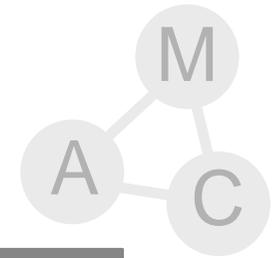


# Software-Ergonomie

---

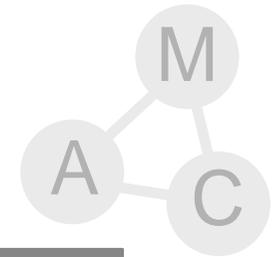


## Vorlesung 13

- ◆ **Dialoghistorie**
- ◆ **Undo**

# Dialoghistorie

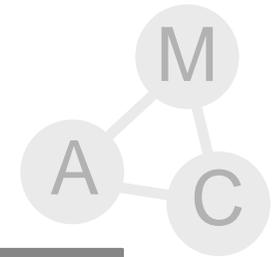
---



- Folge von Zuständen und/oder Benutzeraktionen
  - ◆ als Erinnerungshilfe (**Dialogprotokoll**)
  - ◆ zur Wiederholung von Aktionen (**Again-Funktion**)
  - ◆ zur Definition von **Makros**
  - ◆ um Aktionen zu stornieren (Undo-Funktionen)
  - ◆ um stornierte Aktionen wiederauszuführen (**Redo-Funktion**)
  - ◆ zur Analyse der Aktionen für **aktive Hilfe**
  - ◆ zur Analyse für **Individualisierung**

# History-Komponente

---

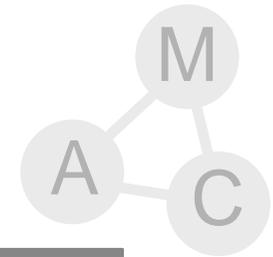


bestehend aus:

- ◆ Dialogprotokollierung
- ◆ Again-Funktion
- ◆ Makros
- ◆ Undo-Funktion
- ◆ Redo-Funktion

# Dialogprotokoll

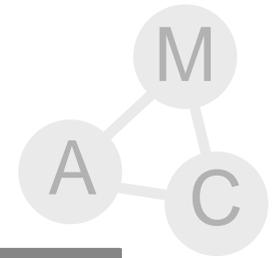
---



Chronologische Auflistung der Benutzeraktionen

Zweck:

- ◆ Kontrolle der ausgeführten Aktionen
- ◆ Rückverfolgung von Fehlern
- ◆ Planung des weiteren Dialogs
- ◆ rechtliche Referenz (Journal, Logbuch)



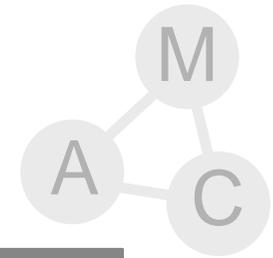
# Ebene der Protokollierung

---

- ◆ **Semantisch:** objektbezogene Operationen im Anwendungsbereich (Bsp: Lösche Datei xy)
- ◆ **Syntaktisch:** von der Interaktionsform abhängige Aktionen (Bsp: Selektiere Widget 345)
- ◆ **Lexikalisch:** Tastatur-, Mausaktionen mit Angabe der absoluten Bildschirmposition (Bsp: Drücke linken Mausbutton an Position 200,350)

# Bsp. für Dialogprotokoll in UNIX

---



> **history**

```
1 12:12 pwd
2 12:12 cd SoftErg98
3 12:12 emacs Vorlesung13.txt
4 13:07 cp Vorlesung13.txt /backup
5 13:08 print Vorlesung13.txt
```

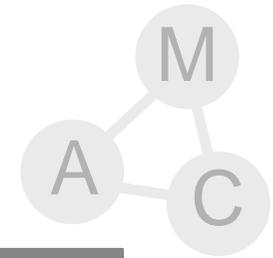
Again-Funktion:

> **!3**

Starting emacs...

# Again-Funktion

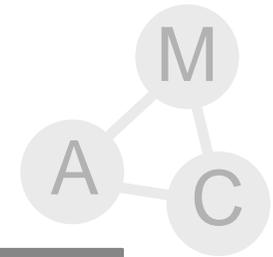
---



- ◆ Wiederholung von Aktionen
  - unverändert
  - geringfügig modifiziert
- ◆ Vorteil: Erspart unnötigen Interaktionsaufwand, vor allem bei sprachlichen, deskriptiven Interaktionsformen

# Makros

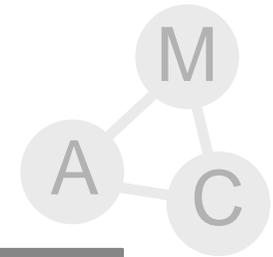
---



- ◆ Zusammenfassung einer Aktionssequenz aus Dialogprotokoll, meist mit Namen versehen
- ◆ Vorteil: Erspart unnötigen Interaktionsaufwand
- ◆ Erweiterungen:
  - Parametrisierung (Zulassen von Argumenten)
  - Kontrollstrukturen wie bedingte Ausdrücke, Schleifen
- ◆ Setzt je nach Ausprägung Vorkenntnisse der Benutzer voraus
- ◆ Bsp: Keyboard-Makros im Emacs-Editor, Skript-Sprachen wie WordBasic, Visual Basic

# Undo

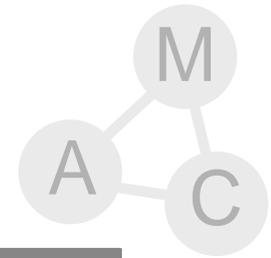
---



Undo = Zurückkehren in einen Systemzustand der Vergangenheit

Gründe für Undo:

- ◆ unbeabsichtigte Aktionen (Typos etc.)
- ◆ nachträglich als ungünstig, nutzlos oder falsch erkannte Aktionen
- ◆ zur Unterstützung explorativen Erlernens und explorativer Arbeitsweise



# Undo-Funktion

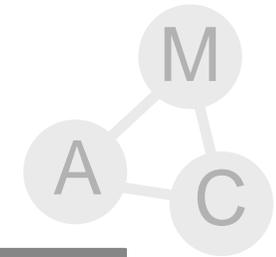
---

Unterscheide:

- ◆ Undo (Benutzerabsicht)
- ◆ Undo-Funktion (Systemfunktion)

Neben der Undo-Funktionen können auch **inverse Funktionen** die Benutzerabsicht Undo unterstützen!

Bsp: Backspace-Taste, um soeben geschriebenen Text zu korrigieren.

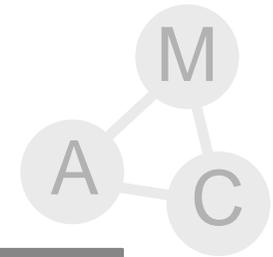


# Undo-Techniken

---

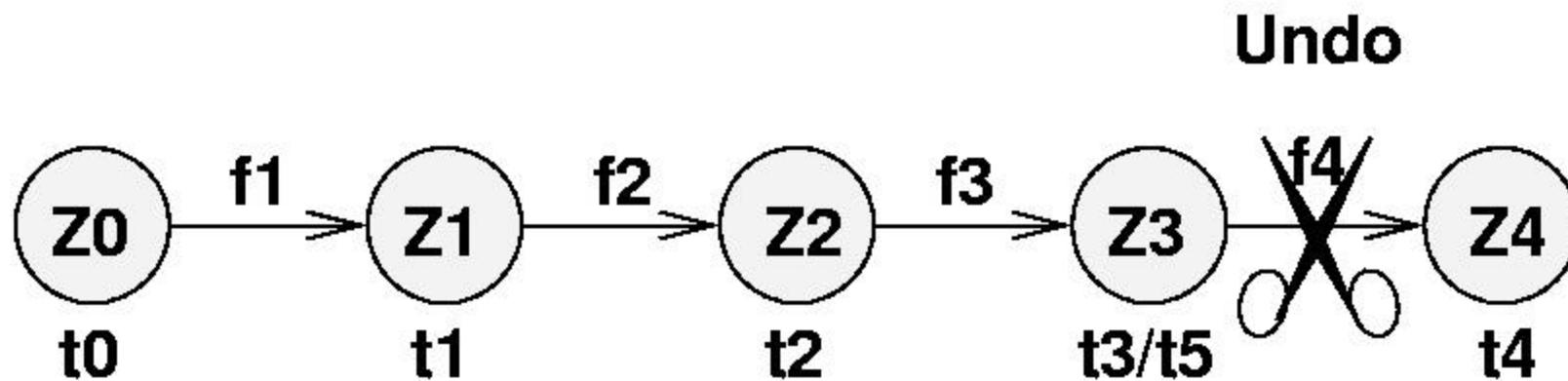
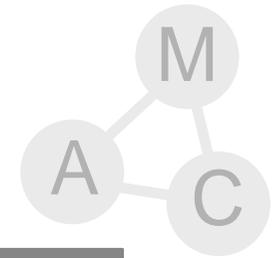
- ◆ technische Realisierung einer Undo-Funktion
  - anwendungsunabhängig
  - anwendungsabhängig
- ◆ setzt voraus:
  - Dialogprotokoll
  - evtl. zusätzliche Speicherung alter Zustände
  - Verfahren zur Neutralisierung von Funktionsausführungen (inverse Funktionen oder Restaurierungsfunktionen)
- ◆ Unterstützung des Benutzers bei der Auswahl des gewünschten Zustands

# Typen von Undo-Funktionen

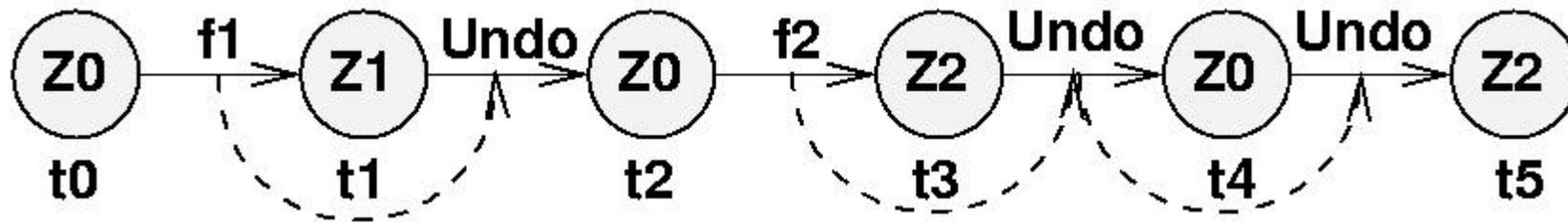
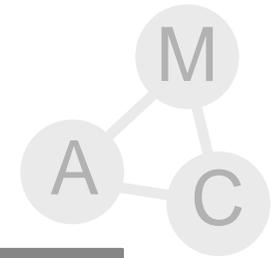


Undo-Form	Mehrstufigkeit	Form der Dialoghistorie	Operationen auf Historie	Selektivität
<b>Retract-Undo</b>	mehrstufig	linear	verkürzen	zeitlich
<b>Recall-Undo</b>	einstufig	linear	erweitern	zeitlich
<b>History-Undo</b>	mehrstufig	linear	erweitern	zeitlich
<b>Lineares Undo</b>	mehrstufig	linear	navigieren, verkürzen	zeitlich
<b>Travel-Undo</b>	mehrstufig	baumförmig	navigieren	zeitlich
<b>Checkpoint-Undo</b>		linear oder baumförmig	verkürzend	
<b>Direktes Undo</b>	einstufig	linear	modifizieren o. erweitern	selektiv

# Retract-Undo



# Recall-Undo



# History-Undo

