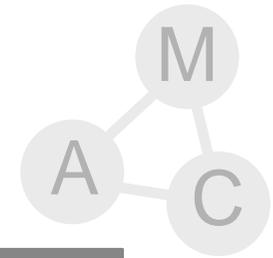
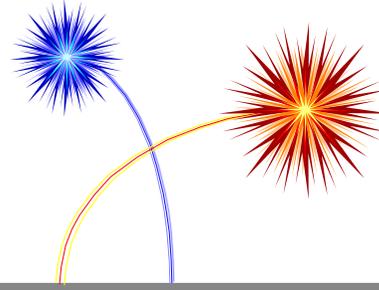


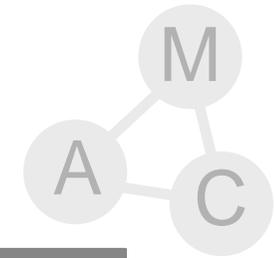
# Software-Ergonomie



## Vorlesung 11

- ∪ **Ein gutes, erfolgreiches neues Jahr 1998**
- ∪ Wiederholung
- ∪ Individualisierung - Adaptierbarkeit & Adaptivität

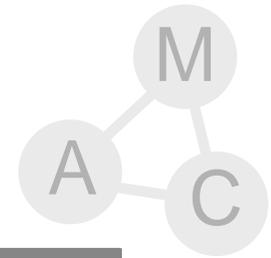




# Wiederholung I

---

- ∪ **Direkte Manipulation** (Shneiderman): ständige Visualisierung, physische Aktionen, schnelle, inkrementelle reversible Aktionen
- ∪ Direktheit vs. Distanz (Ausführungsß, Auswertungskluft), Einbezogenheit (Weltmodell)
- ∪ leichte Erlernbarkeit, geringer Fehleranfälligkeit, rasche Regulation durch sofortige Rückmeldung, Stornierbarkeit, Benutzer behält die Kontrolle
- ∪ problematisch: viele Objekte, unsichtbare Obj., mehrdeutig, nicht ausdrückstark



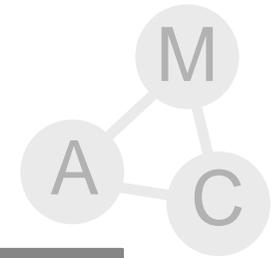
# Wiederholung II

---

- ∩ **Multiple Aktivitäten:** Linearisierung paralleler Aktivitäten, Wechsel der Arbeitskontexte und Werkzeuge, Aktivitätenverwaltung, Informationsaustausch
- ∩ Aktivitätenübersicht, Fenstersysteme, virtuelle Desktops, Split-Screen-Technik (*panes*), generische Werkzeuge

# Wiederholung III

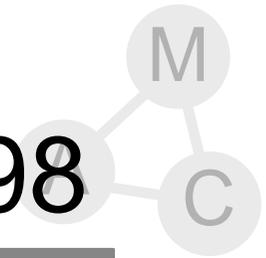
---



- ∪ **Zeitverhalten interaktiver Systeme:**  
kurze Antwortzeit wegen Flüchtigkeit des KZG,  
Antwortzeit ~ Denkzeit  
aber: Zu schnelle Reaktion kann zu Streß führen
- ∪ **Ausgabezeit:**  
Ausgaberate bei Texten beeinflußt Lese-, Denk-  
und Arbeitstempo
- ∪ Anpassung an Benutzererfahrung und Aufgabe  
notwendig

# Software-Ergonomie: Ausblick 1998

---

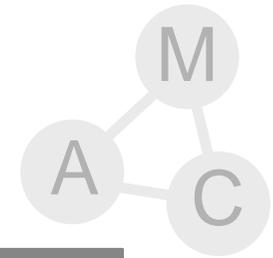


Ausgewählte Themen aus:

- ∪ **Individualisierung** (*adaptiv vs. adaptierbar*)
- ∪ **Gestaltungsgrundsätze** für interaktive Systeme
- ∪ **Dialoghistorie** und **Undo**
- ∪ Spezielle Aspekte von **Mehrbenutzersystemen** (Groupware)
- ∪ **Evaluation** (Benutzertests)

# Individualisierung

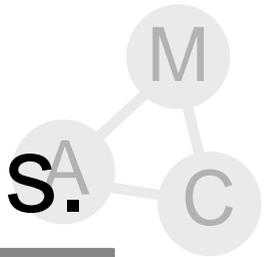
---



**Anpassung** von Benutzungsschnittstellen an **individuellen Benutzer** (Kenntnisse, Fähigkeiten, Präferenzen) und bestimmte **Arbeitssituation** (Aufgabe, Randbedingungen)

- ∪ **Adaptierbarkeit:** Anpassung durch Benutzer
- ∪ **Adaptivität:** automatische Anpassung durch Computer

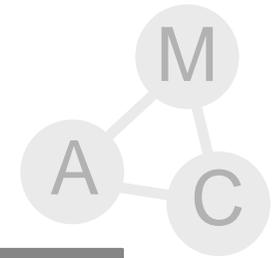
# Gestaltungsebenen der Individualis.



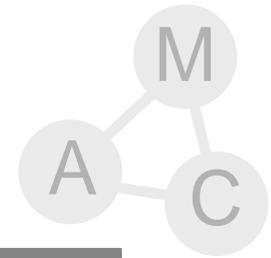
Ebene	Veränderung von	Beispiele
Intentional	Aufgabenstruktur	organisatorische Abläufe (nicht computerbezogen)
Pragmatisch	Prozeduren	Definieren von Aktionssequenzen, Gruppieren von Desktop-Werkzeugen
Semantisch	Arbeitsobjekte	Festlegung von Objekt-Voreinstellungen
Syntaktisch	Interaktionsform	Entscheidung für Interaktionsform, Festlegung von Datenformaten
Lexikalisch	Informationscodierung	Auswahl von Farben, Fonts, Positionen, Namen
Sensomotorisch	Eigenschaften der Ein- & Ausgabegeräte	Anpassung von Bildschirmhelligkeit, Kontrast, Tastenbelegung, Mausgeschwindigkeit

# Verfahren zur Individualisierung

---



- ∪ Physikalische Anpassungen
- ∪ Benutzerprofile
- ∪ Voreinstellungen
- ∪ Constraints
- ∪ Makros
  
- ∪ Adaptivität



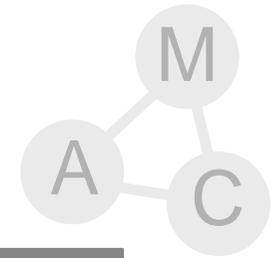
# Physikalische Anpassungen

---

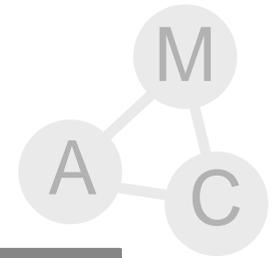
- ∪ Anpassung des **Bildschirmarbeitsplatzes**  
(Arbeitsstuhl, -tisch, Positionierung von Computerperipherie)
- ∪ Anpassung der **Bildschirmcharakteristik**  
(Helligkeit, Kontrast) - wichtig: über Drehregler an der Front
- ∪ Anpassung der **Lautstärke**

# Benutzerprofile

---



- ⌞ Anpassungen auf sensomotorischer und lexikalischer Ebene
- ⌞ meist über Initialisierungsdatei, die beim Systemstart gelesen wird  
Bsp: `.Xdefaults`
- ⌞ Optional und wünschenswert: interaktive Anpassungsprogramme



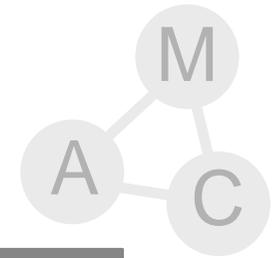
# Voreinstellungen

---

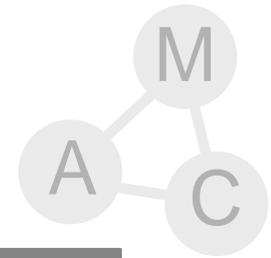
- ∩ auf der semantischen Ebene
- ∩ Festlegung von Voreinstellungen (Defaults) von Objektattributen  
Bsp: Vorbelegung von Formularfeldern
- ∩ Vordefinierte Parameter für Operationen  
Bsp: `alias cprint lp -ddiacolor`

# Constraints

---



- ∪ auf semantischer und pragmatischer Ebene
- ∪ legt Beziehungen zwischen Arbeitsobjekten fest  
Bsp: Tabellenkalkulation (*one-way*)
- ∪ Benutzer kann sich eigene Anwendung bauen
- ∪ einfachste Art der **Endbenutzerprogrammierung**
- ∪ Bsp. (*two-way constraint*):  
$$\text{Temperatur-in-F} = 32 + 9/5 * \text{Temperatur-in-C}$$



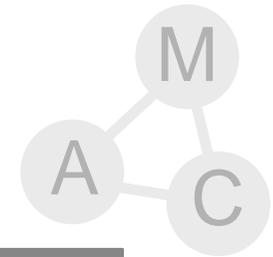
# Makros

---

- ∪ auf **pragmatischer** Ebene
- ∪ Zusammenfassung von immer wiederkehrenden **Aktionssequenzen**  
Bsp: Keyboard-Macros im Emacs-Editor
- ∪ **Skriptsprachen** enthalten zusätzlich **Kontrollstrukturen**  
Bsp: Shells unter UNIX, Bat-Dateien unter DOS

# Adaptivität

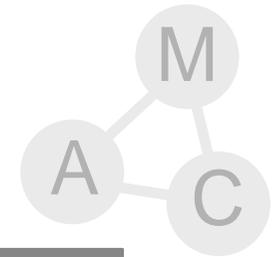
---



- ∪ **adaptive Benutzungsschnittstellen:** Computer nimmt dem Benutzer die Adaptierung ab
- ∪ Computersystem benötigt hierzu **Wissen über die Gestaltungsdimensionen** und deren Wirkung, also Wissen von sich selbst:  $S(S(A))$
- ∪ **Explizites Wissen über Anwendungsbereich:**  $S(A)$
- ∪ **Explizites Wissen über Benutzer:**  $S(B(A))$

# Formen von Adaptivität

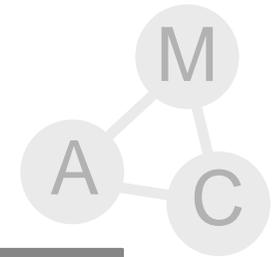
---



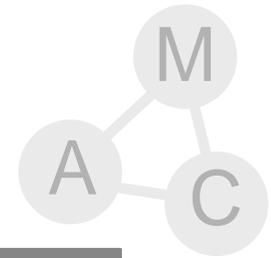
- ∪ aktive Hilfesysteme
- ∪ automatische Rechtschreibkorrektur
- ∪ Sortierung von Menüeinträgen (sinnvoll?)
- ∪ Anpassung von Menühierarchien (sinnvoll?)
- ∪ Anpassung von Helligkeit, Kontrast an Umfeld
- ∪ Reduktion der Skalierung von Mausbewegungen
- ∪ Verlängerung der Zeitintervalle zwischen Doppelklicks
- ∪ Anpassung von Sound an Umgebung

# Probleme adaptiver Systeme

---



- ∪ **verletzen Konsistenz** der Benutzungsschnittstelle (Grundsatz der Erwartungskonformität)  
Ausweg: Benutzer informieren, fragen
- ∪ **erschweren Erlernbarkeit**  
Ausweg: kurze adaptive Phase zu initialen groben Anpassung, gefolgt von Phase der Adaptierbarkeit (Umschaltung benutzer- oder systemgesteuert).
- ∪ aufwendige Spezifikation, Implementierung (Benutzer-, Anwendungsmodelle)



# Stufen der Individualisierung

---

Steigende Anforderung an Fähigkeiten der Benutzer:

- v Voreinstellungen → Benutzerprofile
- v Tabellenkalkulation → Constraints
- v Makros → Skriptsprachen  
→ Programmiersprachen

Wechsel der Gestaltungsart (Vorgaben, Constraints, formale Sprache) bedingt Sprünge, die nahtlosen Übergang erschweren