBOX Format.
- **Mailserver** Semalan kann als normales Mitglied einer Mailingliste angemeldet werden und neue E-Mails über ein eigenes E-Mail-Konto empfangen. Zum Abruf der E-Mails vom Mailserver wird das POP3 Protokoll unterstützt.
- **Newsserver** Es ist ebenfalls möglich, Newsgroupbeiträge in Semalan zu importieren, da diese eine große strukturelle Ähnlichkeit mit E-Mails aufweisen. Hierzu bietet Semalan die Möglichkeit, über das NNTP Protokoll direkt mit einem Newsserver zu kommunizieren.

[1] http://www.qmail.org/man/man5/mbox.html
[2] http://www.greatcircle.com/majordomo/
[3] http://www.gnu.org/software/mailman/
[4] http://www.w3.org/Mail/Archives

In dieser Arbeit wird auf folgende Mailinglisten Bezug genommen, die auch als Grundlage für die präsentierten Statistiken dienen:

- **Linux-Kernel**[5]: 14000 E-Mails aus einem Zeitraum von Oktober 2005 bis Dezember 2005. Mit 5000-6000 E-Mails pro Monat ist dies eine der nach eigenen Angaben aktivsten Mailinglisten weltweit. Diese Mailingliste zeichnet sich auch dadurch aus, dass E-Mails häufig zitiert und diskutiert werden, was zu überdurchschnittlich großen Diskussionsbäumen führt und somit zum Testen der Zitaterkennung geeignet ist. Zudem ist in E-Mails dieser Mailingliste auch häufig Quellcode von Programmen enthalten, was durch Sonderzeichen und im Vergleich zu natürlichsprachlichen Texten ungewöhnliche Strukturierungen eine zusätzliche Herausforderung bei der Textanalyse darstellt.
- **WWW-RDF-Interest**[6]: 13574 E-Mails aus einem Zeitraum von August 1999 bis November 2005. Diese Mailingliste enthält auch ältere E-Mails, die vor der letzten größeren Änderung der E-Mail-Standards im Jahr 2001 erstellt wurden. Dadurch wird ersichtlich, wie Semalan mit den zum Teil veränderten Kopfzeilen dieser älteren E-Mails zurechtkommt.
- **B-Greek**[7]: 11287 E-Mails aus einem Zeitraum von Juni 2003 bis Dezember 2005. Da die beiden anderen genannten Mailinglisten – wie ein Großteil der existierenden Mailinglisten – eher technischen Themen gewidmet sind, wird die *Biblical Greek Mailing List*, auf der griechische Ursprungstexte der Bibel diskutiert werden, als eine Mailingliste mit möglichst wenig technischen Themen herangezogen. Damit kann getestet werden, ob sich die in Semalan verwendeten Algorithmen und Methoden auch in diesem Bereich bewähren. Im Gegensatz zu den anderen beiden Mailinglisten steht von dieser Liste nur ein Archiv zur Verfügung, in dem sehr wenige Kopfzeilen der ursprünglichen E-Mails enthalten sind. So lässt sich auch testen, wie sich Semalan verhält, wenn nur ein minimaler Satz an Metadaten über jede E-Mail vorhanden ist.

3.4.1 Kopfzeilen

Um Nachrichten zuverlässig zu identifizieren, Zusammenhänge zwischen den Nachrichten zu erkennen und diese anschließend in das Diskussionsmodell zu überführen ist es hilfreich, wenn dafür auf standardisierte Strukturen und Metadaten zurückgegriffen werden kann, die in allen Nachrichten vorhanden sind.

Im Falle von E-Mails bieten sich dazu die Kopfzeilen an, deren Struktur weitgehend durch IETF Standards festgelegt ist. Die Kopfzeilen von E-Mails werden

[5] http://www.tux.org/lkml/
[6] http://lists.w3.org/Archives/Public/www-rdf-interest/
[7] http://www.ibiblio.org/bgreek/

in RFC 822[8] (1982), sowie dessen Erweiterungen RFC 1123[9] (1989) und RFC 2822[10] (2001) beschrieben.

Zwingend vorgeschrieben sind danach nur Absendedatum (*Date*), Absender (*From*) und Empfänger (*To*). Daneben existiert eine Vielzahl weiterer standardisierter, aber optionaler Kopfzeilen, von denen einige wie die Betreffzeile (*Subject*) oder *Message-ID* fast immer vorhanden sind, während andere wie die Rücksendeadresse (*Reply-To*) selten genutzt werden. Eine dritte Gruppe von Kopfzeilen bilden die sogenannten *X-Header*. Dies sind frei definierbare Kopfzeilen, die von jedem E-Mail-Programm, Mailserver oder jeder anderen Software, die E-Mails verarbeitet, hinzugefügt werden können.

	WWW-RDF-Interest		Linux-Kernel		B-Greek	
Nachrichtenanzahl	13574		14000		11287	
From	100%	13574	100%	14000	100%	11287
Date	100%	13574	100%	14000	100%	11287
Message-ID	100%	13574	100%	14000	100%	11284
Subject	100%	13574	100%	13991	100%	11284
To	99%	13459	100%	14000	0%	0
Content-Type	96%	12970	93%	13070	0%	0
MIME-Version	95%	12946	93%	13026	0%	0
X-Mailer	63%	8513	25%	3476	0%	0
Content-Transfer-Encoding	60%	8092	52%	7214	0%	0
CC	54%	7356	86%	12098	0%	0
In-Reply-To	43%	5815	76%	10706	45%	5085
References	41%	5553	77%	10808	35%	3944
Return-Path	18%	2390	100%	14000	0%	0
User-Agent	17%	2249	50%	7047	0%	0
Organization	14%	1884	8%	1148	0%	0
Reply-To	2%	225	7%	1049	0%	0

Tabelle 3.1: Die häufigsten Kopfzeilen

Tabelle 3.1 zeigt, wie häufig in den zuvor erwähnten Beispiel-Mailinglisten einzelne Kopfzeilen verwendet werden. Die Auflistung beinhaltet die 15 häufigsten Kopfzeilen, sowie *Reply-To*, das selten genutzt wird, aber dennoch zu den bekannteren Kopfzeilen zählt und auch in RFC 2822 standardisiert sind. Nicht in der Liste aufgeführt ist die Kopfzeile *Received*, die zwar in jeder E-Mail mehr-

[8] http://www.ietf.org/rfc/rfc0822.txt
[9] http://www.ietf.org/rfc/rfc1123.txt
[10] http://www.ietf.org/rfc/rfc2822

fach vorhanden ist, die aber nur Routinginformationen über den Versandweg einer E-Mail enthält. Dies ist für Semalan nicht von Interesse, da darin keine zusätzlichen Informationen über die E-Mails selbst oder die Absender enthalten sind. Eine weitere Verarbeitung der *Received* Kopfzeile würde zudem, aufgrund umfangreichen Routinginformationen, die zu speichernde Datenmenge deutlich erhöhen.

Aus der Tabelle ist ersichtlich, dass zur weiteren Verarbeitung der E-Mails Absender, Sendedatum, Message-ID und Betreffzeile als vorhanden vorausgesetzt werden können.

Umgang mit fehlerhaften oder fehlenden Kopfzeilen Die in den Kopfzeilen enthaltenen Daten entsprechen überwiegend den in RFC 2822 vorgegebenen Standards, wie Tabelle 3.2 fehlerhafter Kopfzeilen zu entnehmen ist. Alle Kopfzeilen aus Tabelle 3.1 die nicht in Tabelle 3.2 enthalten sind, sind nicht fehlerhaft, beziehungsweise weichen nicht von den Vorschriften des E-Mail-Standards ab.

	WWW-RDF-Interest	Linux-Kernel	B-Greek
Nachrichtenanzahl	13574	14000	11287
Fehler in Content-Transfer-Encoding	27	7	0
Fehler in To	10	0	0
Fehler in CC	4	0	0
Fehler in Reply-To	1	0	0
Doppelte Message-ID	8	5	0

Tabelle 3.2: Fehlerhafte Kopfzeilen

Falls eine der für die Weiterverarbeitung in Semalan benötigten vier Kopfzeilen (Absender, Sendedatum, Message-ID und Betreffzeile) dennoch fehlen oder fehlerhaft sein sollte, wird diese durch einen vorgegebenen Wert belegt. Message-ID und E-Mail-Adresse des Absenders werden dabei durch zufällig generierte, standardkonforme Werte ersetzt, Fehlerhafte Datumsangaben werden durch das UNIX Standarddatum (01.01.1970) und fehlende Betreffzeilen durch eine leere Betreffzeile repräsentiert.

In einigen Fällen treten auch doppelte Message-IDs auf. Dies ist eine Verletzung der E-Mail-Standards, da nach RFC 2822 Message-IDs global eindeutig sein müssen. Die Message-ID einer E-Mail wird entweder von dem E-Mail-Programm erzeugt, mit dem die E-Mail geschrieben wird, oder, wenn das nicht der Fall ist, spätestens vom ersten Mailserver der die E-Mail weiterleitet. Es ist nicht vorgeschrieben, wie eine Message-ID erzeugt werden muss. Jedoch werden dafür üblicherweise Zeitstempel, Zufallszahlen, IP-Adressen und Domainnamen verwendet.

Wie der Statistik in Tabelle 3.2 zu entnehmen ist, liegt die Anzahl doppelter Message-IDs in allen getesteten Mailinglisten unter 0,06%. Eine nähere Betrachtung der E-Mails mit doppelter Message-ID in *WWW-RDF-Interest* und *Linux-Kernel* hat gezeigt, dass es sich hierbei fast ausschließlich um doppelt in der Mailingliste vorhandene E-Mails handelt und nur selten um unterschiedliche E-Mails mit gleicher Message-ID.

Trifft eine Nachricht mit einer Message-ID ein, die bereits im Diskussionsmodell vorhanden ist, wird zunächst überprüft, ob es sich dabei um identische Nachrichten handelt. Dazu wird aus Absender, Erstellungsdatum und Message-ID beider Nachrichten je eine Prüfsumme gebildet und diese beiden Prüfsummen verglichen. Da das Erstellungsdatum in E-Mails und Newsgroupbeiträgen sekundengenau vorliegt, kann mit hoher Wahrscheinlichkeit davon ausgegangen werden, dass die beiden Nachrichten identisch sind, wenn sie zum gleichen Zeitpunkt und vom gleichen Absender erstellt wurden und zudem noch die gleiche Message-ID haben. In diesem Fall wird die neu eingetroffene, doppelte Nachricht verworfen.

Sind nur die Message-IDs zweier Nachrichten identisch, jedoch Absender oder Erstellungsdatum verschieden, wird die Message-ID der neuen Nachricht durch eine von Semalan generierte, global eindeutige Message-ID ersetzt. Dies ist notwendig, da die Message-ID intern von Semalan zur eindeutigen Identifikation der Nachrichten genutzt wird und somit keine zwei Nachrichten mit der gleichen Message-ID in das Datenmodell aufgenommen werden können. Durch das Generieren einer neuen Message-ID kann die Nachricht zwar nicht mehr, wie in Abschnitt 3.5.1 beschrieben, über die Message-ID mit anderen Nachrichten verknüpft werden, jedoch ist immer noch die im gleichen Abschnitt erläuterte Verknüpfung über die Betreffzeile möglich. Somit kann die Nachricht trotz neuer Message-ID in die in Abschnitt 3.5.2 beschriebene, inhaltlichen Verknüpfung einbezogen werden. Durch die von Semalan durchgeführte Konfliktbehebung bei doppelten Message-IDs ist daher kein negativer Einfluss auf die Datenqualität im Diskussionsmodell zu erwarten.

Archive (MBOX) Welche Kopfzeilen in einer E-Mail vorhanden sind, hängt auch von der Datenquelle ab, von der die E-Mails bezogen werden. Insbesondere bei Mailinglistenarchiven liegt es im Ermessen des Administrators der Mailingliste, welche Kopfzeilen in den Archiven gespeichert werden und wie diese vor dem Speichern verändert werden. Bei vielen Mailinglisten wie *B-Greek* beschränkt sich die Speicherung auf Absender, Sendedatum, Betreffzeile, Message-ID, Referenzen auf andere E-Mails in Form von *References* und *In-Reply-To* Kopfzeilen, sowie Informationen zur Zeichencodierung. Bei einigen nicht öffentlich zugänglichen Archiven wie denen des W3C bleiben hingegen fast alle Kopfzeilen der ursprünglichen E-Mails auch in den Archiven erhalten, wie die E-Mails der *WWW-RDF-Interest* Mailingliste zeigen. Speziell in öffentlich zugänglichen Mai-

linglistenarchiven werden oft die E-Mail-Adressen in den Kopfzeilen verändert, um das automatisierte Sammeln von E-Mail-Adressen für Werbezwecke zu erschweren. Semalan kann die am häufigsten angewandte Änderung, das Ersetzen von @ durch *at* beim Datenimport rückgängig machen und somit die ursprünglichen E-Mail-Adressen extrahieren.

3.4.2 Zeichencodierung

Der nächste Schritt beim Import der E-Mails ist die korrekte Extraktion des eigentlichen Nachrichtentextes. Im ursprünglichen RFC 822 Standard von 1982 waren nur Texte im 7 Bit ASCII Format vorgesehen, das auf in der englischen Sprache verwendete Zeichen beschränkt ist. Um diese Limitationen zu umgehen, wurden Multipurpose Internet Mail Extensions (MIME) eingeführt, die heute in erster Linie durch RFC 2045-2049[11] spezifiziert werden. Die wichtigsten Neuerungen in MIME sind die Unterstützung internationaler Zeichensätze, sowie die Möglichkeit anstelle eines einzigen Nachrichtentextes mehrere Teile unterschiedlichen Inhalts in einer E-Mail zu verwenden – sogenannte Multipart-Nachrichten. Dies kann unter anderem der Nachrichtentext selbst in verschiedenen Formaten sein, oder auch zusätzliche Dateianhänge die selbst als Text (z.B. andere E-Mails, Visitenkarten) oder in einem beliebigen Binärformat vorliegen können (z.B. Bilder). Die Informationen über den Aufbau einer MIME E-Mail sind in drei Kopfzeilen enthalten.

- **MIME-Version** gibt die verwendete MIME Versionsnummer an.
- **Content-Type** legt fest, welcher Art der Inhalt der E-Mail ist. Insbesondere, ob die Nachricht aus mehreren Teilen besteht. Für den eigentlichen Nachrichtentext sind *text/plain* (unformatierter Text) und *text/html* (Text im HTML Format) häufig anzutreffende Werte, wie Tabelle 3.3 zeigt. Eher selten genutzt werden andere Formate wie *text/enriched*.
- **Content-Transfer-Encoding** bestimmt die Art der Zeichencodierung die benutzt wird, um Sonderzeichen oder Binärdaten im ASCII Format darzustellen.

Während dies für alle E-Mails gilt, lohnt es sich dennoch, den Einsatz und Aufbau von MIME Nachrichten speziell in Mailinglisten genauer anzuschauen. Denn sowohl in Mailinglisten, als auch in Newsgroups wird erwartet, dass sich die Diskussionsteilnehmer an einige grundlegende Regeln beim Verfassen von Nachrichten halten, die in der Netikette[12] beschrieben werden.

[11] http://www.ietf.org/rfc/rfc2045.txt, http://www.ietf.org/rfc/rfc2046.txt, http://www.ietf.org/rfc/rfc2047.txt, http://www.ietf.org/rfc/rfc2048.txt, http://www.ietf.org/rfc/rfc2049.txt

[12] http://www.netplanet.org/netiquette/maillist.shtml, http://www.eschkitai.de/openoffice/netikette.html

Das Nachrichtenformat selbst betreffend sind dies:

- Nachrichten sollen als unformatierter Text gesendet werden. Insbesondere sind HTML E-Mails unerwünscht.
- Nachrichten an Mailinglisten sollen keine Dateianhänge enthalten.

Allerdings obliegt es dem Administrator einer Mailingliste und deren Teilnehmern durchzusetzen, dass diese Regeln eingehalten werden.

Die Statistik 3.3 zeigt, dass sich die meisten Mitglieder von Mailinglisten an die Regeln der Netikette halten und die E-Mails den Nachrichtentext fast immer auch als unformatierten Text (text/plain) enthalten. Der Anteil an reinen HTML E-Mails (text/html) liegt deutlich unter 1%. Da Multipart-Nachrichten den Nachrichtentext gleichzeitig in verschiedenen Formatierungen enthalten können, übersteigt die Summe der in der Statistik nach Textformatierungen aufgelisteten Nachrichten zum Teil die Gesamtanzahl der Nachrichten.

	WWW-RDF-Interest		Linux-Kernel		B-Greek	
Nachrichtenanzahl	13574		14000		11287	
Multipart	9%	1163	5%	732	0%	0
Dateianhänge	<1%	84	<1%	124	0%	0
auch text/plain	99%	13429	99%	13921	99%	11253
nur text/html	<1%	99	0%	0	0%	0
nur text/enriched	<1%	2	0%	0	0%	0
kein Text	<1%	13	<1%	70	<1%	37

Tabelle 3.3: Textformatierungen

Der Inhalt von HTML E-Mails ließe sich im Rahmen der in Abschnitt 3.5.2 beschriebenen inhaltlichen Verknüpfung nur eingeschränkt verarbeiten, da zitierter Text in HTML E-Mails je nach E-Mail-Programm auf verschiedenste Weise kenntlich gemacht ist. Zitate können durch farbliche oder sonstige Hervorhebung, oder ‚mittels Javascript, gar durch interaktive Elemente gekennzeichnet werden und lassen sich so nur schwer identifizieren. Daher verzichtet Semalan auf die Verarbeitung von reinen HTML E-Mails. Da diese wie zuvor erläutert nur einen geringen Anteil aller E-Mails in Mailinglisten ausmachen und reine HTML E-Mails in vielen Fällen Werbemails sind, wie eine Betrachtung der in Mailinglisten enthaltenen HTML E-Mails zeigt, ist von deren weiterer Verarbeitung in Semalan auch keine signifikante Verbesserung der Datenqualität im Diskussionsmodell zu erwarten.

E-Mails, die gar keinen identifizierbaren Textteil oder Text in einem anderen Format beinhalten machen weniger als 0,5% der gesamten E-Mails aus und können daher vernachlässigt werden. Multipart E-Mails sind mit weniger als 10%

eher selten und die einzelnen Teile der Multipart E-Mails bestehen meistens aus dem Nachrichtentext in verschiedenen Formaten oder anderen Textelementen wie Signaturen. Reine Binärdaten in Form von Dateianhängen sind bei weniger als 1% aller E-Mails vorhanden. In vielen Mailinglistenarchiven, wie in dem der *B-Greek* Mailingliste, wurden Dateianhänge schon beim Speichern der E-Mails im Archiv entfernt.

Bei Multipart-Nachrichten wird von Semalan nur der Teil genutzt, der den Nachrichtentext enthält. Andere Teile wie binäre Dateianhänge werden verworfen, da diese die zu speichernde Datenmenge erhöhen würden und zudem nicht zur Erstellung des Diskussionsmodells beitragen.

Die Zahl der E-Mails, die ungültige Angaben zum Content-Type oder zur Zeichencodierung enthalten, ist ebenfalls sehr gering, wie Tabelle 3.4 zeigt. Falls kein Content-Type angegeben ist, geht Semalan den Standards entsprechend von einem US-ASCII Text in 7-Bit Kodierung aus. Dies fällt insbesondere bei älteren Mailinglisten auf, deren E-Mails zum Teil aus einer Zeit stammen, zu der sich der MIME Standard noch nicht durchgesetzt hatte. Auch hier spielen Mailinglistenarchive eine Sonderrolle, da in diesen die Zeichencodierung der E-Mails oft schon vereinheitlicht ist.

	WWW-RDF-Int.		Linux-Kernel		B-Greek	
Nachrichtenanzahl	13574		14000		11287	
falscher Content-Type	<1%	2	0%	0	0%	0
falsche Zeichencodierung	<1%	27	<1%	7	0%	0
kein Content-Type	4%	604	7%	930	100%	11287

Tabelle 3.4: Fehlerhafte Codierung/Content-Type

3.4.3 Beiträge aus Newsgroups

Zu RFC 2822 analoge Standards, die das Format von Newsgroupbeiträgen beschreiben sind RFC 850[13] (1983) und dessen Nachfolger RFC 1036[14] (1987). Newsgroupbeiträge weisen sowohl bei den Kopfzeilen, als auch beim Format des Nachrichtentextes große Ähnlichkeiten mit E-Mails auf. Hier sind die Kopfzeilen Absender (*From*), Betreffzeile (*Subject*), Sendedatum (*Date*), *Message-ID* vorgeschrieben. Die Kopfzeilen *References* und *In-Reply-To* sind wie bei E-Mails optional, werden aber auch von fast allen Newsreadern gesetzt. Das Format der Kopfzeilen entspricht ansonsten dem bei E-Mails verwendeten Format. Da zur Übermittlung des Nachrichtentextes auch die im vorherigen Abschnitt beschrie-

[13] http://www.ietf.org/rfc/rfc0850.txt
[14] http://www.ietf.org/rfc/rfc1036.txt

benen Zeichencodierungen verwendet werden, können Newsgroupbeiträge ohne Änderung an den zugrundeliegenden Datenstrukturen oder an den eingesetzten Algorithmen von Semalan auf gleiche Weise verarbeitet werden wie E-Mails.

3.4.4 Von Semalan nutzbare Daten

Um möglichst viele der in den ursprünglichen Nachrichten enthaltenen Metadaten zu erhalten, werden alle in Tabelle 3.1 aufgeführten Kopfzeilen von Semalan gespeichert. Da das Diskussionsmodell von Semalan dafür ausgelegt ist auch von anderen Programmen gelesen zu werden, könnten später auch Kopfzeilen Verwendung finden, die von Semalan nicht genutzt werden.

Zur weiteren Verarbeitung benötigt Semalan jedoch nur Absender, Sendedatum, Message-ID und Betreffzeile einer Nachricht, die üblicherweise bei Nachrichten aus allen drei nutzbaren Datenquellen vorhanden sind und in denen gegebenenfalls, wie zuvor beschrieben, Fehler behoben werden. Ebenfalls genutzt werden die Kopfzeilen *References* und *In-Reply-To*, die allerdings nicht zwingend vorhanden sein müssen.

Semalan kann alle Nachrichten importieren, die den Nachrichtentext auch als reinen ASCII-Text enthalten und bei denen der Nachrichtentext mit einer gültigen Zeichencodierung versehen ist und die somit auch decodiert werden können. Den zuvor präsentierten Statistiken ist zu entnehmen, dass dies in den untersuchten Mailinglisten bei jeweils über 99% der E-Mails der Fall ist. Somit kann man bei unabhängigem Auftreten dieser Eigenschaften davon ausgehen, dass Semalan über 98% der E-Mails aus Mailinglisten importieren kann.

3.5 Überführung in das Diskussionsmodell

Der zweite Verarbeitungsschritt von Semalan nach dem Bereinigen der importierten Nachrichten besteht darin, diese in das Diskussionsmodell zu überführen. Einzelne Unterschritte dabei sind eine Verknüpfung der Nachrichten aufgrund ihrer Kopfzeilen, die Unterteilung des Nachrichtentextes in einzelne Textabschnitte zur Identifikation von Zitaten, Antworten und semantischen Annotationen, die Zuordnung der Zitate zu den Nachrichten, aus denen der betreffende Text zitiert wurde (im weiteren Ursprungsnachricht genannt), das Verknüpfen einzelner Textabschnitte, sowie das Speichern aller Informationen im zugrundeliegenden Diskussionsmodell.

3.5.1 Verknüpfung der Nachrichten

Jede E-Mail und auch jeder Newsgroupbeitrag enthält in den Kopfzeilen einen Identifikator, genannt Message-ID, der nach den entsprechenden Standards global eindeutig sein muss. Wird eine Antwort auf eine Nachricht mit Hilfe der

Antworten-Funktion eines E-Mail-Programms oder Newsreaders erstellt, fügen diese Programme, wie nachstehend detaillierter beschrieben, die Message-ID der beantworteten Nachricht den Kopfzeilen der Antwort hinzu. Dadurch lässt sich in einer Diskussion erkennen, in welchem äußeren Bezug einzelne Nachrichten zueinander stehen.

Ebenso wird beim Beantworten einer Nachricht üblicherweise deren Betreffzeile mit geringfügigen Änderungen als Betreffzeile der Antwort übernommen.

Dieser zwei Eigenheiten bedient sich Semalan, um einen ersten Zusammenhang zwischen den importierten Nachrichten herzustellen. Diese Vorgehensweise wird auch von Jacob Palme [Pal02] beschrieben, und von vielen E-Mail-Programmen und Newsreadern eingesetzt.

Verknüpfung über Message-IDs

Wie Abbildung 3.5 zeigt, werden beim Beantworten einer Nachricht automatisch Message-IDs aus verschiedenen Kopfzeilen der beantworteten Nachricht übernommen. Zum einen wird die *In-Reply-To* Kopfzeile erstellt, die nur aus der Message-ID der beantworteten Nachricht besteht. Zum anderen wird der Antwort eine *References* Kopfzeile hinzugefügt, die aus der beantworteten Nachricht die Message-IDs vorangehender Nachrichten übernimmt.

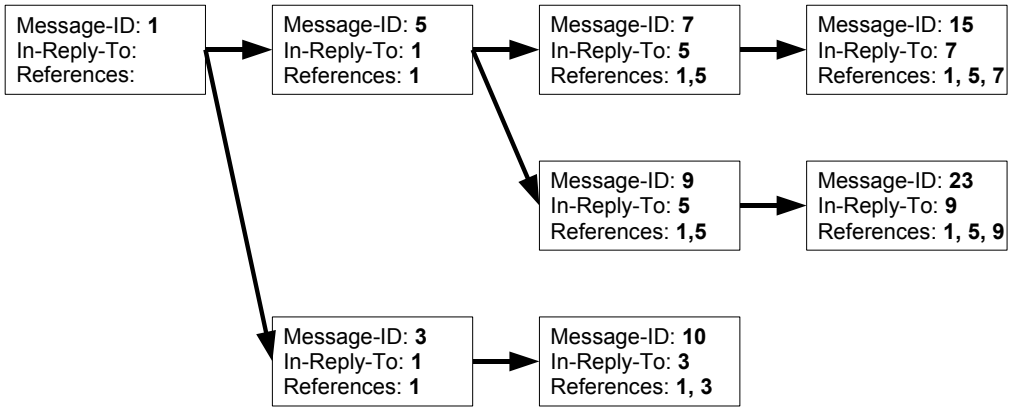

Abbildung 3.5: Message-IDs im Diskussionsbaum

So liegt im Idealfall eine Kette von Message-IDs in der korrekten Reihenfolge in der die Nachrichten beantwortet wurden vor, bis hin zur ersten Nachricht im Diskussionsbaum. Allerdings kann jedes E-Mail-Programm beim Übernehmen der *References* Kopfzeile Message-IDs aus dieser entfernen, falls die Kopfzeile zu lang ist. Ab welcher Länge dies geschieht ist nicht standardisiert und hängt vom jeweiligen E-Mail-Programm ab. Daher liefert die *References* Kopfzeile in der

Praxis zwar Informationen über die Reihenfolge einiger vorangehender E-Mails, weist aber auch oft Lücken auf.

Anhand der in den *In-Reply-To* und *References* Kopfzeilen enthaltenen Message-IDs können somit die Nachrichten in eine Baumstruktur eingeordnet werden, in der eine Nachricht immer direkt unter der von ihr beantworteten Nachricht steht.

Verknüpfung über die Betreffzeile

Eine andere Methode zur Identifikation zusammengehöriger Nachrichten bedient sich der Betreffzeilen, die sich innerhalb einer Diskussion üblicherweise nicht grundlegend ändern. Es werden von E-Mail-Programmen beim Antworten lediglich Zusätze wie *Re:* zu den Betreffzeilen hinzugefügt, durch die sich eine Antwort von der beantworteten Nachricht unterscheidet. Um zusammengehörige Nachrichten zu identifizieren wird versucht, diese Zusätze herauszufiltern und so die zugrundeliegende Basis-Betreffzeile zu finden. Allein aus den Betreffzeilen lässt sich allerdings meist nicht schließen, in welcher Beziehung einzelne Nachrichten zueinander stehen. Insbesondere ob eine Nachricht eine direkte oder nur eine indirekte Antwort auf eine andere Nachricht ist, ist hierbei nicht ersichtlich. Daher wird eine Gruppierung der Nachrichten nach Betreffzeilen von vielen E-Mail-Programmen nur ergänzend zur Verknüpfung über Message-IDs vorgenommen.

Umsetzung in Semalan

Dass eine Verknüpfung zusammengehöriger Nachrichten sowohl über die *In-Reply-To/References* Kopfzeilen, als auch über die Betreffzeile gerechtfertigt ist, zeigt Tabelle 3.5, wonach, je nach Mailingliste, über 20% aller Nachrichten, deren Betreffzeile darauf schließen lässt, dass sie eine Antwort auf eine andere Nachricht sind, weder eine *In-Reply-To*-, noch eine *References*-Kopfzeile enthalten.

	WWW-RDF-Interest		Linux-Kernel		B-Greek	
Nachrichtenanzahl	13574		14000		11287	
In-Reply-To	43%	5815	76%	10706	45%	5085
References	41%	5553	77%	10808	35%	3944
Re: in Betreff	70%	9620	73%	10164	<1%	9
Zitatzeichen in Text	72%	9728	73%	10147	61%	6880
Re: ohne References oder In-Reply-To	21%	2844	3%	380	<1%	1

Tabelle 3.5: Antwort-Indikatoren

Eine Sonderrolle spielen hierbei Mailinglistenarchive, da in diesen zum Teil die Betreffzeilen schon durch das Entfernen von für Antworten typischen Zusätzen vereinheitlicht sind.

Semalan verbindet Nachrichten ähnlich dem von Jamie Zawinski[15] beschriebenen Algorithmus, der seit mehreren Jahren erfolgreich in populären E-Mail-Programmen wie Netscape Mail oder dessen Nachfolger Mozilla Mail eingesetzt wird. Die Hauptschritte des Algorithmus zur Verknüpfung neu eingegangener Nachrichten sind:

- Einordnung einer neuen Nachricht in den Diskussionsbaum als Unterknoten der Nachrichten, deren Message-ID der in der *In-Reply-To*-Kopfzeile enthaltenen Message-ID entspricht. Etwaige, bereits existierende Verknüpfungen, die dem widersprechen werden gelöscht, da die Informationen der *In-Reply-To*-Kopfzeile zuverlässiger sind als die der *References*-Kopfzeile, aus der E-Mail-Programme fast nach belieben Message-IDs entfernen können. Existiert die in der *In-Reply-To*-Kopfzeile referenzierte Nachricht noch nicht im Diskussionsmodell, wird eine leere Nachricht mit dieser Message-ID erstellt, die als Platzhalter dient, bis die echte Nachricht eintrifft.
- Extraktion aller Message-IDs aus der *References*-Kopfzeile in umgekehrter Reihenfolge. Sofern dies nicht im Widerspruch zu bereits existierenden Verknüpfungen steht und falls dadurch keine Zyklen in der Baumstruktur entstehen, werden die Nachrichten entsprechend der Reihenfolge der Message-IDs aus der *References*-Kopfzeile in den Baum eingeordnet.
- Unabhängig davon wird aus jeder Betreffzeile eine Basis-Betreffzeile gebildet und die Nachrichten zu Gruppen mit gleicher Basis-Betreffzeile zusammengefasst. In der Standardeinstellung wird die Basis-Betreffzeile durch das Entfernen von Zusätzen, wie *Re:*, *Re[ ]:* oder *Aw:* und Anhängen wie *(was:)* oder *(war:)* gebildet, die E-Mail-Programme beim Senden von Antworten der Betreffzeile üblicherweise hinzufügen. Zudem werden Leerzeichen entfernt und nur die ersten 50 Zeichen der Betreffzeile für die Basis-Betreffzeile genutzt, da einige E-Mail-Programme zu lange Betreffzeilen automatisch kürzen oder mehrfache Leerzeichen entfernen.

Im Gegensatz zu den meisten E-Mail-Programmen versucht Semalan nicht, bei der ersten Verknüpfung Nachrichten ohne *In-Reply-To-* oder *References*-Kopfzeilen in Diskussionsbäume einzuordnen, die Nachrichten mit ähnlicher Betreffzeile enthalten. Denn allein basierend auf der Betreffzeile lässt sich höchstens ermitteln, zu welcher Diskussion eine E-Mail gehört, jedoch nicht deren Bezug zu anderen Nachrichten und die genaue Position im Diskussionsbaum. Dies könnte zu einer falschen Einordnung im Diskussionsbaum führen.

[15] http://www.jwz.org/doc/threading.html

Da Semalan die Nachrichten in einem zweiten Schritt, wie im nächsten Unterkapitel beschrieben, basierend auf deren Inhalt und enthaltenen Zitaten verknüpft, stellt es aber keinen Nachteil dar, dass die Gruppierung nach Diskussionsbaum und Betreffzeile getrennt erfolgt.

3.5.2 Textanalyse und inhaltliche Verknüpfung

Nach der Verknüpfung über die Kopfzeilen erfolgt eine weitere Verknüpfung basierend auf dem Nachrichteninhalt. Dafür wird zuerst der Text einer Nachricht in einzelne Abschnitte zerlegt, die aus anderen Nachrichten zitiert wurden oder die Antworten auf Zitate darstellen. Anschließend werden die Nachrichten und Textabschnitte aufgrund dieser Zitat-Relationen untereinander verknüpft.

Zitatidentifikation

Um zu sehen, inwiefern Zitate zum Aufteilen von Nachrichten in einzelne Textabschnitte genutzt werden können, soll zunächst ein Blick auf die Zitatkultur in Mailinglisten geworfen werden, die von drei verschiedenen Methoden, Text aus einer anderen Nachricht zu zitieren geprägt ist.

- Die gesamte zitierte Nachricht kann entsprechend den MIME Standards als Teil einer Multipart Nachricht angefügt werden.
- Der Text der zitierten Nachricht kann ohne Zitatzeichen unverändert und komplett am Ende der Antwort eingefügt werden.
- Es werden nur die jeweils zitierten Abschnitte der Nachricht mit einem Zitatzeichen am Zeilenanfang versehen in die Antwort übernommen. Die Antwort auf einen Abschnitt erfolgt direkt im Anschluss daran.

Der schon im Kapitel 1 erwähnten Netikette entsprechend wird in Mailinglisten und Newsgroups erwartet, dass letztere Zitatweise benutzt wird. Um die Nachrichten möglichst kurz und übersichtlich zu halten, sollen danach nur die Abschnitte zitiert werden, auf die tatsächlich geantwortet wird. Tabelle 3.6 zeigt, welche dieser Zitatmöglichkeiten in der Praxis wie oft genutzt wird.

	WWW-RDF-Interest		Linux-Kernel		B-Greek	
Nachrichtenanzahl	13574		14000		11287	
Multipart/Anhang	<1%	17	<1%	18	0%	0
Ohne Zitatzeichen	<1%	39	<1%	53	<1%	27
Mit Zitatzeichen	72%	9728	73%	10147	61%	6880

Tabelle 3.6: Zitatarten

Bei den ersten beiden Zitatarten, die den kompletten Text einer beantworteten Nachricht an die Antwort anfügen, ergibt sich das Problem, dass nicht ersichtlich ist, auf welchen Teil der beantworteten Nachricht sich die Antwort bezieht. Denn die zitierte Nachricht und die Antwort liegen hierbei getrennt voneinander vor. Daher ist in diesem Fall auch keine Unterteilung solch einer Nachricht in separate Textabschnitte unterschiedlichen Zitatursprungs möglich, auf denen die Abbildung auf das feingranulare Diskussionsmodell von Semalan basiert.

Aus diesem Grund beschränkt sich Semalan bei der Zitaterkennung und Zuordnung auf die dritte Zitatweise, bei der die Zitate explizit gekennzeichnet sind. Wie Tabelle 3.6 zu entnehmen ist, kommt diese Zitatweise auch bei fast allen untersuchten Nachrichten zum Einsatz.

Zum Kennzeichnen von Zitaten werden, je nach E-Mail-Programm und Newsreader, die Zeichen "**>**", "**|**" oder "**:**" dem zitierten Text vorangestellt. Wobei sich, wie Tabelle 3.7 zeigt, die Verwendung von "**>**" als Quasistandard etabliert hat. Um eine möglichst vollständige Zitaterkennung zu gewährleisten, berücksichtigt Semalan dennoch alle drei Arten von Zitatzeichen.

	WWW-RDF-Interest		Linux-Kernel		B-Greek	
Nachrichtenanzahl	13.574		14.000		11.287	
Nachrichten mit >	66%	8.994	71%	9.941	61%	6.826
Nachrichten mit |	2%	235	<1%	105	<1%	9
Nachrichten mit :	4%	499	<1%	101	<1%	45
Zeilenanzahl	827.528		1.151.111		155.035	
Zeilen mit >	28%	231.420	16%	187.896	33%	155.035
Zeilen mit |	<1%	2.487	<1%	1030	<1%	574
Zeilen mit :	<1%	2.081	<1%	275	<1%	45

Tabelle 3.7: Zitatzeichen

Extraktion von Zitaten

Wird ein bereits zitierter Textabschnitt von einer nachfolgenden Nachricht nochmals zitiert, fügen E-Mail-Programme den bereits vorhandenen Zitatzeichen jeweils ein zusätzliches Zitatzeichen pro zitierter Textzeile hinzu. Auf diese Weise entstehen bei Mehrfachzitaten Strukturen, die dem in Abbildung 3.6 abgebildeten Beispiel ähneln.

Unterteilung in Textabschnitte Daher unterteilt Semalan zunächst jede Nachricht in einzelne Abschnitte, wobei jeder Abschnitt dadurch gekennzeichnet ist, dass er einen zusammenhängenden Textblock bildet und am Zeilenanfang jeweils

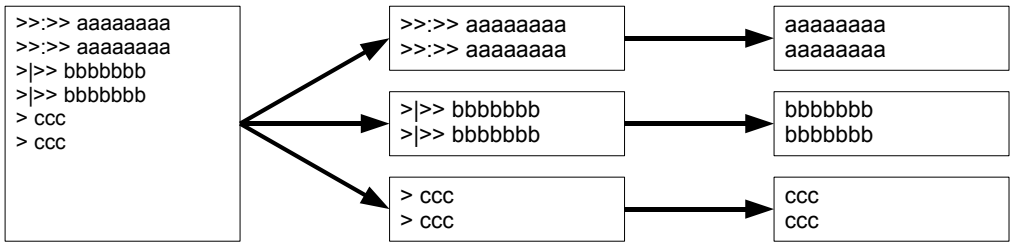

Abbildung 3.6: Zitatkennzeichnung

die gleiche Anzahl an Zitatzeichen enthält. Entsprechend der oben beschriebenen Zitatweise in Mailinglisten hat jeder dieser zitierten Textabschnitte seinen Ursprung in genau einer anderen Nachricht. Abschnitte ohne Zitatzeichen entstammen der aktuell verarbeiteten Nachricht und können Antworten auf Zitate sein. Nachfolgend wird zunächst die Verarbeitung der Textabschnitte, die Zitate aus anderen Nachrichten darstellen behandelt. Auf die Verarbeitung der Textabschnitte ohne Zitatzeichen wird am Ende von Unterkapitel 3.5.2, auf Seite 39 näher eingegangen.

Filtern von Zitaten In einem zweiten Schritt werden die Abschnitte mit Zitatzeichen betrachtet. Viele E-Mail-Programme stellen Zitaten automatisch Zusätze voran, die den Autor des zitierten Textes oder das Sendedatum der zitierten Nachricht enthalten und die fälschlicherweise als eigenständiges Zitat erkannt werden könnten.

Semalan bietet daher die Möglichkeit, nicht relevante Abschnitte auszusortieren, indem nur diejenigen Zitate weiterverarbeitet werden, die einer gegebenen Positivliste von regulären Ausdrücken entsprechen und die einer gegebenen Negativliste von regulären Ausdrücken nicht entsprechen. Um Konflikte zwischen den beiden Listen zu vermeiden, werden nur die Textabschnitte weiter betrachtet und mit der Negativliste verglichen, die erfolgreich den Vergleich mit der Positivliste passiert haben.

In der Standardeinstellung filtert Semalan Textabschnitte heraus, die nur aus einer Textzeile bestehen, mit einem Doppelpunkt enden, sowie die Wörter “wrote” oder “schrieb” enthalten. Dies verhindert größtenteils, dass Zeilen wie “On Thu, Nov 10, 2005 at 08:22:52PM +0100, Michael Meier wrote:”, die die meisten E-Mail-Programme Zitaten automatisch voranstellen, als eigenständiges Zitat interpretiert werden.

Da je nach der in einer Mailingliste vorherrschenden Sprache, dem Thema, sowie den Gepflogenheiten einer Mailingliste recht unterschiedliche Textabschnitte von der Zitaterkennung ausgenommen werden müssen, ist es kaum machbar automatisch alle nicht relevanten Textabschnitte in jeder Mailingliste zu identifizieren.

Daher bietet Semalan auch die Möglichkeit, die regulären Ausdrücke der Positiv- und Negativliste für jede Mailingliste individuell anzupassen.

Vorauswahl beim Zitatvergleich

Die Ursprungsnachricht eines Zitats zu finden erfordert einen, wie auch immer gearteten Vergleich mit anderen Nachrichten. Findet keine Vorauswahl der Nachrichten statt, die in diesen Vergleich mit einbezogen werden, muss jede neue Nachricht mit allen bereits im Diskussionsmodell vorhandenen Nachrichten verglichen werden. Dies führt mit zunehmender Anzahl der Nachrichten im Diskussionsmodell dazu, dass das Hinzufügen neuer Nachrichten immer mehr Zeit beansprucht. Da Semalan auch Mailinglisten mit mehreren zehntausend Nachrichten verwalten kann, ist es unerlässlich eine Vorauswahl zu treffen, innerhalb welcher Gruppe von Nachrichten dieser Vergleich stattfinden soll, um die Anzahl der benötigten Vergleiche zu reduzieren. Semalan bedient sich dabei der schon zuvor erfolgten ersten Verknüpfung der Nachrichten aufgrund von Kopfzeilen und beschränkt die Suche nach dem Ursprung eines Zitats auf zwei Gruppen von Nachrichten.

- Nachrichten, die sich vor der zitierenden Nachricht im gleichen Diskussionsbaum befinden.
- Nachrichten, die die gleiche Basis-Betreffzeile wie die zitierende Nachricht haben und zugleich älter als diese sind.

Die Idee dabei ist, dass eine Nachricht nicht aus einer anderen Nachricht zitieren kann, die erst danach oder als Antwort darauf geschrieben wurde. Tabelle 3.8 zeigt, dass diese Einschränkung der Suche auf eine Untermenge von Nachrichten kaum Auswirkungen auf die Anzahl der Zitate hat, die ihrem Ursprung zugeordnet werden können.

Nachrichtenanzahl	200		400		800		1600	
Ohne Vorauswahl	100%	247	100%	510	100%	1174	100%	2574
Mit Vorauswahl	93%	230	95%	486	97%	1135	97%	2493

Tabelle 3.8: Anzahl der Zitate, deren Ursprung gefunden werden kann

Der Vorteil einer Vorauswahl ist in Tabelle 3.9 und Abbildung 3.7 zu sehen, die zeigen, wieviel Zeit das Füllen des Diskussionsmodells mit neuen Nachrichten benötigt. Dabei wird die Zeit gemessen die benötigt wird, die angegebene Anzahl an Nachrichten in ein zuvor leeres Diskussionsmodell einzufügen. Wird eine Vorauswahl der möglichen Ursprungsnachrichten wie oben beschrieben durchgeführt, steigt der Zeitbedarf für den Import neuer Nachrichten nur linear mit der Anzahl

Nachrichtenanzahl	100	200	400	800	1600
Ohne Vorauswahl	66s	151s	379s	1092s	3573s
Mit Vorauswahl	7s	16s	27s	55s	115s

Tabelle 3.9: Zeitbedarf zum Einfügen neuer Nachrichten in das Diskussionsmodell

der neu zu importierenden Nachrichten und hängt nicht mehr von der Gesamtanzahl der Nachrichten im Diskussionsmodell ab. Somit kann die Zitaterkennung erheblich beschleunigt werden, da sich, insbesondere bei großen Mailinglisten, die Zahl der beim Hinzufügen neuer Nachrichten zu vergleichenden Nachrichten so von mehreren Tausend auf jeweils wenige Dutzend reduziert.

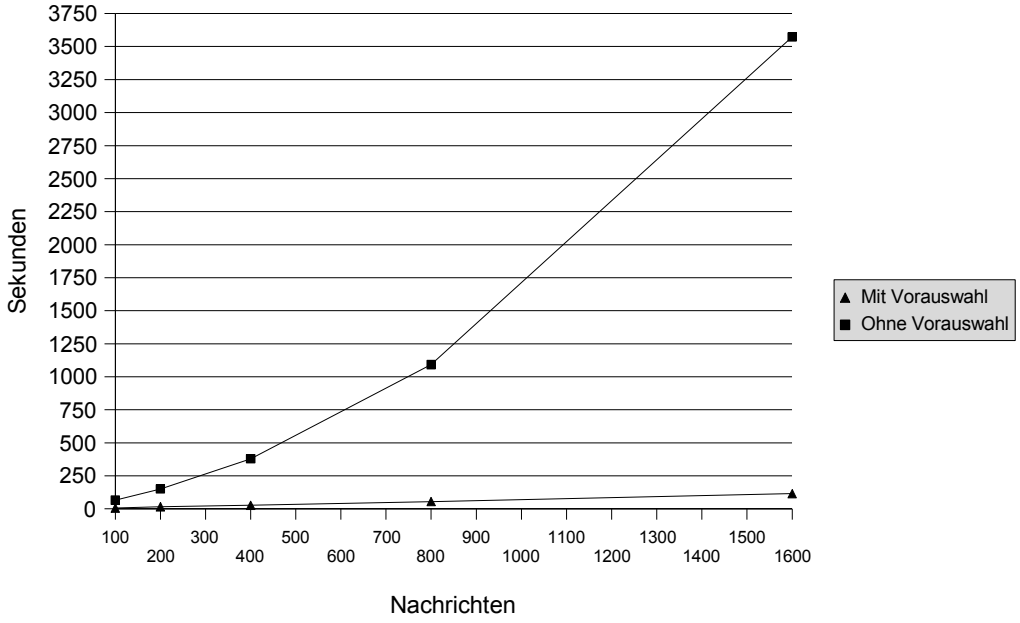

Abbildung 3.7: Zeitbedarf zum Einfügen neuer Nachrichten in das Diskussionsmodell

Suche nach dem Ursprung von Zitaten

Alle aus einer Nachricht extrahierten Zitate werden mit jeder Nachricht aus der in der Vorauswahl eingegrenzten Gruppe möglicher Ursprungsnachrichten verglichen. Um den Vergleich möglichst effizient durchzuführen werden drei Vergleichsmethoden näher betrachtet und daraufhin untersucht, inwieweit sie sich für die Suche nach dem Ursprung eines Zitats in Semalan eignen.

Zeilenbasierter Textvergleich Ein möglicher Ansatz, um den Ursprung eines Zitats zu finden ist, die Zitatzeilen nach dem Entfernen der Zitatzeichen mit den Zeilen der Nachrichten zu vergleichen, die als Ursprung in Frage kommen. Dieser zeilenbasierte Ansatz hat sich in Tests jedoch nicht als praktikabel erwiesen, da viele E-Mail-Programme und Newsreader beim Beantworten einer Nachricht, je nach Länge der zitierten Textzeilen, zusätzliche Zeilenumbrüche in den Zitattext einbauen und somit die Einteilung der Zeilen verändern. Ab welcher Zeilenlänge dies geschieht, unterscheidet sich von E-Mail-Programm zu E-Mail-Programm. Dadurch kann sich eine Textzeile beim Zitieren auf mehrere Zeilen aufteilen, was zeilenbasierte Vergleiche scheitern lässt.

Vergleich von Prüfsummen Beim zweiten untersuchten Ansatz werden aus den Zitaten und den Nachrichtentexten Prüfsummen gebildet und der Ursprung eines Zitats über einen Vergleich dieser Prüfsummen bestimmt. Dabei kann jedoch nicht nur eine Prüfsumme aus dem kompletten Text eines Zitats berechnet werden. Denn dies würde für die Berechnung der zweiten, zu vergleichenden Prüfsumme erfordern, dass schon im Voraus bekannt ist, ob das Zitat aus einer gegeben Nachricht stammt oder nicht und wo in der Nachricht sich das Zitat genau befindet. Aber gerade dies soll durch die Suche ja erst ermittelt werden.

Eine mögliche Lösung dieses Problems basiert auf der von Paula S. Newman [New01] beschriebenen Idee, Vergleiche nach Sätzen anstelle von Zeilen durchzuführen. Dabei wird aus dem Anfang eines Satzes und bei längeren Sätzen auch aus dem Satzende eine Prüfsumme berechnet. Somit wird der zitierte Text durch eine Reihe von Prüfsummen repräsentiert. In gleicher Weise wird mit jeder Nachricht verfahren, die sich in der Gruppe der möglichen Ursprungsnachrichten des Zitats befindet. Falls sich in einer der Nachrichten die gleiche Folge von Prüfsummen findet, aus der das Zitat besteht und dort den entsprechenden Zeilen keine Zitatzeichen vorangestellt sind, ist dies die Quelle des Zitats. Dieser Ansatz liefert wesentlich bessere Ergebnisse als der zuvor beschriebene, zeilenbasierte Ansatz, da die Zitaterkennung hier unabhängig von der Zeilenpartitionierung ist.

Textvergleich ganzer Zitate Wie in Abbildung 3.8 illustriert, werden bei der dritten Methode aus dem zitierten Text zunächst die den Zeilen vorangestellten Zitatzeichen entfernt. Anschließend werden aus dem verbleibenden Text die Zeilenumbrüche und alle Zeichen, mit Ausnahme alphanumerischer Zeichen entfernt. Ebenso wird mit dem Text der zu vergleichenden Nachrichten verfahren mit dem Unterschied, dass hier die Zitatzeichen erhalten bleiben. So wird sichergestellt, dass nicht eine andere Nachricht, die denselben Text zitiert, fälschlicherweise als Quelle des Zitats identifiziert wird. Anschließend wird über einen einfachen Textvergleich überprüft, ob die aus dem Zitat gebildete Zeichenkette in einer der aus den möglichen Ursprungsnachrichten gebildeten Zeichenketten enthalten ist.

Wird eine solche Übereinstimmung gefunden, ist dies die Nachricht, aus der das Zitat ursprünglich stammt.

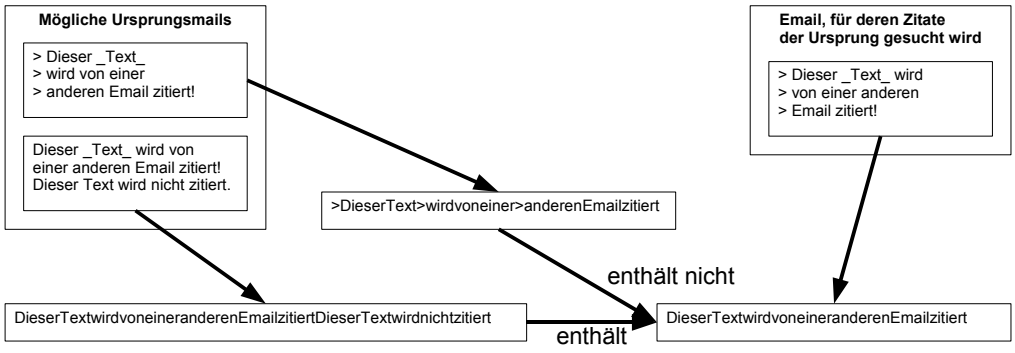

Abbildung 3.8: Textvergleich von Zitaten und Nachrichten

Die in Semalan verwendete Methode Der auf Prüfsummen basierte Vergleich und der Textvergleich ganzer Zitate stützen sich beide allein auf den Zitattext und sind somit unempfindlich gegenüber Änderungen der Zeilenformatierung und Zeilenpartitionierung beim Zitieren. Daher liefern beide Vergleiche gute Ergebnisse. Ein Problem beim Prüfsummen-Vergleich auf Satzebene ist jedoch, dass dieser nicht funktioniert, wenn nur Teile eines Satzes zitiert werden. Eine mögliche Lösung hierfür wäre, Prüfsummen nicht auf Satzebene, sondern für jedes einzelne Wort zu berechnen. Dies ist jedoch nicht praktikabel, da dadurch sowohl die Zahl der zu berechnenden Prüfsummen und somit die zum Berechnen benötigte Zeit, als auch der dafür benötigte Speicherplatz deutlich zunehmen würden. Der Textvergleich ist dagegen, unabhängig von den zugrundeliegenden Satzstrukturen, auf jede Art von Textabschnitt anwendbar.

Ein weiterer Nachteil des Prüfsummen-Vergleichs auf Satzebene zeigt sich bei nicht natürlichsprachlichen Texten, in denen keine Unterteilung in einzelne Sätze vorgenommen werden kann. Ein Beispiel hierfür ist der Quellcode von Programmen, der oft in technisch orientierten Mailinglisten wie *Linux-Kernel* zu finden ist. Hier bleibt nur das Ausschließen dieser Abschnitte von der Zitaterkennung, der Rückfall auf die Berechnung einer Prüfsumme für jedes Wort, oder spezielle Methoden zu Aufteilung nicht natürlichsprachliche Texte. Im Gegensatz dazu funktioniert der Textvergleich gleichermaßen mit nicht natürlichsprachlichen Texten.

Ebenfalls nachteilig bei Prüfsummen-Vergleichen ist die notwendige Speicherung der Prüfsummen. Ein auf Prüfsummen basierter Vergleich macht gegenüber einem reinen Textvergleich nur Sinn, wenn die für einen Text berechneten Prüfsummen zur späteren Wiederverwendung gespeichert werden und somit nicht bei

jedem Vergleich neu berechnet werden müssen. Gegenüber dem Textvergleich, bei dem keine zusätzlichen Daten gespeichert werden müssen, erhöht sich beim Prüfsummen-Vergleich somit sowohl der Speicherplatzbedarf für das Diskussionsmodell, als auch der Aufwand beim Umgang mit den darin enthaltenen Daten.

Daher kommt bei Semalan der in Abbildung 3.8 gezeigte Textvergleich zum Einsatz.

Verknüpfung von Zitaten und Antworten

Als letzter Schritt der inhaltlichen Verknüpfung werden auch einzelne Textabschnitte miteinander verknüpft. Wie bereits zuvor beschrieben, wird von Semalan jede Nachricht in einzelne Textabschnitte unterteilt, die sich durch die Anzahl der vorangestellten Zitatzeichen unterscheiden. Folgt in einer Nachricht auf einen Textabschnitt mit Zitatzeichen direkt ein Textabschnitt ohne Zitatzeichen, wird entsprechend der in E-Mails und Newsgroups üblichen Zitatweise davon ausgegangen, dass der Textabschnitt ohne Zitatzeichen eine Antwort auf den vorherigen Textabschnitt darstellt. Dies wird im Diskussionsmodell durch eine entsprechende Verknüpfung der beiden Textabschnitte repräsentiert.

3.5.3 Auswertung semantischer Annotationen

Eng mit der Zitaterkennung verbunden ist die Auswertung semantischer Annotationen, mit denen einzelne Textabschnitte kommentiert werden können. Im Gegensatz zu den im vorherigen Abschnitt beschriebenen allgemeinen Zitat-Antwort-Relationen zwischen einzelnen Textabschnitten dienen semantische Annotationen dazu, Teilnehmer einer Mailingliste explizit über einen Textabschnitt abstimmen zu lassen.

Der Autor einer Nachricht kann eine solche Annotation vornehmen, indem er direkt im Anschluss an den Textabschnitt den er kommentieren möchte einen einzeiligen Kommentar anfügt, der mit einer speziellen Markierung beginnen muss. Standardmäßig ist diese Markierung *[SEMPOLL]*, kann jedoch in den Einstellungen von Semalan beliebig geändert und so der Sprache und dem Verwendungszweck in jeder Mailingliste angepasst werden.

Erkennt Semalan im Anschluss an einen zuvor identifizierten Textabschnitt eine solche semantische Annotation anhand deren Markierung, wird zunächst geprüft, ob der gleiche Textabschnitt bereits einmal kommentiert worden ist. Wenn dies der Fall ist, wird die neue Annotation der existierenden Abstimmung hinzugefügt und vermerkt, vom wem die Annotation stammt. Sonst wird eine neue Abstimmung erstellt. Es wird immer sichergestellt, dass zu einer Abstimmung nicht mehr als eine Annotation von einem Mailinglisten-Teilnehmer vorliegt, indem mehrfache Annotationen eines Teilnehmers zum gleichen Textabschnitt verworfen werden. Dies funktioniert allerdings nur, wenn ein Teilnehmer nicht unter

verschiedenen Namen und mit unterschiedlichen E-Mail-Adressen auftritt.

Die oben beschriebene Vorgehensweise hat den Vorteil, dass die Mailinglisten-Teilnehmer nicht explizit neue Abstimmungen erstellen müssen. Es genügt, als Erster einen Textabschnitt zu annotieren, um automatisch eine neue Abstimmung zu erstellen. Auch können die Annotationen flexibel interpretiert werden und sind nicht nur auf Abstimmungen beschränkt. Semalan aggregiert bei der Präsentation identische Annotationen und gibt an, wie oft die jeweilige Annotation vorkommt. Dadurch sind nicht nur Abstimmungen mit Antworten wie *Ja* und *Nein* oder einer kleinen Anzahl an Antwortmöglichkeiten denkbar. Es können zum Beispiel auch Ideensammlungen durchgeführt werden, bei denen von jedem Teilnehmer unterschiedliche Antworten gegeben werden.

3.6 Benutzungsschnittstellen

Da Semalan den Nutzern von Mailinglisten und Newsgroups in erster Linie ergänzende Informationen liefern soll, die herkömmliche Darstellungen nicht bieten, stehen bei den Abfrage- und Präsentationsmöglichkeiten der Benutzungsschnittstellen Zusammenhänge zwischen Nachrichten und Zitaten, sowie Abstimmungen im Vordergrund. Insbesondere ist Semalan in der Lage folgende, zum Teil bereits in Kapitel 1 erwähnte Fragen zu beantworten:

- Wie oft, und von welchen anderen Nachrichten wird eine Nachricht zitiert?
- Welche anderen Nachrichten zitiert eine Nachricht?
- Welche Textabschnitte werden in einer Gruppe von Nachrichten besonders häufig zitiert?
- Welche anderen Nachrichten haben denselben Textabschnitt zitiert?
- Wie wurde ein Textabschnitt oder eine Abstimmung von anderen Diskussionsteilnehmern kommentiert, beziehungsweise beantwortet?
- Welche Diskussionsteilnehmer haben wie viele Nachrichten zu einer Gruppe von Nachrichten beigesteuert?

Die Strukturierung und Verknüpfung der aufbereiteten Nachrichten durch das zugrundeliegende Diskussionsmodell erlaubt, die darin enthaltenen Informationen auf vielfältige Weise miteinander zu kombinieren und unter Betrachtung verschiedener Teilaspekte zu präsentieren. Semalan bietet hierfür drei verschiedene, einander ergänzende Benutzungsschnittstellen an.

- Eine textbasierte Abfrage per E-Mail.
- Eine Web-Schnittstelle mit REST-artiger Adressierung und HTML-Darstellung.
- Eine interaktive, grafische Web-Schnittstelle zur gleichzeitigen Visualisierung verschiedener Zusammenhänge.

Die Gemeinsamkeiten, sowie Vor- und Nachteile der einzelnen Benutzungsschnittstellen werden nachfolgend näher beschrieben.

3.6.1 E-Mail-Abfragen

Mailinglisten und Newsgroups werden primär in Verbindung mit E-Mail-Programmen oder Newsreadern genutzt. Daher ist es sinnvoll auch eine Möglichkeit zu bieten, die von Semalan aufbereiteten Informationen auf dem gleichen Weg abzurufen. Dies soll mit jedem E-Mail-Programm möglich sein und keine Modifikationen an den Programmen erfordern. Aus diesem Grund kommt in Semalan eine eigene, leicht erlernbare Abfragesprache zum Einsatz, bei der alle für eine Abfrage notwendigen Informationen im Nachrichtentext und der Betreffzeile enthalten sind. Die Abfrage-E-Mails sind an eine gesonderte E-Mail-Adresse zu senden, die von Semalan in regelmäßigen Abständen auf neue Nachrichten hin überprüft wird. Die Antwort erfolgt ebenfalls per E-Mail.

Eine Abfrage-E-Mail bietet den Vorteil, dass das Ergebnis der Abfrage vom E-Mail-Programm des Empfängers automatisch gespeichert wird und durch Weiterleiten der E-Mail auch einfach an andere Personen weitergegeben werden kann.

Die Abfragesprache ist möglichst einfach gehalten, da zu erwarten ist, dass die Bereitschaft der meisten Mailinglisten-Nutzer, zu diesem Zweck eine komplexe Abfragesprache zu erlernen gering ist. Die Abfrage per E-Mail ist in erster Linie dazu gedacht, einen ersten Überblick über einzelne Diskussionen, Nachrichten oder Textabschnitte zu geben. Da sich aufgrund der eingeschränkten Strukturierungsmöglichkeiten in E-Mails komplexe Zusammenhänge und größere Datenmengen nur schlecht darstellen lassen, enthalten die Antworten auf E-Mail-Abfragen zusätzlich URLs zu der in Abschnitt 3.6.2 beschriebenen REST-artigen Schnittstelle, die nähere Informationen zu einzelnen Textabschnitten, Nachrichten und Abstimmungen bietet.

Auswahl der Mailingliste Die Betreffzeile einer Abfrage-E-Mail darf nur den Namen der Mailingliste enthalten, auf die sich die Abfrage bezieht. Dies ermöglicht Semalan, das in der Lage ist mehrere Mailinglisten gleichzeitig zu verwalten, eine Zuordnung einer Abfrage zur richtigen Mailingliste.

Auswahl der angeforderten Daten Der Nachrichtentext einer Abfrage-E-Mail ist in drei Bereiche unterteilt:

- Die erste Zeile muss aus einem der vier Schlüsselwörter *Getmessages*, *Getcitations*,*Getpolls* oder *Getparticipants* bestehen, die festlegen, welche Art von Information angefordert wird.

- Die zweite Zeile der E-Mail muss ein weiteres Schlüsselwort enthalten, das vom ersten Schlüsselwort abhängt und über das aus diesen Informationen eine Untermenge ausgewählt wird.
- Alle weiteren Zeilen der E-Mail enthalten, je nach Art der verwendeten Schlüsselwörter, Message-IDs, Betreffzeilen oder Textabschnitte, auf die sich die angeforderten Informationen beziehen sollen.

Um die Abfrage fehlertoleranter zu gestalten, wird bei den Schlüsselwörtern nicht zwischen Groß- und Kleinschreibung unterschieden und innerhalb eines Schlüsselwortes dürfen auch beliebig Leerzeichen eingefügt werden.

Schlüsselwörter der Abfragesprache

Getmessages Das *Getmessages* Schlüsselwort gibt, wie in Abbildung 3.9 illustriert, eine Liste von Nachrichten zurück, die den Auswahlkriterien eines der folgenden sekundären Schlüsselwörter entsprechen. In der Antwort sind für jede Nachricht Betreffzeile, Absender, Sendedatum und Message-ID enthalten, Informationen darüber, wie häufig diese Nachricht zitiert wird und wie häufig sie andere Nachrichten zitiert, sowie eine URL, unter der noch mehr Informationen zu der Nachricht und den Zitatzusammenhängen abgerufen werden können.

Id wählt eine Nachricht zu einer angegebenen Message-ID aus.

Threadup wählt zu einer angegebenen Message-ID alle in direkter Linie vorangehenden Nachrichten im Diskussionsbaum aus.

Threaddown wählt zu einer angegebenen Message-ID alle direkt oder indirekt nachfolgenden Nachrichten im Diskussionsbaum aus.

Sendername wählt zu einem angegebenen Namen alle Nachrichten dieser Person aus.

Senderemail wählt zu einer angegebenen E-Mail-Adresse alle von dieser Adresse gesendeten Nachrichten aus.

Subject wählt zu einer angegebenen Betreffzeile alle Nachrichten mit ähnlicher Betreffzeile aus.

Getcitations Dieses Schlüsselwort liefert eine Liste von zitierten Textabschnitten, sowie eine URL auf jedes dieser Zitate in der REST-Schnittstelle. Sekundäre Schlüsselwörter für *Getcitations* sind:

Bytext gibt Zitate mit dem als drittem Argument angegebenen Zitattext zurück.

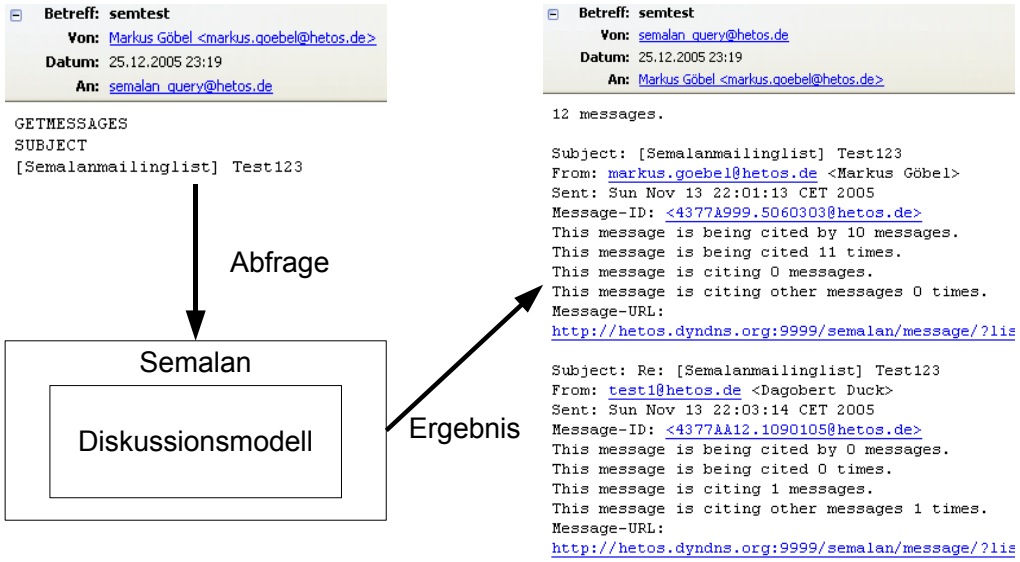

Abbildung 3.9: Beispiel einer Abfrage per E-Mail

Threadup liefert zu einer angegebenen Message-ID alle Zitate, die in direkter Linie vorangehend im Diskussionsbaum vorkommen.

Threaddown liefert zu einer angegebenen Message-ID alle Zitate, die direkt oder indirekt nachfolgend im Diskussionsbaum vorkommen.

Subject gibt alle Zitate zurück, die in E-Mails vorkommen, deren Betreffzeile der angegebenen Betreffzeile ähnelt.

Getpolls Das *Getpolls* Schlüsselwort liefert Informationen zu Abstimmungen, sowie zugehörige URLs, unter denen Details zu jeder Abstimmung und deren Ergebnisse über die REST-Schnittstelle eingesehen werden können. Die sekundären Schlüsselwörter zu *Getpolls* lauten:

All gibt eine Liste aller existierenden Abstimmungen zurück.

Bytext liefert in Verbindung mit einem darauffolgenden Textabschnitt die zu diesem Text gehörige Abstimmung, falls schon Antworten dazu existieren.

Getparticipants In Verbindung mit einem der sekundären Schlüsselwörter *Subject*, *Threadup*, *Threaddown* wird eine aggregierte Liste der Personen zurückgegeben, die Nachrichten mit ähnlicher Betreffzeile oder Nachrichten im Diskussionsbaum aufwärts oder abwärts geschrieben haben.

3.6.2 REST-Schnittstelle

Ein Nachteil einer rein textbasierten Abfrage per E-Mail ist, dass die in der Antwortmail enthaltenen Informationen statisch sind. Verändert sich das Ergebnis der Abfrage, müsste diese jedes Mal wiederholt werden, um ein aktuelles Ergebnis zu erhalten. Im Falle von Semalan tritt dies zum Beispiel ein, wenn zu einer Abstimmung neue Antworten hinzukommen oder eine Nachricht erneut zitiert wird. Daher bindet Semalan, wie schon im vorangehenden Abschnitt erwähnt, URLs in die Antwort-E-Mails ein, die eine Verbindung zur zweiten, REST-ähnlichen Benutzungsschnittstelle schaffen, mit der sich dieses Unterkapitel befasst.

Da die Semalan Datenverwaltung auf RDF basiert und auch RDF, ebenso wie REST, durch URIs adressierbare Ressourcen verwendet, bietet es sich an, die RDF-Ressourcen direkt über REST-ähnliche Webservices zugänglich zu machen.

Aus dem REST-Konzept übernimmt Semalan die eindeutige Adressierung von Ressourcen über URLs, die direkt aus den zugrundeliegenden RDF-URIs gebildet werden. Zudem sind zusammenhängende Ressourcen, wie nachfolgend beschrieben, auch untereinander verlinkt. Weitere REST-Funktionalität wie das Hinzufügen, Verändern oder Löschen von Ressourcen bietet Semalan nicht, da die primären Datenquellen zum Importieren neuer Nachrichten, wie in Kapitel 3.1 beschrieben, Mailserver, Newsserver und Mailinglistenarchive sind und derzeit keine Anwendungen oder Webservices existieren, die einen Zugriff auf Mailinglisten und Newsserver per REST-Schnittstelle anbieten. Die Ressourcen werden von Semalan mittels XHTML Seiten präsentiert und können somit mit jedem gängigen Webbrowser betrachtet und auch einfach maschinell ausgelesen und weiterverarbeitet werden.

Nachrichten Die Nachrichtenansicht der REST-Schnittstelle ist in vier Bereiche unterteilt.

- Im ersten, unveränderlichen Bereich wird der Nachrichtentext, sowie die wichtigsten Kopfzeilen (Absender, Sendedatum, Betreff und Message-ID) angezeigt.
- Im zweiten, ebenfalls statischen Bereich werden die Nachrichten aufgelistet, die von der aktuell angezeigten Nachricht zitiert werden.
- Der dritte Bereich zeigt, welche anderen Nachrichten zur Zeit im Diskussionsmodell gespeichert sind, die Textabschnitte aus dieser Nachricht zitieren. Dadurch bekommt man bei jedem Aufruf einer Nachrichten-URL einen aktuellen Überblick, welche direkten oder indirekten Antworten auf diese Nachricht vorliegen. Sämtliche Nachrichten und Zitate auf die in diesem Zusammenhang verwiesen wird, sind durch ihre URLs referenziert. Dies ermöglicht eine komfortable Navigation innerhalb mehrerer, einander zitierender Nachrichten. Zudem wird aufgelistet, welche Textabschnitte jede

Nachricht aus der aktuell angezeigten Nachricht zitiert.
- Im vierten Bereich sind URLs zu allen Abstimmungen enthalten, auf die in der aktuell angezeigten Nachricht geantwortet wird.

Zitate Die in Abbildung 3.10 gezeigte Zitatansicht enthält über einen in einer Nachricht zitierten Textabschnitt die folgenden Informationen:

- Es werden sowohl die zitierende Nachricht, als auch die zitierte Nachricht inklusive der wichtigsten Kopfzeilen angezeigt. Der zitierte Textabschnitt wird in beiden Nachrichten hervorgehoben. Dies soll helfen das betreffende Zitat zu identifizieren, falls mehrere Textabschnitte unabhängig voneinander aus derselben Nachricht zitiert werden.
- Alle Zitatstellen aus anderen Nachrichten werden aufgelistet, die denselben Textabschnitt zitieren wie das aktuell angezeigte Zitat. Für jede dieser Zitatstellen ist eine URL vorhanden, über die man zu deren REST-Darstellung gelangt. Des weiteren wird für jede Zitatstelle angezeigt, was in der zitierenden Nachricht auf das Zitat geantwortet wird. Somit kann man direkt vergleichen, was verschiedene Diskussionsteilnehmer zu dem gleichen zitierten Textabschnitt geäußert haben.
- Es werden alle Zitatstellen aus anderen Nachrichten aufgelistet, die zwar nicht genau den denselben Textabschnitt zitieren, die jedoch einen Teil dieses Textabschnittes enthalten. Auch hier sind wieder auf die REST-Darstellung der Zitatstellen verweisende URLs vorhanden. Zudem wird angegeben, welcher Teil des Textabschnittes dort zitiert wird. Somit hat man eine direkte Verbindung zu anderen Diskussionssträngen, die einzelne Aspekte des aktuell angezeigten Textabschnittes thematisieren und daher mit diesem wahrscheinlich thematisch verwandt sind.

Abstimmungen Die Abstimmungs- oder Kommentaransicht besteht aus dem Text der Abstimmung, beziehungsweise des kommentierten Abschnittes, einer aggregierten Liste der aktuell dazu vorliegenden Kommentare, sowie einer Liste der Diskussionsteilnehmer, die an der Abstimmung teilgenommen haben.

Citing Message:

From: test2@hetos.de(Donald Duck)
Date: Mon Jan 02 19:21:31 CET 2006
Subject: Re: [Semalanmailinglist] Thunderbird Spamfilter
Message-ID: <43B96F2B.2040608@hetos.de>

0001: Dagobert Duck schrieb:
0002: **> gibt es eine Möglichkeit, den Spamfilter in Thunderbird auch auf**
0003: **> Newsgroup-Beiträge anzuwenden? Ich benutze Thunderbird 1.5 und bekomme**
0004: **> immer eine komische Fehermeldung, wenn ich dort einen Newsgroup-Beitrag**
0005: **> als Spam markiere.**
0006:
0007: Nein, das funktioniert afaik nicht.
0008:
0009: Mfg,
0010: Donald

Original Message:

From: test1@hetos.de(Dagobert Duck)
Date: Mon Jan 02 19:20:26 CET 2006
Subject: [Semalanmailinglist] Thunderbird Spamfilter
Message-ID: <43B96EEA.80304@hetos.de>

0001: Hallo,
0002:
0003: **gibt es eine Möglichkeit, den Spamfilter in Thunderbird auch auf**
0004: **Newsgroup-Beiträge anzuwenden? Ich benutze Thunderbird 1.5 und bekomme**
0005: **immer eine komische Fehermeldung, wenn ich dort einen Newsgroup-Beitrag**
0006: **als Spam markiere.**
0007:
0008: Gruß,
0009: Dagobert

The following citations are citing/answering the same text:

Mon Jan 02 19:23:16 CET 2006
From: test3@hetos.de(Gustav Gans)
Answer to Citation:

Wie soll das denn auch gehen? Thunderbird kann ja schließlich nicht
einfach irgendwelche Beiträge aus der Newsgroup löschen.

ciao,
Gustav

The following citations from the same message are not identical to, but a subset of this citation:

Mon Jan 02 19:23:55 CET 2006
From: markus.goebel@hetos.de(Markus Göbel)
Cited Text:
> Ich benutze Thunderbird 1.5 und bekomme
> immer eine komische Fehermeldung, wenn ich dort einen Newsgroup-Beitrag
> als Spam markiere.

Abbildung 3.10: Zitatansicht der REST-Schnittstelle von Semalan

Die in der Abbildung unterstrichen dargestellten URLs verweisen auf andere, über die REST-Schnittstelle zugängliche Ressourcen.

3.6.3 Grafische Weboberfläche

Die dritte Benutzungsschnittstelle ist eine grafische Web-Schnittstelle die dazu dient, Informationen darzustellen, die sich nur schwer über die beiden anderen, rein textbasierten Schnittstellen vermitteln lassen. Insbesondere dynamisch auf Benutzereingaben reagierende Darstellungen oder die Visualisierung von Zusammenhängen zwischen Nachrichten ist mit grafischen Mitteln leichter möglich, als mit einer Textdarstellung.

In der in Abbildung 3.11 gezeigten grafischen Web-Schnittstelle von Semalan werden eine sequentielle Darstellung und eine Graphendarstellung zur Visualisierung von Nachrichten und Zusammenhängen zwischen Nachrichten kombiniert.

Verwaltet Semalan mehrere Mailinglisten, kann über das Auswahlfeld im unteren Teil der Weboberfläche gewählt werden, welche dieser Mailinglisten dargestellt werden soll.

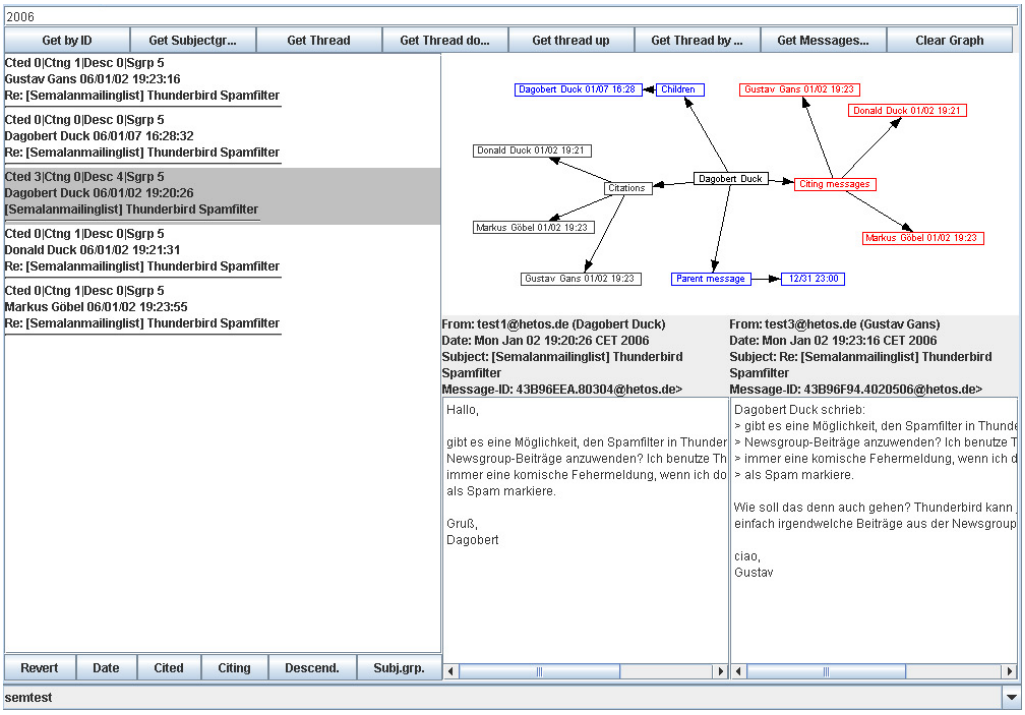

Abbildung 3.11: Grafische Benutzungsoberfläche von Semalan

Sequentielle Darstellung Die Liste links im Bild enthält Nachrichten in einer sequentiellen Darstellung, wie sie von vielen E-Mail-Programmen verwendet wird und die schon in Kapitel 2.1.1 beschrieben wurde.

Welche Nachrichten in der Liste enthalten sind, kann der Benutzer über Suchanfragen selbst festlegen. Dabei ist es möglich, Nachrichten aus einer bestimmten Diskussion, über die sich der Benutzer informieren möchte auszuwählen. Hierfür können, ausgehend von einer durch Angabe der Message-ID auszuwählenden Nachricht, alle darauffolgenden Nachrichten im gleichen Diskussionsbaum oder alle Nachrichten mit ähnlicher Betreffzeile in die Liste übernommen werden. Es kann aber auch ganz allgemein, ohne Fokus auf eine bestimmte Diskussion durch die Mailingliste geblättert werden, indem alle Nachrichten oder Diskussionen einer anzugebenden Zeitspanne angezeigt werden.

Die einzelnen Nachrichten werden in der Liste durch ihre wichtigsten Kopfzeilen, den Absender, das Sendedatum und die Betreffzeile repräsentiert. Darüber hinaus werden, was normalerweise bei sequentiellen E-Mail-Darstellungen nicht der Fall ist, für jede Nachricht Informationen über deren Bedeutung im Diskussionsbaum angezeigt. Dies sind:

- Die Anzahl der Nachrichten, aus denen diese Nachricht Textabschnitte zitiert.
- Die Anzahl der Nachrichten, die aus dieser Nachricht Text zitieren.
- Die Anzahl der nachfolgenden Nachrichten im gleichen Diskussionsbaum.
- Die Anzahl der Nachrichten mit ähnlicher Betreffzeile.

Dies soll helfen, auf einen Blick für die Diskussion relevante von weniger relevanten Nachrichten zu unterscheiden. Dem liegt die Annahme zugrunde, dass für eine Diskussion insbesondere diejenigen Nachrichten wichtig sind, die besonders häufig zitiert und somit diskutiert werden. Auch die Nachrichten, die viele Zitate aus anderen Nachrichten enthalten und sich so zu diesen äußern, werden hierdurch besonders betont.

Nachrichten in der Liste können nach verschiedenen Kriterien, wie dem Absendedatum, der Zitathäufigkeit oder der Anzahl verwandter Nachrichten sortiert werden, um es dem Benutzer zu erleichtern, die Minima und Maxima dieser Werte in der Liste zu finden.

Die sequentielle Darstellung wurde gewählt, weil sich so eine größere Anzahl an Nachrichten kompakt darstellen lässt, als bei stärker strukturierten Darstellungsweisen. Zudem kann eine Liste einfach nach unterschiedlichen Kriterien sortiert werden, was bei Baum- und Graphendarstellungen nicht immer möglich ist. Der in Abschnitt 2.1.1 erwähnte Nachteil der sequentiellen Darstellung, die mangelnde Strukturierung, spielt hierbei keine Rolle, da die Diskussionsstruktur in der anschließend erläuterten Graphendarstellung zum Ausdruck kommt.

Graphendarstellung Die zweite, in der grafischen Benutzungsschnittstelle verwendete Darstellungsweise kombiniert eine Graphendarstellung mit zwei Textfeldern und ist in Abbildung 3.12 in einer Detailansicht zu sehen. Wählt der

Benutzer in der sequentiellen Darstellung aus der Liste von Nachrichten eine Nachricht aus, wird deren Position in der Diskussionsstruktur mit Hilfe des Graphen visualisiert.

Dazu wird diese Nachricht als zentraler Knoten des Graphen dargestellt und der Nachrichtentext, sowie einige Kopfzeilen im linken der beiden Textfelder angezeigt. Alle anderen Nachrichten und Zitate, die mit der Nachricht in Verbindung stehen, werden zu Gruppen zusammengefasst und als Unterknoten dem Graphen hinzugefügt. Dies sind Nachrichten, aus denen die ausgewählte Nachricht Textstellen zitiert, von denen sie zitiert wird, sowie im Diskussionsbaum direkt vorangehende oder nachfolgende Nachrichten, die in keiner Zitat-Relation zu der Nachricht selbst stehen. Des Weiteren werden auch Textstellen, die von anderen Nachrichten aus der gewählten Nachricht zitiert werden, durch Knoten im Graphen repräsentiert.

Führt der Benutzer den Mauszeiger über einen Knoten im Graphen, der ein Zitat repräsentiert, wird direkt im Graphen angezeigt, welcher Textabschnitt darin zitiert wird. Bei einem Mausklick auf einen Zitatknoten werden hingegen nähere Informationen zu diesem Zitat im rechten Textfeld dargestellt. Dies sind, wie Abbildung 3.12 zu entnehmen ist, der zitierte Text selbst, sowie eine Liste aller Nachrichten, die den gleichen Text zitiert haben. Bei den Nachrichten, die eine direkte Antwort auf das Zitat enthalten, wird zudem auch diese Antwort angezeigt.

Analog wird bei einem Klick mit der linken Maustaste auf einen Knoten, der eine Nachricht repräsentiert, der Text der Nachricht im rechten Textfeld dargestellt, wobei Zitate zwecks besserer Übersichtlichtkeit nur gekürzt angezeigt werden. Wird hingegen mit der mittleren Maustaste auf einen Nachrichtenknoten geklickt, wird diese Nachricht zur neuen, zentralen Nachricht des Graphen.

Mit der Grapenhdarstellung kann somit schnell ein Überblick gewonnen werden, welche Teile einer Nachricht wo im Diskussionsverlauf diskutiert werden und wie darauf geantwortet wird.

Zusammenfassung Zusammenfassend lässt sich sagen, dass über die Listendarstellung diejenigen Nachrichten identifiziert werden können, die zentrale Punkte einer Diskussion darstellen und man sich dann über die Graphendarstellung schnell informieren kann, wie und wo Textstellen aus solchen zentralen Nachrichten weiter diskutiert werden.

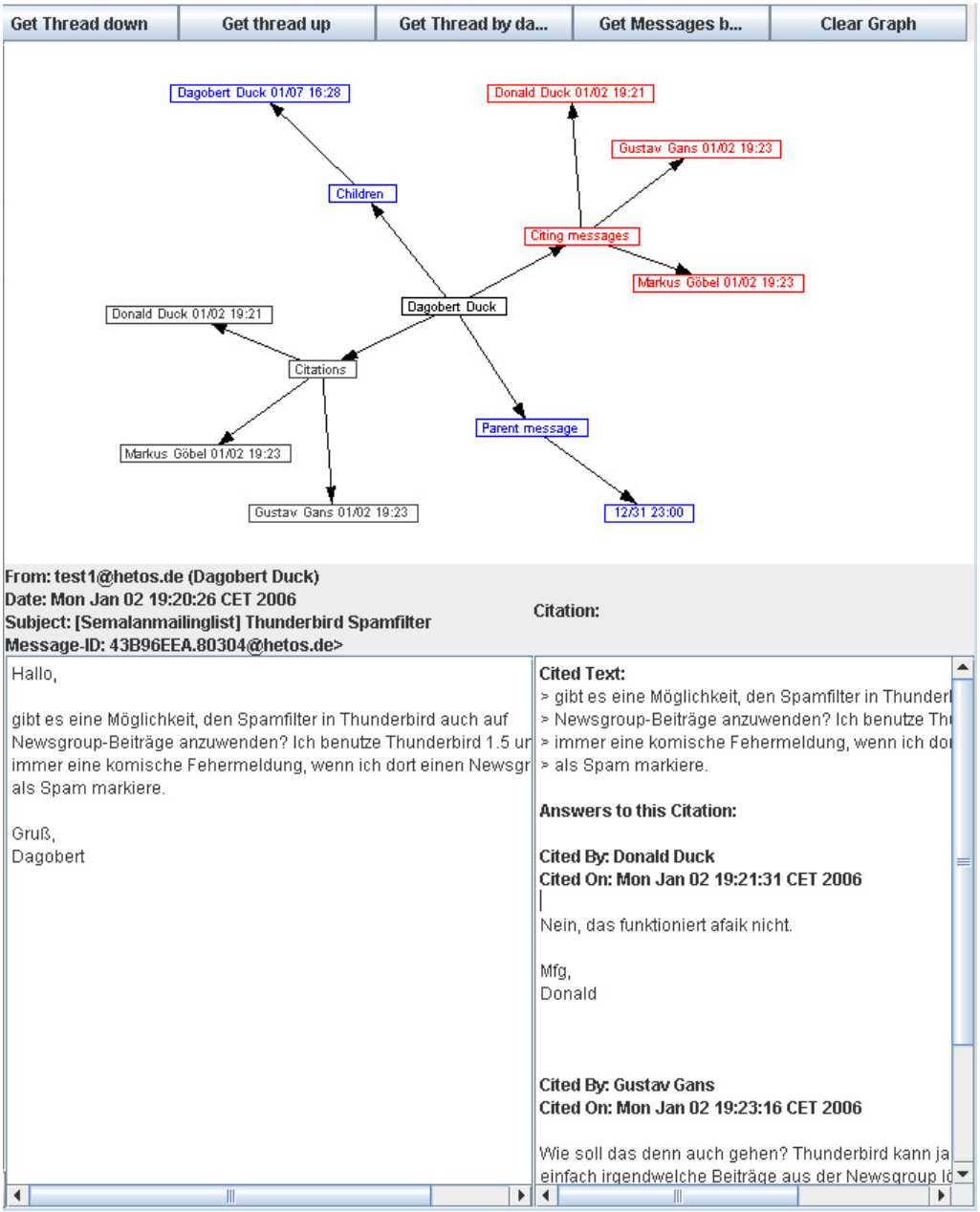

Abbildung 3.12: Detailansicht der Graphendarstellung

4 Implementierung

Dieses Kapitel gibt einen groben Überblick über den Aufbau der Semalan Java-Implementierung und erläutert diesbezüglich einige zentrale Details und Implementierungsentscheidungen.

4.1 Aufbau der Anwendung

Die Java Implementierung von Semalan besteht aus einem Server und einem Client, wobei ein Großteil der Funktionalität im Server enthalten ist und der Client, der als Java-Applet realisiert ist, nur die grafische Benutzungsoberfläche bereitstellt.

Der Semalan Server besteht aus vier Arten von Hauptkomponenten, die alle unabhängig voneinander in eigenen Prozessen laufen.

- Die Hauptanwendung, die dafür zuständig ist, die anderen Komponenten zu starten, zu überwachen, sowie gemeinsam genutzte Funktionalität wie Bericht- und Statistikfunktionen bereitzustellen.
- Ein Modul, das Abfragen per E-Mail entgegennimmt, bearbeitet und das Ergebnis an den Absender der Abfrage-E-Mail sendet.
- Ein Webserver, der sowohl die REST-artige Schnittstelle bereitstellt, als auch eine Webseite anbietet, über die das Java-Applet der grafischen Benutzungsoberfläche gestartet werden kann. Zudem ist der Webserver für die Kommunikation zwischen dem Semalan Server und dem Java-Applet zuständig

  Als Webserver kommt dabei Jetty[1] zum Einsatz, das ebenfalls in Java implementiert ist und sich daher zur vollständigen Integration in andere Java-Anwendungen eignet. Sowohl für die REST-artige Schnittstelle, als auch für die Client-Server Kommunikation nutzt der Webserver Java Servlets.
- Für jede von Semalan überwachte Mailingliste oder Newsgroup wird eine eigene Instanz einer Listenverwaltung erzeugt, die neue Nachrichten empfängt, sie in das Diskussionsmodell überführt und im zugrundeliegenden RDF Datenmodell speichert. Allein diese Listenverwaltung fügt dem Diskussionsmodell der ihr zugeordneten Mailingliste neue Daten hinzu. So wird

[1] http://jetty.mortbay.org/jetty/

gewährleistet, dass es nicht zu Kollisionen und Inkonsistenzen im Datenbestand kommt, die entstehen können, wenn verschiedene Prozesse gleichzeitig schreibend auf das Diskussionsmodell zugreifen. Komplexere Abfragen, die einen direkten Zugriff auf das RDF Datenmodell erforderlich machen, werden ebenfalls von der Listenverwaltung durchgeführt. Dagegen können verschiedene Prozesse gleichzeitig lesend auf das Diskussionsmodell zugreifen und einfache Abfragen direkt, ohne Hilfe der Listenverwaltung durchführen.

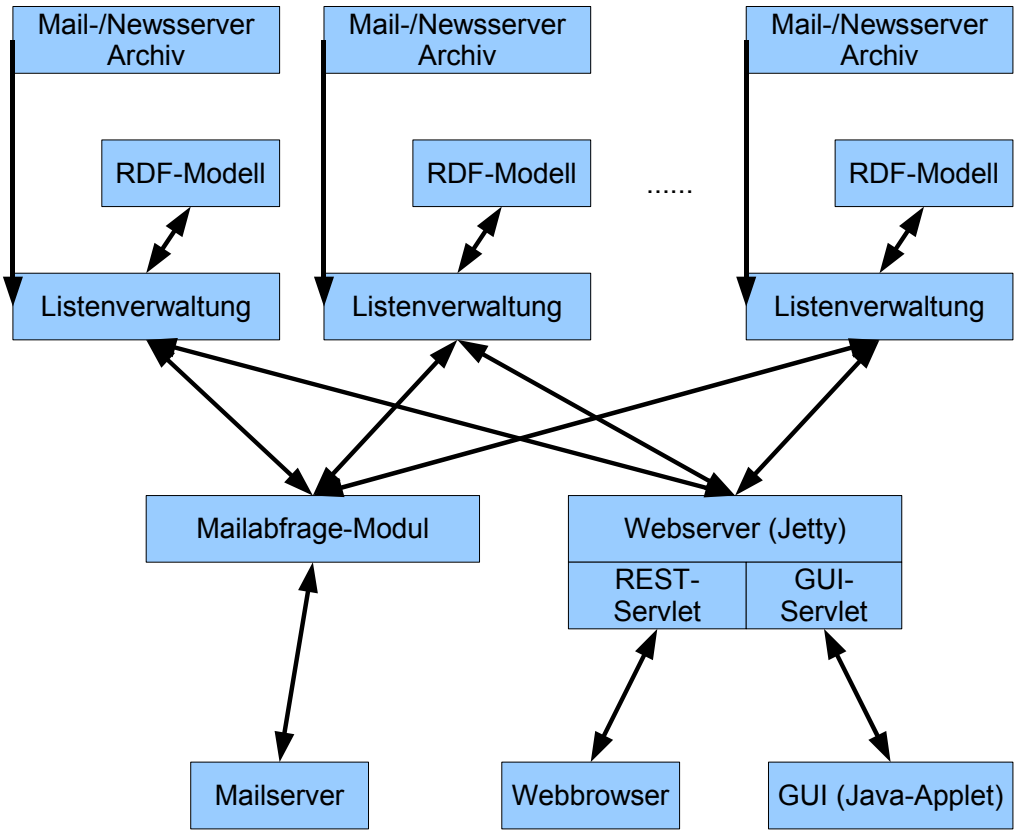

Abbildung 4.1: Implementierung der internen Kommunikation in Semalan

4.2 Datenimport

Der Import neuer Nachrichten in Semalan basiert auf der Javamail[2] Schnittstelle von Sun. Diese ist zwar weniger mächtig als einige andere E-Mail Schnittstellen für Java, wie zum Beispiel Ristretto[3], bietet aber den Vorteil, dass eine Vielzahl, sowohl von Sun selbst, als auch von Drittanbietern entwickelte Erweiterungen existieren, mit denen verschiedene E-Mail-Protokolle wie POP3, IMAP, SMTP und verwandte Kommunikationsprotokolle wie NNTP mit Javamail genutzt werden können. Auch ist über Javamail ein Zugriff auf Archivformate wie das verbreitete MBOX Format und proprietäre Speicherformate einiger E-Mail-Programme wie *Microsft Outlook* möglich. Ein Großteil der Funktionalität zum Importieren von Nachrichten ist in Semalan in einer abstrakten Klasse implementiert, die nur vom zugrundeliegenden Kommunikationsprotokoll unabhängige Javamail-Funktionen nutzt. Für jedes von Semalan verwendete Protokoll – dies sind MBOX, POP3 und NNTP – existiert eine eigene Klasse, die die zuvor erwähnte abstrakte Klasse erweitert und die selbst nur wenige, für das jeweilige Protokoll benötigte Zusatzfunktionen enthält. Auf diese Weise kann Semalan mit minimalem Aufwand um zusätzliche Protokoll- und Nachrichtenformate erweitert werden, sofern dafür Javamail Implementierungen existieren und das Nachrichtenformat die gleichen Kopfzeilen und eine ähnliche Textcodierung wie E-Mails nutzt.

Der Aufwand bei der Verarbeitung von Nachrichten wird dadurch reduziert, dass Javamail die Art des Nachrichteninhaltes (reiner Text, HTML-Text, Dateianhang etc.) und die verwendete Zeichencodierung automatisch erkennt, sofern darüber korrekte Informationen in den Kopfzeilen einer Nachricht vorhanden sind. Die Nachrichtentexte stellt Javamail unabhängig von deren vorheriger Codierung im Unicode-Format bereit. Gleiches gilt für die Kopfzeilen, auf die, unabhängig vom zugrundeliegenden Nachrichtenformat, auf einheitliche Art und Weise zugegriffen werden kann.

Die Notwendigkeit einer zusätzlichen Aufbereitung, Decodierung und Fehlerkorrektur besteht daher nur bei den Nachrichten, die absichtlich, wie bei einigen Mailinglistenarchiven zur Vermeidung von Werbemails, oder unabsichtlich, durch fehlerhafte E-Mail-Programme oder Mailserver, unvollständige oder falsche Kopfzeilen enthalten.

4.3 Datenspeicherung

Zur Datenspeicherung nutzt Semalan ein RDF-basiertes Datenmodell. Das Datenmodell selbst ist in RDF-Schema beschrieben und ist dieser Arbeit im Anhang

[2] http://java.sun.com/products/javamail/
[3] http://columbamail.org/

beigefügt. RDF[4] (Ressource Description Framework) ist ein vom W3C standardisiertes Beschreibungsformat, um Informationen und Metadaten, insbesondere über Web-Ressourcen wie Webseiten, aber auch über beliebige andere Ressourcen zu speichern und auszutauschen. Mittels RDF-Schema können semantische Relationen innerhalb eines RDF Datenbestandes beschrieben, die inhaltliche Bedeutung der Daten dargelegt und Ontologien definiert werden.

Durch die Nutzung von RDF können im Semalan Diskussionsmodell gespeicherte Daten problemlos von anderen RDF-fähigen Programmen gelesen und weiterverarbeitet werden kann.

Zur einfacheren und flexibleren Handhabung der RDF Daten werden in Semalan zwei zusätzliche Abstraktionsebenen verwendet.

- **Jena** Das zugrundeliegende RDF Datenmodell wird von Jena[5] verwaltet. Jena erleichtert das Erstellen von Semantic-Web Anwendungen, indem es Schnittstellen zum Zugriff auf RDF, RDFS und OWL Modelle, sowie Methoden zur Speicherung und Verwaltung der zugrundeliegenden Daten anbietet. Jena hat sich für den Einsatz in Semalan als geeignet erwiesen, da damit die Geschwindigkeit beim Hinzufügen auch einer größeren Anzahl neuer Nachrichten zum Diskussionsmodell und beim Durchführen von Abfragen zufriedenstellend ist. Zudem ermöglicht Jena, die Daten auf verschiedene Weise und in unterschiedlichen Formaten persistent zu Speichern. Dies kann unter anderem in einer Datenbank oder als Datei im N3 oder XML Format geschehen, was die Wiederverwendung in anderen Programmen vereinfacht.
- **RDF$_2$GO** Der Zugriff auf die RDF-Daten erfolgt in Semalan nicht direkt mittels Jena, sondern ausschließlich über RDF$_2$GO[6]. rdf2go stellt eine einheitliche Schnittstelle bereit, über die verschiedene darunterliegende RDF-Datenspeicher wie Jena, NG4J oder YARS genutzt werden können. Dies ermöglicht es, Jena mit nur wenigen Änderungen an der Semalan Implementierung gegen eine andere Lösung zur Speicherung der RDF-Daten auszutauschen.
- **RDFReactor** RDF-Daten liegen als Aussagen der Form Subjekt-Prädikat-Objekt vor. Dies erschwert die Nutzung in objektorientierten Sprachen wie Java. Denn hierbei repräsentieren RDF-Ressourcen im RDF-Datenmodell zwar einzelne Objekte, können jedoch im Objektmodell von Java nicht als Objekte angesprochen werden. Eine Lösung für dieses Problem bietet RDFReactor[7] [VS05], das Stellvertreterobjekte für RDF-Ressourcen zur Verfügung stellt, mit deren Hilfe direkt auf die Eigenschaften einer

[4] http://www.w3.org/RDF/
[5] http://jena.sourceforge.net
[6] http://rdf2go.ontoware.org/
[7] http://rdfreactor.ontoware.org/

Ressource zugegriffen werden kann. Zur einfacheren Nutzung werden RDF-Ressourcen daher in Semalan größtenteils durch RDFReactor Objekte repräsentiert. Lediglich bei komplexeren Abfragen auf dem Diskussionsmodell ist ein direkter Zugriff auf Aussagen des RDF-Datenmodells notwendig, der über die unter RDFReactor liegende RDF_2GO Abstraktionsebene erfolgt.

Das in RDF vorliegende Diskussionsmodell von Semalan kann wahlweise im Arbeitsspeicher gehalten und beim Beenden von Semalan auf die Festplatte geschrieben, oder direkt in einer MySQL Datenbank angelegt werden.

4.4 Client-Server Architektur

Den Designanforderungen entsprechend soll die grafische Benutzungsoberfläche von Semalan:

- nicht nur lokal, sondern auch über ein Netzwerk nutzbar sein,
- ohne Installation und mit den auf den meisten Computern vorhandenen Hilfsmitteln genutzt werden können,
- auch anspruchsvollere Visualisierungen ohne Performanceeinbußen ermöglichen.

Um diesen Anforderungen gerecht zu werden, ist die Benutzungsoberfläche in Form eines Java-Applets realisiert. Dadurch stehen auf Clientseite fast alle Möglichkeiten einer vollwertigen Java Anwendung zur Verfügung und zugleich ist die Benutzungsoberfläche über jeden gängigen Webbrowser aufrufbar. Im Gegensatz zu einer rein serverseitigen Darstellung, zum Beispiel durch Servlets oder Java-Server Pages kann durch das Applet die Hauptlast der Visualisierung zum Client ausgelagert und so der Server entlastet werden. Im Vergleich zu den meisten anderen clientseitigen Methoden zur Visualisierung dynamischer Inhalte wie Javascript, bieten Applets mehr Möglichkeiten zur Visualisierung, insbesondere mit Graphen, da hierbei auf existierende Java Bibliotheken wie Jung[8], Jgraph[9] oder das von Semalan verwendete Prefuse[10] zurückgegriffen werden kann.

Die Kommunikation zwischen Client und Server ist derart gestaltet, dass die Benutzungsoberfläche auch durch Firewalls hindurch benutzbar ist. Anfragen an den Server werden über das HTTP Protokoll an einen in Semalan integrierten Webserver gesendet, wobei alle Parameter der Anfrage direkt in der URL enthalten sind. Die serverseitig als Javaobjekte vorliegenden Ergebnisse werden in Form eines XML Dokuments serialisiert und als Resultat des HTTP Aufrufs

[8] http://jung.sourceforge.net/
[9] http://www.jgraph.com/
[10] http://prefuse.sourceforge.net/

an das Applet übergeben. Durch die Beschränkung auf das HTTP Protokoll als Transportmedium ist die Semalan Benutzungsoberfläche überall dort einsetzbar, wo auch normale Webseitenaufrufe möglich sind. Da die Daten mit XML in einem strukturierten Format vorliegen, ermöglicht dies auch die Nutzung der Schnittstelle durch andere Webapplikationen.

Die XML Schnittstelle ist so konzipiert, dass nur das gerade vom Client benötigte Mindestmaß an Daten übertragen wird. Will der Client eine Liste von Nachrichten anzeigen, werden nur einige wichtige Kopfzeilen der Nachrichten wie Betreffzeile, Absendedatum und Absender übertragen. Die Nachrichtentexte werden erst dann vom Server nachgeladen, wenn diese ebenfalls angezeigt werden sollen. So wird gewährleistet, dass auch über langsame Internetverbindungen größere Mailinglisten im Client visualisiert werden können.

Im Gegensatz dazu überträgt die im Abschnitt 3.6.2 beschriebene REST-artige Schnittstelle immer alle zu einer Nachricht gehörigen, relevanten Informationen wie den Nachrichtentext, Verknüpfungen mit anderen Nachrichten oder enthaltene Zitate auf einmal. Da die REST-artige Schnittstelle dem Benutzer aber jeweils nur eine Nachricht, ein Zitat oder eine Abstimmung zugleich präsentiert, fallen bei der Datenübertragung zum Webbrowser des Benutzers dennoch keine größeren Datenmengen an.

Nachteile anderer Kommunikationsformen

Es hat sich gezeigt, dass eine alternative Kommunikation über RMI (Remote Method Invocation) Aufrufe, die ebenfalls die Möglichkeit bieten, komplette Javaobjekte zwischen Client und Server zu übergeben, zwar einfacher zu implementieren ist als eine Kommunikation mittels XML, da hierbei keine explizite Serialisierung der Objekte erforderlich ist. Jedoch treten mit RMI erhebliche Probleme auf, wenn auf Seite des Clients eine Firewall oder NAT zum Einsatz kommt. Da nicht jeder Nutzer in der Lage ist oder die notwendigen Berechtigungen besitzt, eventuell vorhandene Firewalls für den Einsatz von RMI zu konfigurieren, ist diese Kommunikationsform in Semalan nicht nutzbar.

Eine weitere Alternative wäre eine direkte Übertragung von Teilen des im RDF Format vorliegenden Datenmodells zum Client, wie sie von Joseki[11], einem zum Jena Projekt gehörenden RDF Server unterstützt wird. Dies bereitet zwar weniger Probleme mit Firewalls als RMI, hat jedoch den Nachteil, dass das Applet zuerst sämtliche für den Umgang mit RDF benötigten Bibliotheken vom Server laden muss, was insbesondere bei langsameren Internetverbindungen nicht praktikabel ist. Daher ist auch diese Lösung wenig geeignet für den Einsatz in Semalan.

[11] http://www.joseki.org/

5 Evaluation

Dieses Kapitel bewertet die Performance der Java Implementierung von Semalan, beschreibt die Ergebnisse eines nicht repräsentativen Tests unter Mailinglistennutzern und erläutert die implementierte Funktionalität von Semalan.

5.1 Performance

Da der Aufwand beim Hinzufügen neuer Nachrichten zum Diskussionsmodell, wie nachfolgend erläutert, nur von der Anzahl der neuen Nachrichten, nicht aber von der Zahl der bereits im Diskussionsmodell vorhandenen Nachrichten abhängt, erfährt Semalan die stärkste Belastung beim Importieren neuer Nachrichten aus Mailinglistenarchiven, wobei bei großen Mailinglisten oft ein Nachrichtenvolumen von mehreren zehntausend Nachrichten zu verarbeiten ist. Im laufenden Betrieb treffen neue Nachrichten von den Mailinglisten hingegen über den Tag verteilt nur einzeln oder in kleineren Gruppen ein, wodurch beim Importieren dieser Nachrichten keine nennenswerte Last entsteht, zumal sich auch bei großen Mailinglisten die Anzahl neuer Nachrichten pro Tag maximal im dreistelligen Bereich bewegt. Entscheidende Kenngrößen für die Leistungsfähigkeit der Semalan-Implementierung sind dabei die benötigte Zeit zum Importieren der Nachrichten, sowie der Arbeitsspeicherbedarf.

5.1.1 Geschwindigkeit beim Nachrichtenimport

Tabelle 5.1 illustriert die benötigte Zeit zum Importieren neuer Nachrichten[1] und deren Aufnahme in das Diskussionsmodell, sowie den Unterschied zwischen einem im Arbeitsspeicher liegenden und einem in einer MySQL Datenbank gespeicherten Diskussionsmodell.

Nachrichtenanzahl	5000	10000	15000
RAM	10 min	21 min	33 min
MySQL	112 min	257 min	-

Tabelle 5.1: Geschwindigkeit beim Nachrichtenimport

[1] Testsystem: Pentium 4, 3GHz

Auch wenn diese Werte je nach verwendeter Hardware variieren können und es auch je nach Anzahl der in den Nachrichten enthaltenen Zitate geringfügige Unterschiede gibt, zeigt sich, dass sich die zum Datenimport benötigte Zeit, zumindest bei einer Speicherung im Arbeitsspeicher, in einem für den praktischen Einsatz vertretbaren Rahmen bewegt. Zumal ein Mailinglistenarchiv für jede Mailingliste nur einmalig importiert werden muss.

Die Tabelle zeigt auch, dass die Importdauer linear mit der Anzahl der zu importierenden Nachrichten steigt. Dies ist insbesondere auf die bereits im Abschnitt 3.5.2 erläuterte Beschränkung bei der Suche nach dem Ursprung eines Zitats auf verwandte, ältere Nachrichten zurückzuführen. Dadurch liegt die Anzahl der mit jeder neuen Nachricht zu vergleichenden Nachrichten, unabhängig von der Gesamtanzahl der Nachrichten in der Mailingliste, meist in einer Größenordnung von wenigen Dutzend. Im Gegensatz dazu hat sich, wie in Tabelle 3.9 illustriert, gezeigt, dass bei einem Zitatvergleich mit jeweils allen in der Mailingliste enthaltenen Nachrichten, die für den Import benötigte Zeit mit der Anzahl der bereits im Diskussionsmodell enthaltenen Nachrichten deutlich zunimmt, was insbesondere bei größeren Mailinglisten nicht praktikabel ist.

Ein erheblicher Performanceunterschied besteht auch zwischen einem im Arbeitsspeicher gehaltenen Datenmodell und der Speicherung desselben in einer MySQL Datenbank. Neben dem grundsätzlichen Geschwindigkeitsunterschied zwischen einer Datenhaltung im Arbeitsspeicher und der Speicherung der Daten auf Festplatte, ist dies ist auch darauf zurückzuführen, dass die MySQL Schnittstelle des zugrundeliegenden RDF-Speichers Jena nach Aussage von dessen Entwicklern noch nicht nach Performancegesichtspunkten optimiert wurde. Wie folgendes Unterkapitel zeigt, lässt sich jedoch auch eine größere Anzahl an Nachrichten problemlos im Arbeitsspeicher halten, wodurch der Speicherung in einer MySQL Datenbank keine große Rolle zukommt.

5.1.2 Arbeitsspeicherverbrauch

Tabelle 5.2 zeigt die Arbeitsspeicherbelegung bei unterschiedlicher Anzahl im Diskussionsmodell enthaltener Nachrichten.

Nachrichtenanzahl	10000	20000	30000
Speicherverbrauch	235 MB	483 MB	725 MB

Tabelle 5.2: Arbeitsspeicherverbrauch

Der belegte Arbeitsspeicher nimmt ebenfalls linear mit der Anzahl der Nachrichten zu, wodurch sich selbst mit einer von heutigen Arbeitsplatzrechnern leicht erreichbaren Speichergröße von 2 GB bis zu 80.000 Nachrichten im Arbeitsspeicher halten lassen. Die in der Tabelle aufgeführten Werte gelten bei Deaktivieren

der internen Statistikfunktionen von Semalan. Sind die Statistikfunktionen, die in erster Linie zu Entwicklungszwecken nützlich sind, aktiviert, verdoppelt sich der Arbeitsspeicherverbrauch.

5.2 Tests mit Mailinglistennutzern

Ein nicht repräsentativer Test wurde mit Mailinglistennutzern durchgeführt, bei dem die Teilnehmer über die drei in Kapitel 3.6 beschriebenen Benutzungsschnittstellen auf den Semalan Server zugreifen konnten.

Bewertungen Es wurde die Möglichkeit positiv bewertet, sich über eine statische URL per REST-Schnittstelle regelmäßig darüber zu informieren, von welchen anderen Nachrichten eine Nachricht aktuell zitiert wird und welche Textabschnitte daraus zitiert werden. Dies wurde von den Testern als Verbesserung gegenüber der normalen Darstellungsweise von E-Mail-Programmen empfunden, bei denen nur ersichtlich ist, welche neuen, direkten Antworten auf eine E-Mail vorliegen, die sich zudem nicht immer inhaltlich auf die beantwortete E-Mail beziehen.

Die Möglichkeit, in der Zitatdarstellung der REST Schnittstelle auf einen Blick zu sehen, was in verschiedenen Nachrichten auf denselben zitierten Textabschnitt geantwortet wurde, ist von den Testern ebenfalls als hilfreiche Funktion betrachtet worden. Insbesondere wurde genannt, das dies einen einfacheren und schnelleren Zugriff auf die Antworten erlaubt, als in herkömmlichen E-Mail-Programmen, in denen die einzelnen Antworten auf einen Textabschnitt nur über mehrere Nachrichten verstreut vorhanden sind und bei längeren Nachrichten zudem noch im Text der Nachricht gesucht werden müssen.

Auch die Durchführung von Abstimmungen und Darstellung der Abstimmungsergebnisse über die REST Schnittstelle wurde als nützliche Funktion gesehen. Inwieweit sie diese Funktion in der Praxis nutzen würden, konnten die Tester nicht einschätzen und machten dies insbesondere davon abhängig, wie die Abstimmungs-Funktion von anderen Mailinglistenteilnehmern akzeptiert wird. Hier zeigt sich das Henne-Ei Problem, dass Abstimmungen nur sinnvoll eingesetzt werden können, wenn eine größere Anzahl an Personen daran teilnimmt. Andererseits machen die Mailinglistennutzer ihre Teilnahme an einer Umfrage davon abhängig, ob diese Funktion bereits von vielen anderen Nutzern akzeptiert und genutzt wird.

Das Konzept, Abfragen in einer eigenen Abfragesprache per E-Mail an Semalan zu stellen fand nur teilweise Anklang. Es wurde zwar als geeignet empfunden, über einfache Abfragen einen Einstiegspunkt in die REST-Schnittstelle zu bieten und auch um die Ergebnisse von Abfragen dauerhaft im E-Mail-Programm zu speichern. Jedoch wünschten sich die meisten Tester keinen weiteren Ausbau

des Funktionsumfangs der E-Mail-Abfrageschnittstelle. Eine Rolle spielt hierbei jedoch auch die Zusammensetzung des Testergruppe, die nach eigenen Angaben nur wenig Erfahrung mit anderen Abfragesprachen wie zum Beispiel SQL hat und eher den Umgang mit grafischen Benutzungsoberflächen gewohnt ist.

Positiv bewertet wurde das Einbeziehen von Zitathäufigkeiten in die Listendarstellung der grafischen Benutzungsoberfläche. Die Tester waren der Meinung, dass ihnen dadurch nützliche Informationen darüber zugänglich gemacht werden, wie stark eine Nachricht in einer Diskussion eingebettet ist.

Bei der grafischen Benutzungsoberfläche haben die Tester die gleichzeitige Darstellung mehrerer Antworten aus verschiedenen Nachrichten auf einen Textabschnitt als vorteilhaft empfunden. Als Verbesserung gegenüber herkömmlichen E-Mail-Programmen wurde hierbei genannt, dass nicht erst mehrere andere Nachrichten gelesen werden und dort die Antworten zudem noch im Nachrichtentext gesucht werden mussten. Auch die gekürzte Darstellung von Zitaten in Nachrichten wurde geschätzt, da so auch bei Nachrichten, die längere Zitate enthalten, der neue Text in der Nachricht schneller identifiziert werden konnte.

Die Graphendarstellung wurde als interessante Neuerung bewertet, jedoch waren die Tester durch diese Darstellungsweise auch irritiert, da diese erheblich von der üblicherweise in E-Mail-Programmen genutzten Darstellungsweise abweicht. Daher konnten die Tester noch nicht abschließend entscheiden, ob sie diese auch dauerhaft nutzen würden.

Verbesserungsvorschläge Als Verbesserungsvorschlag wurde ein größerer Funktionsumfang der grafischen Benutzungsschnittstelle genannt, um diese mehr wie ein E-Mail-Programm nutzen zu können. Gewünscht wurde die Möglichkeit, von dort aus direkt neue E-Mails oder Antworten auf E-Mails zu verfassen oder ein Adressbuch mit Kontakten verwalten zu können.

Ebenfalls gewünscht wurde eine Integration der Funktionalität von Semalan in die von den Testern üblicherweise genutzten E-Mail-Programme.

Ein weiterer Vorschlag bestand darin, die über die REST-artige Schnittstelle verfügbaren Informationen so anzubieten, dass diese im RSS Format abonniert werden können.

Fazit Die von Semalan über per E-Mail geführte Diskussionen zur Verfügung gestellten Informationen wurden von den Testern überwiegend als nützlich und als Neuerung im Vergleich zu herkömmlichen Darstellungen von Mailinglisten empfunden. Die Tester hatten den Eindruck, dass ihnen dadurch zentrale Aspekte einer Diskussion leichter zugänglich sind als mit derzeitigen E-Mail-Programmen. Die Tester sahen Semalan nicht als Ersatz aktuell genutzter Darstellungen von Mailinglisten an, sondern als zusätzliche Informationsquelle, die neben den primär eingesetzten E-Mail-Programmen genutzt werden kann, wenn in diesen der

Überblick über eine umfangreiche Diskussion verloren geht. Gewünscht wurde die Erweiterung von Semalan um viele für E-Mail-Programme typische Funktionen.

5.3 Realisierte Funktionalität

Semalan ist in der Lage, wie in Abschnitt 3.4 erläutert, einen Großteil (>98%) der in Mailinglisten und Newsgroups enthaltenen Nachrichten zu verarbeiten und einige Fehler wie verstümmelte E-Mail Adressen zu korrigieren. Noch nicht möglich ist die Verarbeitung spezieller Textformatierungen wie HTML E-Mails, die aber nur einen geringen Anteil aller E-Mails ausmachen.

Die erste Einordnung der Nachrichten in einen Diskussionsbaum aufgrund der in den Kopfzeilen enthalten Metadaten basiert auf einer erprobten und weit verbreiteten Vorgehensweise und liefert daher gute Resultate. Voraussetzung dafür ist, dass das in Kapitel 3.4 beschriebene und in E-Mails und Newsgroupbeiträgen üblicherweise enthaltene Mindestmaß an Kopfzeilen vorhanden ist.

Der Algorithmus zum Herstellen inhaltlicher Verknüpfungen zwischen Textabschnitten und Nachrichten funktioniert ebenfalls zuverlässig, wenn die in Mailinglisten übliche Zitatweise genutzt wird. Noch nicht vollständig unterstützt wird das Filtern von Elementen wie Grußformeln oder Signaturen, die nichts zum eigentlichen Inhalt der Nachricht beitragen.

Alle Aspekte des Diskussionsmodells sind über mindestens eine der drei Benutzungsschnittstellen zugänglich. Aufgrund der eingeschränkten Strukturierungsmöglichkeiten reiner Textdarstellungen können einige Informationen nicht über die Schnittstelle für E-Mail-Abfragen, sondern nur über die REST-artige Schnittstelle und die grafische Benutzungsoberfläche dargestellt werden. Noch nicht implementiert in der grafischen Benutzungsoberfläche ist die Darstellung von Abstimmungen, die aber über die REST-artige Schnittstelle eingesehen werden können.

6 Zusammenfassung und Ausblick

Im ersten Kapitel wurde ein Überblick über die heutige Nutzung von Mailinglisten und Newsgroups gegeben. Es wurde dargelegt, dass das Lesen von Mailinglisten-Diskussionen mit aktuellen E-Mail-Programmen ineffizient ist, da diese keine inhaltlichen Zusammenhänge zwischen Nachrichten darstellen. Eine Lösung des Problems wurde vorgeschlagen, bei der die Nachrichten auf ein Diskussionsmodell abgebildet werden, mit dem eine feingranulare Darstellung von E-Mail-Diskussionen möglich ist.

Ansätze aktueller Software zur Visualisierung von Mailinglisten-Diskussionen wurden in Kapitel zwei beschrieben. Es wurden Arbeiten vorgestellt, die diese Ansätze, insbesondere durch Einbeziehung von Zitatzusammenhängen, verbessern. Es wurden verschiedene Techniken wie REST und eine Graphendarstellung vorgestellt, die kombiniert und in ähnlicher Form in Semalan zum Zugriff auf das Diskussionsmodell genutzt werden.

Im dritten Kapitel wurde eine Architektur konzipiert, mit der Diskussionen aus Mailinglisten und Newsgroups in ein Diskussionsmodell überführt werden können. Es wurde analysiert, inwieweit sich Nachrichten aus Mailinglisten von den Ausgangsdaten her zur Abbildung auf ein solches Diskussionsmodell eignen und welche Vorverarbeitung der Nachrichten dazu erfolgen muss. Das zugrundeliegende Diskussionsmodell wurde entworfen, sowie Algorithmen zur Abbildung von Nachrichten auf das Diskussionsmodell beschrieben. Um einen Zugriff auf das Diskussionsmodell zu ermöglichen, wurden drei Schnittstellen geschaffen. Dazu wurde eine Abfragesprache für Abfragen per E-Mail, eine REST-artige Schnittstelle, sowie ein Konzept für eine grafische Benutzungsoberfläche entworfen.

Diese Architektur wurde als Prototyp in Java implementiert und einige wichtige Details der Implementierung, insbesondere im Bereich der Datenspeicherung und der Client-Server Kommunikation wurden in Kapitel vier erläutert.

Im fünften Kapitel wurden Aussagen über die Leistungsfähigkeit der Semalan Implementierung getroffen. Es wurde erläutert, welches Urteil Testnutzer über Semalan abgegeben haben und wozu Semalan in der Praxis eingesetzt werden kann, beziehungsweise wo noch Verbesserungsmöglichkeiten bestehen.

6.1 Ausblick

Beim Erstellen dieser Arbeit und durch das Feedback von Testern haben sich Möglichkeiten zur Erweiterung und Verbesserung der Semalan-Architektur gezeigt, die den Rahmen dieser Arbeit sprengen würden, die aber noch abschließend Erwähnung finden sollen.

6.1.1 Textanalyse

Die in Semalan verwendeten Methoden zur Analyse und Zuordnung von Zitaten basieren nur auf Textvergleichen. Dies ist ausreichend, um inhaltliche Zusammenhänge zwischen Nachrichten herzustellen, kann jedoch keine darüber hinausgehenden Informationen über den Inhalt einer Nachricht bieten. Eine sinnvolle und wünschenswerte Ergänzung von Semalan wäre daher eine tiefergehende linguistische Analyse der Nachrichtentexte, mit deren Hilfe automatisch inhaltlich korrekte Zusammenfassungen von einzelnen Nachrichten oder ganzen Diskussionen erstellt werden können. Auch könnten damit die für Abstimmungen zu ermittelnden Standpunkte und Meinungen der Diskussionsteilnehmer bezüglich eines Themas direkt aus dem Nachrichtentext und nicht wie bei Semalan über spezielle semantische Annotationen gewonnen werden. Es ist allerdings fraglich, ob in absehbarer Zeit ein Grad maschinellen Textverständnisses erreicht werden kann, der für diesen Zweck zufriedenstellende Resultate liefert. Aktuell verwendete Methoden zum automatischen Erstellen von Textzusammenfassungen, wie sie zum Beispiel in *Microsoft Word* zum Einsatz kommen, sind zumindest noch weit von brauchbaren Ergebnisse entfernt.

6.1.2 Alternative Benutzungsschnittstellen

Von Semalan Testern wurde der Kritikpunkt vorgebracht, dass sie es bevorzugen würden, wenn die Darstellungsmöglichkeiten von Semalan anstatt über gesonderte Schnittstellen, direkt in den von ihnen üblicherweise genutzten E-Mail-Programmen zur Verfügung stehen würden. Da andererseits die Bereitschaft das E-Mail-Programm zu wechseln eher gering ist, ist dies aufgrund der Vielzahl an existierenden E-Mail-Programmen nur schwer zu realisieren. Dennoch könnte durch die Implementierung einer Schnittstelle zum Semalan Server, zumindest in einige populäre E-Mail-Programme, ein größerer Benutzerkreis erreicht werden. Dazu bieten sich Programme wie *Mozilla Thunderbird*[1] an, deren Quellcode frei verfügbar ist und die bereits Mechanismen zur Erweiterung um zusätzliche Funktionen bieten.

[1] http://www.thunderbird-mail.de/

6.1.3 Verbindung mit anderen RDF Daten

Semalan selbst stehen nur die Informationen zur Verfügung, die in den Kopfzeilen, sowie dem Text der importierten Nachrichten enthalten sind. Über den Nachrichtentext hinausgehend sind dies in erster Linie einige Informationen über die Absender und Empfänger der Nachrichten, wie deren Name oder E-Mail-Adresse.

Um den Benutzern darüber hinaus zusätzliche Informationen zur Verfügung zu stellen, könnte Semalan auf ebenfalls in RDF gespeicherte Daten anderer Semantic-Web-Anwendungen zurückgreifen. Vorstellbar wäre hier, mehr Informationen über die an einer Diskussion beteiligten Personen anzubieten, wie ein Bild jeder Person oder ein Verweis auf deren Homepage. Dazu könnte zum Beispiel auf Daten zurückgegriffen werden, die andere Programme mittels des FOAF[2] RDF Vokabulars gespeichert haben, das Semalan selbst schon zur Speicherung personenbezogener Daten nutzt.

6.1.4 Erweiterung um andere Diskussionsformen

Neben E-Mails und Newsgroups existieren noch andere textbasierte Diskussionsformen in Internet. Dazu zählen sowohl Medien zur Kommunikation in Echtzeit wie Chats und Instant-Messaging, als auch Medien, bei denen die Nachrichten zeitlich versetzt verfasst werden wie bei Diskussionen in Wikis. Wie bei E-Mails liegen auch hier sich aufeinander beziehende Nachrichten vor, die zusammen eine Diskussionsstruktur bilden. Daher liegt es nahe, auch diese Diskussionsformen auf das Semalan Diskussionsmodell abzubilden. Allerdings bestehen einige strukturelle Unterschiede zwischen Wikis und Instant-Messaging auf der einen und E-Mails auf der anderen Seite. Im Gegensatz zu E-Mails sind in den einzelnen Nachrichten weder Kopfzeilen vorhanden, die Metainformationen über die Nachrichten enthalten, noch werden Textabschnitte aus anderen Nachrichten zitiert. Daher wäre die Integration dieser Diskussionsformen nur mit Schwierigkeiten zu realisieren. Dies könnte nur über eine inhaltliche Verknüpfung erfolgen und würde Änderungen an der Semalan-Architektur und auch an dem zugrundeliegenden Diskussionsmodell erfordern.

[2] http://www.foaf-project.org/

A Das Diskussionsmodell in RDFS

Das in Abschnitt 3.1 näher erläuterte und, wie in Kapitel 4 beschrieben, in RDF-Schema modellierte Diskussionsmodell von Semalan ist nachfolgend abgebildet:

```
#Das Semalan Diskussionsmodell in RDFS

@prefix rdf : <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix foaf: <http://xmlns.com/foaf/0.1/#>.
@prefix     : <http://semalan.ontoware.org/ns/2005/semalan#>.

#---Klassen--

:Message                rdf:type        rdfs:Class.
:Citation               rdf:type        rdfs:Class.
:Client                 rdf:type        rdfs:Class.
:Poll                   rdf:type        rdfs:Class.
:Response               rdf:type        rdfs:Class.

#--Person--

foaf:name               rdf:type        rdfs:Property;
                        rdfs:domain     foaf:Person;
                        rdfs:range      rdfs:Literal.

foaf:mbox               rdf:type        rdfs:Property;
                        rdfs:domain     foaf:Person;
                        rdfs:range      rdfs:Literal.

#--Client

:Clientname             rdf:type        rdfs:Property;
                        rdfs:domain     :Client;
                        rdfs:range      rdfs:Literal.
```

```
#--Nachricht--

:Uniquemessageid        rdf:type        rdfs:Property;
                        rdfs:domain     :Message;
                        rdfs:range      rdfs:Literal.

:Messagetype            rdf:type        rdfs:Property;
                        rdfs:domain     :Message;
                        rdfs:range      rdfs:Literal.

:Messagestatus          rdf:type        rdfs:Property;
                        rdfs:domain     :Message;
                        rdfs:range      rdfs:Literal.

:Parent                 rdf:type        rdfs:Property;
                        rdfs:domain     :Message;
                        rdfs:range      :Message.

:Child                  rdf:type        rdfs:Property;
                        rdfs:domain     :Message;
                        rdfs:range      :Message.

:Subjectgroup           rdf:type        rdfs:Property;
                        rdfs:domain     :Message;
                        rdfs:range      rdfs:Literal.

:Sentdate               rdf:type        rdfs:Property;
                        rdfs:domain     :Message;
                        rdfs:range      rdfs:Literal.

:Receiveddate           rdf:type        rdfs:Property;
                        rdfs:domain     :Message;
                        rdfs:range      rdfs:Literal.

:Year                   rdf:type        rdfs:Property;
                        rdfs:domain     :Message;
                        rdfs:range      rdfs:Literal.

:Month                  rdf:type        rdfs:Property;
                        rdfs:domain     :Message;
                        rdfs:range      rdfs:Literal.
```

```
:Day                    rdf:type        rdfs:Property;
                        rdfs:domain     :Message;
                        rdfs:range      rdfs:Literal.

:Subjectline            rdf:type        rdfs:Property;
                        rdfs:domain     :Message;
                        rdfs:range      rdfs:Literal.

:From                   rdf:type        rdfs:Property;
                        rdfs:domain     :Message;
                        rdfs:range      foaf:Person.

:Organization           rdf:type        rdfs:Property;
                        rdfs:domain     :Message;
                        rdfs:range      rdfs:Literal.

:Returnpath             rdf:type        rdfs:Property;
                        rdfs:domain     :Message;
                        rdfs:range      foaf:Person.

:Replyto                rdf:type        rdfs:Property;
                        rdfs:domain     :Message;
                        rdfs:range      foaf:Person.

:To                     rdf:type        rdfs:Property;
                        rdfs:domain     :Message;
                        rdfs:range      foaf:Person.

:Cc                     rdf:type        rdfs:Property;
                        rdfs:domain     :Message;
                        rdfs:range      foaf:Person.

:Bcc                    rdf:type        rdfs:Property;
                        rdfs:domain     :Message;
                        rdfs:range      foaf:Person.

:Messageid              rdf:type        rdfs:Property;
                        rdfs:domain     :Message;
                        rdfs:range      rdfs:Literal.
```

```
:Inreplyto              rdf:type        rdfs:Property;
                        rdfs:domain     :Message;
                        rdfs:range      :Message.

:References             rdf:type        rdfs:Property;
                        rdfs:domain     :Message;
                        rdfs:range      :Message.

:Xmailer                rdf:type        rdfs:Property;
                        rdfs:domain     :Message;
                        rdfs:range      :Client.

:Xnewsreader            rdf:type        rdfs:Property;
                        rdfs:domain     :Message;
                        rdfs:range      :Client.

:Useragent              rdf:type        rdfs:Property;
                        rdfs:domain     :Message;
                        rdfs:range      :Client.

:Contenttype            rdf:type        rdfs:Property;
                        rdfs:domain     :Message;
                        rdfs:range      rdfs:Literal.

:Contransenc            rdf:type        rdfs:Property;
                        rdfs:domain     :Message;
                        rdfs:range      rdfs:Literal.

:Mimeversion            rdf:type        rdfs:Property;
                        rdfs:domain     :Message;
                        rdfs:range      rdfs:Literal.

:Body                   rdf:type        rdfs:Property;
                        rdfs:domain     :Message;
                        rdfs:range      rdfs:Literal.

:Citedtext              rdf:type        rdfs:Property;
                        rdfs:domain     :Message;
                        rdfs:range      :Citation.
```

```
#---Textabschnitt---

:Citationfrom           rdf:type        rdfs:Property;
                        rdfs:domain     :Citation;
                        rdfs:range      :Message.

:Citedby                rdf:type        rdfs:Property;
                        rdfs:domain     :Citation;
                        rdfs:range      :Message.

:Citationstartline      rdf:type        rdfs:Property;
                        rdfs:domain     :Citation;
                        rdfs:range      rdfs:Literal.

:Citationendline        rdf:type        rdfs:Property;
                        rdfs:domain     :Citation;
                        rdfs:range      rdfs:Literal.

:Answerstartline        rdf:type        rdfs:Property;
                        rdfs:domain     :Citation;
                        rdfs:range      rdfs:Literal.

:Answerendline          rdf:type        rdfs:Property;
                        rdfs:domain     :Citation;
                        rdfs:range      rdfs:Literal.

:Citationlength         rdf:type        rdfs:Property;
                        rdfs:domain     :Citation;
                        rdfs:range      rdfs:Literal.

:Originoffset           rdf:type        rdfs:Property;
                        rdfs:domain     :Citation;
                        rdfs:range      rdfs:Literal.

:Citationtype           rdf:type        rdfs:Property;
                        rdfs:domain     :Citation;
                        rdfs:range      rdfs:Literal.

:Citationhash           rdf:type        rdfs:Property;
                        rdfs:domain     :Citation;
                        rdfs:range      rdfs:Literal.
```

```
#---Abstimmung---

:Pollhash               rdf:type        rdfs:Property;
                        rdfs:domain     :Poll;
                        rdfs:range      rdfs:Literal.

:Polltext               rdf:type        rdfs:Property;
                        rdfs:domain     :Poll;
                        rdfs:range      rdfs:Literal.

:Pollanswer             rdf:type        rdfs:Property;
                        rdfs:domain     :Poll;
                        rdfs:range      :Response.

:Polldate               rdf:type        rdfs:Property;
                        rdfs:domain     :Poll;
                        rdfs:range      rdfs:Literal.

#---Antwort auf Abstimmung---

:Respondent             rdf:type        rdfs:Property;
                        rdfs:domain     :Response;
                        rdfs:range      foaf:Person.

:Responsetext           rdf:type        rdfs:Property;
                        rdfs:domain     :Response;
                        rdfs:range      rdfs:Literal.

:Responsemessage        rdf:type        rdfs:Property;
                        rdfs:domain     :Response;
                        rdfs:range      :Message.
```

Literaturverzeichnis

[Dod05] Dodds, Leigh: *Connecting Social Content Services using FOAF, RDF and REST.* Paper at XTech2005, 2005

[EH02] Eugene, Eric K. ; Holman, Ken: *Interoperability Between Collaborative Knowledge Applications.* Paper at EML2002, 2002

[Eng90] Engelbart, Douglas C.: Knowledge-Domain Interoperability and an Open Hyperdocument System. In: *CSCW2000*, 1990, S. 143–156

[Eng00] Engelbart, Dougla C.: *Draft OHS-Project Plan.* Article at Bootstrap Institute. `http://www.bootstrap.org/augdocs/bi-2120.html`. Version: Oktober 2000

[Fie00] Fielding, R.: *Architectural Styles and the Design of Network-based Software Architectures.* Irvine, USA, University of Califormia, Diss., 2000. – `http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm`

[Gie04] Giereth, Mark: *Foafscape - a Browser for Friend of a Friend Documents.* Paper at Foaf Galway 2004, 2004

[Ibi02] Ibidunni, Olu: Supporting workgroups collaborating via email using the semantic web and RDF / Hewlett Packard Laboratories. Version: November 22 2002. `http://www.hpl.hp.com/techreports/2002/HPL-2002-316.pdf` (HPL-2002-316). – Forschungsbericht. – Online–Ressource. – 16 S

[KR70] Kunz, Werner ; Rittel, Horst: *Issues as Elements of Information Systems.* Working Paper 131. `http://www-iurd.ced.berkeley.edu/pub/WP-131.pdf`. Version: 1970

[MEHL04] McDowell, Luke ; Etzioni, Oren ; Halevy, Alon ; Levy, Henry: Semantic email. In: *WWW2004*, 2004, S. 244–254

[New01] Newman, Paula S.: Treetables and Other Visualizations for Email Threads / Xerox PARC. Version: 2001. `http://www2.parc.com/istl/groups/hdi/papers/psn_emailvis01.pdf`. – Forschungsbericht. – Online–Ressource

[New02] Newman, Paula S.: Exploring discussion lists: steps and directions. In: *JCDL2002*, 2002, S. 126–134

[Pal02] Palme, Jacob: *Message Threading in E-mail Software.* Article at K2Lab Laboratory, DSV University Department. `http://www.dsv.su.se/jpalme/ietf/message-threading.html`. Version: Juli 14 2002

[PCL00] Popolov, Dimitri ; Callaghan, Michael ; Luker, Paul: Conversation Space: Visualising Multi-threaded Conversation. In: *AVI2000*, 2000, S. 246–249

[SF01] Smith, Marc A. ; Fiore, Andrew: Visualization Components for Persistent Conversation. In: *CHI2001*, 2001, S. 136–143

[VN03] Venolia, Gina D. ; Neustaedter, Carman: Understanding sequence and reply relationships within email conversations: a mixed-model visualization. In: *CHI2003*, 2003, S. 361–368

[VS05] Völkel, Max ; Sure, York: *RDFReactor - From Ontologies to Programmatic Data Access.* Poster and Demo at ISWC2005, Nov 2005

[YST00] Yabe, Jun ; Shibayama, Etsuya ; Takahashi, Shin: Automatic animation of discussions in USENET. In: *AVI2000*, 2000, S. 84–91

Erklärung

Ich versichere hiermit wahrheitsgemäß, die Arbeit bis auf die dem Aufgabensteller bereits bekannte Hilfe selbstständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde.

Bad Schönborn, den 10.01.2006