

Groupware

Andreas Neu
RWTH Aachen

Sören Busch
RWTH Aachen

1. EINLEITUNG

Im Zeitalter der Globalisierung und Vernetzung ist der englische Begriff "Teamwork" allgegenwärtig. Es ist keine Seltenheit, dass große Unternehmen weltweit rund um die Uhr an Projekten arbeiten. Arbeit die tagsüber in Europa fertiggestellt wird, kann Nachts in Asien weiterbearbeitet werden. Darüber hinaus müssen ausgelagerte Arbeiten ständig eingesehen werden können, was ein hohes Maß an Kommunikation und geteilten Ressourcen erfordert.

Durch Automatisierung solcher Arbeiten lässt sich viel Zeit sparen und Fehler vermeiden, was im Endeffekt auch eine Einsparung an Finanzmitteln bedeutet. Darüber hinaus werden die Mitarbeiter an einem Projekt entlastet, sodass ihre Arbeitsleistung effektiver ausfällt.

Es ist also eine Notwendigkeit stabile Systeme zu entwickeln, welche sich den Prozessen innerhalb eines Teams anpassen und darüber hinaus so benutzerfreundlich gestaltet sind, dass sie die Mitglieder entlasten.

Systeme für viele Benutzer, welche wir von nun an mit "Groupware" bezeichnen werden, haben ihre eigenen Problematiken. Während visuelle Darstellung und weitere Bereiche der Systeme noch mit den bisher im Seminar vorgestellten Regeln behandelt werden können, müssen die speziellen Probleme der Groupware anders gelöst werden.

Mit einer Vorstellung einzelner Systemarten, ihrer Probleme und möglichen Lösungen beschäftigt sich unsere Arbeit.

2. KLASSIFIZIERUNG VON SYSTEMARTEN

Wie in der Einleitung beschrieben, sollte sich ein gutes System den Arbeitsprozessen eines Teams anpassen, um somit ein hohes Maß an Effizienz erzielen zu können.

Da jedoch jedes Team unterschiedlich ist und somit andere Anforderungen an ein System stellt, muss schon während der Design-Phase ein hohes Maß an Aufmerksamkeit auf das Umfeld des Einsatzbereiches gelegt werden.

Darüber hinaus wird in vielen Firmen ein hoher Aufwand an Aufzeichnungsarbeit betrieben, um somit den Arbeitsablauf protokollieren zu können. Auch solche Dinge müssen vom Designer der Groupware-Applikation beachtet werden, da

ansonsten wieder aufwändige Arbeitsschritte an die Mitarbeiter zurückfällt.

Wichtige Fragen die während des Design auftauchen sollten sind:

- Wie ist das Team strukturiert? Gibt es Hierarchien, welche man beachten sollte?
- Wie ist der Kommunikationsfluss gestaltet?
- Was wird zeitgleich und was zeitlich versetzt gemacht?
- Sind verschiedene Mitarbeiter an räumlich entfernten Punkten?

Durch Gespräche mit den Kunden des Systems können solche Fragen leicht geklärt werden. Eine direkte Einbindung der Zielgruppe kann nur Schwächen im eigenen Design aufzeigen und somit frühzeitig Fehler ausbessern.

2.1 Die Zeit- /Raum Matrix

Als Hilfsmittel eignet sich die sehr einfache, aber übersichtliche Zeit-/Raum Matrix. Sie besteht aus zwei Zeilen und Spalten.

	beisammen	entfernt
synchron		
asynchron		

Abbildung 1: 1. Zeit- /Raum Matrix

Die Zeilen sind mit einer zeitlichen Unterscheidung beschriftet, während die Spalten eine räumliche Differenz zeigen.

Während es sehr oft möglich ist, diese Unterscheidung klar vorzunehmen, kann es dennoch vorkommen, dass sich einzelne Bereiche des Systems sowohl asynchron, als auch synchron benutzen lassen. Eine frühzeitige Erkennung dieses Sachverhalts kann bei der technischen Umsetzung helfen unnötigen Arbeitsaufwand einzusparen, da Gemeinsamkeiten beider Modi zusammen implementiert werden könnten.

Diese Unterscheidung durch die Zeit-/Raum Matrix wird im weiteren Verlauf des Textes als Beschreibungskriterium genutzt, da sie intuitiv ist.

2.2 Wichtige Systemarten

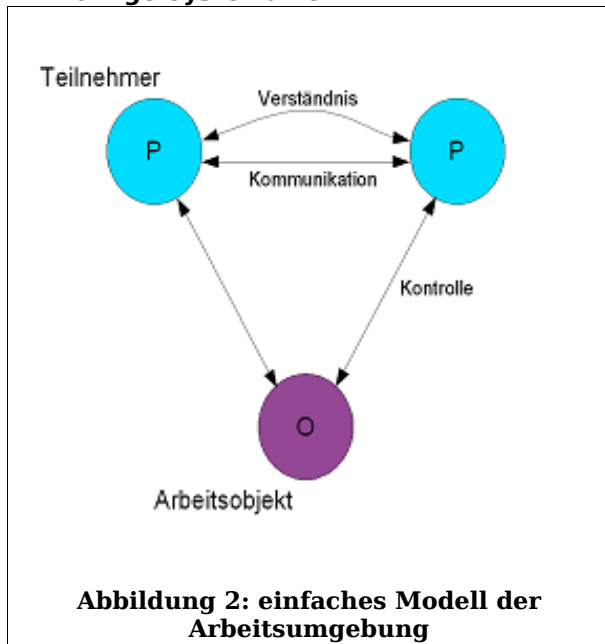


Abbildung 2: einfaches Modell der Arbeitsumgebung

Diese einfache Grafik wirft einige der schon aufgelisteten Design-Fragen auf. Sie ist eine vereinfachte Darstellung der Arbeits-umgebung. Es existieren mehrere Nutzer, kenntlich gemacht durch die Kreise mit der Beschriftung P und ein Objekt (Kreis mit Beschriftung O). Beide Nutzer arbeiten an oder mit dem Objekt und müssen sich dabei über den Prozess austauschen können. Darüber hinaus sollten sich beide Nutzer immer über die Ziele im Klaren sein. Ihr Verhältnis untereinander, jedoch auch zu Objekt wird durch die Pfeile dargestellt. Welchen Einfluss haben sie auf das System oder das Produkt und wie verhält es im Gegenzug darauf?

Im normalen Arbeitsalltag großer Firmen, wird das gemeinsame Verständnis über die Ziele durch Meetings festgehalten. Es ist also durchaus lohnenswert dahingehend automatisierende Systeme zu erschaffen, welche den Mitgliedern eines Teams die Aufzeichnungen einfach und schnell bereitstellen.

Das Zusammenspiel von Objekt und Nutzern kann wie folgt erklärt werden: Oft müssen mehrere Mitgliedern an ein und demselben Dokument Änderungen vornehmen. Hier ist ein hohes Maß an Sicherheit erforderlich, dessen Einhaltung durchaus dem Groupware-System überlassen werden kann.

Von dieser Grafik ausgehend, unterscheiden wir zwischen folgenden System-Arten:

- Systeme zur Kommunikation

- Systeme zur Unterstützung von Meetings
- geteilte Arbeitsumgebungen und Werkzeuge

3. VERSCHIEDENE SYSTEM-ARTEN

In diesem Abschnitt wollen wir nun Vertreter dieser 3 System-Arten vorstellen. Dabei soll auch ein Blick auf die Probleme und ihre möglichen Lösungen gelegt werden.

3.1 Systeme zur Kommunikation

Jede gemeinschaftliche Arbeit benötigt eine gewisse Form der Kommunikation. Je nach Struktur der Arbeitsumgebung wird eine andere Form bevorzugt.

Da eine computer-gestützte Kommunikation in sofern überflüssig wäre, wenn die Mitglieder der Arbeitsgemeinschaft direkt nebeneinander sitzen würden, können wir gemäß der vorgestellten Zeit- / Raum Matrix unser Augenmerk auf die zeitliche Unterscheidung legen.

3.1.1 Asynchrone Kommunikationssysteme

Diese Form der Kommunikation ist ideal für Strukturen geeignet, in denen keine direkte Unterhaltung nötig ist. Dies umfasst reine Dinge wie Informations-Weitergabe, da hier nicht unbedingt ein direktes Feedback des Empfängers nötig ist.

Große Firmen, mit vielen Unterabteilungen profitieren von solch einer Art des Kommunikationswegs. Es wäre einfach zu aufwändig auf die Anwesenheit jedes Empfängers zu warten. Auch Unternehmen mit weltweiten Vernetzungen dienen asynchrone Systeme, da aufgrund der Zeitzonen ein Gesprächspartner nachts wach bleiben müsste. Bekannte Vertreter asynchroner Kommunikationssysteme sind:

- E-Mail
- SMS

Im weiteren wollen wir unsere Betrachtung auf die Vor- und Nachteile des E-Mail Systems richten.

Die Möglichkeit E-Mails zu versenden ist einer der ältesten Dienste des Internets. Die Nachrichten, welche versandt werden basieren auf einem sehr leicht-gewichtigen System. Jede Nachricht besteht nur aus 3 Komponenten:

- Kopf (Header)
- Körper (Body)
- Inhalt

Da der Inhalt nur aus ASCII-Symbolen besteht und somit unformatiert bleibt, wird pro Zeichen nur 8 Byte verwendet. Somit ist jede Nachricht sehr leicht und kann sehr schnell übertragen werden.

Ein weiterer Vorteil liegt in der Möglichkeit die Nachricht an viele Menschen gleichzeitig zu versenden. Die meisten Applikationen, welche zum Verschicken und Abrufen von E-Mails dienen, erlauben es Gruppen anzulegen, sodass per Knopfdruck die Nachricht an alle Personen versandt werden kann, welche auf der Liste stehen.

Ein weiterer Vorteil liegt in der leichten Erweiterbarkeit und Nutzung in anderen Systemen. Aufgrund ihres einfachen Aufbaus können sie leicht Archiviert werden und belasten aufgrund ihrer Größe nicht die Datenbanken. Darüber hinaus können die Nachrichten nach Wörtern durchsucht werden, was mit formatierten Texten Schwierigkeiten bereiten könnte.

Am Beispiel des Systems "SaveMe" [1] sieht man, dass E-Mails als Grundbaustein von Archivierungs-Software genutzt werden kann. Nachrichten werden klassifiziert und per E-Mail an einen konfigurierten Server geschickt. Dieser behandelt die Nachricht aufgrund gegebener Informationen und legt sie in bestimmte Fächer ab. Fächer sind dabei wie Dateisysteme in einstellbaren Hierarchien strukturiert.

Die Nachteile von E-Mail als Kommunikationsmittel liegen in Arbeitsumgebungen, in denen direkte Gespräche nötig sind. Obwohl es rein theoretisch möglich ist, E-Mails wie in einem Chat zu versenden, bleibt es jedoch eine unangenehme Form. Deshalb sollte man in solchen Fällen auf synchrone Systeme zurückgreifen.

3.1.2 Synchrone Kommunikationssysteme

Diese Form der Kommunikation ist heutzutage wichtiger denn je. Aufgrund der weltweiten Vernetzung ist das Leben für Privatmenschen, als auch für Unternehmen hektischer geworden. Über weite Strecken müssen in kürzester Zeit Entscheidungen getroffen und Prozesse koordiniert werden.

Schon seit einigen Jahrzehnten ist und bleibt das Telefon wohl das wichtigste System zu direkten Kommunikation über weite Strecken. Doch durch die weltweite Akzeptanz des Internets beginnen kostengünstigere Dienste den Markt zu erobern:

- Instant Messaging
- Voice & Video Chat
- virtuelle Realitäten

Obwohl virtuelle Welten wie "Second Life" aus dem Hause Linden Lab, immer mehr Menschen und Unternehmen in ihren Bann ziehen, wollen wir eine Betrachtung dieser Systeme außen vor lassen. Sie sind unserer Meinung nach noch zu spielerisch, als das sie im Alltag der Unternehmen einen Platz finden könnten.

Deshalb wollen wir unser Hauptaugenmerk auf die

sehr oft kombinierten Instant Messaging und Voice/Video Chat Systeme richten.

Vertreter 1 Instant Messaging:

Das seit ICQ weltweit bekannt gewordene Instant Messaging ist ein sehr stabiler Kommunikationsweg. Zwei oder mehr Kontakte können sich dabei über Text-Nachrichten bequem austauschen.

Um es zu nutzen können, benötigt jeder Teilnehmer einen Client. Gespräche laufen zumeist "Peer to Peer", also ohne Einbindung eines Servers. Der Server dient meist nur zur Verwaltung der Benutzer und hält fest, welche Benutzer sich gerade eingeloggt haben. Der Client des Nutzers kann dann per Anfrage an den Server eine Verbindung zu einem anderen Nutzer eröffnen. Ab hier sind dann die Nutzer im direkten Gespräch, ohne den Umweg über einen Server nehmen zu müssen.

Instant Messaging Systeme eignen sich sehr gut für Teamarbeit, da sie zumeist sehr stabil und verlustfrei sind. Obwohl wie schon beschrieben, eine direkte Kommunikation über asynchrone Systeme wie E-Mail theoretisch möglich ist, bleiben Vertreter wie ICQ in der Gunst der Nutzer unangefochten. Die Nutzung ist bequemer, weil zusätzlicher Aufwand, wie das Adressieren und Verschicken ausbleibt.

Eine weitere Verbesserung gegenüber Telefongesprächen und E-Mails, stellt die Transparenz des Instant-Messaging dar: Die meisten Clients unterstützen verschiedene Modi, welche den aktuellen Status des Nutzer widerspiegeln. Durch diese Funktionalität ist es jedem Kontakt möglich, sich darauf einzustellen, wann überhaupt eine Antwort möglich sein wird: Wenn der Gegenüber nicht da ist, so wird er auch nicht sofort antworten. Darüber hinaus können selbst geschriebene Status Nachrichten Gründe für Abwesenheit usw. offenbaren.



Vertreter 2 Voice & Video Chat:

Die direkte Kommunikation über das Internet wird in naher Zukunft von Unterhaltungen per Voice &

Video Chat dominiert werden. Dieses System kommt dabei unseren Vorstellungen eines persönlichen Gesprächs am nächsten.

Gestik und Mimik sind in unserer Interaktion mit anderen Menschen unerlässlich, da sie jede Botschaft unbewusst unterstützen.

Jeder Teilnehmer nutzt dabei eine Kamera und ein Mikrofon. Bild und Ton Signale werden dabei digitalisiert und über das Internet an den Computer des Gesprächspartner geschickt. Dessen Client nimmt die Daten an und gibt sie über die entsprechenden Geräte wie Sound-Boxen aus.

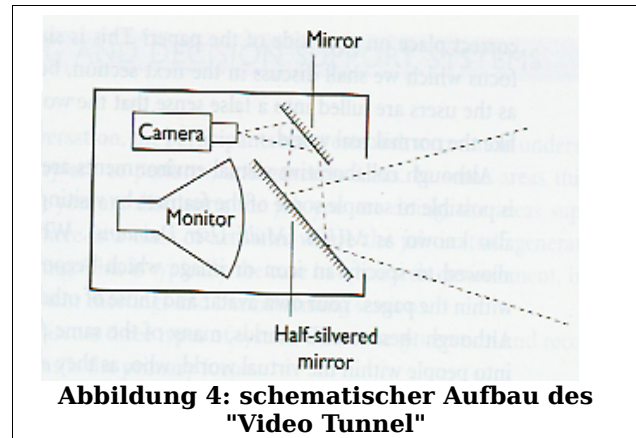
Die schon beim Instant-Messaging angesprochene Technik, zwei oder mehr Teilnehmer direkt zu verbinden und dabei den Server nur zum Verbindungsaufbau zu nutzen, wird auch hierbei genutzt. Es wäre einfach zu langsam die Daten in einer ständigen Verbindung über den Server zu übertragen.

So schön diese Technik auch im Ideal klingen mag. Sie hatte deutliche Nachteile zum jetzigen Zeitpunkt. Aufgrund der noch zu geringen Bandbreiten, können nicht alle Daten unverändert an den anderen Nutzer gesandt werden, da es viel zu lange dauern würde. Aus diesem Grunde werden Algorithmen verwendet, welche versuchen die Menge an Daten zu verringern, indem sie die Qualität der Bilder und des Tons mit heuristischen Annahmen mindern. So werden zum einen wie bei der Konvertierung in das mp3 Tonformat Feinheiten herausgenommen, welche der Benutzer sowieso nicht bewusst wahrnimmt.

Doch wir können davon ausgehen, dass in Zukunft durch immer bessere Bandbreiten eine Übertragung ohne qualitative Einbußen möglich sein wird.

Ein anderes Problem des Voice & Video Chats liegt vielmehr in der Wahrnehmung seiner Nutzer. Obwohl wir die Stimme unseres Gesprächspartners hören und sogar sein Bild sehen, sind wir nicht zufrieden. Die eingesetzten Kameras verbleiben starr an einer Position. Da der Gegenüber auch auf unser Abbild auf dem Bildschirm fixiert ist, wird er selten Augenkontakt mit der Kamera haben. Als Nutzer nimmt man dies wahr und empfindet es als störend. Abhilfe sollen hier spezielle Kameras mit dem Namen "Video Tunnel" schaffen, welche durch Spiegel-Technik dem Blick seines Nutzers folgen.

Trotz guter Ergebnisse besitzen wenige Anwender diese Art von Kameras, da sie aufgrund des technischen Aufwands noch zu teuer sind.



3.2 Systeme zur Unterstützung von Meetings

Wie schon in Kapitel 1.2 beschrieben, benötigt eine erfolgreiche Arbeit in einem Team nicht nur Kommunikation während des Arbeitsprozesses, sondern auch eine klare Vorstellung über die Ziele des Projekts.

Oft werden in Design-Phasen eines Projektes Beratungen abgehalten, wo die Mitglieder eines Teams Ideen zusammentragen. Des weiteren finden während späterer Phasen in regelmäßigen Abständen Konferenzen statt, um die bisherigen Ergebnisse zusammenzutragen. Die Teilarbeiten einzelner Gruppen oder Personen wird aus Sicht des gesamten Projektes betrachtet und analysiert.

Techniken, welche wir im Kapitel über Voice & Video Chat angesprochen haben, könnten sogar eingesetzt werden, um Meetings trotz verschiedener Standorte ihrer Teilnehmer zu vernetzen.

Aus der Beschreibung solcher Treffen und Konferenzen lassen sich leicht die Anforderungen an ein dort verwendetes Groupware System klassifizieren: Egal wo es stattfindet, ein Meeting kann nicht asynchron verlaufen. Aufgrund dieser Feststellung brauchen wir nur räumliche Unterscheidungen zu treffen und dahingehend zu betrachten.

Groupware Systeme können unter solchen Voraussetzungen verschiedene Einsatz-Gebiete haben. Auf der einen Seite können sie dazu genutzt werden, mehrere Räume miteinander zu verbinden, während andere wiederum die einzelnen Teilnehmern beim Zusammentragen der Informationen unterstützen. So wurden Räume entwickelt, in denen jeder Teilnehmer einen eigenen Computer an seinem Sitzplatz benutzt, um von dort aus, das auf einem Haupt-Bildschirm angezeigte Dokument zu verändern. Es können aber auch einzelne Dokumente bequem von derzeitigen Sitzplatz aus an alle anderen verschickt werden, um

somit Ideen zu vergleichen.



Abbildung 5: Computer-gestützter Konferenzsaal

Ein weiterer wichtiger Punkt sind soziale Regeln bei Meetings. Jeder Teilnehmer sollte das Recht haben an Ideen und Änderungen teilzuhaben, ohne dabei von anderen unterbrochen zu werden. Wenn nun wie oben beschrieben jeder Teilnehmer das Dokument einfach so verändern kann, wird es unweigerlich zu Kollisionen kommen: Beispielsweise könnten zwei Personen denselben Eintrag verändern wollen. Auch hier kann ein Groupware System eingesetzt werden, indem es Bearbeitungsrechte verteilt. Diese Bearbeitungsrechte sollten nicht als Beschränkung durch das System, sondern vielmehr als Unterstützung betrachtet werden, welche den Arbeitsprozess und somit dem gesamten Team helfen.

Wie zu Beginn des Kapitels schon beschrieben, werden Meetings auch noch nach der Design-Phase abgehalten, um alle Mitglieder auf dem aktuellsten Stand der Entwicklung zu bringen. Da in der Entwicklung selten alles reibungslos funktioniert kann es zu Problemen mit dem bisherigen Design kommen. Dinge, die noch während der Design Phase als machbar angenommen wurden, stellen sich vielleicht plötzlich als unmöglich oder schlecht heraus. Es müssen also Änderungen vorgenommen, Datenbanken erneuert und ganze Dokumenten-Berge verteilt werden.

Auch hier können computer-gestützte Systeme Abhilfe schaffen, indem sie solche lästigen Arbeiten automatisieren. Dem Menschen können häufig Fehler bei sturen Überarbeitungen von alten Dokumenten unterlaufen. Systeme die automatisch Änderungen festhalten sind wesentlich besser dafür geeignet, was im Endeffekt auch einen Produktionsgewinn des Teams mit sich bringt.

Während ein komplett ausgestatteter Raum sehr

viel Geld kostet gibt es für kleiner Budgets auch einfachere Möglichkeiten, welche die oben beschriebenen Arbeiten des Digitalisierens und Automatisieren übernehmen können: Es handelt sich dabei um kleine Aufsätze für weiße Tafeln namens "Whiteboards". Sie sollen die Bewegungen des Benutzers aufzeichnen und als digitalisierte Dokumente verarbeiten. Der Benutzer muss dabei lediglich einen speziellen Stift verwenden, der von dem System erkannt wird. Klare Vorteile dieser Lösung sind:

- Portabilität
- Kostenfaktor
- nicht aufdringlich



Abbildung 6: eBeam System 3 als kostengünstiges Beispiel

Der wichtigste Faktor bei all diesen Systemen zur Unterstützung von Meetings ist, dass sie wenn möglich nicht als Computer wahrgenommen werden sollten. Sie sollten sich wunderbar in die Umgebung des Meetings einfügen, zum Beispiel durch Einbau der Arbeitsstation in die kleinen Tische am Stuhl. Mit dem Fortschreiten der Nano-Technologie werden wir wohl in Zukunft ganze Räume erleben, welche wie in Science-Fiction teilweise von Computern gesteuert werden.

3.3 geteilte Arbeitsumgebungen und Werkzeuge

Alle Systeme, welche wir bisher vorgestellt haben, dienen hauptsächlich zur Unterstützung von Arbeitsabläufen zwischen Mitgliedern im Team. Nun wollen wir jedoch eine Art von Systemen beschreiben, welche die Produktion, sei es nun Schreiben oder Programmieren als gemeinschaftlichen Prozess in den Vordergrund rücken.

Bis jetzt wurde die Arbeit einzelner Mitglieder durch Kommunikation und Meetings mit den der

anderen verbunden. Geteilte Werkzeuge schlagen nun eine Brücke zwischen unterschiedlichen Arbeiten, indem sie Ressourcen, Fenster-Systemen oder sogar ganze Workstation teilen. Das entstehende Produkt kann nun von allen Seiten bearbeitet werden und ist somit wirklich ein gemeinschaftliches Erzeugnis.

Um diese Systeme klassifizieren zu können bedarf es einer reinen zeitlichen Unterscheidung. Entfernungen verändern nur den technischen Aufwand. Es ändert nichts daran, dass die Ressourcen geteilt werden.

Die HCI Forschung versucht auf verschiedene Arten solche Systeme zu verbessern. Neben den generellen Regeln für GUIs müssen hierbei andere Dinge als in Einzel-Nutzer Systemen beachtet werden.

Sobald mehrere Menschen auf dieselben Ressourcen zugreifen, kann es automatisch zu Kollisionen kommen, wenn beispielsweise zwei Nutzer gleichzeitig genau dasselbe ändern wollen. Man spricht dabei von einem "concurrent write", der beispielsweise auch bei der Benutzung paralleler Prozessorarchitekturen auftauchen kann. Fehler die hierdurch entstehen können über lange Zeit unbemerkt Nachwirkungen hervorrufen und sollten deshalb durch Sicherheitsmechanismen unmöglich gemacht werden.

Dafür benutzt man in geteilten Werkzeugen einen ähnlichen Ansatz wie bei den Systemen zur Unterstützung von Meetings. Während dort eine Kollision noch belächelt werden kann, sollte dies in geteilten Arbeitsumgebungen zum Standard werden, weil hierbei neben System-Abstürzen auch wirklich unerkennbare Fehler auftauchen können.

Um dies zu vermeiden wird auch hier normalerweise ein Verschluss-Mechanismus eingesetzt, der jene Daten vor weiteren Zugriffen schützt, welche gerade von einer Person verändert werden.

Während dieser Punkt eine reine Notwendigkeit solcher Systeme darstellen sollte, gibt es andere Punkte, welche mehr vom Einsatzbereich des Systems bestimmt werden.

Zum Beispiel könnte die Frage "Sollen alle Benutzer haargenau das gleiche sehen?" auftauchen. Obwohl dieser Ansatz für viele Privat-Anwender auf den ersten Blick unsinnig erscheinen mag, kann es für bestimmte Arbeitsbereiche äußerst wichtig sein. Wenn zwei Benutzer sich über weite Strecken beispielsweise daran versuchen einen bestimmten Textabschnitt zu bearbeiten, wäre es durchaus von Vorteil, wenn ein jeder dem anderen ohne Umstände zeigen könnte, auf welchen Teil er sich gerade bezieht. Dieser Mechanismus wird gerne in geteilten Editoren benutzt, damit beide Nutzer wirklich nach dem Grundsatz "Was ich sehe, siehst

auch du" arbeiten können.

Solche Funktionalität nicht das Design des geteilten Werkzeug bestimmen, sondern vielmehr als nützliche Eigenschaft vorhanden sein. Ein Modi-Wechsel könnte somit auch die Arbeit an verschiedenen Abschnitten ermöglichen und nur wenn benötigt dasselbe auf dem Bildschirm anzeigen.

Eine weitere Funktionalität, welche durchaus als vorteilhaft angesehen werden kann, ist die automatische Buchführung über Änderungen durch das System. Wie schon im Kapitel über Kommunikationssysteme beschrieben, ist es manchmal erforderlich Arbeitsschritte zu dokumentieren. Während eine einzelne Person über Änderungen noch gut Buch führen kann, kann es in Team-Arbeiten zu Problemen kommen. Eine Person nimmt Änderungen an einem Dokument vor, ohne dabei diese aufzuzeichnen. Der ehemalige Autor des Abschnittes wird sich wundern, was passiert ist und möglicherweise alle Änderungen rückgängig machen.

Wenn das Team mit handelsüblichen Editoren arbeitet, wird es notwendig solche Änderungen in anderer Form festzuhalten, weil solche Systeme eine solche Funktionalität automatisiert nicht anbieten.

Genau mit dieser Dokumentationsform macht der an der University of British Columbia geteilte Editor [2] ein Ende. Sie benutzten E-Mails als Grundlage ihrer Beobachtungen. Autoren in Teams benutzen zumeist E-Mails, um über den Stand an Änderungen Buch zu führen. Sie analysierten diese und konnten somit Einteilungskriterien aufstellen. In diesem System werden nun Änderungen der Autoren mit Namen, Datum und Anmerkungen versehen, welche sich in bestimmte Klassen einordnen lassen. Dadurch kann über lange Zeit die Arbeit dokumentiert und nachvollzogen werden.



Abbildung 7: Anmerkungen in einem kleinen Fenster unter dem Text

Nach einem ähnlichen System funktionieren offene Systeme wie Wikipedia. Jede Änderung wird festgehalten und kann als Grundlage für Verbesserungen und Diskussionen dienen.

4. RAHMENBEDINGUNGEN ZUM DESIGN

Nun sollen einige Punkte aufgeworfen werden, die beim Design eines Groupware-Systems zu bedenken sind und durchaus auch Relevanz in der Einteilung schon bekannter Systeme haben.

4.1 Zeit-/Raum Matrix

Das alte Modell der Zeit-/Raum Matrix ist zwar sehr intuitiv und bei der Einordnung von Systemen hilfreich, allerdings ist es an einigen Stellen sehr ungenau und zur Betrachtung von Groupware-Systemen wenig hilfreich

	beisammen	entfernt
synchron	Konferenzraum	Videokonferenz
asynchron	Argumentationswerkzeuge	E-Mail

Abbildung 8: die Raum-/Zeit Matrix nun mit den bereits vorgestellten Systemen

Nun spielt es für asynchrone Systeme allerdings prinzipiell keine Rolle ob die Teilnehmer sich am gleichen Ort aufhalten oder nicht, weshalb diese Unterteilung im folgenden weglassen wird. Im Gegenzug soll die zeitliche Unterteilung genauer betrachtet werden.

Dazu unterscheidet man zwischen simultanen Systemen (alle Teilnehmer können gleichzeitig arbeiten), seriellen (nur ein aktiver Teilnehmer zu jeder Zeit), gemischten (beides möglich) und unsynchronisierten Systemen.

Daraus ergibt sich nun die neue Matrix:

	beisammen	entfernt
simultan	Konferenzraum geteilte Editoren / Fenstersysteme	Videokonferenz
seriell	Argumentationswerkzeuge	
gemischt	Co-Autor Systeme	
unsynchronisiert	E-Mail	

Abbildung 9: zwar nicht mehr so übersichtlich, dafür aber genauer in der Klassifizierung

4.2 Granularität

Für viele Systeme ist es wichtig zu beachten, wie lange es dauert bis Veränderungen, die von anderen Teilnehmern vorgenommen wurden angezeigt werden.

Eine häufige Aktualisierung muss nicht zwingend nötig sein. Sperrt ein System z.B. eine Datei während diese bearbeitet wird für andere Teilnehmer, so ist es vollkommen ausreichend die Änderungen auch erst bei Abschluss anzuzeigen.

Bei einem Editor hingegen, der es Teilnehmern ermöglicht sogar am gleichen Wort zu arbeiten wäre eine Verzögerung von nur Sekunden absolut inakzeptabel, da so ein kollisionsfreies miteinander Arbeiten ausgeschlossen ist.

Für Systeme in denen simultane Arbeit nicht als grundlegendes Design-Element gilt, jedoch auch nicht ausgeschlossen werden kann, wird häufig ein Mittelweg gewählt. Dieser besteht darin verschiedene Modi anzugeben, zwischen welchen die Nutzer flexibel schalten können.

4.3 Präsentation

Bei der Betrachtung von Systemen die keine simultane Arbeit unterstützen, erscheint es offensichtlich, dass die Art in der den verschiedenen Teilnehmern die Daten präsentiert werden, keine Rolle spielt.

In einem simultanen System hingegen hat weitreichende Auswirkungen, ob alle das Gleiche vorgezeigt bekommen oder nicht. Als Beispiele seien hier zum einen ein geteiltes Fenstersystem genannt. Da hier einfach nur das normale Computerbild an mehrere Stationen verteilt wird, sehen auch alle Teilnehmer jederzeit das gleiche. Geteilte Editoren hingegen ermöglichen es den Teilnehmern getrennt durch das Dokument zu navigieren und an verschiedenen Stellen zu arbeiten.

Beide Ansätze bringen ihre eigenen Probleme mit sich. Einerseits reduziert eine einheitliche Präsentation Kommunikationsprobleme (schließlich ist es einfacher sich auf etwas gezeigtes zu beziehen, wenn auch alle das gleiche sehen). Andererseits kann es leicht zu Konflikten zwischen den Betrachtern kommen, da nicht unbedingt bei allen der Fokus an der gleichen Stelle ist.

4.4 Eingabe

Direkt mit der Präsentation hängt auch die Art der Eingabe zusammen. Hier wird zwischen vier Abstufungen unterschieden:

- Gemeinsame virtuelle Tastatur

Hierbei teilen sich alle Teilnehmer ein Eingabe-Medium. Aufgrund von Kollisionen erscheint ein computergesteuertes Regelsystem zweckmäßig.

- Andere Teilnehmer sichtbar

Jeder Teilnehmer hat seine eigene Eingabe und kann die Aktivitäten der anderen Teilnehmer ebenfalls sehen. Da Konflikte so

nicht stattfinden können ist so etwas wie Sperrung auch nicht notwendig.

- **Gruppenzeiger**

Die Teilnehmer können die fremden Eingaben nicht sehen, aber es gibt einen Gruppenzeiger, den jeder bedienen kann um auf Objekte zu verweisen. Da auch hier Konfliktpotenzial vorhanden ist werden ebenfalls Regeln zur Sperrung benötigt

- **Andere Teilnehmer nicht sichtbar**

Alle Teilnehmer haben ihre eigenen Eingaben und können die der anderen auch nicht sehen. Dies kommt bei allen nicht-simultanen Systemen zum Einsatz

4.5 Transparenz

Für eine produktive Zusammenarbeit ist es nötig, dass die Teilnehmer auch erfahren, was außerhalb ihrer eigenen Aktivitäten geschieht.

Eine Frage die sich oft gestellt wird ist "Wer ist gerade erreichbar oder am arbeiten?" Der bekannteste Ansatz diese zu beantworten stammt sicherlich aus der Welt der Instant-Messenger. Fast alle diese Programme bieten die Möglichkeit eine Status-Nachricht zu hinterlassen, so dass jeder erfahren kann, womit man denn gerade beschäftigt ist. Eine Funktion die sicherlich auch in anderen Programmen, die es mehreren Menschen ermöglichen, gleichzeitig zu arbeiten, denkbar ist um etwa andere Teilnehmer über etwaige Inaktivitäten zu informieren oder ähnliches.

Im Bereich des nicht-simultanen Arbeitens tritt hingegen öfter die Frage "Was ist mit einem Objekt passiert und wie?" auf. Um dem Benutzer die Mühe zu ersparen ständig verschiedene Versionen einer Datei von Hand zu vergleichen bieten v.a. Ko-Autoren-Systeme vorgefertigte Funktionen zum Versionsabgleich.

5.IMPLEMENTATION SYNCHRONER SYSTEME

Während die vorhergegangenen Kapitel sich stark auf idealisierte Systeme bezogen, wollen wir nun auf technische Probleme aufmerksam machen, welche bei der Implementation synchroner Systeme zu beachten sind.

Die meisten Systeme im Bereich Groupware werden in Netzwerken benutzt, da die Verbindung innerhalb eines Gebäudes oder mit dem Internet in den meisten Unternehmen und Privathaushalten zum Standard geworden ist. Es ist also kein Wunder, dass die Entwickler von Groupware Systemen sich dieser Marktstruktur anpassen.

5.1 Verzögerungen

Durch die Einbindung einer Netzwerkarchitektur ist der von einer Applikation geleistete Aufwand wesentlich größer, als bei normalen Einzelnutzer

Systemen. Die Programme müssen folgende Dienste ansprechen:

- Grafische Oberfläche
- Betriebssystem
- Netzwerk – API

Da diese Schichten auch andersherum durchlaufen werden müssen, entsteht wie schon gesagt ein beträchtlicher Mehraufwand. Dies kann zu Verzögerungen führen, welche die Interaktion des Benutzers mit seinem System beeinträchtigen würden. Beispielsweise könnten bei der Bild- oder Textbearbeitung Änderungen an dem Dokument langsamer auf dem Bildschirm als Feedback erscheinen, als der Nutzer es in einem optimalen System gewöhnt ist. Solche Dinge werden als äußerst störend aufgefasst, was im kommerziellen Markt beachtliche Nachteile haben würde.

Natürlich können solche Störungen nie wirklich ausgeschlossen werden, doch es sollte wenigstens eins der Ziele des Designers sein, diese wenn möglich gering zu halten. Ein Blick auf oft verwendete Netzwerkarchitekturen kann daher nicht schaden:

5.1.1 Client-Server Architektur

Diese klassische Netzwerkarchitektur baut darauf auf, dass die Nutzer des Systems sich mit leichtgewichtigen Programmen, den so genannten "Clients", per Anfragen an einen Server wenden. Dieses mächtigere Programm dient als verwaltende Stelle der Netzwerk-Applikation. Er kennt alle angemeldeten Clients und bearbeitet ihre Anfragen. So kann ein jeder Client eine Aktion durchführen, welche vom Server als Anfrage aufgenommen und ausgeführt wird. Sobald der Server die Aufgabe abgearbeitet hat schickt er die Ergebnisse als Updates an alle Clients. Diese brauchen nur noch den neuen Stand des Systems auf dem Bildschirm des einzelnen Nutzers ausgeben.

Der Vorteil dieser Architektur liegt in der Gewissheit, dass alle Clients im gleichen Maße vom Server mit Updates versorgt werden, weshalb hier Systemstatus untereinander nicht abweicht.

Ihre Problematik liegt viel mehr auf der lokalen Ebene. Weil der Server die gesamte Arbeit ausführt und ihr Ergebnis an alle Clients sendet, kann es durchaus dazu kommen, dass jener Nutzer, welcher die Aktion-Anfrage stellte selbst erst als letztes das Ergebnis auf seinen Bildschirm sehen wird. Damit gerät die Feedback Struktur zwischen aktivem Nutzer und System aus der Balance.

5.1.2 Peer to Peer Architektur

In dieser Form der Netzwerkarchitektur wird auf den Server verzichtet. Die Clients sind weiter ausgebaut als in der Client-Server Architektur, da

sie nun ein jeder von ihnen über die aktuell angemeldeten Clients informiert sein müssen. Da nun kein Client mehr vorhanden ist, um Aktionen durchzuführen und ihre Ergebnisse an die Nutzer zu senden, müssen nun die Clients selbst dafür Sorge tragen. Eine Aktion des Nutzers wird lokal vom eigenen Client ausgeführt und die Veränderung der betroffenen Daten an alle anderen Clients gesandt. Lokal gesehen ist damit das Problem der Verzögerungen im Keim erstickt, da der ausführende Nutzer das Ergebnis sofort sieht, jedoch kann es auch hier wieder zu Kollisionen mehrerer Nutzer kommen. Sobald zwei Nutzer ohne verwaltende Instanz wie einem Server geteilte Daten verändern können, sind solche Probleme vorprogrammiert.

5.2 Netzwerkbeschränkungen

Ein häufiges Problem sind die netzwerktechnischen Verzögerungen die bei der Kommunikation mehrerer Instanzen eines Programms. Wenn bei gleichzeitiger Arbeit mehrerer Benutzer an einem Dokument zu starke Latenzen auftreten kann dies die Arbeit natürlich stark behindern.

Eine Grundlage dieses Problems ist, dass viele Programme über Nachrichten im Klartext kommunizieren. Diese haben den Vorteil einfach zu schreiben und wieder zu empfangen zu sein. Allerdings können sie leicht unnötig viel Platz einnehmen und somit für Datenstau sorgen. Ein Komprimierungsverfahren welches, z.B. Redundanzen in den Nachrichten vermeidet könnte hier Abhilfe schaffen.

Ein System welches genau dieses versucht ist "General Message Compression" (GMC) [3]. Es kann ohne großen Aufwand in Groupware-Applikationen eingebunden werden und so den Datenverkehr beträchtlich reduzieren.

5.3 Häufige Fehlerquellen

Während der Entwicklung eines kommerziellen Produkts sollten frühzeitig und regelmäßig Test-Phasen durchgeführt werden. Wenn Fehler früh erkannt werden, können wichtige Teile des Systems umgeändert werden. Der anfangs zusätzliche Invest führt am Ende zu Gewinnen:

Das Produkt wird fehlerfreier ausgeliefert, was die Akzeptanz der Nutzer steigert und die Zahl unnötiger Updates minimiert.

Des weiteren können am Ende erkannte Probleme starke Nachwirkungen haben, da möglicherweise große Einheiten des Produkts verändert werden müssen.

In synchronen Groupware Systemen sind folgende Dinge häufig der Ursprung von Fehlern:

- Äußere Fehler wie Probleme des

Betriebssystems

- das System hat programmiertechnische Fehler
- unerwartete Ereignisse wie "Deadlocks" oder veränderte Anzahl von Nutzern

5.4 Äußere Fehlerquellen

Manchmal liegt die Fehlerquelle nicht bei dem selbst erschaffenen Produkt, sondern viel mehr außerhalb. So können Fehler im Netzwerk, im Computer oder wie oben schon angedeutet im Betriebssystem liegen.

Solche Fehler können durch den Wechsel einer Komponente in Zielsystem entstehen. Als Beispiel nehme man die Architektur der weit verbreiteten Personal-Computer. Im Gegensatz zu anderen Computer-Systemen liegt hier ein offener Standard vor, für den verschiedene Firmen ihre Produkte schaffen können. Die große Anzahl verschiedener Hardware macht es für viele Entwickler schier unmöglich genaue Angaben über das vom Nutzer verwendete System zu machen.

5.5 Programmiertechnische Fehler

Die wohl schwierigste Fehlerquellen sind algorithmischer Natur. So kann es passieren, dass ein Algorithmus ohne Wissen des Entwicklers Ergebnisse liefert, die minimal falsch sind, jedoch auf lange Sicht den Fehler vergrößern.

Beispielsweise stellen wir uns eine Rechnung vor, welche höchste Präzision erwartet. Sie wird 3000 mal benutzt und addiert jeden Durchlauf 0,001 zu viel auf das Ergebnis. Während es bei Tests mit kleiner Durchlaufzahl noch alles im Rahmen bleibt, wird in unserem Szenario schon eine Abweichung von 30 erschaffen.

Solche Fehler können verhindert werden, indem man den Algorithmus vorher mathematisch verifiziert und Fehlerabfragen im Quellcode einbettet.

5.6 Unerwartete Ereignisse

Viele Fehler können durch eine unerwartete Folge von Benutzersritten entstehen. Nicht umsonst, werden in anderen Industrie-Sparten wie bei den Entwicklern von Video-Spielen ganze Abteilungen damit beauftragt eigentlich unsinnige Dinge zu machen wie "3 mal im Kreis drehen und an die Wand springen". Auch wenn man denken könnte, dass der Nutzer so etwas nie machen würde, zeigt die Realität wie häufig Murphys Gesetz eintritt: Irgendein Nutzer wird es genauso durchführen.

Es sollte sich also auch in der Entwicklung anderer Software durchsetzen, solche Tests durchzuführen.

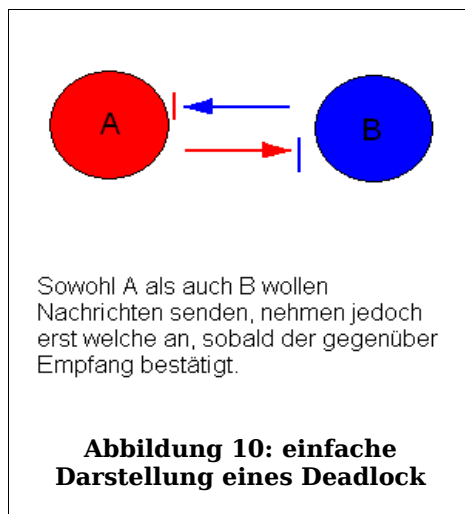
Um einen solchen Fehler im Bereich synchroner Groupware Systeme vorzustellen, möchten wir hier beispielhaft den "Deadlock" beschreiben.

Genauso wie im Bereich der parallelen Programmierung gibt es auch in synchronen Netzwerksystemen den äußerst kritischen Punkt namens "Deadlock".

Dieser tritt im auf, wenn zwei Teilnehmer auf eine Bestätigung des anderen warten. Wie im Bild schematisch dargestellt, wollen die Clients A und B einander Daten zusenden. Sie selbst akzeptieren erst dann eingehende Nachrichten, wenn ihr Versandziel die Ankunft der Daten bestätigt.

Somit wird keiner der beiden die Arbeit weiter aufnehmen und das System bleibt in diesem Zustand.

Wie schon zu Beginn geschrieben, lassen sich solche Fehler meistens nur durch Testen auffinden. Im Gegensatz zu algorithmischen oder äußeren Fehlern liegt hier die Problematik der Kombinationsmöglichkeit vor. Die vom Nutzer gemachten Schritte müssen ganz genau nachgestellt werden, da viele Faktoren mit einwirken. Es ist also nicht ratsam das System nur aufgrund von Simulationen als stabil einzustufen.



3. Support über lange Zeitspannen

Entwickler müssen von vorne herein sicherstellen, dass die geplante Software alle möglichen Nutzer zufrieden stellen kann. Gruppen haben eine eigene Dynamik und erfordern gründliche Analysen.

Auf der Basis heutiger Systeme lässt sich sehr gut aufbauen und Erfahrungen gewinnen.

6.FAZIT

Abschließend betrachtet kann man sagen, dass Groupware-Systeme in naher Zukunft immer mehr an Bedeutung gewinnen werden. Viele Systeme, welche anfänglich im Bereich der Universitäten genutzt wurden, schaffen langsam aber sicher den Einzug in die Privat-Haushalte. Dieser Sprung ändert die Anforderungen an Entwickler erheblich. Stabilität und Wartung rücken somit in den Vordergrund.

Groß-Unternehmen mit vielen Mitarbeitern und private Anwender fordern neben Funktionalitäten viel mehr von Groupware-Systemen:

1. einfache Handhabung
2. Sicherheit

7. LITERATUR

[1] C-Authoring with Structured Annotations.

Proceedings of the SIGCHI conference on Human Factors in computing systems CHI '06

[2] SaveMe: A System for Archiving Electronic Documents Using Messaging Groupware.

SIGSOFT Software Engineering Notes, Proceedings of the international joint conference on Work activities coordination and collaboration WACC '99

[3] Improving Network Efficiency in Real-Time Groupware with General Message Compression.

Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work CSCW '06

[4] Human Computer Interaction

Third Edition published by Pearson Education Limited 2004

8. BILDER QUELLEN

Abbildung 1: Figure 19.1 aus [4]

Abbildung 2: Figure 19.2 aus [4]

Abbildung 4: Figure 19.5 aus [4]

Abbildung 5: <http://www.isrc.umbc.edu/HCIHandbook/figures/29-02.jpg>

Abbildung 6: http://www.creative-products.de/presentationssysteme/gfx/ebeam_s3_blue.jpg

Abbildung 7: http://www.nectar-research.net/news/wp-content/ZhengCoauthoring_01.png

Abbildung 8: Figure 19.9 aus [4]

Abbildung 9: Figure 19.10 aus [4]