

# Human Computer Interaction

## Kapitel 19 Groupware

# Was ist Groupware?

- Systeme mit mehr als einem Nutzer
- Unterstützen eine Gruppe während des Arbeitsprozess, indem
  - Sie sich der Arbeitsweise anpassen
  - Sie Gruppenprozesse leichter und schneller gestalten
  - Sie Verwaltungsaufgaben übernehmen

# Themenübersicht

1.Einteilungsmodelle

2.Darstellung verschiedener Systemarten

2.1 Kommunikationssysteme

2.2 Systeme für Konferenzen und Meetings

2.3 Geteilte Werkzeuge und Arbeitsumgebungen

3.Rahmenbedingungen zum Design

4.Probleme bei Umsetzungen synchroner Systeme

# Kapitel 1

# Einteilungsmodelle

# Ziele dieses Themas

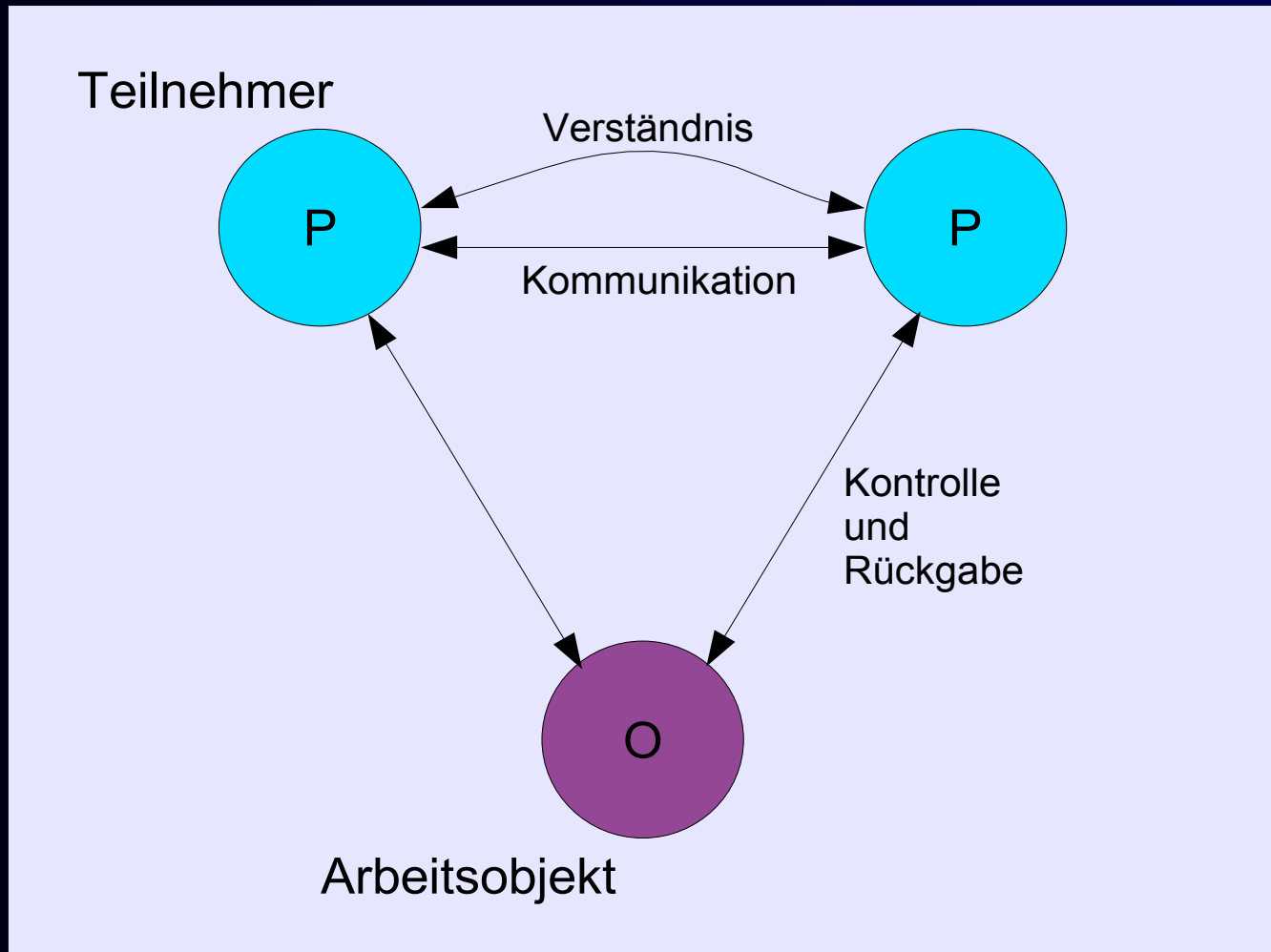
- Wie können wir Anforderungen an ein System klassifizieren?
- In wie weit unterstützt es uns in der Designanalyse?

# Zeit- / Raummatrix

|           | beisammen   | entfernt |
|-----------|-------------|----------|
| synchron  | Gespräch    | Telefon  |
| asynchron | Notizzettel | Brief    |

dem Bild 19.1 aus dem Buch Human-Computer Interaction nachempfunden

# Betrachtung der Arbeitsumgebung eines Teams



dem Bild 19.2 aus dem Buch Human-Computer Interaction nachempfunden

# Wichtige Fragen beim Design

- Was ist das Ziel der Benutzergruppe?
- Wie nimmt der Einzelne seine Aufgaben wahr?
- Wie ist die Arbeitsumgebung aufgebaut?
  - Wie kommuniziert die Gruppe?
  - Wie wirkt der Einzelne auf das System ein?
  - Finden Änderungen häufig statt?



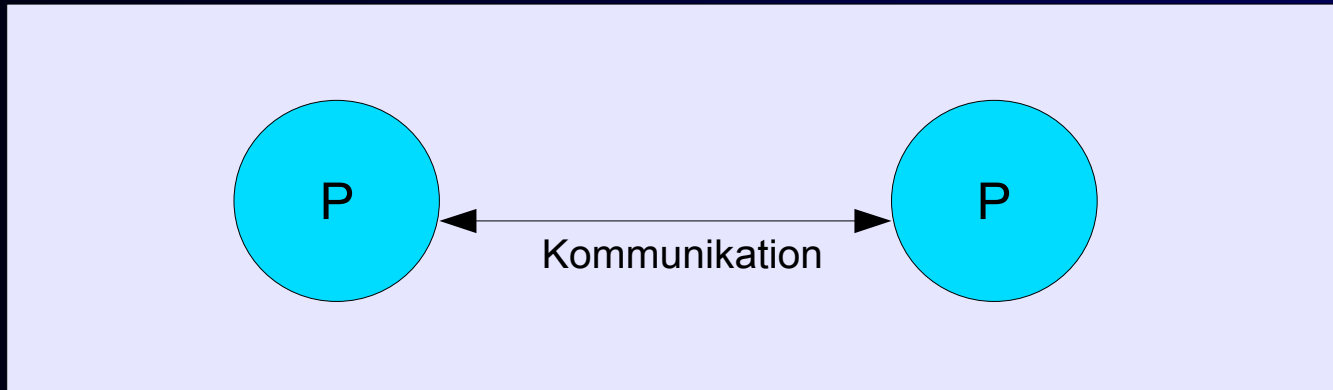
# Kapitel 2

## Darstellung verschiedener Systemarten

# Unterabschnitte

- Wir betrachten nun folgende Felder in denen Groupware-Systeme eine Rolle spielen
  - Computer-gestützte Kommunikation
  - Computer-gestützte Konferenzen
  - Geteilte Werkzeuge und Arbeitsumgebungen

# Kommunikation



- Warum benötigen wir dafür die Hilfe von Computersystemen?
  - In großen Firmen können alle Mitarbeiter nicht persönlich informiert werden
  - Im Zeitalter der Globalisierung wird rund um die Welt an gemeinsamen Projekten gearbeitet
  - Hobby-Projekte laufen zumeist dezentralisiert ab

# Erkenntnisse

|           | beisammen | entfernt |
|-----------|-----------|----------|
| synchron  |           | ✓        |
| asynchron |           | ✓        |

- erst bei entfernten Arbeitsplätzen sinnvoll
- Hauptunterscheidungsmerkmal: Zeitliche Differenz

# Asynchrone Vertreter



**E-Mail**



**SMS**

- Vielseitige Systeme, welche sich in vielen Bereichen des täglichen Lebens durchgesetzt haben
- Sind nicht für direkte Gespräche gedacht, sondern ähneln dem traditionellen Brief

# E-Mail

- Vor dem Aufkommen des World Wide Web war die elektronische Post der wichtigste Dienst des Internets.

# E-Mail

- Vor dem Aufkommen des World Wide Web war die elektronische Post der wichtigste Dienst des Internets.
- Vorteile:
  - Einfacher Aufbau: nur ASCII-Symbole, Header und Body
  - Massenverteilung an alle Mitarbeiter möglich
  - Sortierung und Archivierung einfach
  - Leicht erweiterbar als Grundbaustein anderer Systeme

# E-Mail

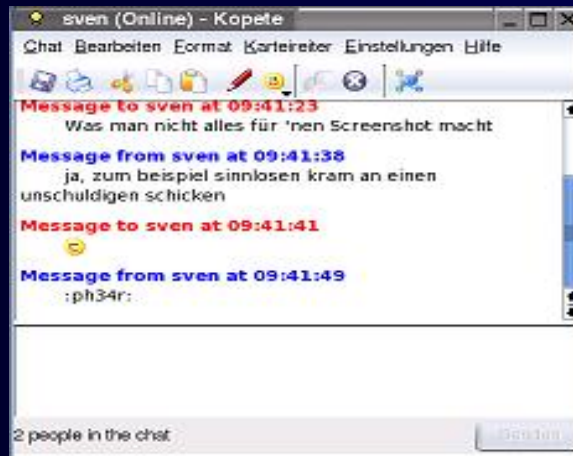
- Vor dem Aufkommen des World Wide Web war die elektronische Post der wichtigste Dienst des Internets.
- Vorteile:
  - Einfacher Aufbau: nur ASCII-Symbole, Header und Body
  - Massenverteilung an alle Mitarbeiter möglich
  - Sortierung und Archivierung einfach
  - Leicht erweiterbar als Grundbaustein anderer Systeme
- Nachteile
  - Für den direkten Austausch ungeeignet



# Synchrone Vertreter



**Audio- / Video Chat**



**Instant Messaging**



**Virtuelle Welten**

### Quellen:

Bild1: [http://blog.cws.net/uploaded\\_images/skype-796446.bmp](http://blog.cws.net/uploaded_images/skype-796446.bmp)

Bild2: [http://www.linux-user.de/ausgabe/2004/01/062-drlinux/kopete\\_message\\_window.png](http://www.linux-user.de/ausgabe/2004/01/062-drlinux/kopete_message_window.png)

Bild3: <http://secondlife.com/world/de/whatis/>

# Instant Messaging



# Instant Messaging



- Vorteile

- Für viele mittlerweile Austauschmedium schlecht hin
- Systeme sind recht stabil
- Bieten direkte Kommunikation auf Textbasis
- Transparenz wer online ist

# Instant Messaging



- Vorteile
  - Für viele mittlerweile Austauschmedium schlecht hin
  - Systeme sind recht stabil
  - Bieten direkte Kommunikation auf Textbasis
  - Transparenz wer online ist

- Nachteile
  - Aufgrund schlechter Archivierung-Möglichkeiten sind sie nicht zur Dokumentation der Arbeitsschritte geeignet
  - Durch informelle Gespräche können Missverständnisse auftreten, welche nicht durch Gestik und Mimik aufgehoben werden

# Audio- / Video Chat

- Aufbau:
  - Alle Gesprächspartner nutzen eine Kamera und ein Mikrofon
  - Aufgezeichnete Daten werden bearbeitet und dann als Pakete über das Internet übertragen

# Audio- / Video Chat

- Aufbau:
  - Alle Gesprächspartner nutzen eine Kamera und ein Mikrofon
  - Aufgezeichnete Daten werden bearbeitet und dann als Pakete über das Internet übertragen
- Vorteile:
  - Über weite Strecken ist die Kommunikation mit Bild und Ton der beste Ersatz für das persönliche Gespräch
  - Gestik, Mimik sowie Stimmlagen sind wichtiger Bestandteil unseres Umgangs mit anderen Menschen

# Probleme und Lösungsversuche

- Technische Probleme:
  - Bandbreiten können oft nicht die Datenmenge in der benötigten Geschwindigkeit übertragen
  - Gespräche wirken oft noch unnatürlich durch starre Kamera-Position

# Probleme und Lösungsversuche

- Technische Probleme:
  - Bandbreiten können oft nicht die Datenmenge in der benötigten Geschwindigkeit übertragen
  - Gespräche wirken oft noch unnatürlich durch starre Kamera-Position
- Lösungsversuche:
  - Algorithmen, welche Daten komprimieren und dabei heuristische Techniken anwenden (der Mensch nimmt bestimmte Dinge nicht wahr)
  - Spezielle Kameras, welche sich selbst der Position des Gesprächspartner anpassen, um direkten Augenkontakt zu ermöglichen



# Der Video-Tunnel

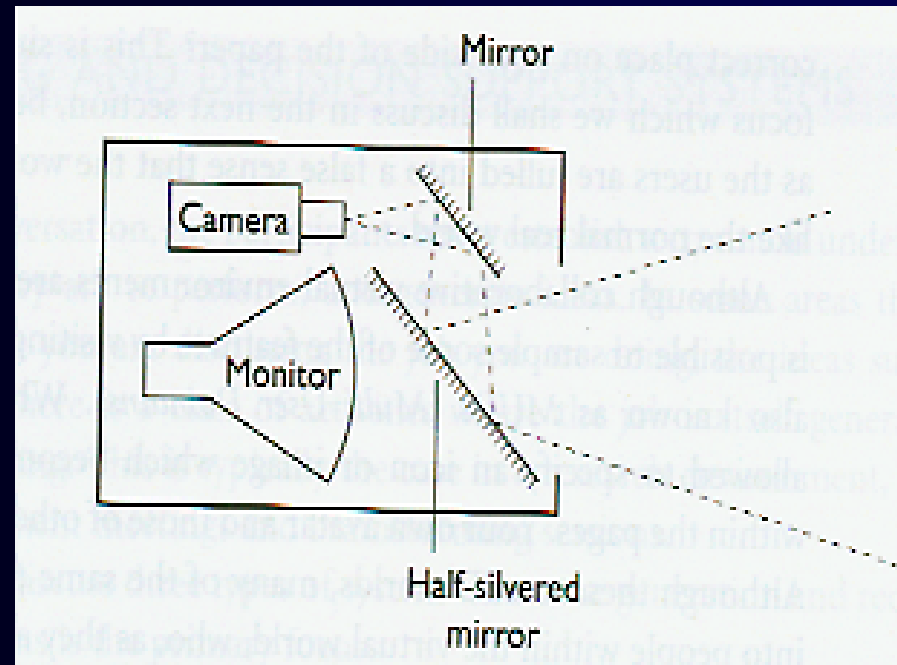
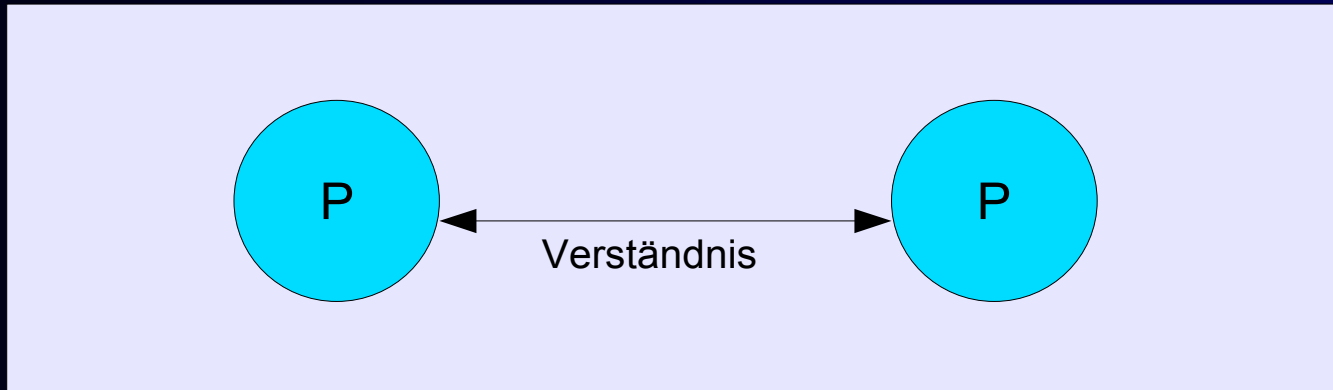


Bild 19.5 aus dem Buch Human-Computer Interaction

# Konferenzen und Meetings



- Die Arbeit im Team erfordert neben der Kommunikation auch ein gemeinsames Verständnis über die Ziele und das Design
- Konferenzen dienen
  - Zum Austausch verschiedener Arbeitsgruppen
  - Um Übersicht über den aktuellen Stand des Projekts zu schaffen

# Wie kann Groupware helfen?

- Groupware Systeme können lästige Aufgaben übernehmen
  - Archivierung einzelner Konferenzen
  - Änderungen an Aufzeichnung automatisieren
  - Verteilung neuer Projektpläne
  - Fehler und Gemeinsamkeiten einzelner Aufzeichnungen darstellen

# Wie kann Groupware helfen?

- Groupware Systeme können lästige Aufgaben übernehmen
  - Archivierung einzelner Konferenzen
  - Änderungen an Aufzeichnung automatisieren
  - Verteilung neuer Projektpläne
  - Fehler und Gemeinsamkeiten einzelner Aufzeichnungen darstellen
- Groupware Systeme können jedem Teilnehmer die Arbeit von seinem Platz aus ermöglichen

# Wie kann Groupware helfen?

- Groupware Systeme können lästige Aufgaben übernehmen
  - Archivierung einzelner Konferenzen
  - Änderungen an Aufzeichnung automatisieren
  - Verteilung neuer Projektpläne
  - Fehler und Gemeinsamkeiten einzelner Aufzeichnungen darstellen
- Groupware Systeme können jedem Teilnehmer die Arbeit von seinem Platz aus ermöglichen
- Groupware Systeme können die Einhaltung sozialer Regeln unterstützen

# Klassifikation

|           | beisammen | entfernt |
|-----------|-----------|----------|
| synchron  | ✓         | ✓        |
| asynchron |           |          |

- Konferenzen finden nicht zeitlich versetzt statt
- Jedoch könnten mit Hilfe von Groupware Systemen Teams aus Tokyo und Kalifornien zusammenarbeiten

# Spezielle Konferenzräume



Quellen:

Bild1: <http://www.isrc.umbc.edu/HCIHandbook/figures/29-02.jpg>

Bild2: Bild 19.6 aus dem Buch Human Computer Interaction

# Spezielle Konferenzräume

- Aufbau:
  - Sitzplätze sind mit Rechnern ausgestattet
  - Großer Bildschirm visualisiert die Arbeit des Einzelnen
  - Teilnehmer können Dokument auf dem großen Bildschirm verändern
  - Bei Vorträgen können Teilnehmer die Präsentationen auf ihren Bildschirmen verfolgen
  - System könnte auf verteilte Räume erweitert werden



# Spezielle Konferenzräume

- Aufbau:
  - Sitzplätze sind mit Rechnern ausgestattet
  - Großer Bildschirm visualisiert die Arbeit des Einzelnen
  - Teilnehmer können Dokument auf dem großen Bildschirm verändern
  - Bei Vorträgen können Teilnehmer die Präsentationen auf ihren Bildschirmen verfolgen
  - System könnte auf verteilte Räume erweitert werden
- Probleme:
  - Ausstattung sehr teuer
  - Teilnehmer können sich gegenseitig in die Quere bei der Benutzung kommen

# Lösungsmöglichkeiten

- Kostenfaktor senken
  - eBeam System 3 benötigt nur einen weißen Hintergrund
  - Bewegungen des Stift werden aufgezeichnet



Quelle [http://www.creative-products.de/presentationssysteme/gfx/ebeam\\_s3\\_blue.jpg](http://www.creative-products.de/presentationssysteme/gfx/ebeam_s3_blue.jpg)

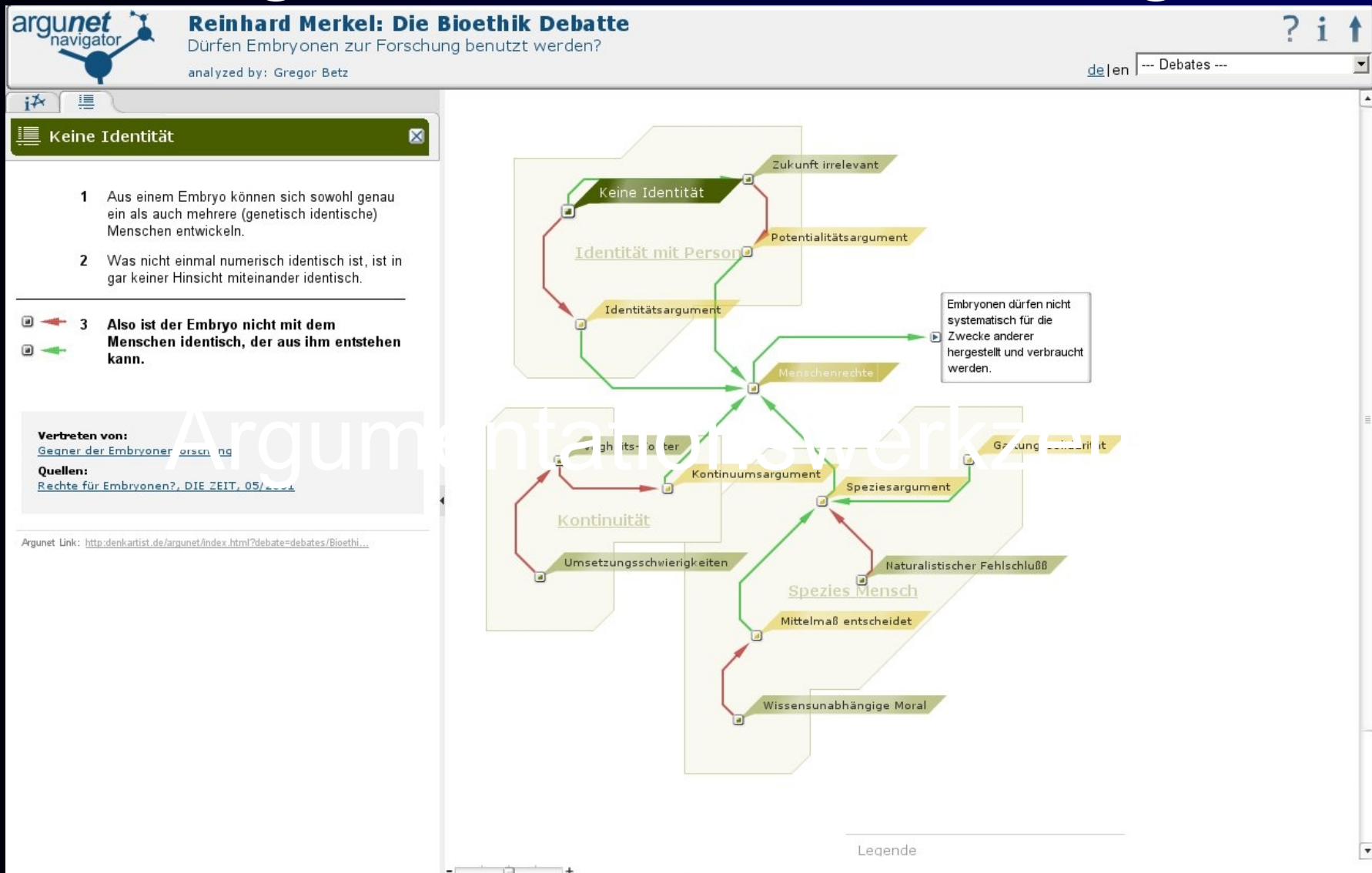
# Lösungsmöglichkeiten

- Kostenfaktor senken
  - eBeam System 3 benötigt nur einen weißen Hintergrund
  - Bewegungen des Stift werden aufgezeichnet
- Benutzerkollision verhindern
  - das System könnte jeweils nur einem Teilnehmer das Verändern des bearbeiteten Dokuments zur gleichen Zeit erlauben
  - wenn der bearbeitende Teilnehmer eine gewisse Zeitlänge inaktiv bleibt, so könnte das System automatisch den Zugriff freigeben



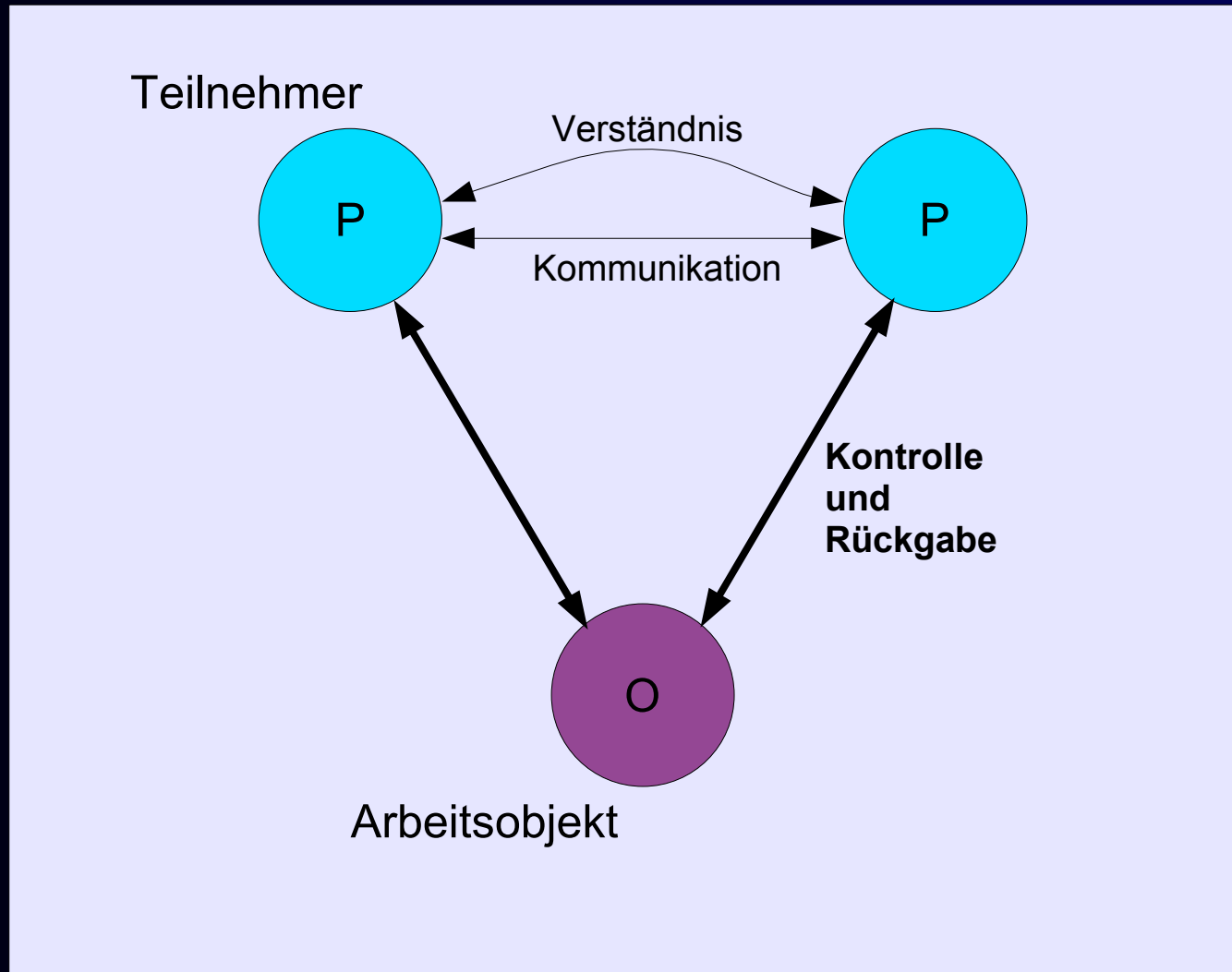
Quelle [http://www.creative-products.de/presentationssysteme/gfx/ebeam\\_s3\\_blue.jpg](http://www.creative-products.de/presentationssysteme/gfx/ebeam_s3_blue.jpg)

# Argumentationswerkzeuge



Quelle: [http://denkartist.de/argunet/index.html?debate=debates/Bioethik/merkel\\_html.xml](http://denkartist.de/argunet/index.html?debate=debates/Bioethik/merkel_html.xml)

# Geteilte Werkzeuge



# Geteilte Werkzeuge

- bisher betrachtete Systeme unterstützten nur den Arbeitsablauf zwischen Mitgliedern eines Teams
  - Kommunikation unter Mitgliedern
  - Zielsetzung der Mitglieder

# Geteilte Werkzeuge

- bisher betrachtete Systeme unterstützten nur den Arbeitsablauf zwischen Mitgliedern eines Teams
  - Kommunikation unter Mitgliedern
  - Zielsetzung der Mitglieder
- geteilte Arbeitsumgebungen gehen weiter:
  - Produktion als solche steht im Vordergrund
  - gemeinsame Ressourcen und Produktionsobjekte

# Geteilte Werkzeuge

- bisher betrachtete Systeme unterstützten nur den Arbeitsablauf zwischen Mitgliedern eines Teams
  - Kommunikation unter Mitgliedern
  - Zielsetzung der Mitglieder
- geteilte Arbeitsumgebungen gehen weiter:
  - Produktion als solche steht im Vordergrund
  - gemeinsame Ressourcen und Produktionsobjekte
- Idee dahinter:
  - einzelne Fenster, Applikationen oder sogar gesamte Rechnersysteme werden regelrecht geteilt
  - was einer sieht, sehen alle Mitarbeiter



# Klassifikation

|           | beisammen | entfernt |
|-----------|-----------|----------|
| synchron  | ✓         | ✓        |
| asynchron |           |          |

- geteilte Werkzeuge sind erst klassifizierbar, wenn mehrere Menschen wirklich zur selben Zeit Dinge teilen
- Entfernungen spielen nur im technischen Aufwand eine Rolle

# Wie kann Groupware helfen?

- Groupware Systeme können auch hier durch Regelsysteme die gemeinsame Arbeit unterstützen
- Automatisierte Routinen um verschiedene Rechner auf dem selben Stand zu halten
- durch verschiedene Systemmodi könnte das System sich den verschiedenen Bedürfnissen jedes Arbeitsschrittes anpassen
  - nur wenn verlangt sehen alle dasselbe
  - Verschluss-Mechanismen je nach Wunsch abstellbar

# Ko-Autoren Systeme

- Ziel: Unterstützung langfristiger Bearbeitung eines Projektes
- Möglichkeiten der Umsetzung:
  - 1. jeder Autor verändert nur einen bestimmten Textbereich
  - 2. alle Autoren haben gleichzeitigen Zugriff zur selben Datenbasis

# Ko-Autoren Systeme

- Ziel: Unterstützung langfristiger Bearbeitung eines Projektes
- Möglichkeiten der Umsetzung:
  - 1. jeder Autor verändert nur einen bestimmten Textbereich
  - 2. alle Autoren haben gleichzeitigen Zugriff zur selben Datenbasis
- 1.Fall:
  - einfacher Verschlussmechanismus
  - System muss nur Daten abgleichen und verteilen

# Ko-Autoren Systeme

- Ziel: Unterstützung langfristiger Bearbeitung eines Projektes
- Möglichkeiten der Umsetzung:
  - 1. jeder Autor verändert nur einen bestimmten Textbereich
  - 2. alle Autoren haben gleichzeitigen Zugriff zur selben Datenbasis
- 1.Fall:
  - einfacher Verschlussmechanismus
  - System muss nur Daten abgleichen und verteilen
- 2.Fall:
  - aufwendiger Verschlussmechanismus
  - Änderungen müssen festgehalten werden

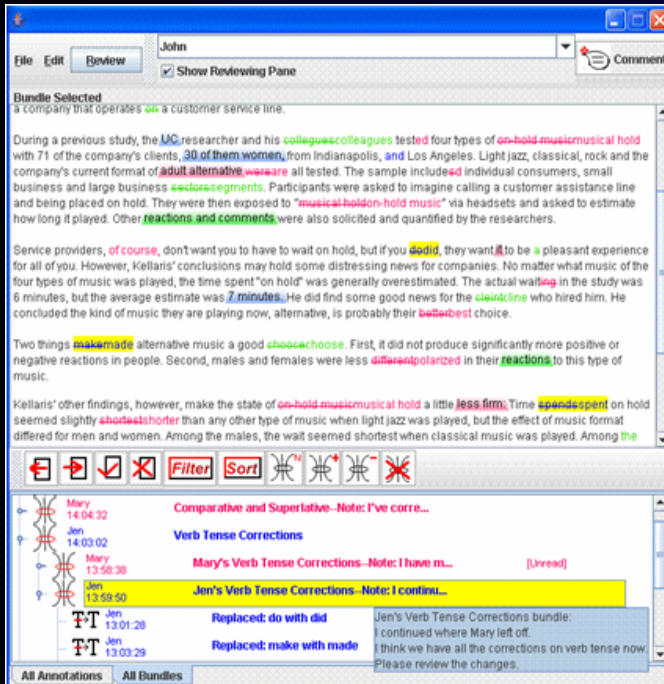
# Beispiel der University of British Columbia

- Problem:
  - Oft werden Kommentare verschiedener Autoren in separierten Dateien abgelegt (z.B. E-Mails)
  - Ineffizienz und Fehleranfälligkeit aufgrund des Suchaufwands
  - Missverständnisse, wenn jemand Dinge einfach verändert

# Beispiel der University of British Columbia

- Problem:
  - Oft werden Kommentare verschiedener Autoren in separierten Dateien abgelegt (z.B. E-Mails)
  - Ineffizienz und Fehleranfälligkeit aufgrund des Suchaufwands
  - Missverständnisse, wenn jemand Dinge einfach verändert
- Analyse:
  - Entwickler überprüften E-Mails von Autoren die zusammen eine wissenschaftliche Arbeit schrieben
  - Man erkannte 10 Klassen, in die sich jeder Kommentar einteilen ließ
  - Man entwickelte einen Editor, welcher den zu bearbeitenden Text mit Kommentaren verbindet

# Die Software



- Das entwickelte System verbindet das zu bearbeitende Dokument mit Anmerkungen
- Anmerkungen besitzen Pflichtdaten und optionale Felder
- Pflichtdaten, die vom System automatisch angelegt werden sind:
  - Name des Autors einer Veränderung
  - Zeitmarke
  - Status (gelesen / nicht gelesen sowie Zustimmung / Ablehnung)

Quelle: “Co-Authoring with Structured Annotations“ von Zheng, Booth und McGrenere



# Vergleich

- Wikipedia bietet den gemeinsamen Autoren ein ähnliches System
- Über jede Änderung wird Buch geführt
- Jede Änderungen kann überprüft, diskutiert und korrigiert werden

|                           |   |
|---------------------------|---|
| ■ (Aktuell) (Vorherige) ○ | 23:56, 17. Sep. 2006 Guidod (Diskussion   Beiträge) (→ <i>Die Rache der Rōnin</i> - voll übersetzt)                         |
| ■ (Aktuell) (Vorherige) ○ | 22:15, 17. Sep. 2006 Guidod (Diskussion   Beiträge) (→ <i>Vorgeschichte</i> - übersetzt)                                    |
| ■ (Aktuell) (Vorherige) ○ | 14:38, 16. Sep. 2006 88.112.49.117 (Diskussion) (+fi)   |
| ■ (Aktuell) (Vorherige) ○ | 16:28, 31. Aug. 2006 Pacifier (Diskussion   Beiträge)   |
| ■ (Aktuell) (Vorherige) ○ | 15:19, 31. Aug. 2006 Asthma (Diskussion   Beiträge) K   |
| ■ (Aktuell) (Vorherige) ○ | 21:07, 14. Jun. 2006 YurikBot (Diskussion   Beiträge) K (Bot: Ergänze: <i>fr:47 rōnin</i> , <i>it:Quarantasette Ronin</i> ) |
| ■ (Aktuell) (Vorherige) ○ | 15:01, 12. Jun. 2006 Shikeishu (Diskussion   Beiträge) (→ <i>Verfilmung</i> )   |
| ■ (Aktuell) (Vorherige) ○ | 17:04, 21. Mai 2006 Dergreg: (Diskussion   Beiträge) K  |

Quelle: [http://de.wikipedia.org/w/index.php?title=Geschichte\\_der\\_47\\_R%C5%8Dnin&limit=500&action=history](http://de.wikipedia.org/w/index.php?title=Geschichte_der_47_R%C5%8Dnin&limit=500&action=history)

# Kapitel 3

## Rahmenbedingungen zum Design

# Zeit- /Raummatrix

Alt:

|           | beisammen               | entfernt       |
|-----------|-------------------------|----------------|
| synchron  | Konferenzraum           | Videokonferenz |
| asynchron | Argumentationswerkzeuge | E-Mail         |

dem Bild 19.9 aus dem Buch Human-Computer Interaction nachempfunden

# Zeit- /Raummatrix

Alt:

|           | beisammen               | entfernt       |
|-----------|-------------------------|----------------|
| synchron  | Konferenzraum           | Videokonferenz |
| asynchron | Argumentationswerkzeuge | E-Mail         |

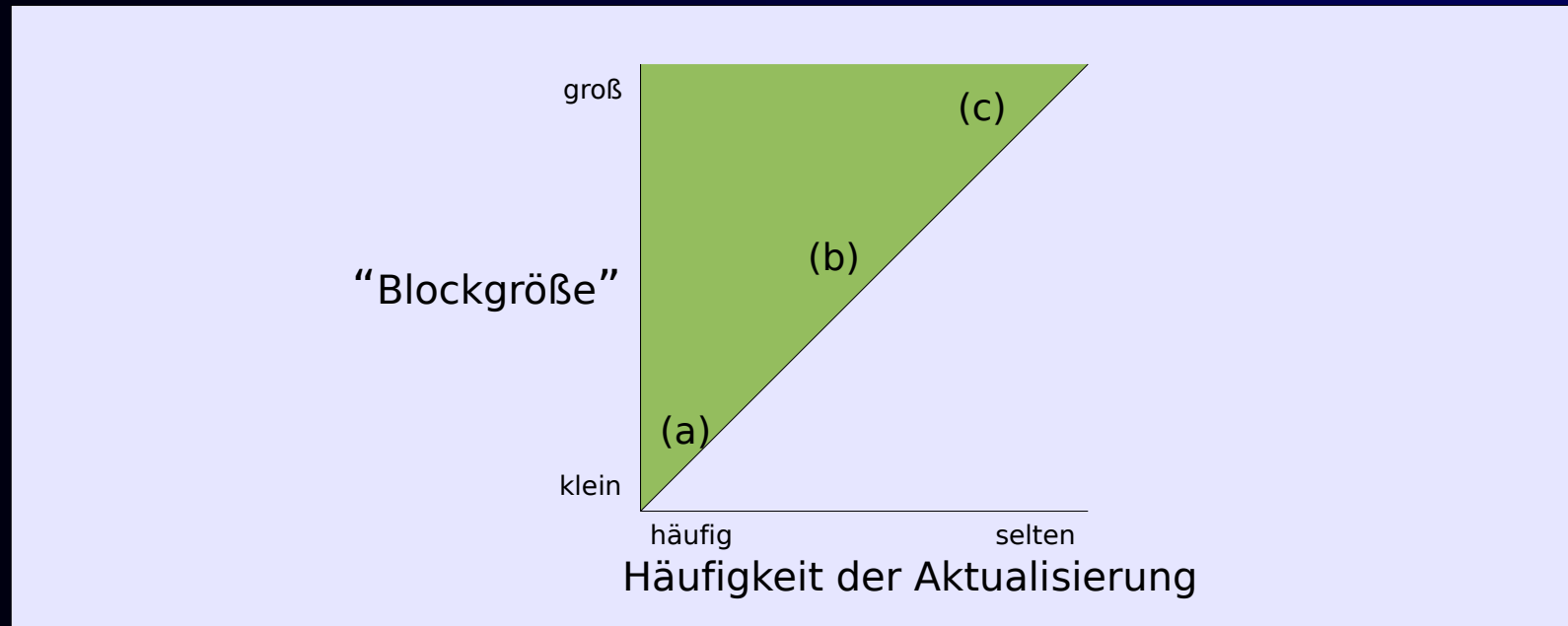
dem Bild 19.9 aus dem Buch Human-Computer Interaction nachempfunden

Neu:

|           | beisammen   | entfernt       |
|-----------|---|----------------|
| simultan  | Konferenzraum<br>geteilte Editoren / Fenstersysteme | Videokonferenz |
| seriell   | Argumentationswerkzeuge                             |                |
| gemischt  | Co-Autor Systeme                                    |                |
| asynchron | E-Mail  |                |

dem Bild 19.10 aus dem Buch Human-Computer Interaction nachempfunden

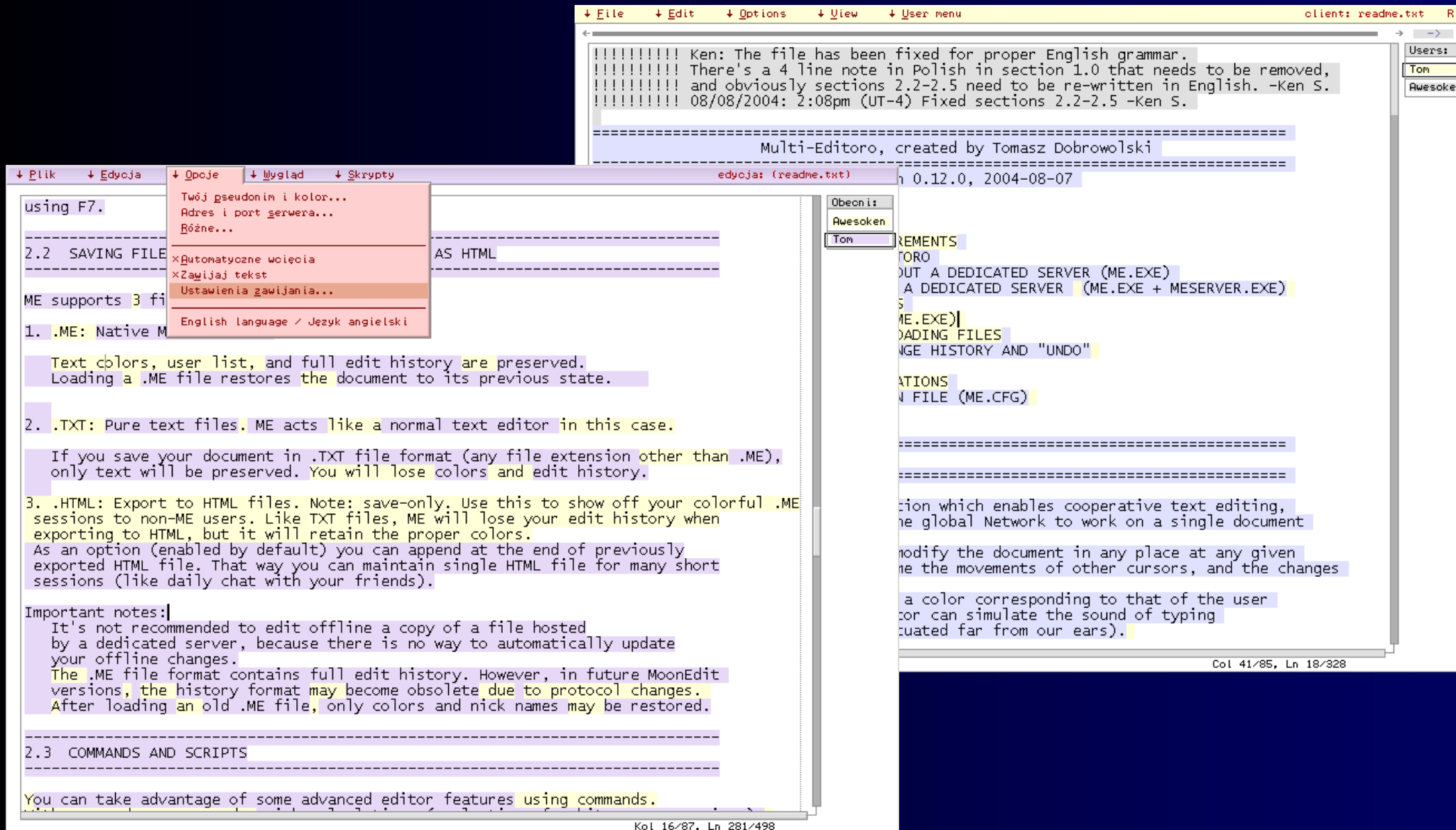
# Granularität



dem Bild 19.11 aus dem Buch Human-Computer Interaction nachempfunden

- (a) geteilter Editor
- (b) Co-Autor System
- (c) geteiltes Dateisystem mit Sperren

# Präsentation

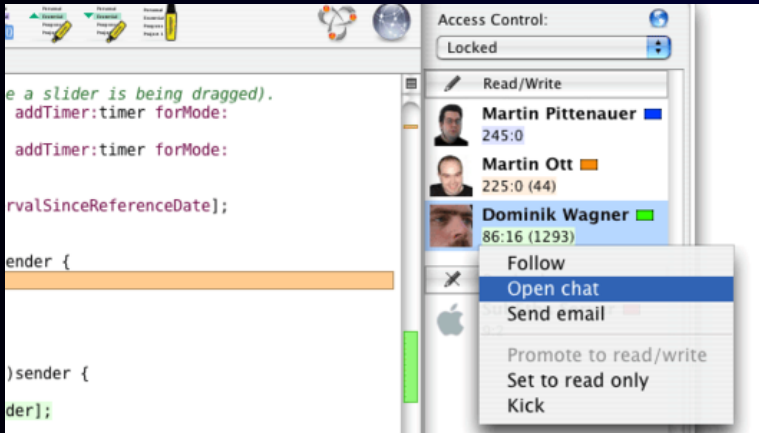


# Eingabe

- Vier Abstufungen:
  - Geteiltes Keyboard
  - Andere Teilnehmer sichtbar
  - Gruppenzeiger
  - Andere Teilnehmer nicht sichtbar

# Bewusstheit

- Was machen andere Teilnehmer? Sind sie da?



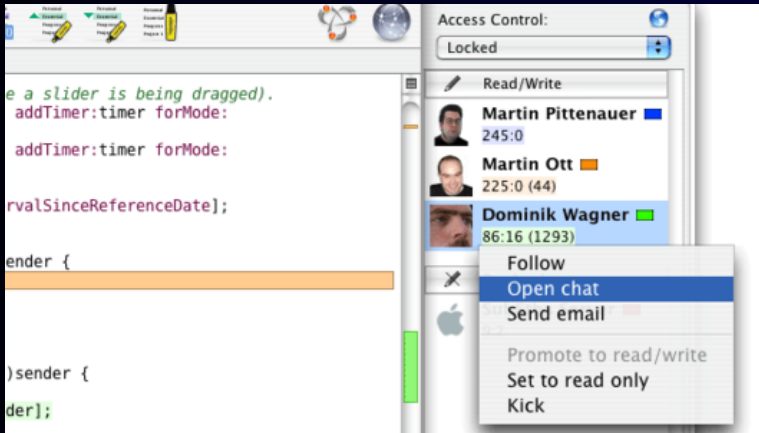
Quelle:

<http://en.wikipedia.org/wiki/Image:SubEthaEditExtremeProgramming.png>



# Bewusstheit

- Was machen andere Teilnehmer? Sind sie da?



Quelle:

<http://en.wikipedia.org/wiki/Image:SubEthaEditExtremeProgramming.png>

- Was passiert mit Objekten und wie?

- (Aktuell) (Vorherige) ○ 23:56, 17. Sep. 2006 Guidod (Diskussion | Beiträge) (→ *Die Rache der Rōnin* - voll übersetzt)
- (Aktuell) (Vorherige) ○ 22:15, 17. Sep. 2006 Guidod (Diskussion | Beiträge) (→ *Vorgeschichte* - übersetzt)
- (Aktuell) (Vorherige) ○ 14:38, 16. Sep. 2006 88.112.49.117 (Diskussion) (+fi)
- (Aktuell) (Vorherige) ○ 16:28, 31. Aug. 2006 Pacifier (Diskussion | Beiträge)
- (Aktuell) (Vorherige) ○ 15:19, 31. Aug. 2006 Asthma (Diskussion | Beiträge) K
- (Aktuell) (Vorherige) ○ 21:07, 14. Jun. 2006 YurikBot (Diskussion | Beiträge) K (Bot: Ergänze: fr:47 rōnin, it:Quarantasette Ronin)
- (Aktuell) (Vorherige) ○ 15:01, 12. Jun. 2006 Shikeishu (Diskussion | Beiträge) (→ *Verfilmung*)
- (Aktuell) (Vorherige) ○ 17:04, 21. Mai 2006 Dergreg: (Diskussion | Beiträge) K

Quelle: [http://de.wikipedia.org/w/index.php?title=Geschichte\\_der\\_47\\_R%C5%8Dnin&limit=500&action=history](http://de.wikipedia.org/w/index.php?title=Geschichte_der_47_R%C5%8Dnin&limit=500&action=history)

# Kapitel 4

## Probleme bei der Umsetzung synchroner Systeme

# Wozu dient dieses Kapitel?

- Der Bau von Groupware Applikationen birgt gewisse Schwierigkeiten
  - es hilft bekannte Fehlerquellen zu kennen
  - frühe Testphasen können bekannte Probleme offenbaren

# Wozu dient dieses Kapitel?

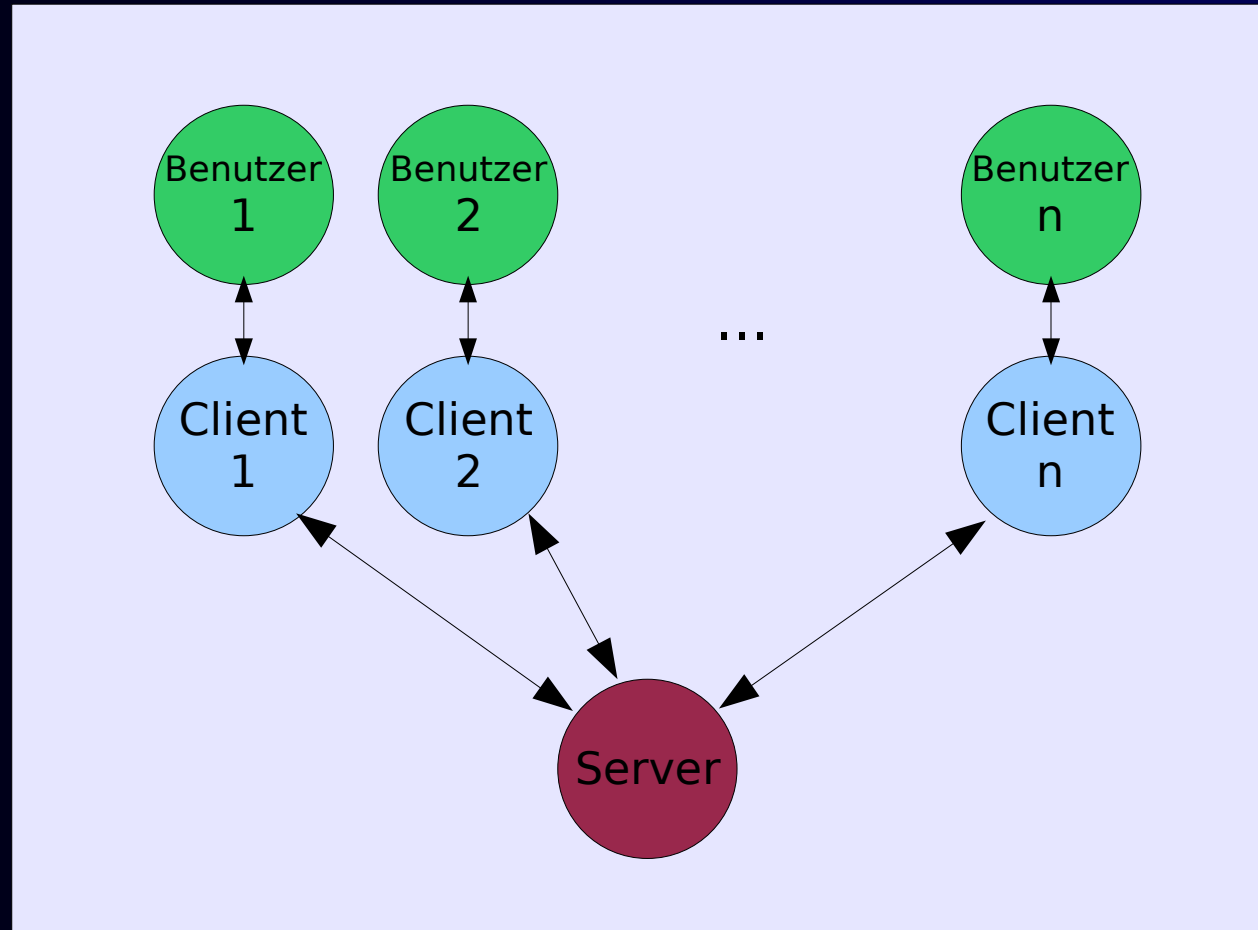
- Der Bau von Groupware Applikationen birgt gewisse Schwierigkeiten
  - es hilft bekannte Fehlerquellen zu kennen
  - frühe Testphasen können bekannte Probleme offenbaren
- technische Details können HCI Richtlinien blockieren
  - In der Design-Phase müssen auch solche Dinge berücksichtigt werden
  - gutes Design kann technische Grenzen umgehen

# Bekannte Probleme

- Viele Groupware Systeme arbeiten auf Basis von Netzwerken
- Wir unterscheiden nun zwischen
  - Probleme durch Nutzung der Netwerk-Architektur
  - äußere Fehler
  - Programmiertechnische Fehler
  - Unerwartete Ereignisse

# bekannte Architekturen

- Client/Server:

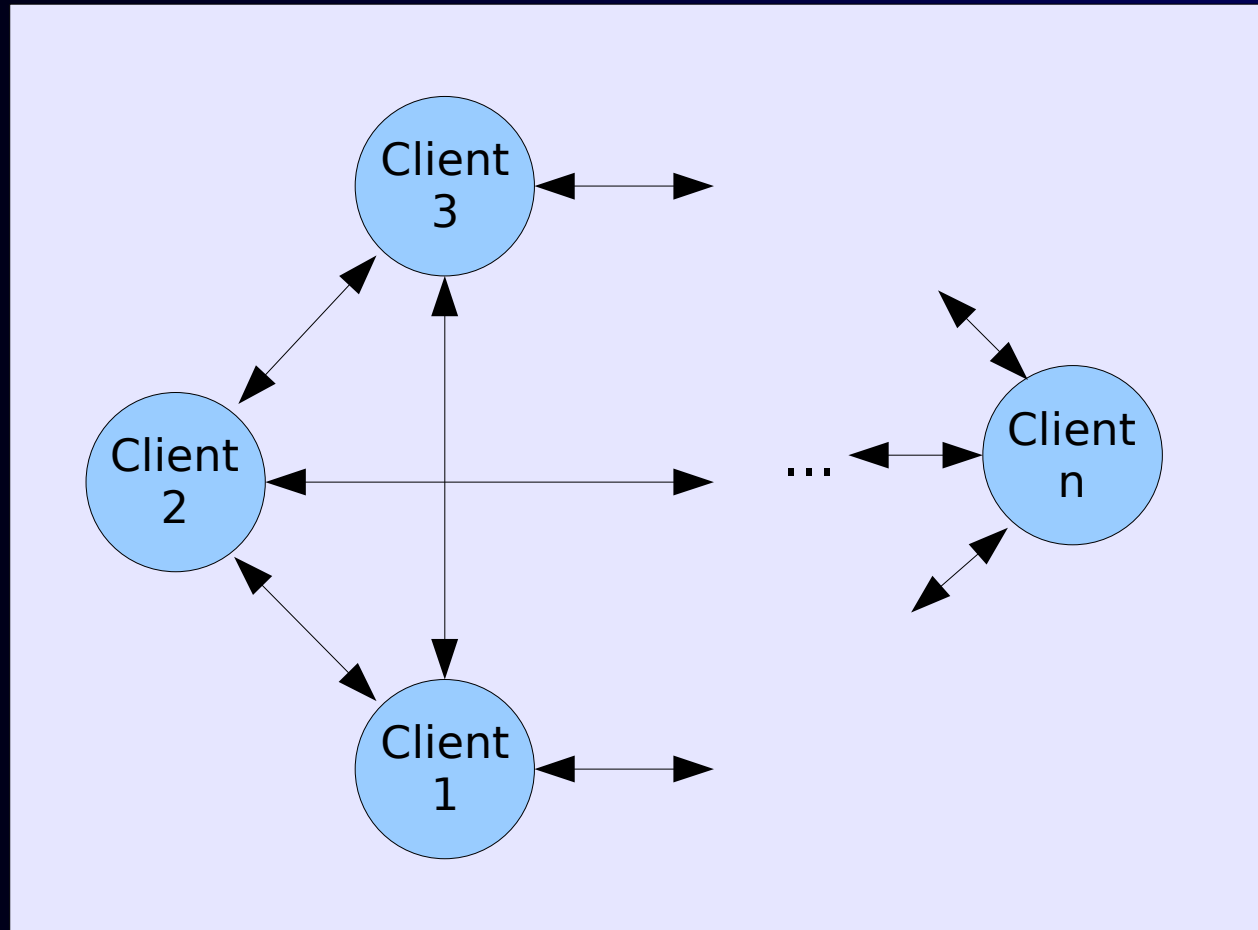


# Client/Server Pro und Contra

- Pro:
  - Datenabgleich sehr einfach, da nur auf einer Maschine wirklich etwas verändert wird
- Contra:
  - Aufgrund des Umwegs über den Server sehen Teilnehmer das Ergebnis ihrer Aktion verzögert
  - Wenn Server ausfällt kann niemand weiter arbeiten bis anderer Teilnehmer die Rolle übernimmt

# bekannte Architekturen

- Replicated:





# Replicated Pro und Contra

- Pro:
  - Datenveränderungen durch Teilnehmer sind direkt auf seiner Maschine sichtbar
  - also gutes Feedback
- Contra:
  - Bei Datenabgleich können durch gleichzeitige Veränderungen ungeahnte Fehler folgen

# Netzwerkbeschränkungen

- Programme kommunizieren häufig über textbasierte Nachrichten
- Große Nachrichten werden langsam verschickt und Verzögerungen können entstehen
- Lösung: Nachrichten komprimieren
- General Message Compression (GMC) kann Nachrichten um bis zu 90% verkleinern und kann mit minimalem Aufwand eingebunden werden

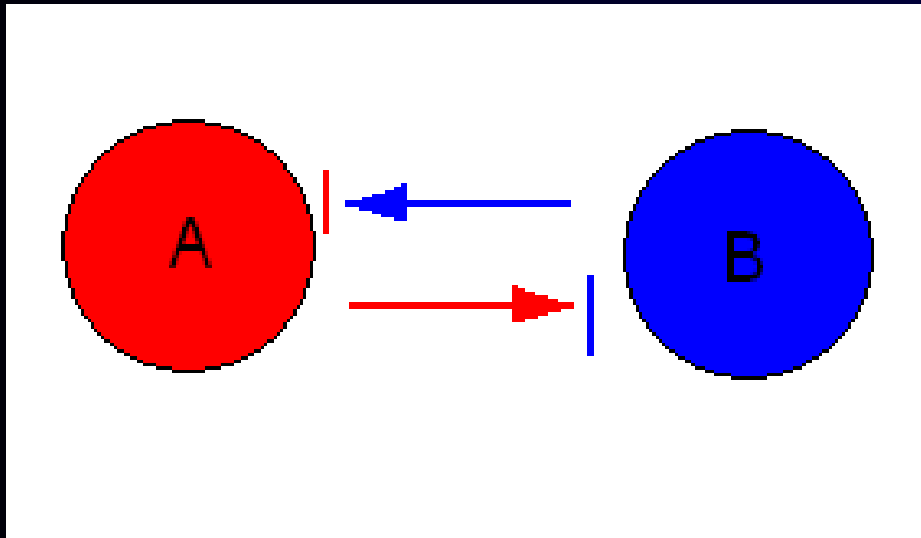
# Programmiertechnische Fehler

- Algorithmen können für Fehler verantwortlich sein
  - schlechtes Design
  - wenig getestet
  - ein großer Algorithmus anstatt kleinerer Einheiten

# Programmiertechnische Fehler

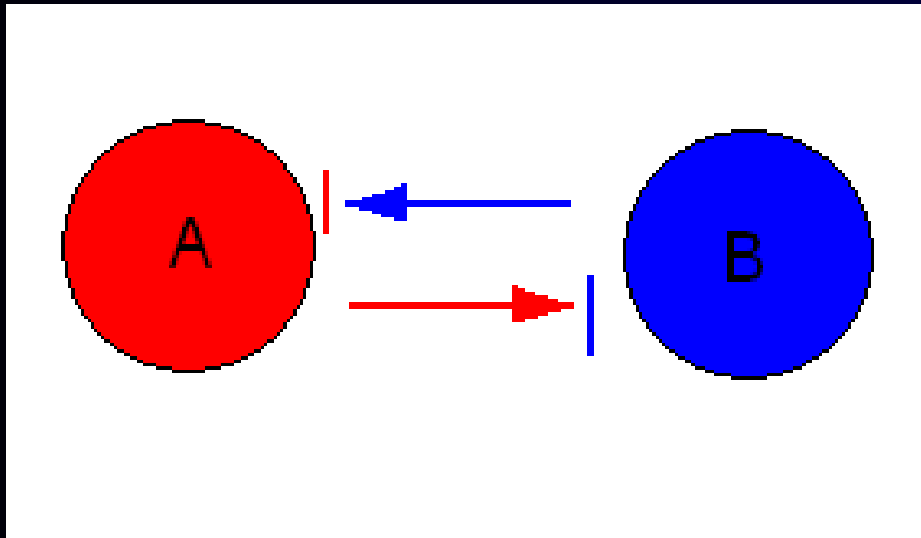
- Algorithmen können für Fehler verantwortlich sein
  - schlechtes Design
  - wenig getestet
  - ein großer Algorithmus anstatt kleinerer Einheiten
- Wie man dies umgehen könnte
  - Algorithmus formal verifizieren lassen
  - viele Testdurchläufe mit unterschiedlichen Eingaben
  - Fehler-Abfragen während der Laufzeit

# Unerwartete Ereignisse



- Beispiel Deadlock
- Sowohl A als auch B senden sich Nachrichten und warten auf Antwort
- Viele Fehler entsprechen diesem Muster in synchronen Anwendungen

# Unerwartete Ereignisse



- Beispiel Deadlock
- Sowohl A als auch B senden sich Nachrichten und warten auf Antwort
- Viele Fehler entsprechen diesem Muster in synchronen Anwendungen

- Lösungsmöglichkeiten
  - Nie Ein- bzw. Ausgabe blocken
  - Nutzung von Ereignissen, welche vom System behandelt werden können
  - Nie Annahmen aufstellen, sondern flexible Systeme gestalten

# Fehlerbeseitigung durch Tests

- frühzeitig regelmäßige Tests
  - nur augenscheinlich Mehrkosten: Langzeitfaktor bedenken
  - bei unausweichlichen Änderungen ist der Aufwand geringer
  - stabile Systeme haben höhere Nutzer-Akzeptanz
- nicht nur Simulieren
  - unerwartete Ereignisse meist nicht unter anderen Bedingungen auffindbar