

# **Computergestützte Gruppenarbeit**

## **9. Visualisierung semantischer Konflikte**

*Dr. Jürgen Vogel*

*European Media Laboratory (EML)  
Heidelberg*

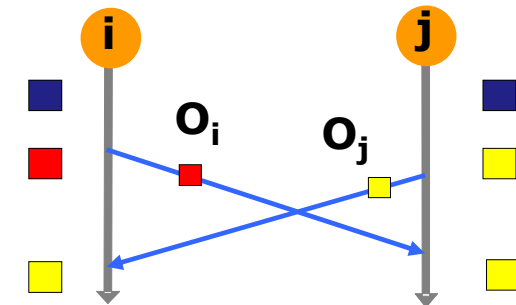
*SS 2006*

# 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

# Motivation

- falls Groupware nebenläufige Operationen zulässt, sind semantische Konflikte möglich
- Konsistenzerhaltungs-Mechanismen lösen solche Konflikte durch Gewährleistung syntaktischer Kriterien auf
- aber: der aus Benutzersicht wünschenswerte Zustand kann i.d.R. nicht automatisch bestimmt werden
- mögliche Probleme bei semantischen Konflikten
  - Effekt konfliktärer Operation wegen Überschreiben unsichtbar  
→ Benutzer j ist sich des Konflikts nicht bewusst
  - Artefakte bei Korrektur des Zustands  
→ z.B. springende Objekte bei Timewarp
  - korrigierter Zustand widerspricht Benutzerabsicht  
→ z.B. beim Serialisieren relativer Operationen



# Semantische Konflikte

- Wiederholung:  $O_{i,t^0_i,t^*_i}$  und  $O_{j,t^0_j,t^*_j}$  ( $i \neq j$ ) sind konfliktär, wenn sie dieselben Attribute eines Objekts modifizieren
- aus Benutzersicht sind solche Operationen vor allem dann kritisch, wenn sie zeitlich nahe beinander liegen:  $t_{cr} = |t^*_j - t^*_i|$
- $t_{cr}$  hängt von der Reaktionszeit des Benutzers ab, d.h., nicht notwendigerweise  $O_{i,t^0_i,t^*_i} \parallel O_{j,t^0_j,t^*_j}$
- kritisches Zeitfenster wird durch die Notification Time bestimmt:
  - synchrone Groupware:  $t^*_{\text{lokal}} \leq t^*_{\text{remote}}$  ( $\rightarrow$  Local Lag)
  - asynchrone Groupware:  $t^*_{\text{lokal}} \ll t^*_{\text{remote}}$
- automatische Entdeckung semantischer Konflikte: Analyse der Historie

# Umgang mit semantischen Konflikten

1. Verhindern: keine konfliktären Operationen
  - verbietet generell nebenläufige Schreibzugriffe (potentiell konfliktär)
2. Benutzer-Aktion: Nachfrage beim Benutzer bei Konflikt
  - erfordert manuelles Abstimm-Verfahren
  - kann störend wirken
3. Awareness: Visualisierung von Konflikten
  - stellt dem Benutzer manuellen Eingriff frei
  - kann störend wirken
  - kann übersehen werden

(2) und (3) sollten alle beteiligten Benutzer einbinden.

# Abgrenzung zur Intentions-Erhaltung

Wiederholung: Intentions-Erhaltung kann automatisch gewährleistet werden (z.B. Operations-Transformation)

Aber: automatische Intentions-Erhaltung funktioniert nicht für

- absolute Operationen → "entweder-oder-Zustand"
- semantisch widersprüchliche Operationen (z.B. Bedeutung einer Textpassage)

# Visualisierung semantischer Konflikte (1)

Ziel: Information des Benutzers über semantische Konflikte

- ➔ Benutzer sind sich Konflikten bewusst, können diese auflösen und zukünftige möglicherweise verhindern (Awareness)

Relevante Informationen

- Konfliktanzeige
- Welche Objekte sind betroffen? welche Aktionen? Position im gemeinsamen Arbeitsbereich?
- Welche Benutzer sind involviert? Was ist deren Absicht? Was die übergeordnete Aufgabe?
- Welche Aktionen wurden früher durchgeführt? Gibt es Zusammenhänge?
- Welche alternativen Zustände gibt es?

# Visualisierung semantischer Konflikte (2)

## Design-Optionen

1) Platzierung (wo?): innerhalb des Arbeitsbereichs vs. separat

2) Repräsentation (wie?)

- symbolisch (Icon) vs. deskriptiv
- graphisch vs. textuell
- statisch vs. animiert
- singulär vs. zusammenhängend
- Farbe, Form und Größe
- Abstufung (diskret oder stetig) und Ordnung (z.B. Größe)

3) Anlass (wann?) und Zeitdauer (wie lange?)

- automatische Anzeige vs. Benutzeranfrage
- permanent vs. temporär



# Objekt-Duplikation

## Prinzip

- erzeuge zusätzliche Objekt-Version, bei Entdeckung eines semantischen Konflikts
- Benutzer ist sich der Existenz verschiedener Versionen bewusst
- implizite oder explizite Konflikthanzeige

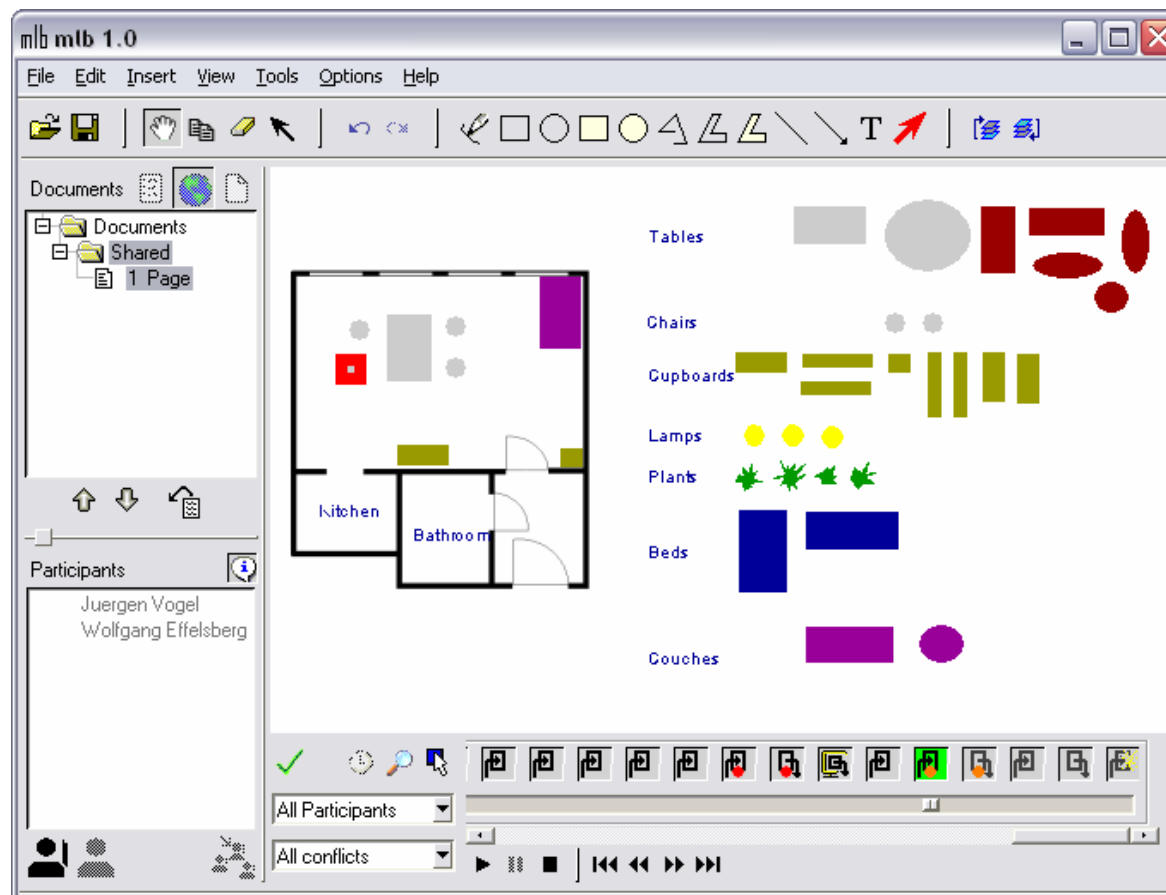
## Bewertung

- + gleichzeitige Visualisierung unterschiedlicher Sichtweisen
- keine Hilfsmittel zur Konfliktauflösung (z.B. Mergen von Versionen)
- bei vielen Versionen unübersichtlich
- kontinuierliche Anwendungen

# Visualisierung der Historie (1)

## Prinzip

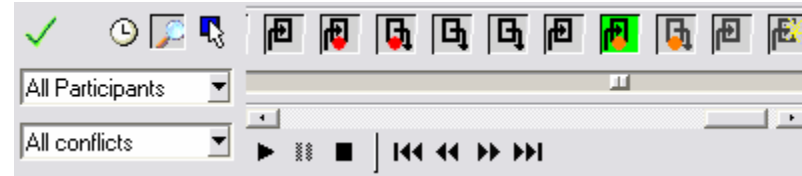
- Visualisierung der Operations-Historie zur Konfliktbehandlung: wer veränderte wann welches Objekt mit welchen konfliktären Operationen?



## Visualisierung der Historie (2)

### Darstellung der Historie

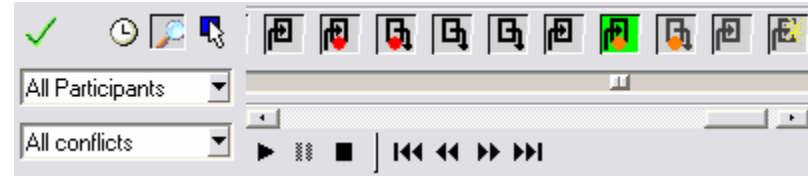
- Icon für jede Operation
- Ordnung nach der Ausführungszeit
- Icons: Form, Farbe und Hintergrundfarbe
  - lokale vs. empfangene Operation
  - konfliktäre Operation → Konfliktsequenzen
- Konfliktanzeige
  - letzter Konflikt
  - alle Konflikte
  - bestimmter Konflikt (Zeitpunkt)
- Verbindung Historie und gemeinsamer Arbeitsbereich
  - automatisches Update der Historie bei Aktionen
  - Selektion: Objekte und Operationen



## Visualisierung der Historie (3)

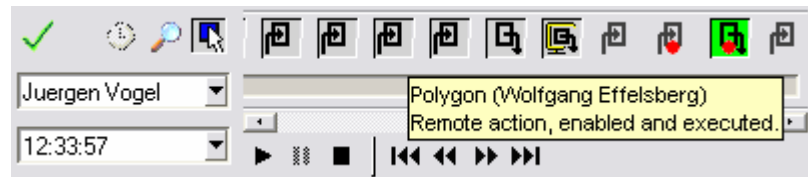
Erkunden der Zustandsentwicklung

- Abspielfunktion
  - Echtzeit vs. schnell
  - beliebiger Anfangszeitpunkt und Springen
- schneller Durchlauf per Slider
- Abspielen ist rein lokal



Erkunden alternativer Zustände

- Untersuchung der Frage: "Was wäre wenn?"
- Selektion eines bestimmten Benutzers
  - zeigt Absicht des selektierten Benutzers
- Ausschalten einzelner Operationen
  - erlaubt die Erzeugung eines alternativen Zustands
- Übernahme eines alternativen Zustands: lokal → allgemein gültig



## Visualisierung der Historie (4)

### Bewertung

- + schnelle Entdeckung von Konflikten
- + flexibel
- + keine erzwungene Benutzeraktionen
- + schnelle Rekapitulation der Historie (→ Slider)
- Lernaufwand für Icon-Bedeutungen
- Historie mit vielen Operationen unübersichtlich (trotz Zusammenfassung)
- zeitaufwendige Analyse bei komplexen Konflikten
- Ändern des Zustands kann zu neuen Konflikten führen

# Literaturhinweise

- Sun, C., Jia, X., Zhang, Y., Yang, Y., and Chen, D. *Achieving Convergence, Causality Preservation and Intention Preservation in Real-Time Cooperative Editing Systems*. In: ACM Transactions on Computer-Human Interaction, Vol. 5, No. 1, pages 63–108, 1998
- Sun, C. and Chen, D. *Consistency Maintenance in Real-Time Collaborative Editing Systems*. In: ACM Transactions on Computer-Human Interaction, Vol. 9, No. 1, pages 1–41, 2002. J.
- Tam, J. R. Change Awareness in 2D Graphical Workspaces. Master's thesis, Department of Computer Science, University of Calgary, Alberta, Canada, 2002.
- McCaffrey, L. *Representing Change in Persistent Groupware Environments*. Technical report, Grouplab Report, Department of Computer Science, University of Calgary, Canada, 1998.
- J. Vogel, Consistency Algorithms and Protocols for Distributed Interactive Applications, PhD Thesis, University of Mannheim, 2004