

Multiple Visualisierungsformen für Beiträge in Chat-Systemen

Stephan Lukosch¹, Torsten Holmer², Verena Kunz¹

Fakultät für Mathematik und Informatik, FernUniversität in Hagen¹

Fakultät für Informatik, Kommunikation und Medien, FH-Oberösterreich²

Zusammenfassung

In diesem Artikel adressieren wir das Problem des Co-Text-Verlustes in der Chat-Kommunikation, formulieren Anforderungen an eine Lösung, diskutieren verwandte Arbeiten und entwickeln einen neuen Lösungsansatz. Wir berichten über erste Erfahrungen und präsentieren Erweiterungsideen. Die Kernidee von MuViChat (multiple-visualization chat) besteht in der Bereitstellung multipler Visualisierungsmöglichkeiten von referenzierten Chat-Protokollen, bei denen die Anwender je nach Nutzungskontext die angemessene Form der Darstellung wählen können. Durch die Möglichkeit des Aufnehmens und Abspielens von Chat-Diskussionen und die erweiterbare Konzeption der Software werden die Evaluation und die Weiterentwicklung von Visualisierungskonzepten unterstützt.

1 Einleitung

Textbasierte Konferenzsysteme, auch Chat-Systeme genannt, unterstützen die schriftliche Kommunikation und sind sehr verbreitet, weil sie geringe Anforderungen an den Rechner und die Übertragungsgeschwindigkeit stellen. In neuerer Zeit können Text-Chats auch parallel mit Audio-Chats eingesetzt werden, was neue Probleme mit sich bringt (z.B. Was soll geschrieben und was gesprochen werden? Wie wird der Bezug zwischen Text und Sprache hergestellt?). Im Rahmen dieser Arbeit konzentrieren wir uns jedoch auf die Standardsituation, bei dem die Teilnehmer nur durch eine Text-Chat-Anwendung vermittelt miteinander kommunizieren. Als ein Medium synchroner computervermittelter Kommunikation ist Chat Gegenstand des interdisziplinären Forschungsfeldes der computergestützten Gruppenarbeit (Computer Supported Cooperative Work, CSCW) sowie des computergestützten kooperativen Lernens (Computer Supported Cooperative Learning, CSCL). Mit Chat-Kommunikation beschäftigen sich außerdem die Medien- und Kommunikationswissenschaft sowie die Sprachwissenschaft (Linguistik). Daher gibt es mehrere Untersuchungen, die sich mit den sprachlichen Besonderheiten der Chat-Kommunikation und der besonderen Stellung der Chat-Kommunikation zwischen Schriftlichkeit und Mündlichkeit beschäftigen (vgl. z. B. Storrer, 2001; Beißwenger, 2000).

Die Vorteile der Chat-Kommunikation sind die ortsunabhängige und kostengünstige Kommunikationsform und das es beim Chat nicht unhöflich ist, sich gleichzeitig noch mit anderen Dingen zu beschäftigen. Die gleichzeitige Teilnahme an mehreren Unterhaltungen ist möglich und die Möglichkeiten des humorvollen Austauschs und der spielerische Umgang mit der Sprache machen für viele Benutzer den Reiz der Kommunikationsform Chat aus. Zudem hat man, im Gegensatz zum mündlichen Gespräch, bei der Chat-Kommunikation die Möglichkeit, frühere Aussagen nochmals nachzulesen (vgl. Herring, 1999).

Die Chat-Kommunikation hat aber auch einige Nachteile. So listen Smith et al. (2000) fünf wesentliche Probleme der Chat-Kommunikation auf:

- (P1)** Mangel an Bezug zwischen Personen und deren Nachrichten
- (P2)** Keine Sichtbarkeit des Zuhörers
- (P3)** Kein wahrnehmbarer Antwortprozess
- (P4)** Mangel an Kontrolle über die Positionierung der Nachrichten (Co-Text-Verlust)
- (P5)** Mangel an brauchbaren Aufzeichnungen

In welchem Umfang diese Schwächen auftreten, hängt maßgeblich von den Eigenschaften und dem Funktionsumfang der jeweils eingesetzten Software ab. Während P1-P3 vor allem die Dynamik der Gesprächssituation beeinflussen, wirken sich P4 und P5 primär auf die Darstellung des Gesprächsinhaltes in Form des Chat-Protokolls aus.

In diesem Artikel untersuchen wir aktuelle Chat-Systeme im speziellen daraufhin, wie sie den Mangel an Kontrolle über die Positionierung der Nachrichten adressieren und wie sich dieses in den Aufzeichnungen bzw. Chat-Protokollen darstellt. Im Anschluss daran stellen wir ein Chat-System vor, das alternative Visualisierungen von Chat-Beiträgen zur Verfügung stellt, um das Problem des Co-Text-Verlusts zu adressieren. Wir berichten über erste Erfahrungen bei dem Einsatz unseres Chat-Systems und schließen mit einer Zusammenfassung und einem Ausblick auf zukünftige weitere Arbeiten.

2 Anforderungsanalyse

Standard-Chat-Systeme stellen eingehende Nachrichten in der Reihenfolge dar, in der sie auf dem Server eintreffen. Da alle Chat-Teilnehmer gleichzeitig Nachrichten verfassen und senden können, haben alle Teilnehmer gleichzeitig das Rederecht. Die Reihenfolge, in der die Nachrichten später beim einzelnen Benutzer auf dem Bildschirm erscheinen, wird vom Server nach dem sog. Mühlenprinzip "wer zuerst kommt, mahlt zuerst" festgelegt (vgl. Beißwenger, 2000). Dies hat zur Folge, dass der angezeigte Gesprächsverlauf nicht linear ist: Nicht jeder Beitrag bezieht sich jeweils auf den vorherigen Beitrag. Es entstehen mehrere parallele ineinander verflochtene gleichzeitige Gesprächsstränge, bei denen oftmals auch mehrere Themen parallel diskutiert werden. Deshalb kann es passieren, dass ein Leser nicht weiß, welcher der vorherigen Beiträge die Informationen enthält, die notwendig sind, um die aktuelle Nachricht zu verstehen. Diese Situation wird von Pimentel et al. (2003) als Co-Text-Verlust bezeichnet. Tabelle 1 zeigt ein Beispiel für einen Co-Text-Verlust.

Schwarz:	Hast Du schon den neuen Mel Gibson Film gesehen: erheißt Payback?
Rosa:	Ich habe die Oskar-Verleihung gesehen, Du auch?
Rosa:	Ja, genau.
Rosa:	Etwas brutal, aber lustig.
Schwarz:	Du hast sie gesehen, war sie gut?
Schwarz:	Wie ging es aus, wer hat gewonnen?
Rot:	Soll gut sein, habe ich gehört.
Rosa:	War OK. Titanic hat zum Glück nicht alles abgeräumt.

Tabelle 1: Beispiel entnommen aus Vronay et al. (1999)

Unter "Co-Text" wird in der Linguistik derjenige Text verstanden, der jeweils vor und nach einer Nachricht geschrieben wurde und dessen Inhalt zum Verstehen einer Nachricht hilfreich ist. Nach Pimentel et al. (2003) verlieren Anwender, die den Co-Text suchen, den Gesprächsrhythmus. Sobald Anwender aufgeben, den Co-Text zu suchen, verstehen sie Teile der Konversation nicht. In der letzten Stufe geht der Gesprächsfluss verloren, wenn Anwender deklarieren, den Co-Text nicht gefunden zu haben. Insbesondere bei vielen Gesprächsteilnehmern benötigen die Chat-Nutzer ein hohes Maß an Konzentrationsfähigkeit und Reaktionsgeschwindigkeit, um die verschiedenen Gesprächsstränge zu entwirren, auftretenden Co-Text-Verlust zu beseitigen und somit der Konversation folgen zu können. Dabei wird es umso schwieriger die Zuordnung zu finden, je größer der Abstand zwischen den zusammengehörenden Nachrichten ist.

Durch den Co-Text-Verlust können Mehrdeutigkeiten entstehen, wenn nicht klar ist, welche Nachricht sich auf welche andere bezieht. Dabei sind insbesondere kurze Antworten, wie "ja" oder "nein" problematisch. Neben dem Begriff des Co-Text-Verlusts finden sich in der Literatur noch weitere Bezeichnungen für die durch das Mühlenprinzip verursachten Probleme der Chat-Kommunikation. Vronay et al. (1999) unterteilen es in "intention confusion" bei Unklarheit des Adressaten und als "thread confusion" bei Mehrdeutigkeit in Bezug auf den Themenstrang. Weitere Bezeichnungen für das Problem sind "chat confusion" (Pimentel et al., 2006) und "interactional incoherence" (Herring, 1999).

Co-Text-Verlust entsteht vor allem dann, wenn nicht allein aus dem Inhalt einer Nachricht klar ist, auf welche vorherige Nachricht sich eine Nachricht bezieht. Das zu entwickelnde Chat-System soll den Benutzer dabei helfen, Bezüge eines Gesprächsbeitrags zu anderen Gesprächsbeiträgen zu erkennen und so dem Gesprächsverlauf zu folgen. Das System muss daher dem Benutzer die Möglichkeit bieten, Referenzen auf frühere Chat-Beiträge zu setzen, indem es bei der Eingabe einer Nachricht möglich ist, eine Nachricht auszuwählen, auf die sich der aktuelle Beitrag bezieht. Dies führt zu folgender Anforderung:

(A1) Herstellung von Bezügen zwischen Beiträgen.

Die Chat-Beiträge und die dazwischen gesetzten Beziehungen müssen allen Chat-Teilnehmern in geeigneter Weise angezeigt werden. Vorrangig soll die Darstellung den Co-Text-Verlust minimieren. Dabei soll darauf geachtet werden, dass der Bezug zwischen Personen und ihren Nachrichten übersichtlich dargestellt wird (vgl. P1). Dem Benutzer sollen

mehrere Visualisierungsalternativen zur Verfügung stehen. Im Rahmen von Benutzertests kann dann geprüft werden, welche davon den Benutzer am besten dabei unterstützt, Zusammenhänge zwischen den Beiträgen zu erkennen und welche Nachteile einzelne Visualisierungen aufweisen. Dies führt uns zur folgenden Anforderung:

(A2) Multiple Visualisierungsformen der Beiträge und der Bezüge zwischen den Beiträgen.

Um den Co-Text-Verlust in Anschluss an eine Diskussion auflösen zu können, müssen Benutzer den Gesprächsverlauf speichern und zu einem späteren Zeitpunkt nachvollziehen können. Bei der Anzeige der gespeicherten Gespräche sollen alle Visualisierungsalternativen zur Verfügung stehen, und es soll die Möglichkeit geben, das Gespräch in seinem zeitlichen Verlauf wiederzugeben (Replay-Modus). Dadurch, dass der zeitliche Ablauf sowie die Bezüge zwischen den Nachrichten auch im Protokoll im Chat-Protokoll noch nachvollziehbar sind, kann dem Problem des Mangels an brauchbaren Aufzeichnungen (vgl. P5) entgegengewirkt werden. Für die Speicherung sollte ein vom Benutzer lesbares Format verwendet werden, das auch als Importformat für auf anderem Wege aufgezeichnete oder fiktive Gesprächsverläufe verwendet werden kann. So kommen wir zu den folgenden Anforderungen:

(A3) Speicherung und Wiedergabe von Konversationen.

3 Verwandte Arbeiten

Vorschläge und Konzepte zur Verbesserung des herkömmlichen Chat-Interfaces sind in großer Zahl entwickelt worden, von denen aber nur wenige auf das Problem des Co-Text-Verlustes eingehen. Im Folgenden geben wir einen Überblick über diejenigen Ansätze, die die Anforderungen (A1)-(A3) adressiert haben.

Die Systeme Threaded Chat (Smith et al. 2000), HyperDialog (Pimentel et al., 2003, 2006) und ThreadChat (Holmer & Wessner 2005) stellen Referenzierungsfunktionen bereit und bieten eine Visualisierung des Chat-Protokolls in Form einer eingerückten Liste. Neue Beiträge werden in die Baumstruktur eingeordnet und können daher an verschiedenen Stellen erscheinen, wenn die Teilnehmer mehrere Themen parallel diskutieren. Die Organisation der Beiträge ist damit zwar sehr strukturiert und der Co-Text eines Beitrages klar erkennbar, dennoch kann es im Gespräch kann es zu Unklarheiten kommen, wenn z.B. vergessen wurde, eine Referenz zu setzen und diese in einen neuen Baum eingeordnet wird. Ein weiterer Punkt ist der ständige Wechsel des Ortes, an dem neue Beiträge erscheinen, so dass die Gefahr des Übersehens neuer Beiträge steigt.

Diese Gefahr wird bei KOLUMBUS (Kienle & Holmer 2006) und ConcertChat (Mühlpfordt & Wessner 2005) dadurch verhindert, indem das Chatprotokoll in der herkömmliche linearen Weise angezeigt wird, aber erstens an jedem Beitrag ein Indikator angebracht ist, der zeigt, ob dieser einen Co-Text besitzt. Darüber hinaus wird der entsprechende Co-Text angezeigt, wenn der Anwender den entsprechenden Beitrag mit der Maus anwählt. Dadurch ist immer eindeutig, an welcher Stelle neue Beiträge erscheinen werden und der Co-Text kann bei

Bedarf leicht erschlossen werden. Allerdings fehlt die Möglichkeit, sich einen größeren Überblick über die Gesprächsstruktur zu verschaffen, weil diese nicht visualisiert wird.

Einen ganz anderen Ansatz verfolgt das System factChat (Harnoncourt et al. 2005), bei dem die Beiträge auf einer Fläche platziert werden und damit der Co-Text durch die Positionierung des Beitrages in der Nähe des referenzierten Beitrages hergestellt werden kann. Darüber hinaus ist es auch möglich, explizite Referenzen auf vergangene, auf der Fläche nicht mehr sichtbare Beiträge zu setzen, die der Anwender nach verfolgen kann. Zwar bietet factChat die Möglichkeit, den Verlauf der bisherigen Konversation mittels einer Zeitleiste abzuspielen, die Visualisierung der Gesprächsstruktur wird aber nicht unterstützt.

Zusammenfassend ist festzustellen, dass es bereits verschiedene Umsetzungsversuche gibt. Diese sind aber in Bezug auf unsere Anforderungen, im Speziellen der multiplen Visualisierungsformen (A2), noch unvollständig.

4 Lösungsansatz

Das zu entwickelnde Chat-System soll Lösungen für das Problem des Co-Text-Verlusts aufzeigen. Es sind daher, neben den Standard-Chat-Funktionalitäten, Mechanismen zu implementieren, die es zunächst ermöglichen, den Bezug zwischen einzelnen Gesprächsbeiträgen herzustellen, um diesen dann in geeigneter Form zu visualisieren. In den folgenden Abschnitten beschreiben wir unseren auf XMPP¹ basierenden Ansatz eines Chat-Clients, der die von uns identifizierten Anforderungen erfüllt.

4.1 Herstellung von Bezügen zwischen Beiträgen (A1)

Ein Chat-Beitrag kann sich inhaltlich auf keinen anderen Beitrag, auf genau einen früheren Beitrag oder auf mehrere frühere Beiträge beziehen. Ein Gesprächsbeitrag hätte beispielsweise dann keinen Bezugsbeitrag, wenn der Verfasser ein vollkommen neues Thema anspricht. Genau ein Bezugsbeitrag wäre typisch für die Antwort auf eine Frage bzw. den Standardfall eines Dialoges zwischen zwei Personen. Mehrere Bezugsbeiträge wären dann vorhanden, wenn ein Gesprächsteilnehmer mehrere Beiträge zusammenfasst, ein Fazit zieht oder versucht, abschweifende Diskussionszweige wieder zum Hauptthema zurückzuführen. Da letzteres Verhalten in den meisten Situationen nur selten und von wenigen Teilnehmern gezeigt wird, um die Benutzerschnittstelle einfach zu gestalten und die Anforderungen an die Visualisierungen in realisierbarem Rahmen zu halten, wird zunächst keine Mehrfachreferenzierung unterstützt.

Eine weitere Frage in Zusammenhang mit der Referenzierung von Beiträgen ist, ob das Absenden einer Nachricht nur dann zulässig ist, wenn eine Referenz zu einer Bezugsnachricht gesetzt ist. Die Erfahrungen bei der Nutzung des Chat-Systems HyperDialog (Pimentel et al.,

¹ Extensible Messaging and Presence Protocol (XMPP) (<http://www.xmpp.org/>)

2003, 2006) haben jedoch gezeigt, dass fehlende und falsch gesetzte Bezüge die Hauptursache dafür waren, dass Co-Text-Verlust durch die Verwendung von Referenzierungen nicht beseitigt werden konnte. Des Weiteren schränkt ein Referenzierungszwang die inhaltlichen Freiräume eines Gesprächs stark ein. Aus diesem Grund entscheiden wir uns dafür, die Referenzierung nicht zwingend vorzuschreiben. Stattdessen können Anwender vor dem Verfassen, spätestens jedoch vor Versand eines Beitrags, einen Chat-Beitrag im Anzeigebereich auswählen und so einen Bezug zwischen den ausgewählten und neuen Beitrag herstellen.

4.2 Multiple Visualisierung der Beiträge sowie der Bezüge zwischen den Beiträgen (A2)

Das System wurde so gestaltet, dass mehrere Visualisierungsalternativen ausgewählt werden können. Dadurch kann jeder Benutzer, unabhängig von anderen Benutzern, die von ihm favorisierte Alternative verwenden. Alle Visualisierungsalternativen bieten in einem Konfigurationsbereich weitere visualisierungsspezifische Einstellmöglichkeiten.

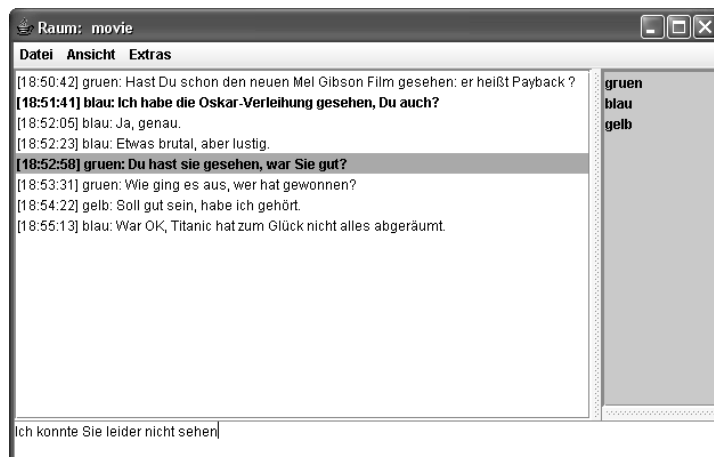


Abbildung 1: Benutzeroberfläche der Listenansicht mit Hervorhebung

Die *klassische Visualisierung* entspricht einem Standard-Chat-Client. Das System stellt diese neben den experimentellen neuen Visualisierungen zur Verfügung, damit eine schnelle Vergleichsmöglichkeit besteht. In der klassischen Visualisierung können Benutzer die Schriftgröße, die Schriftfarbe und die Hintergrundfarbe konfigurieren.

Eine Abwandlung der klassischen Visualisierung stellt die *Listenansicht mit Hervorhebung* dar. Hier werden die eingehenden Chat-Nachrichten chronologisch in einer Liste angezeigt. Die neueste Nachricht wird fett formatiert. Falls die Nachricht einer Bezugsnachricht zugeordnet ist, wird auch diese fett dargestellt. Falls diese wiederum eine Bezugsnachricht hat, wird diese ebenfalls fett markiert usw. So werden alle Bezugsnachrichten auf dem Pfad von der aktuellen Nachricht bis zur Wurzel hervorgehoben (vgl. Abbildung 1).

Abgesehen von den textuellen Varianten zur Visualisierung der Chat-Beiträge, stellt das von uns entwickelte Chat-System eine Reihe von baumbasierten Darstellungen zur Verfügung. All diesen Baumdarstellungen ist gemeinsam, dass die Chat-Beiträge als farbige Rechtecke und die Bezüge als Linien zwischen diesen Rechtecken dargestellt werden, jeder Benutzer durch eine bestimmte Farbe repräsentiert wird, Beiträge durch Klicken mit der linken Maustaste als Bezugsbeitrag ausgewählt werden können und danach eine spezielle Auswahlkennzeichnung haben und die Größe des Chat-Fensters nach Belieben verändert werden kann. Der Benutzer kann selbst festlegen, ob die Ansicht beim Eintreffen einer neuen Nachricht per Autofokus auf diese neue Nachricht positioniert werden soll. Da die Baumansichten einen größeren Platzbedarf haben, haben wir die Benutzeroberfläche um einen Überblicksbereich, in der immer der komplette Gesprächsverlauf zu sehen ist, ergänzt. Insgesamt unterstützen wir die folgenden vier Baumvisualisierungen:

- *Einfache Baumdarstellung*: Diese Baumvisualisierung verzichtet auf eine gesonderte Betrachtung des Parameters "Zeit" und ordnet die in Bezug stehenden Nachrichten in einem Baum an (siehe Abbildung 2).

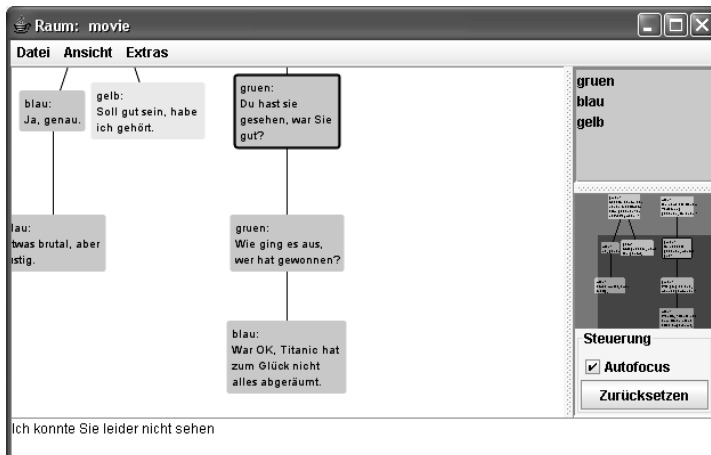
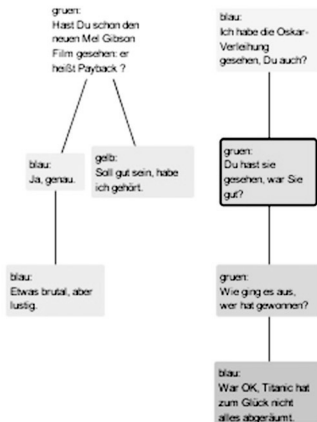


Abbildung 2: Benutzeroberfläche der einfachen Baumdarstellung

- *Baumdarstellung mit Visualisierung der Zeit durch hellere Farben*: Die Zeit wird dadurch berücksichtigt, dass ältere Nachrichten in helleren Farben und neuere Nachrichten in dunkleren Farben dargestellt werden (siehe Abbildung 3a)
- *Baumdarstellung mit Visualisierung der Zeit durch das Layout*: Die Zeit kann sich auch durch die proportionale Position der Nachrichten widerspiegeln: je mehr Zeit zwischen zwei Beiträgen vergangen ist, desto größer ist die Distanz. In diesem Fall entspricht die y-Achse der Darstellung der Zeitachse (Abbildung 3b).
- *Sequenzielle Baumdarstellung*: Bildet man die Zeit linear auf der y-Achse der zweidimensionalen Darstellung ab, wird durch größere Zeitabstände im Gesprächsverlauf viel Platz bei der Darstellung verschwendet. Aufgrund dieser Tatsache haben wir eine weitere Variante realisiert, bei der auf der y-Achse nicht die Zeit, sondern

die Reihenfolge der Beiträge dargestellt ist. Der Abstand zwischen den Nachrichten kann vom Benutzer festgelegt werden (vgl. Abbildung 3c).

a) Baumdarstellung mit Visualisierung der Zeit durch hellere Farben



b) Baumdarstellung mit Visualisierung der Zeit durch das Layout



c) Sequenzielle Baumdarstellung



Abbildung 3: Gegenüberstellung der weiteren Visualisierungen

4.3 Speicherung und Wiedergabe von Konversationen (A3)

Um einen Gesprächsverlauf nachvollziehen zu können, können Benutzer diesen im XML-basierten Format TreeML² speichern. In diesem Format gespeicherte Konversationen können vom Benutzer geöffnet und abgespielt werden. Das Replay-Fenster enthält dafür einen Bereich für die Steuerung der Wiedergabe der abgespeicherten Chat-Protokolle. In diesem Steuerungsbereich hat der Benutzer die Möglichkeit, die Wiedergabe der Konversation in seinem zeitlichen Verlauf zu starten, zu pausieren oder zu beenden. Die Geschwindigkeit der Wiedergabe kann darüber hinaus variabel verändert werden (von Echtzeit bis Faktor 60).

5 Erste Erfahrungen

Erste Einsätze von MuViChat, in denen Gruppen von 18 Personen für 90 Minuten miteinander kommuniziert haben, haben gezeigt, dass die Technologie auch in dieser Größe funktioniert und somit in mittelgroßen Diskussionsgruppen eingesetzt werden kann. Die Bedienung wurde unmittelbar verstanden und die Referenzen wurden wie vorgesehen eingesetzt. Alle Visualisierungen wurden ausprobiert und die Anwender entwickelten eigene Strategien,

² TreeML ist ein XML-Format zur Speicherung von Baumstrukturen, das für den InfoVis Contest 2003 der IEEE entwickelt wurde. Die Document Type Definition (DTD) für TreeML findet sich unter der folgenden Webadresse: <http://www.nomenclator.org/InfoVis2003/download/treeml.dtd>

welche Visualisierung sie wann einsetzen. Um den Gesprächsverlauf zu verstehen, wurde bevorzugt die einfache Baumdarstellung (Abbildung 2) oder die Baumdarstellung (Abbildung 3a) genommen, da hier die Distanzen zwischen referenzierten Beiträgen am kleinsten sind und der Nachvollzug am leichtesten fällt. Um aktiv an der Diskussion teilzunehmen und die aktuellen Gesprächsstränge zu verfolgen, wurde vor allem die sequentielle Baumdarstellung verwendet (Abbildung 3c). Diese vorläufigen Beobachtungen zeigen, dass die multiplen Visualisierungen von den Anwendern angenommen und ihren Bedürfnissen gemäß eingesetzt wurden.

In einer weiteren Pilotstudie wurde der Einsatz der Replay-Funktionalität zum Testen des Auftretens von Co-Text-Verlust beim Lesen eines Chat-Protokolls getestet. Dabei wurden jeweils drei Probanden in einer 30-minütigen Sitzung dasselbe Chat-Protokoll in unterschiedlichen Visualisierungen vorgespielt und zu den Bezügen zwischen den Nachrichten Fragen gestellt. Während in der Baum-Visualisierung alle Fragen korrekt beantwortet wurden, gab es bei der traditionellen Listendarstellung ohne Visualisierung der Referenzen mehrere Fälle, in denen die Referenz erst nach längerer Suche korrekt identifiziert werden konnte. Dabei tauchten Probleme auf, die auch in den Untersuchungen von Smith et al. (2000) beobachtet wurden: entweder das Übersehen von neuen Beiträgen, weil sie an verschiedenen Stellen auftauchen oder bei Einsatz des Autofokus der Effekt, dass Beiträge zu kurz erscheinen, um sie lesen zu können, weil der Autofokus wieder auf einen neuen Beitrag zentriert. Die Lösung dieser Probleme wird eines der Ziele zukünftiger Erweiterungen sein.

6 Zusammenfassung und Ausblick

MuViChat ist sowohl ein neuartiges Chat-Tool, als auch eine Umgebung zum Experimentieren und Evaluieren von Chat-Visualisierungsvarianten. Durch die Verwendung von verbreiteten Standards (Java und Jabber) ist MuViChat auf jedem Java-fähigen Computer mit Internetanschluss ohne Installation lauffähig. Die Logfiles können als XML-Datei abgespeichert und in unterschiedlichen Visualisierungen angezeigt werden (auch in Visualisierungen, die erst in Zukunft noch entwickelt werden). Dadurch werden mit geringem Aufwand Experimente möglich, bei denen Anwendern unter kontrollierten Bedingungen neuartige Visualisierungen von Chat-Protokollen vorgespielt werden, um die Nützlichkeit und Lesbarkeit der Visualisierungen zu testen. Da MuViChat Open-Source ist, kann von interessierten Forschern eine Vielzahl von Visualisierungsvarianten erzeugt und durch die modulare Architektur mit geringem Aufwand in MuViChat integriert werden.

Literaturverzeichnis

Beißwenger, M. (2000). *Kommunikation in virtuellen Welten: Sprache, Text und Wirklichkeit. Eine Untersuchung zur Konzeptionalität von Kommunikationsvollzügen und zur textuellen Konstruktion von Welt in synchroner Internet-Kommunikation, exemplifiziert am Beispiel eines Webchats*. ibidem-Verlag, Stuttgart.

- Harnoncourt, M., Holzhauser, A., Seethaler, U. & Meinel, P. (2005). Referenzierbarkeit als Schlüssel zum effizienten Chat. In: *Chat-Kommunikation in Beruf, Bildung und Medien: Konzepte – Werkzeuge – Anwendungsfelder*. Stuttgart, S.161-179.
- Herring, S. (1999). Interactional Coherence in CMC. *Journal of Computer-Mediated Communication*, 4(4).
- Holmer, T. & Wessner, M. (2005). Gestaltung von Chat-Werkzeugen zur Verringerung der Inkohärenz. In: *Chat-Kommunikation in Beruf, Bildung und Medien: Konzepte – Werkzeuge – Anwendungsfelder*. Stuttgart, 181-199.
- Kienle, A. & Holmer, T. (2006). Zur Nutzung von Referenzierungsfunktionalitäten in Lernchats. In: DeLFI 2006, 4. e-Learning Fachtagung Informatik, 11.-14. September 2006, Darmstadt, (DeLFI), 2006, 87, S. 111-122
- Mühlpfordt, M. & Wessner, M. 2005. Explicit referencing in chat supports collaborative learning. In: *Proceedings of the 2005 Conference on Computer Support for Collaborative Learning: Learning 2005: the Next 10 Years!* (Taipei, Taiwan, May 30 – June 04, 2005), S. 460-469.
- Pimentel, M. G., Fuks, H. & de Lucena, C. J. P. (2003). Co-text Loss in Textual Chat Tools. In: *Proceedings of Fourth International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT 2003)*, LNAI 2680, S. 483–490. Springer-Verlag Berlin Heidelberg.
- Pimentel, M. G., Fuks, H. & de Lucena, C. J. P. (2006). R-U-Typing-2-Me? Evolving a chat tool to increase understanding in learning activities. In: *International Journal of Computer-Supported Collaborative Learning*, 1(1), S. 117-142.
- Storrer, A. (2001). Getippte Gespräche oder dialogische Texte? Zur kommunikationstheoretischen Einordnung der Chat-Kommunikation. In: *Sprache im Alltag. Beiträge zu neuen Perspektiven in der Linguistik*. Berlin.
- Smith, M., Cadiz, J. J. & Burkhalter, B. (2000). Conversation trees and threaded chats. In: *Proceedings of the 2000 ACM Conference on Computer supported cooperative work*, S. 97– 105. ACM Press, New York, NY.
- Vronay, D., Smith, M. & Drucker, S. (1999). Alternative interfaces for chat. In: *Proceedings of the 12th annual ACM symposium on User interface software and technology*, S. 19–26. ACM Press, New York, NY.