

Computergestützte Gruppenarbeit

13. Ausgewählte Groupware

Dr. Jürgen Vogel

*European Media Laboratory (EML)
Heidelberg*

SS 2006

Inhalt der Vorlesung

1. Einführung
2. Grundlagen von CSCW
3. Gruppenprozesse
4. Benutzerschnittstelle
5. Zugriffsrechte und Sitzungskontrolle
6. Architektur
7. Konsistenz
8. Undo von Operationen
9. Visualisierung semantischer Konflikte
10. Late-Join
11. Netzwerk-Protokolle
12. Entwicklung von Groupware
- 13. Ausgewählte Groupware**

Inhalt

- Asynchrone Kommunikationssysteme
 - Email
 - BSCW
- Synchrone Kommunikationssysteme
 - computergestützte Videokonferenzen
 - Instant Messaging
- Gemeinsame Arbeitsbereiche
 - TeCo3D
 - mlb
 - Spiele
- Synchrone und asynchrone Interaktion
 - Activity Explorer
- Software-Entwicklung
 - CVS
 - Jazz
- Gemeinschaften ("Online Communities")
 - Wiki

Kommunikation

Kommunikation = Informationsaustausch

- zeitlicher Ablauf: synchron vs. asynchron
- Medium: verbal vs. nonverbal (Gesten, Blickkontakt, Stimmlage)
- Art: direkt vs. indirekt
- Assoziierung: 1:1, 1:N, N:1, N:M
- Informationsfluss: unidirektional vs. bidirektional
- Phasen: Begrüßung, Konversation, Verabschiedung
- Strukturierung der Konversation durch explizite oder implizite Sprecher-Reihenfolge ("Turn-taking")
- Ritualisierung (abhängig vom Zweck): formell vs. informell
- verwobene Konversationsstränge ("Threads")

Computergestützte Kommunikationssysteme

- Abbildung der sozialen Mechanismen bei Face-to-Face-Kommunikation, z.B. Emoticons für non-verbale Kommunikation per IM
- Technik hat starken Einfluss auf die Kommunikation, z.B. SMS

Email (1)

Beschreibung und Anwendungsbereiche

- flexible Kommunikations-Anwendung für asynchrone Textnachrichten mit optionalen Attachements

Klassifikation und technische 3K-Protokolle

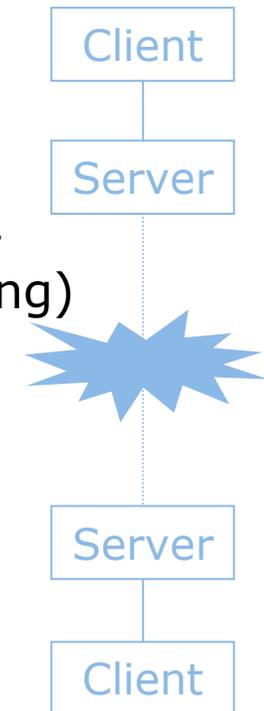
- Groupware: asynchron
- Kommunikation: verbal, direkt oder indirekt, 1:1 oder 1:N, unidirektional oder bidirektional
- Koordination und Kooperation nur über soziale Protokolle, z.B. "Netiquette" in RFC 1855:
 - keine vorschnelle Beleidigung
 - keine Veränderung des Inhalts beim Weiterleiten
 - kein Spam
 - schnelle Antwort
 - Lesen des Archivs vor dem Senden von Fragen an Mailingliste
 - ...



Email (2)

Daten, Datenmodell und Architektur

- diskretes Email-Objekt wird als State versendet
- Objektstruktur (RFC 2822): Sender, Empfänger (To, CC, BCC), Zeit, Betreff (Antwort auf), Priorität, Empfangsbestätigung, Nachricht, Attachment
- Nachricht selbst ist unstrukturiert
- keine Zustandsänderungen bei erzeugten Objekten
- aktive Objekte: im Email-Client betrachtet
- Hierarchie durch Threads
- verteilte Architektur mit replizierter Datenhaltung (Sender- und Empfänger-Server, Nachrichten ohne Server-Verbindung)
- Email-Clients (auch "Message User Agent MUA")
 - Erstellung, Anzeige und Verwaltung, automatische Aktionen (z.B. Junk Filter), Awareness, lokaler Speicher
 - z.B. MS Outlook, Lotus Notes, Thunderbird, Pine, ...
- Email-Server (auch "Message Transfer Agent MTA")
 - Übertragung ("Store and Forward"-Prinzip), Speicher
 - z.B. Sendmail, MS Exchange Server, Lotus Domino, ...



Email (3)

WYSIWIS und Awareness

- relaxiertes WYSIWIS mit individueller Ansicht
- Awareness: Eingang neuer Nachricht, Empfangsbestätigung

Floor und Session Control

- Floor Control nicht erforderlich
- eng gekoppelte Sitzungskontrolle
 - geschlossene Gruppe mit expliziter Mitgliedschaft
 - definiert durch den Sender (explizite Adressierung) oder den Empfänger (Mitgliedschaft in Mailing-Liste)
- Rollen: Sender (einmaliges Schreibrecht) und Empfänger (Leserecht)

Konsistenzerhaltung und semantische Konflikte

- Konsistenz durch zuverlässige Übertragung
- keine Konflikte
- (Konfliktauflösung für IMAP)

Email (4)

Undo von Operationen

- kein Undo

Late-Join

- manueller LJ: Weiterleiten des Email-Objekts durch ein Mitglied an den LJ-Client → Initialisierung per State

Netzwerk-Protokolle (Internet-basiert)

- SMTP ("Simple Mail Transfer Protocol", RFC 2821)
 - Transfer vom Client zum Server und von Server zu Server
 - ASCII-Kodierung
 - TCP Port 25
- POP3 ("Post Office Protocol", RFC 1939)
 - Transfer vom Server zum Client
 - "Abhol"-Metapher mit temporärer Verbindung
 - ASCII-Kodierung
 - TCP Port 110

Email (5)

- IMAP ("Internet Message Access Protocol", RFC 3501)
 - Mailzugriff vom Client auf den Server
 - dauerhafte oder temporäre Verbindung
 - Mail bleibt meist auf dem Server
 - multiple Clients mit nebenläufigem Zugriff
 - zusätzliche Flags auf dem Server (nicht Teil des Zustands!)
- MIME ("Multipurpose Internet Mail Extensions", RFC 2045-2049)
 - Kodierung multimedialer Inhalte als ASCII für SMTP

Analyse, Herausforderungen und Verbesserungen: Übung 1 und 2

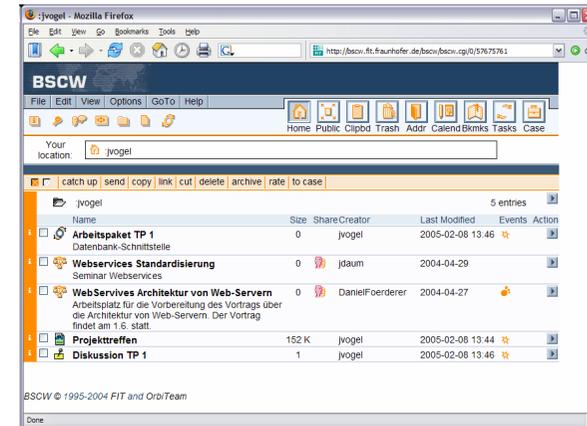
Inhalt

- Asynchrone Kommunikationssysteme
 - Email
 - BSCW
- Synchrone Kommunikationssysteme
 - computergestützte Videokonferenzen
 - Instant Messaging
- Gemeinsame Arbeitsbereiche
 - TeCo3D
 - mlb
 - Spiele
- Synchrone und asynchrone Interaktion
 - Activity Explorer
- Software-Entwicklung
 - CVS
 - Jazz
- Gemeinschaften ("Online Communities")
 - Wiki

BSCW (1)

Beschreibung und Anwendungsbereiche

- Plattform zum asynchronen Austausch von Dokumenten und zur Koordination ihrer Bearbeitung
- unterstützende Diskussion, Aufgaben- und Terminverwaltung
- Zugriff per Web-Client
- entwickelt von GMD/Fraunhofer; kommerziell



Klassifikation und technische 3K-Protokolle

- Groupware: asynchron
- Kommunikation: verbal; direkt oder indirekt; 1:1, 1:N, N:1 oder N:M; unidirektional oder bidirektional
- Koordination durch Setzen von Zugriffsrechten und Diskussion
- Kooperation durch Aufgaben- und Terminverwaltung, Bildung von Subgruppen

BSCW (2)

Daten, Datenmodell und Architektur

- diskrete Objekte mit hierarchischer Anordnung
- C/S-Architektur

WYSIWIS und Awareness

- relaxiertes WYSIWIS (auf den Benutzer angepasste Icons)
- Awareness-Informationen über Änderungen (wer, wann und was) per Icon seit dem letzten Einloggen oder zeitnahe Nachrichten
- individuelle Meldungen und Ereignisfilter
- keine Benachrichtigung über zufälligen synchronen Zugriff

Floor und Session Control

- geschlossene/offene Gruppe mit eng gekoppelter Verwaltung
- Teilnehmer-Rollen: Administrator, Benutzer, Gast, ...
- feingranulare Definition der Zugriffsrechte

BSCW (3)

Konsistenzerhaltung und semantische Konflikte

- Serialisierung der Schreibzugriffe und Versionsverwaltung
- keine Überprüfung auf semantische Konflikte

Inhalt

- Asynchrone Kommunikationssysteme
 - Email
 - BSCW
- Synchroner Kommunikationssysteme
 - computergestützte Videokonferenzen
 - Instant Messaging
- Gemeinsame Arbeitsbereiche
 - TeCo3D
 - mlb
 - Spiele
- Synchroner und asynchroner Interaktion
 - Activity Explorer
- Software-Entwicklung
 - CVS
 - Jazz
- Gemeinschaften ("Online Communities")
 - Wiki

Computergestützte Videokonferenzen (1)

- synchrone Kommunikation mit mehreren Medien: Audio, Video, Shared Whiteboard, Chat, ...
- ersetzt oder ergänzt Face-to-Face-Meetings
- Kommunikation: verbal und nonverbal; direkt; 1:1 oder N:M; bidirektional
- Video ist ein effektives Mittel zur Übertragung von Awareness-Informationen via Gestik und Mimik (trotz begrenzter Auflösung)
- meist je ein Kanal pro Medium und Gegenstelle
 - mehrere Videokanäle verbessern die Wahrnehmung, z.B. gleichzeitige Sprecher und Raum-Ansicht
 - mehrere Audiokanäle ermöglichen Raumklang oder einfachen Sprecherwechsel
- wichtige Synchronisation zwischen Audio und Video
- vielfältige Formen bzgl. Ausstattung, Gruppengröße, Sitzungskontrolle, ...
- permanente Videokonferenz für spontane Interaktion ("Media Space")

Computergestützte Videokonferenzen (2)

- Gruppenkommunikation
 - Konferenzbrücke bei ITU-Protokollen ("Multipoint Control Unit MCU")
 - IP oder Application-Level Multicast bei IP-Protokollen
- Videokonferenzsysteme
 - kommerziell: Polycom, Tandberg, CISCO MCU, ...
 - frei: MS NetMeeting, SunForum, MBone-Software vic/vat/rat, ...

Blickkontakt (1)

Problemstellung

- Blickkontakt ist eine wichtige Geste während der Kommunikation, z.B. zur Bestimmung des nächsten Sprechers ("Turn-taking")
- die menschliche Wahrnehmung ist empfindlich bzgl. der Blickrichtung mit einer Toleranz von wenigen Grad (1° - 5°)
- platziert man die Kamera neben dem Monitor, sieht man die Blickrichtung entfernter Benutzer daher verschoben, d.h. direkter Blickkontakt ist nicht möglich (Parallaxe)
- Abhilfemöglichkeiten
 - Erkennung der Blickrichtung und Korrektur der Augenstellung per Bildverarbeitung
 - Erkennung der Blickrichtung und Benachrichtigung/Markierung des Angesehenen ("Gaze Awareness")
 - Platzierung der Kamera hinter einem halbtransparenten Bildschirm

Blickkontakt (2)

Das Gaze-2-System

- halbtransparenter Bildschirm, dahinter 3 Kameras (eine aktiv) und ein Eye Tracker
- automatische Auswahl der Kamera mit der geringsten Parallaxe
- Darstellung der Gesprächspartner durch gekippte 2D-Videos in virtuellem 3D-Konferenzraumraum
- die Videos sind so gekippt, dass sie die momentane Blickrichtung wiedergeben
- Veränderung der Blickrichtung kann zu einem Wechsel der aktiven Kamera führen und passt die Darstellungen an
- Entfernung des Bildhintergrunds zur Ablenkungsvermeidung bei Kamerawechsel
- Verwendung der Blickrichtung zur Übertragungsoptimierung: nur der Audio-/Videostrom des momentan angeschauten Teilnehmers muss in hoher Qualität übertragen/empfangen/dargestellt werden
- Benutzerstudie mit 10 Teilnehmern: keine störende Ablenkung beim Wechsel der aktiven Kamera



Inhalt

- Asynchrone Kommunikationssysteme
 - Email
 - BSCW
- Synchroner Kommunikationssysteme
 - computergestützte Videokonferenzen
 - Instant Messaging
- Gemeinsame Arbeitsbereiche
 - TeCo3D
 - mlb
 - Spiele
- Synchroner und asynchroner Interaktion
 - Activity Explorer
- Software-Entwicklung
 - CVS
 - Jazz
- Gemeinschaften ("Online Communities")
 - Wiki

Instant Messaging (1)

Beschreibung und Anwendungsbereiche

- synchrone textbasierte Kommunikation
- Zusatzfunktionen: Gesten, Konferenzen mit mehreren Teilnehmern, asynchrone Nachrichten, Dateiaustausch, Audio-Konferenzen



Klassifikation und technische 3K-Protokolle

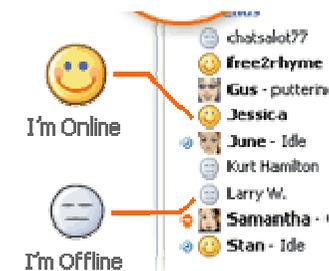
- reine Kommunikationsanwendung ohne technische Protokolle
- verbal und nonverbal; direkt; 1:1 oder N:M; bidirektional

Daten, Datenmodell und Architektur

- Text: diskret, unveränderlich
- meist C/S, aber auch P2P oder hybrid

WYSIWIS und Awareness

- relaxiertes WYSIWIS
- explizite und implizite Awareness-Information über den Benutzerzustand (anwesend, ansprechbar)



Instant Messaging (2)

Konsistenzerhaltung und semantische Konflikte

- meist keine Konsistenzerhaltung, daher Inkonsistenzen in der Form abweichender Zeilenreihenfolge möglich
- semantische Konflikte können nicht auftreten

Evaluation

- IM wird in vielen Unternehmen sehr erfolgreich eingesetzt, da:
 - geringer Ressourcenverbrauch (vgl. Videokonferenz)
 - einfache Verwaltung
 - auf vielen Systemen verfügbar
 - automatische Historie
 - gut für kurze, informelle und nebenläufige Kommunikation
 - geringer Störfaktor (kontaktiere Person und Unbeteiligte)
- Feldstudie mit 437 Benutzern und 21.000 Gesprächen
- $\bar{\delta}$ 6,2 Buddies und 1,7 Gespräche pro Tag (max. 15,5)
- $\bar{\delta}$ -Gespräch dauert 4min 23s mit 17,2 Nachrichten
- in 85% der Gespräche arbeitet mindestens ein Partner an mehreren unabhängigen Aufgaben ("Multitasking")

Instant Messaging (3)

- fortgeschrittene Benutzer besprechen häufig mehrere Themen ("Threads") parallel, in kurzen abgehakten Nachrichten mit Abkürzungen, Symbolen und Rechtschreibbefehlen und über einen längeren Zeitraum
- sporadische Nutzer formulieren ganze Sätze und diskutieren nur selten mehrere Themen parallel
- Verwendung (Mehrfachnennung möglich)
 - 30,8%: zur Terminabsprache und zur Vereinbarung von Meetings
 - 27,8%: zum schnellen Fragen und Antworten (kurze Dauer, wenige Nachrichten, ein Thema)
 - 61,8%: zur Diskussion von Aufgaben und Problemen (lange Dauer, komplexe Themen, mehrere Threads)
 - 13%: zum informellen Kontakthalten über einen längeren Zeitraum (sporadische Interaktion mit kurzen Statusmeldungen)
 - 6,4%: rein privat
- 23,6% aller initialen Nachrichten (Kontaktaufnahme) bleiben unbeantwortet (im Vergleich: Telefon 62%)

Inhalt

- Asynchrone Kommunikationssysteme
 - Email
 - BSCW
- Synchrone Kommunikationssysteme
 - computergestützte Videokonferenzen
 - Instant Messaging
- Gemeinsame Arbeitsbereiche
 - TeCo3D
 - mlb
 - Spiele
- Synchrone und asynchrone Interaktion
 - Activity Explorer
- Software-Entwicklung
 - CVS
 - Jazz
- Gemeinschaften ("Online Communities")
 - Wiki

Gemeinsame Arbeitsbereiche

- synchrone Erstellung und Manipulation von Artefakten (Texte, Graphiken, Animationen, ...)
- unterstützt durch zusätzliche Kommunikationskanäle (Video, Audio, IM)
- meist kleine Gruppen (< 10 Mitglieder)

TeCo3D (1)

Beschreibung und Anwendungsbereiche

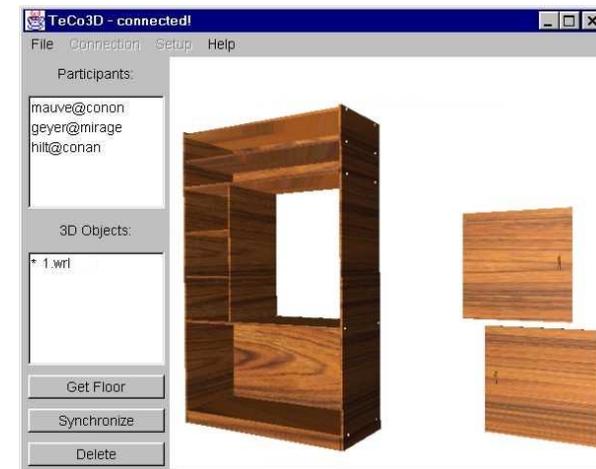
- Virtual Reality-Umgebung
- interaktive 3D-Modelle ohne Avatare
- Einsatz in Videokonferenzen
- entwickelt an der Universität Mannheim

Klassifikation und technische 3K-Protokolle

- synchrone Anwendung für kleine Gruppen
- implizite Kommunikation über Objekte
- Koordination per Floor Control

Daten, Datenmodell und Architektur

- unpartitionierter Anwendungszustand (ein aktives Objekt)
- kontinuierliche Anwendung
- Unterstützung von States und Events
- verteilte Architektur mit replizierter Datenhaltung



TeCo3D (2)

WYSIWIS und Awareness

- gemeinsamer Arbeitsbereich
- relaxiertes WYSIWIS mit individueller Perspektive auf das Modell
- Awareness: anwesende Teilnehmer, Floor Holder

Floor und Session Control

- mutually-exclusive Floors mit expliziter Kontrolle
- leichtgewichtige Sitzungskontrolle (offene/dynamische Gruppen)

Konsistenzerhaltung und semantische Konflikte

- Konfliktvermeidung mit Floor Control
- Local Lag und Timewarp

Undo von Operationen

- kein Undo/Redo

TeCo3D (3)

Late-Join

- sofortige Initialisierung des aktiven Objekts
- LJ-Server = Floor Holder
- Initialisierung per Zustand ohne zusätzliche Konsistenzerhaltung
- 2 Multicast-Gruppen

Netzwerk-Protokolle

- IP-Multicast ← UDP ← RM ← RTP/I ← RTP/I-VRML-Payload
- ALF/ILP mit State, Event und Query ADUs

Evaluation

- komponentenbasierte Entwicklung mit Rapid Prototyping
- hauptsächlich informelle Evaluierung mit Befragung von Experten
- technische Evaluierung

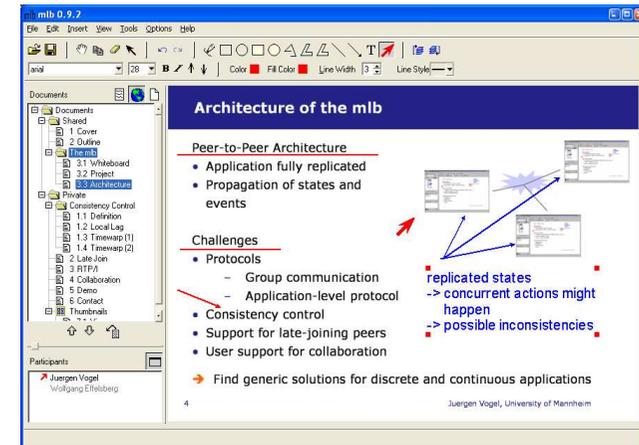
Inhalt

- Asynchrone Kommunikationssysteme
 - Email
 - BSCW
- Synchrone Kommunikationssysteme
 - computergestützte Videokonferenzen
 - Instant Messaging
- **Gemeinsame Arbeitsbereiche**
 - TeCo3D
 - **mlb**
 - Spiele
- Synchrone und asynchrone Interaktion
 - Activity Explorer
- Software-Entwicklung
 - CVS
 - Jazz
- Gemeinschaften ("Online Communities")
 - Wiki

mlb (1)

Beschreibung und Anwendungsbereiche

- synchrones Präsentieren und Editieren von Dokumenten und Folien
- Einsatz meist in Videokonferenzen kleiner Gruppen
- entwickelt an der Universität Mannheim



Klassifikation und technische 3K-Protokolle

- implizite Kommunikation über Objekte und explizite per Chat
- Koordination über elektronisches Melden
- Kooperation über elektronisches Abstimmen

Daten, Datenmodell und Architektur

- hierarchisch partitionierter Anwendungszustand mit Dokumenten, Ordnern, Seiten, Text, graphischen Objekten und Bildern
- diskrete Objekte
 - aktiv: alle Objekte der aktuellen Seite (Shared Workspace)
 - passiv: Rest

mlb (2)

- Unterstützung von States, Events und Cues
- verteilte Architektur mit replizierter Datenhaltung

WYSIWIS und Awareness

- gemeinsamer und private Arbeitsbereiche
- relaxiertes WYSIWIS mit personalisierter Ansicht und privaten Annotationen
- Awareness über anwesende Teilnehmer und deren Aktionen: Teilnehmerliste mit Hervorheben (Aktivität, Konflikt, Telepointer, Austritt) und Objektmarkierung
- multiple Telepointer zum Zeigen und Markieren

Floor und Session Control

- kein Floor Control
- leichtgewichtige Sitzungskontrolle (offene/dynamische Gruppen)
- alle Teilnehmer sind gleichberechtigt (→ gleiche Rolle)
- Bildung von Sub-Gruppen

mlb (3)

Konsistenzerhaltung und semantische Konflikte

- Verwaltung einer limitierten ADU-Historie mit zeitlicher Ordnung
- Kausalität per Zustandsvektor
- Local Lag (Default: 100ms), Timewarp (rundenbasiert und gefiltert) und Zustandsanfragen (iterative Anfrage)
- Visualisierung semantischer Konflikte: interaktive Operations-Historie, Markierung von Teilnehmern und Objekten

Undo von Operationen

- relaxiertes Undo: lokal, linear und semantisch
- separate Undo- und Redo-Historien

Late-Join

- Late-Join mit Politiken: aktuelle Seite sofort, sonst bei Ereignis
- 3 Multicast-Gruppen
- Initialisierung mit Sequenz aus Zustand und Operationen
- Konsistenz durch iterative Anfragen

mlb (4)

Netzwerk-Protokolle

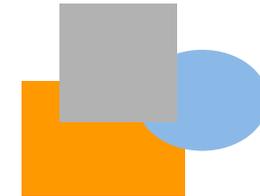
- IP-Multicast ← UDP ← SMP ← RTP/I ← RTP/I-Whiteboard-Payload
- ALF/ILP mit State, Event, Cue und Query ADUs

Evaluation

- komponentenbasierte Entwicklung mit Rapid Prototyping
- hauptsächlich informelle Evaluierung mit Befragung von Experten und Benutzern
- Usability-Tests mit Dozenten und Studenten in Seminar per Fragebogen
- Bedienbarkeit der GUI sowie Auswirkungen von Konsistenzzerhaltung und Late-Join
- antizipierende Evaluierung mit Simulationsstudien
- umfangreiche technische Evaluierung

Anzeige-Reihenfolge (1)

- Anzeige-Reihenfolge bestimmt die Ordnung von Objekten bei der Darstellung:
 - graphische Objekte im Arbeitsbereich
 - Ordner und Seiten
- Definition der Reihenfolge
 - Erzeugung von Objekten: neuere Objekte werden zuletzt gezeichnet
 - Zustandsänderungen durch Verschieben: Vordergrund/Hintergrund, Drag&Drop
- Teil des Objekt-Zustands: Ordnungszahl ord



Algorithmus

- $0 \leq \text{ord}(\text{Objekt}) \leq 2^{32} - 1$
- Container (Seite, Ordner) verwalten Objektliste, die aufsteigend nach ord sortiert ist

Anzeige-Reihenfolge (2)

- Bestimmung von ord
 - Erzeugung
 $\text{ord}(\text{erstes Objekt}) = (2^{32} - 2) / 2$
 - Erzeugung oder in den Vordergrund verschieben
 $\text{ord}(\text{neues letztes Objekt}) = \text{ord}(\text{bisheriges letztes Objekt}) + \Delta$
 - in den Hintergrund verschieben
 $\text{ord}(\text{neues erstes Objekt}) = \text{ord}(\text{bisheriges erstes Objekt}) - \Delta$
 - zwischen zwei Objekte i und j verschieben
 $\text{ord}(\text{Objekt}) = (\text{ord}(i) + \text{ord}(j)) / 2$
- Wahl von Δ : 2^{16}
- ist der Namensbereich aufgebraucht, vergebe neue Ordnungszahlen (im Abstand Δ)

Kollision von Ordnungszahlen

- nebenläufige Erzeugung / Verschiebung von Objekten
- Konsistenzerhaltung stellt korrekte (eindeutige) Operations-Reihenfolge fest
- inkrementiere die Ordnungszahl der späteren Operation

Inhalt

- Asynchrone Kommunikationssysteme
 - Email
 - BSCW
- Synchrone Kommunikationssysteme
 - computergestützte Videokonferenzen
 - Instant Messaging
- **Gemeinsame Arbeitsbereiche**
 - TeCo3D
 - mlb
 - **Spiele**
- Synchrone und asynchrone Interaktion
 - Activity Explorer
- Software-Entwicklung
 - CVS
 - Jazz
- Gemeinschaften ("Online Communities")
 - Wiki

Spiele

Herausforderungen bei Multiplayer-Spielen

- geringe Response und Notification Time für direkte Interaktion
- große Spieleranzahl (insbesondere bei Rollenspielen)
- Konsistenz, Fairness und Absicherung gegen Manipulationen
- Einbeziehen von Kontextinformationen (z.B. Ort)

Architektur (1)

1) Client-Server

- Verwaltung des Spielzustands auf Server/Server-Farm
 - + einfache Konsistenzerhaltung
 - + Manipulationsabsicherung
 - + Geschäftsmodell
 - Server-Überlastung und Konzentration des Netzverkehrs
 - Responsiveness
 - Fairness
- praktisch in allen kommerziellen Spielen (z.B. Ultima Online, Sims, Diabolo, Quake, ...)

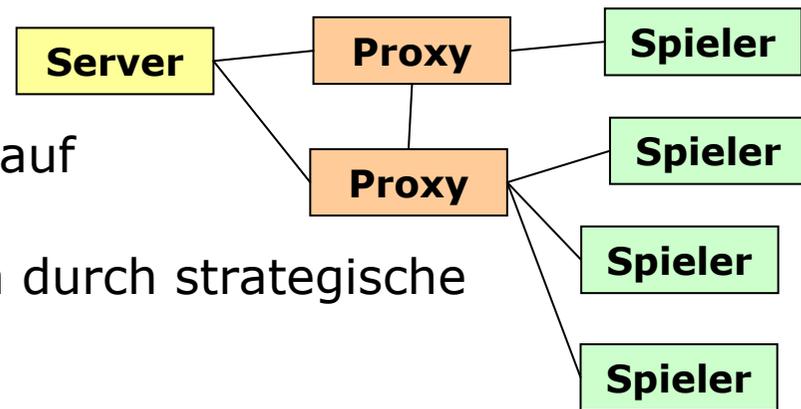
2) Peer-to-Peer

- lokale Verwaltung des replizierten Spielzustands
- Geschäftsmodell und Verwaltung aus Sicht des Spielbetreibers
- akademische und Open Source-Projekte (z.B. MIMAZE, MASSIVE)

Architektur (2)

3) Hybrides System mit Proxies

- Verwaltung des gesamten Zustands auf einem Server und Teile auf Proxies
- + Optimierung der Verzögerungszeiten durch strategische Platzierung der Proxies
- + verbesserte Fairness durch
 - (1) Vergabe der Operations-Zeitstempel durch einen Proxy und
 - (2) Angleichung der Verzögerungszeiten
(Local Lag im Vergleich ist anfällig für Manipulationen)
- + Schutz gegen Netzwerküberlast und -ausfälle durch Proxy-Overlay
- + Entlastung des Servers von bestimmten Aufgaben (z.B. Manipulationskontrolle)
- zusätzliche Kosten durch Proxies
- erhöhte Komplexität bei der Zustandsverwaltung



Untersuchung von Netzwerkeffekten (1)

Auswirkungen von Netzwerkeigenschaften

- Netzwerkverzögerung erhöht Notification Time (alle Architekturen) und Response Time (C/S und hybrid)
- Paketverluste führen u.U. zu temporären Inkonsistenzen
- ➔ anfällig sind insbesondere Spiele mit schnellem Ablauf, z.B. First Person Shooter (FPS)

Untersuchung von Netzwerkeffekten (2)



Untersuchung von Netzwerkeffekten (3)

Studie zur Untersuchung der Auswirkungen

- Unreal Tournament 2003: ca. 1700 Server und 4400 Spieler zu jedem Zeitpunkt
- Spielmodi: Deathmatch, Team Deathmatch, Capture the Flag
- Spieleraktionen: bewegen (laufen, ducken, ...) und schießen
- alle untersuchten Paketverlustraten bis 6% haben keinen signifikanten Einfluss
- Netzverzögerung bis zu 100ms verändern die Schussgenauigkeit kaum, bei 300ms ist die Genauigkeit dagegen schon um 50% schlechter → abnehmende Präzision
- mit zunehmender Netzverzögerung verringern sich die von den Spielern erzielten Punkte leicht

Untersuchung von Netzwerkeffekten (4)

Interpretationsmöglichkeiten

- FPS kombinieren verschiedene Techniken zur Kompensation von Verzögerung und Verlust: Dead Reckoning, lokale Antizipation, ...
- menschliche Spieler können Effekte (bis zu einem gewissen Grad) kompensieren: Waffenwahl, Taktik, ...

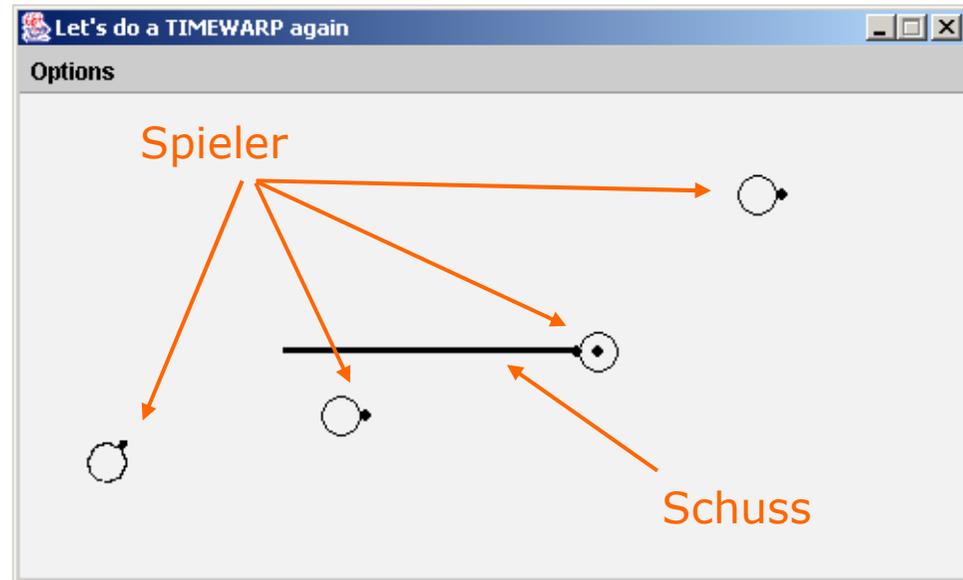
Schlussfolgerungen

- Spiele können auch in nicht-optimalen Netzwerken erfolgreich realisiert werden → mobile Netzwerke
- Präferenz für niedrige Verzögerung gegenüber niedriger Verlustrate
- ggf. vorhandene "Qualitäts-Überschüsse" können für Systemzwecke verwendet werden, z.B. künstliche Verzögerung durch Local Lag

Spaceshooter (1)

Beschreibung

- jeder Benutzer kontrolliert Raumschiff mit begrenzter Lebensenergie
- kleine Gruppen
- Forschungsprototyp der Universität Mannheim zur Erforschung von Konsistenz und Application-Level Multicast



Klassifikation und technische 3K-Protokolle

- synchrone Anwendung mit indirekter Kommunikation
- keine Koordination/Kooperation

Spaceshooter (2)

Daten, Datenmodell und Architektur

- P2P mit replizierter Datenhaltung
- Modellierung der Raumschiffe als Objekte mit kontinuierlicher Zustandsänderung
- Unterstützung von States und Events

WYSIWIS und Awareness

- striktes WYSIWIS

Floor und Session Control

- keine

Konsistenzerhaltung und semantische Konflikte

- (rundenbasierter) Timewarp und Local Lag
- semantische Konflikte können nicht auftreten

Spaceshooter (3)

Netzwerk-Protokolle

- IP-Multicast ← UDP ← RTP/I ← RTP/I-Spaceshooter-Payload
- ALF/ILP mit State und Event ADUs

Evaluation

- Untersuchung der Auswirkungen von Local Lag und Timewarp
- Usability-Test (Beobachtung, Fragebogen) mit 19 Teilnehmern
- Local Lag wurde im Schnitt erst ab 160 ms erkannt
- ohne Local Lag entstehen ab 140 ms Netzverzögerung störende Artefakte durch den Timewarp-Algorithmus
- mit Local Lag von 150 ms entsteht ab 290 ms Netzverzögerung störendes Spielverhalten

Ortsbewusste Spiele ("Location-Awareness")

Idee: mobile Multiplayer-Spiele, die den momentanen Aufenthaltsort der Spieler einbeziehen → Augmented Reality

- technisches Umfeld
 - Endgerät: Laptop, PDA, Handy, spezialisierte Hardware
 - Mobilnetz: GPRS/UMTS, WLAN, Bluetooth, Infrarot
 - Ortsbestimmung: GPS, Netzwerk (Funkzelle, WLAN Access Point), Triangulation (z.B. zu Ultraschall-Sendern), markierte Orte (RFID), Bilderkennung
 - Benutzerschnittstelle: (überblendete) Grafik, Eingabe am Endgerät, Gesten, Interaktion mit physischen Objekten, Sprachein- und -ausgabe, Audio-Hinweise
- ortsbezogene Awareness: Position der Spieler und relevanter Objekte
- Beispiele: Human Pacman, Capture the Flag, Real Tournament, Can You See Me Now?

Capture the Flag (1)

Spiel-Idee

- 2 Teams spielen gegeneinander mit dem Ziel, die gegnerische Flagge von deren Basis zur eigenen Basis zu transportieren (bzw. das gegnerische Team daran zu hindern)
- alle Objekte sind virtuell (Flaggen, Minen, Waffen)
- Aufnahme einer Flagge durch Vorbeilaufen
- dynamisches Spiel: geringe Verzögerung und genaue Lokalisierung (hier über Ultraschallsensoren (Indoor))

Umsetzungen

- 1) einfaches Endgerät: Eingabe mit 2 Buttons, Sound-Ausgabe (verschiedene Samples für Objekt aufgenommen, getroffen, ...)
- ➔ Endgerät ist zu einfach für flüssigen Spielablauf: Sound-Events sind leicht zu überhören, hohe Merkleistung für Positionen und vergangene Ereignisse

Capture the Flag (2)

- 2) zusätzlich fest installierte Monitore an den Basen und global gültige Sound-Ausgabe
 - ➔ Orientierung deutlich Erleichtert, Steuerung des Teams durch einen Spieler nahe der Basis (Monitor)

- 3) Handy als Endgerät mit Visualisierung der Karte und der Spieler- und Objektpositionen
 - ➔ genaues individuelles Wissen schränkt soziale Interaktion ein

Human Pacman (1)

Spiel-Idee

- Augmented Reality: Teams aus Pacmen und Ghosts
- Interaktion mit der physischen und der virtuellen Welt:
 - physisches Spielfeld mit virtuellen Cookies: Ziel der Pacmen ist das Aufsammeln aller Cookies
 - physische Objekte auf dem Spielfeld ("Magic Cookie"): ermöglichen zeitlich limitierte Attacke auf Ghosts
 - physische Berührungen zwischen Pacmen und Ghosts: Anfassen des Berührungssensors = Gefangen
 - Partizipation entfernter Spieler per Virtual Reality ("Helper"): Richtungsanweisungen an Pacman/Ghost

Human Pacman (2)



Human Pacman (3)



Human Pacman (4)

Technik

- C/S-Architektur
- Position per GPS und Schrittzähler (→ Dead Reckoning)
- Soft State mit periodischem Update von Spielerposition und -status
- TCP über WLAN (tragbare Clients, Server) und Bluetooth (physische Objekte)

Evaluation

- Usability-Test (Multiple Choice-Fragebogen) mit 23 Teilnehmern
- 87% finden das Durchlaufen von Cookies als Metapher zum Aufsammeln sinnvoll
- 78% empfinden die Interaktion mit physischen Objekten als Bereicherung
- 96% Zustimmung zum Fangen von Spielern per Berührung
- 65% finden die Zusammenarbeit mit Helfern gut
- 96% wären bereit, eine geringe Gebühr zu entrichten

Soziale Interaktion in Rollenspielen (1)

Massively Multiplayer Online Role-Playing Game (MMORPG)

- Spielziel: Charakterentwicklung durch Gewinnen von Kämpfen, Lösen von Rätseln, ...
- soziale Interaktion zwischen Spielern ist ein zentrales Element in vielen MMORPG's (SIMS, EverQuest, Ultima Online, etc.) und der entscheidende Unterschied zu Einbenutzerspielen
- Spielelemente mit hoher Schwierigkeit erfordern die Zusammenarbeit von Spielern

Star Wars Galaxies

- soziale Interaktion durch räumlich begrenzten (privaten oder öffentlichen) Chat und 340 verschiedene Gesten
- Charakterentwicklung wird durch den Beruf bestimmt
- die 38 verschiedenen Berufe bilden eine komplexe Volkswirtschaft mit gegenseitiger Abhängigkeit: z.B. Kämpfer benötigen Mediziner und Entertainer zur Heilung, beide benötigen Handwerker... → häufige soziale Interaktion unabdingbar
- Automatisierung bestimmter Handlungen durch Makros

Soziale Interaktion in Rollenspielen (2)

- spezielle Gebäude zum Treffen von Interaktionspartnern, z.B. "Cantinas": nur hier können Entertainer heilen, Krankenhäuser und Raumhäfen
- Dienstleistungen (wie Heilung) nehmen eine gewisse Zeit in Anspruch → Zeit für Smalltalk
- ca. 400.000 registrierte Spieler



Evaluation

- Feldstudie durch aktive Partizipation über 3 Monate, Videoaufzeichnung
- passive Charaktere an zwei populären Orten zur Aufzeichnung aller Aktivitäten
- Cantina
 - 3.564 unterschiedliche Spieler in 26 Tagen (meist sporadischer Aufenthalt)
 - ein Drittel der Aktionen sind (überwiegend freundliche) Gesten: häufigste Geste ist "Lächeln" mit 18% Anteil

Soziale Interaktion in Rollenspielen (3)

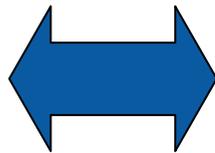
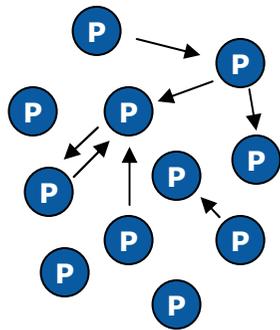
- durchschnittliche Interaktion: je eine Geste und 4 Sätze
→ kurze und flache Interaktionen überwiegen
- viele Spieler verwenden Makros zur (unbeaufsichtigten) Automatisierung der "Interaktion" → oft bemängelt
- sporadisch echte Mensch-zu-Mensch-Interaktion
- Raumhafen
 - 4.668 unterschiedliche Spieler in 26 Tagen
 - Interaktion stärker textorientiert im Vergleich zur Cantina
 - häufigerer Besuch im Vergleich zur Cantina
 - auch hier hoher Anteil automatisierter Spieler (Verkäufer)
- Ergebnisse und Diskussion
 - Abhängigkeiten führen zur häufigen Interaktion
 - spezialisierte Orte fördern Interaktion
 - Interaktion ist meist zielorientiert (Anbieter und Verbraucher) und minimalistisch
 - Makros verhindern echte Interaktion und werden als Spam empfunden
 - Abhilfe: keine Makros oder mehr Punkte für echte Interaktion

Inhalt

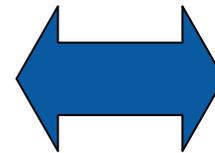
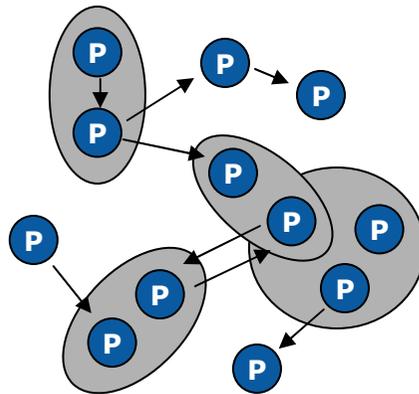
- Asynchrone Kommunikationssysteme
 - Email
 - BSCW
- Synchrone Kommunikationssysteme
 - computergestützte Videokonferenzen
 - Instant Messaging
- Gemeinsame Arbeitsbereiche
 - TeCo3D
 - mlb
 - Spiele
- **Synchrone und asynchrone Interaktion**
 - **Activity Explorer**
- Software-Entwicklung
 - CVS
 - Jazz
- Gemeinschaften ("Online Communities")
 - Wiki

Activity Explorer (1) - Motivation

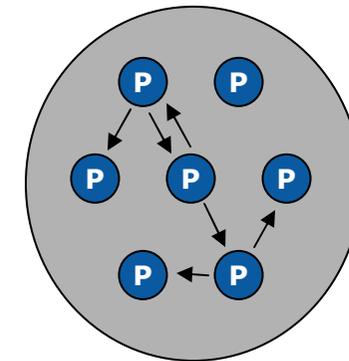
Ad Hoc
Kommunikations-
systeme



Activity Explorer
(Minuten bis Wochen)



Formelle Groupware
(Wochen bis Jahre)



z.B. Email, IM

- spontane Kommunikation
- einfache Handhabung
- unstrukturiert
- Email-Overload

Charakteristika

- kleine Gruppen mit dynamischer Mitgliedschaft
- geringer Datenumfang
- oft nur wenige Vorgänge/Iterationen
- Potential für längere Projekte

z.B. QuickPlace, BSCW

- Handhabung umfangreicher Daten und großer Gruppen
- fehlende Integration
- keine spontane Interaktion
- oft keine Unterstützung für synchrone Szenarien
- aufwendiges Setup und Administration

Activity Explorer (2) - Motivation

Motivation

- zu Beginn eines Gruppenprozesses sind dessen Eigenschaften (Dauer, Daten, Mitglieder, ...) oft noch nicht absehbar
- bei unklarem Nutzen ist die Verwendung formeller Groupware zu aufwendig
- statt dessen Email oder IM
- sollte sich doch eine längerfristige/größere Aktivität entwickeln, ist Email schlecht geeignet und eine Migration zu formeller Groupware schwierig

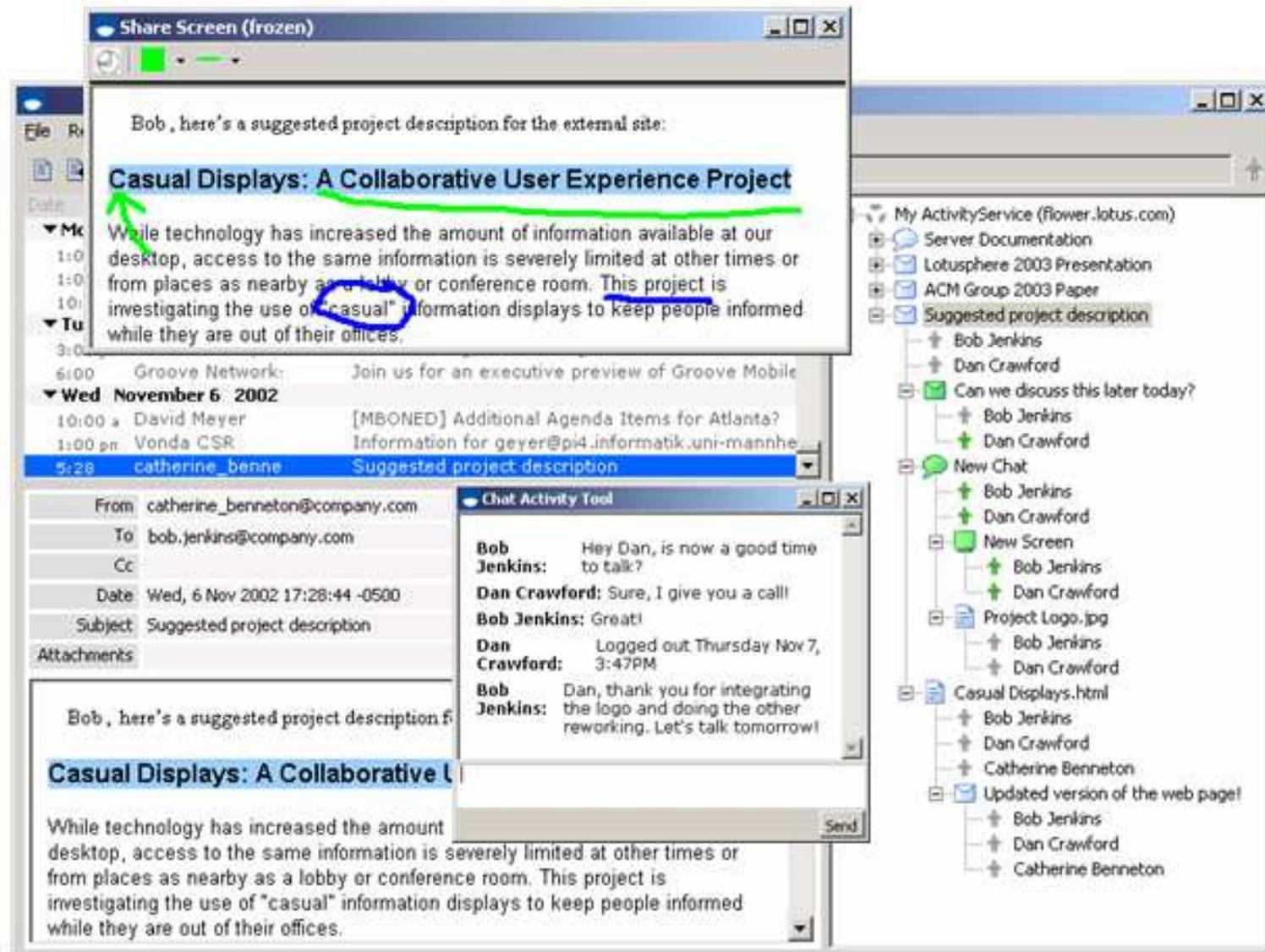
Activity Explorer

- flexible Groupware im Ausgleich von
 - formell vs. informell
 - kleine vs. große Gruppen
 - kurze vs. langfristige Aktivitäten
- entwickelt von IBM Research, erstes Release auf LotusSphere '05

Activity Explorer (3) – Design-Prinzipien

1. keine Trennung zwischen Objekten und Arbeitsbereichen
→ Interaktion auf Objektebene: individuelle Mitglieder, Awareness und Gruppenkommunikation
 2. keine Trennung zwischen synchroner und asynchroner Interaktion mit fließendem und mühelosen Übergang
➔ Objekte sind persistente Sitzungen mit unbegrenzter Dauer
- Objekt-Typen: Nachricht, Chat, Shared Whiteboard, Aufgabe, Kalender, Dokument (Datei), ...
 - Strukturierung mit beliebigen Objekthierarchien (ggf. mit unterschiedlichen Mitgliedern) → Aktivitäten
 - geschlossene Gruppen mit expliziter Einladung (auch nachträglich) und Ausschlussmöglichkeit
 - Integration in Email-Client

Activity Explorer (4) - GUI



Activity Explorer (5)

WYSIWIS und Awareness

- gemeinsame und private Objekte
- relaxiertes WYSIWIS mit individueller Objekthierarchie und unabhängigen aktiven Objekten
- Visualisierung aktiver Benutzer und geänderter Objekte

Daten, Datenmodell und Architektur

- verschiedene (diskrete) Objekt-Typen mit hierarchischer Strukturierung
 - aktiv: geöffnete Objekte
 - passiv: Rest
- Unterstützung von States, Events und Cues
- hier: verteilte Architektur mit replizierter Datenhaltung

Netzwerk-Protokolle

- IP ← TCP ← ALM (sternförmig) ← XML-Anwendungsprotokoll
- ALF/ILP mit State, Event, Cue und Query ADUs

Activity Explorer (6)

Floor und Session Control

- keine Floor Control
- eng gekoppelte Sitzungskontrolle mit hierarchischer Mitgliedschaft und nachträglicher Einladung / Ausschluss
- Kontrolle beim Besitzer des Objekts

Konsistenzerhaltung und semantische Konflikte

- limitierte Operations-Historie mit Zustandsvektor-Ordnung
- Einhaltung kausaler Ordnung (per SV)
- gefilterter Timewarp mit nachgeführten Zuständen
- Ausnahmesituationen mit Zustandsanfragen
- Visualisierung semantischer Konflikte per Operations-Historie

Undo von Operationen

- nicht unterstützt

Activity Explorer (7) – Late Join

Rahmenbedingungen

- häufige Late-Joins
 - zur Initialisierung neuer Objekt-Mitglieder
 - beim Übergang von asynchronen zu synchronen Phasen
 - (1) Offline-Mitglied hat Zustandsänderungen verpasst
 - (2) Offline-Mitglied hat Zustandsänderungen vorgenommen
- LJ-Server nicht immer bzw. nur vorübergehend verfügbar
- divergierende Zustände sehr wahrscheinlich (je nach Szenario)

Design des LJ-Algorithmus

- "Greedy": schneller Austausch / schnelle Konsistenz der LJ-Daten
 - beim Phasenübergang: Replay der Operations-Historie
 - neue Mitglieder: alter Zustand und Operations-Sequenz
 - sofortige LJ-Anfrage für alle Objekte: Vergleich der Zustandsvektoren zur Identifikation von Lücken in den Historien
 - LJ-Anfrage an alle Online-Mitglieder → erhöhte Robustheit und geringere Initialisierungsverzögerung vs. Overhead

Activity Explorer (8) – Late Join

- LJ für neue Objekte: (1) neues Objekt erzeugt oder (2) Einladung für existierendes Objekt
 - erzeugendes oder neues Mitglied könnte offline sein
 - LJ-Client kennt das neue Objekt (oder zusätzlich auch das erzeugende Mitglied!) nicht
 - LJ muss vom erzeugenden Mitglied (LJ-Server) ausgehen
 - Verfahren: zusätzliche periodische Auslieferungsversuche des LJ-Servers
 - Auslieferungsversuche auch durch alle Mitglieder, die zum Zeitpunkt der Erzeugung online waren
- Einfügen empfangener ADUs in die Historie (ggf. Timewarp erforderlich)
- doppelte ADUs werden per SV identifiziert
- Übertragung der LJ-Anfragen und Daten per Unicast (bzw. ALM)

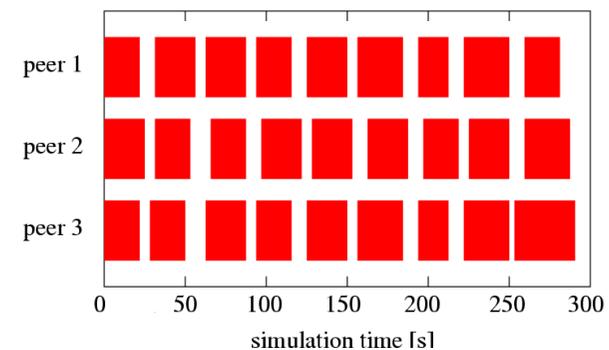
Activity Explorer (9) - Evaluation

- iterative komponentenbasierte Entwicklung mit mehreren Prototypen
- umfangreiche technische Evaluierung
- informelle Evaluierung mit Experten-Befragung

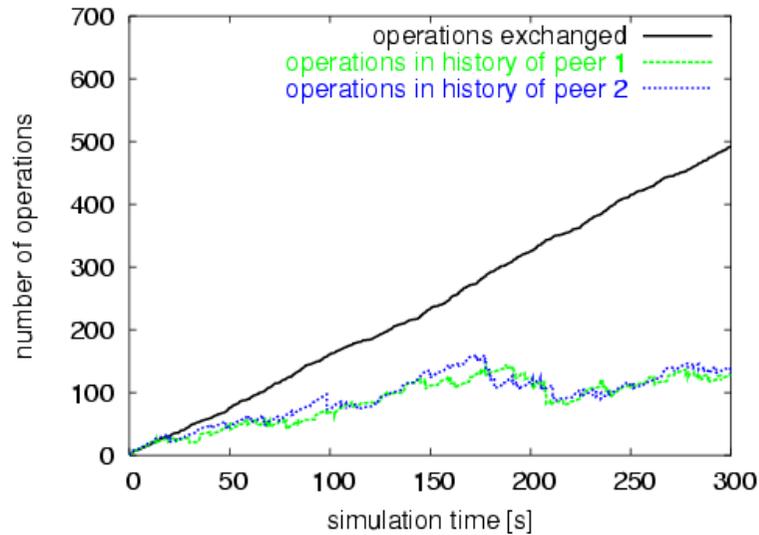
Antizipierende Evaluierung mit Simulationsstudien

- Timewarp und Historien-Verwaltung und LJ-Verfahren
- Szenarien mit 2-3 Mitgliedern über Arbeitswoche (5T): pro Tag 3 neue Objekte, sowie 15 lesende / 10 schreibende Objektzugriffe pro Mitglied
- verschiedene Verteilungen der Online-/Offline-Zeiten: 100%, 80%, 50% und 20 % der Zeit online
- Timewarp-Anzahl: 5, 27, 118 und 44

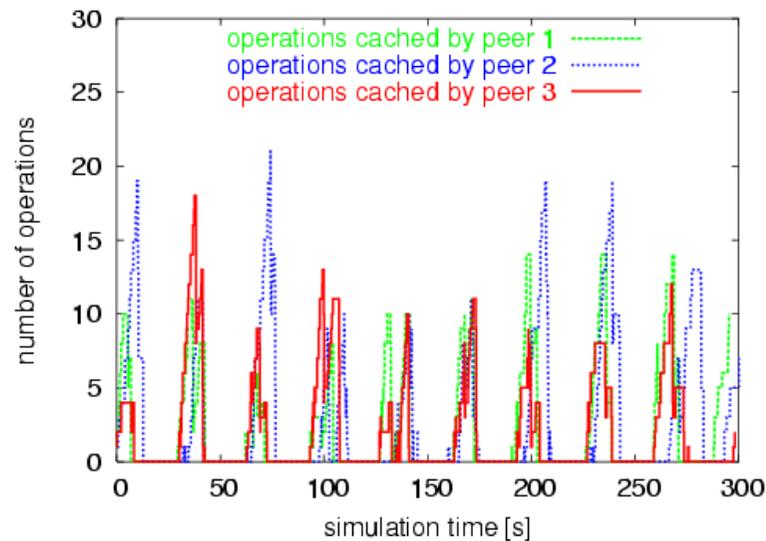
80% Online-Zeit



Activity Explorer (10) - Evaluation



Entwicklung der Operations-Historie bei 80% Online-Zeit



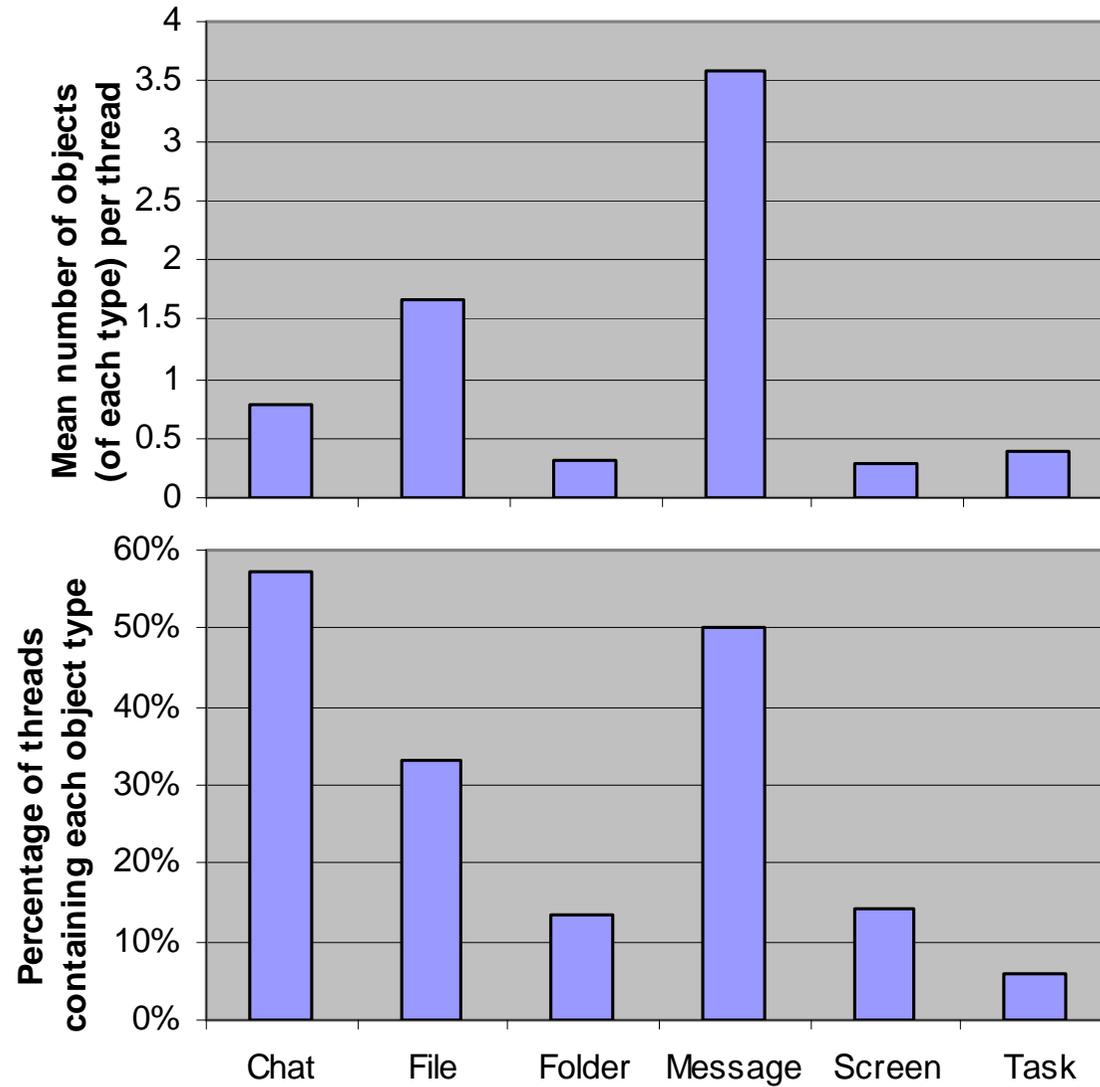
Auszuliefernde Operationen bei 80% Online-Zeit

Activity Explorer (11) - Evaluation

Feldstudie mit 30 Benutzern über 100 Tage

- Benutzer: Studenten und wissenschaftliche Mitarbeiter bei IBM Research in Boston
- keine konkrete Aufgabe vorgegeben → Evaluierung von Einsatzbereichen
- Techniken: Fragebögen und Aufzeichnung des Systemverhaltens (Objektnamen, Thread, Mitglieder, erzeugt wann und von wem, ...) → qualitative und quantitative Daten
- zentrale Ergebnisse der Aufzeichnung
 - von 1.487 Objekten waren 753 Nachrichten, 348 Dateien, ...
 - pro Objekt: \emptyset 3 Mitglieder; 7 Tage (bzw. 11,4 Tage) zwischen Erzeugung und letztem Schreibzugriff (bzw. Lesezugriff)
 - alle Ergebnisse mit hoher Standard-Abweichung
 - AE wird sowohl für kurz- als auch langfristige Interaktion verwendet → breites Interaktions-Spektrum

Activity Explorer (12) - Evaluation



Activity Explorer (13) - Evaluation

Zentrale Ergebnisse der Fragebögen

- vielfältige Einsatzbereiche: Projektarbeit, Austausch von Tips und Tricks, Abstimmung sozialer Aktivitäten, Awareness über die Aktivitäten anderer
- automatische Awareness-Benachrichtigungen nur für kleine Aktivitäten geeignet
- innerhalb der Benutzergruppe nur spärliche Verwendung anderer Groupware (IM, Email)

Inhalt

- Asynchrone Kommunikationssysteme
 - Email
 - BSCW
- Synchrone Kommunikationssysteme
 - computergestützte Videokonferenzen
 - Instant Messaging
- Gemeinsame Arbeitsbereiche
 - TeCo3D
 - mlb
 - Spiele
- Synchrone und asynchrone Interaktion
 - Activity Explorer
- **Software-Entwicklung**
 - CVS
 - Jazz
- Gemeinschaften ("Online Communities")
 - Wiki

Wiederholung: Software-Entwicklung

CSCW in der Software-Entwicklung

- Unterstützung und Strukturierung in allen Phasen des Entwicklungsprozesses (Analyse, Design, Implementierung, Evaluation)
- Interaktion zwischen Anwendern, Entwicklern, Projektleitung, ...

Groupware

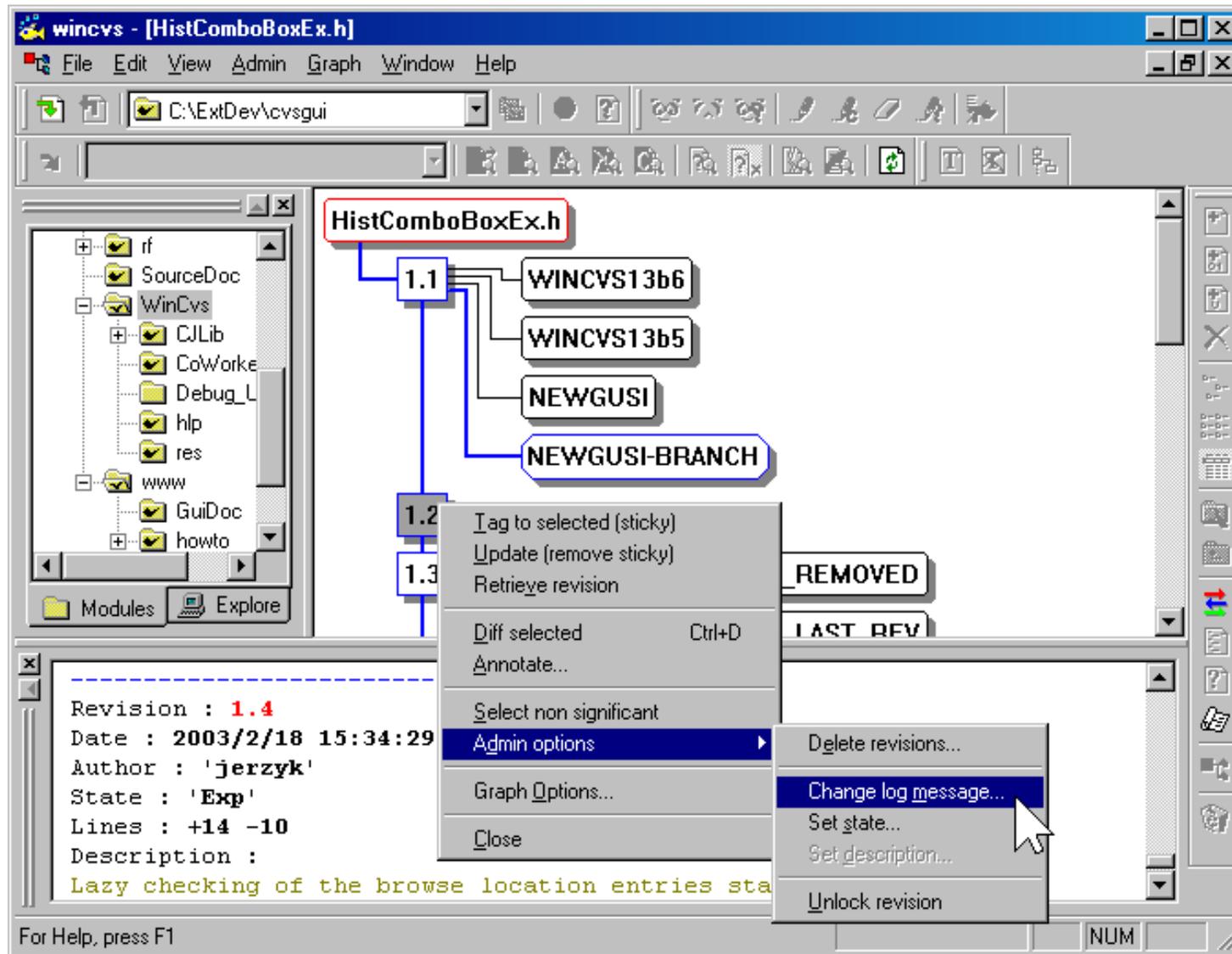
- Kommunikations-Anwendungen (Email, IM, Videokonferenz)
- Dokumentenaustausch (Email, BSCW, Wiki)
- Projekt-Management und Zeitplanung (Email, Kalender, Workflow)
- Programmierung (Gruppeneditor, Versionsverwaltung, Bug Tracking)

CVS ("Concurrent Versioning System") (1)

Beschreibung und Anwendungsbereiche

- Verwaltung von Quellcode (und anderen Textdokumenten) in einem zentralen Repository
- Änderungen werden lokal vorgenommen und dann explizit an das Repository übergeben → ermöglicht Offline-Arbeiten
- Ablauf:
 - (1) Anlegen einer lokalen Kopie ("Checkout")
 - (2) Änderung der lokalen Kopie
 - (3) Abgleich lokaler Version mit Repository-Version ("Update"),
 - (4) Ablegen der neuen Version im Repository ("Commit")
- automatische Erzeugung eines Versions-Baums
- textbasierter Vergleich verschiedener Dokumentenversionen
- Informationen über Änderungshistorie
- Anpassung über Scripte
- für große Gruppen und Projekte geeignet
- CVS-Clients: WinCVS, Eclipse, ...

CVS (2)



CVS (3)

Klassifikation und technische 3K-Protokolle

- asynchron
- explizite (Kommentare) und implizite (Änderungen) Kommunikation
- Koordination und Kooperation über soziale Protokolle

Daten, Datenmodell und Architektur

- Client-Server-Architektur
- hierarchischer Anwendungszustand: Ordner → Dateien → Zeilen
- Unterstützung von States

WYSIWIS und Awareness

- kein WYSIWIS im engeren Sinne: keine impliziten Zustandsupdates
- implizite Awareness durch Änderungs- und Konflikthinweise auf Benutzeranfrage
- Email-Benachrichtigungen (nach Commit)

CVS (4)

Floor und Session Control

- Floor Control durch Schreib- und Leserechte auf der Dateistruktur
- geschlossene Gruppen durch Serverzugang
- Teilnehmer-Rollen durch Schreib- und Leserechte auf der Dateistruktur

Konsistenzerhaltung und semantische Konflikte

- Konsistenz durch Serialisierung
- Benachrichtigung über Konflikte: Änderungen derselben Quellcodezeilen, Änderungen der Dateistruktur
- manuelle Konfliktauflösung

```
/* Look up the IP address of the host. */
host_info = gethostbyname (hostname);

<<<<<<< httpc.c
    if (! host_info) {
        fprintf (stderr, "%s: host not found:
%s\n", progname, hostname);
        exit (1);
    }
=====
    if (! host_info) {
        printf ("httpc: no host");
        exit (1);
    }
>>>>>>> 1.9
    sock = socket (PF_INET, SOCK_STREAM, 0);
```

CVS (5)

Undo von Operationen

- kein automatisches Undo

Late-Join

- durch Checkout der aktuellen Repository-Version

Netzwerk-Protokolle

- TCP, ssh

Inhalt

- Asynchrone Kommunikationssysteme
 - Email
 - BSCW
- Synchrone Kommunikationssysteme
 - computergestützte Videokonferenzen
 - Instant Messaging
- Gemeinsame Arbeitsbereiche
 - TeCo3D
 - mlb
 - Spiele
- Synchrone und asynchrone Interaktion
 - Activity Explorer
- **Software-Entwicklung**
 - CVS
 - **Jazz**
- Gemeinschaften ("Online Communities")
 - Wiki

Jazz (1)

Beschreibung und Anwendungsbereiche

- Integration von CSCW-Funktionalität in die Entwicklungsumgebung (IDE): Awareness und Ad-Hoc-Kommunikation (IM, Screen Sharing, VoIP)
- IM-Besonderheiten: Objekt-Bezug ("Anchored Chat"), Unterstützung von Threads, nachträgliches Editieren der IM-Historie (z.B. Verschieben von Nachrichten)
- Screen Sharing für gemeinsames Programmieren, Debuggen, ...
- kleine Entwicklungs-Teams
- Umsetzung als Plugins für die Eclipse-Plattform
- entwickelt von IBM Research

Klassifikation und technische 3K-Protokolle

- synchrone und asynchrone Kommunikation
- Koordination und Kooperation über soziale Protokolle

Jazz (2)

The screenshot shows the Eclipse IDE with the 'ActionPane.java' file open. A Jazz chat window is overlaid on the code editor. Annotations 'a' through 'f' point to specific features:

- a**: Points to a member's profile picture in the chat window.
- b**: Points to the chat window title bar, which includes the member's name and status.
- c**: Points to the package explorer on the left, highlighting the 'ActionPane' package.
- d**: Points to the 'import' statements in the code editor.
- e**: Points to the 'import java.util.Vector;' statement in the code editor.
- f**: Points to a status message in the chat window: 'Isaac: I am active (as of 5:31 PM EST, Feb 26/04) working on ActionPaneItem.java'.

The chat window displays a list of members: John, Steve, Cleidson, Bryan, Kushal, Susanne, Isaac, Li-Te, and Sooz. Isaac is highlighted as the active member.

- a) Mitglied mit Status
- b) Kommunikationskanäle
- c) Objektinformationen
- d) Anker für IM-Sitzung
- e) Änderungsmarkierung
- f) Statusnachricht

Jazz (3)

Daten, Datenmodell und Architektur

- Client-Server-Architektur
- States und Events

WYSIWIS und Awareness

- relaxiertes WYSIWIS (Code wird per Versioning-System verwaltet)
- personenorientierte Awareness
 - Zusammensetzung der Gruppe und den Status der Mitglieder
 - vergangene und aktuelle Änderungen am Quellcode
 - teilnehmerbezogene IDE-Informationen: aktuelles Projekt, Perspektive, Datei, ...
- ressourcenorientierte Awareness
 - wer greift lesend oder schreibend zu
 - wer ist für welche Änderungen verantwortlich
 - Verhältnis jeder einzelnen Ressourcen-Kopie zur Repository-Version

Jazz (4)

Floor und Session Control

- geschlossene Gruppe aus gleichberechtigten Teilnehmern (Programmierern)
- Zugehörigkeit per Einladung

Konsistenzerhaltung und semantische Konflikte

- IM-Konsistenz durch Server-Serialisierung
- Benachrichtigung über potentielle Konflikte beim nebenläufigen Editieren von Quellcode

Undo von Operationen

- kein automatisches Undo

Late-Join

- Initialisierung des LJ-Clients über den Server

Inhalt

- Asynchrone Kommunikationssysteme
 - Email
 - BSCW
- Synchrone Kommunikationssysteme
 - computergestützte Videokonferenzen
 - Instant Messaging
- Gemeinsame Arbeitsbereiche
 - TeCo3D
 - mlb
 - Spiele
- Synchrone und asynchrone Interaktion
 - Activity Explorer
- Software-Entwicklung
 - CVS
 - Jazz
- Gemeinschaften ("Online Communities")
 - Wiki

Gemeinschaften ("Online Communities")

- Gemeinschaftsbildung: Groupware als sozialer Lebensraum und Begegnungsstätte für Menschen mit gemeinsamen Interessen
- Fragestellungen:
 - Austausch von Informationen und Meta-Informationen (z.B. Bewertungen)
 - personalisierte Informationssuche
 - Repräsentation und Individualisierung von Benutzern
 - Wahrnehmung und Auffinden anderer Benutzer
 - Anreize und Sanktionen
 - Ordnung und ethische Normen

Groupware

- Wiki
- Newsgroup
- Chatraum
- Spiele
- Blog
- Bulletin Board
- Virtual-Reality-Umgebung
- Beziehungsnetzwerk

Literaturhinweise (1)

Email

- RFC 1855, 1931, 2045-2049, 2821, 2822, 3501
- Ducheneaut, N. and Bellotti, V. *E-Mail as a Habitat: An Exploration of Embedded Personal Information Management*. In: ACM Interactions, Vol. 8, No. 5, pages 30–38, 2001
- D. Gruen, S.L. Rohall, S. Minassian, B. Kerr, P. Moody, B. Stachel, M. Wattenberg and E. Wilcox, Lessons from the ReMail Prototypes, In: Proc. ACM CSCW, Chicago, IL, USA, November 2004.

BSCW

- www.bscw.de
- bscw.fit.fraunhofer.de

Videokonferenzen

- R. Vertegaal, I. Weevers, C. Sohn and C. Cheung, GAZE-2: Conveying Eye Contact in Group Video Conferencing Using Eye-Controlled Camera Direction, In: Proc. ACM CHI, Ft. Lauderdale, FL, USA, April 2003

Literaturhinweise (2)

Instant Messaging

- E. Isaacs, A. Walendowski, S. Whittaker, D.J. Schiano and C. Kamm, The Character, Functions, and Styles of Instant Messaging in the Workplace, In: Proc. ACM CSCW, New Orleans, LO, USA, November 2002

TeCo3D

- Mauve, M. Distributed Interactive Media. Ph.D. thesis, Department for Mathematics and Computer Science, University of Mannheim, Germany, August 2000

mlb

- www.informatik.uni-mannheim.de/pi4/projects/mlb
- J. Vogel, Consistency Algorithms and Protocols for Distributed Interactive Applications, PhD Thesis, University of Mannheim, 2004

Literaturhinweise (3)

Spiele

- M. Mauve, S. Fischer, J. Widmer, A Generic Proxy System for Networked Computer Games, In: Proc. NetGames , Braunschweig, 2002.
- Mauve, M., Vogel, J., Hilt, V., and Effelsberg, W. *Local-lag and Timewarp: Providing Consistency for Replicated Continuous Applications*. In: IEEE Transactions on Multimedia, Vol. 6, No. 1, pages 45–57, 2004
- T. Beigbeder, R. Coughlan, C. Lusher, J. Plunkett, E. Agu and M. Claypool, The Effects of Loss and Latency on User Performance in Unreal Tournament 2003, in Proc. ACM SIGCOMM Workshops, Portland, OR, USA, August 2004
- K. Mansley, D. Scott, A. Tse, A. Madhavapeddy, Feedback, Latency, Accuracy: Exploring Tradeoffs in Location-Aware Gaming, In: Proc. ACM SIGCOMM Workshops, Portland, OR, USA, August 2004
- A.D. Cheok, K.H. Goh, W. Liu, F. Farbiz, S.W. Fong, S.L. Teo, Y. Li and X. Yang, Human Pacman: a mobile, wide-area entertainment system based on physical, social, and ubiquitous computing, In. Personal Ubiquitous Computing, 8, 71-81, 2004.
- N. Ducheneaut, R.J. Moore, The social side of gaming: a study of interaction patterns in a massively multiplayer online game, In: Proc. ACM CSCW, Chicago, IL, USA, November 2004

Literaturhinweise (4)

Activity Explorer

- W. Geyer, J. Vogel, L.-T. Cheng, M. Muller; Supporting Activity-centric Collaboration through Peer-to-Peer Shared Objects, In: Proc. ACM SIGGROUP, Sanibel Island, FL, USA, 2003
- J. Vogel, W. Geyer, L.-T. Cheng, M. Muller, Consistency Control for Synchronous and Asynchronous Collaboration Based on Shared Objects and Activities, Springer Journal CSCW, 13, pp. 573-602, 2004
- M. Muller, W. Geyer, B. Brownholtz, E. Wilcox, D.R. Millen, One Hundred Days in an Activity-centric Collaboration Environment, In: Proc. ACM CHI, Vienna, Austria, 2004
- J. Vogel, Consistency Algorithms and Protocols for Distributed Interactive Applications, PhD Thesis, University of Mannheim, 2004

CVS

- www.cvshome.org

Literaturhinweise (5)

Jazz

- L.-T. Cheng, S. Hupfer, S. Ross, J. Patterson, Jazzing up Eclipse with Collaborative Tools, In: Proc. OOPSLA, Anaheim, CA, USA, 2003
- S. Hupfer, L.-T. Cheng, S. Ross, J. Patterson, Introducing Collaboration into an Application Development Environment, In: Proc. ACM CSCW, Chicago, IL, USA, November 2004

Wiki

- en.wikipedia.org/wiki/Wiki
- P. Aigrain, The Individual and the Collective in Open Information Communities, Invited Talk: 16th Bled Electronic Commerce Conference, Bled, Slovenija, June 2003
- F.B. Viegas, M. Wattenberg, K. Dave, Studying Cooperation and Conflict between Authors with history flow Visualizations, In: Proc. ACM CHI, Vienna, Austria, April 2004
- <http://www.research.ibm.com/history/>