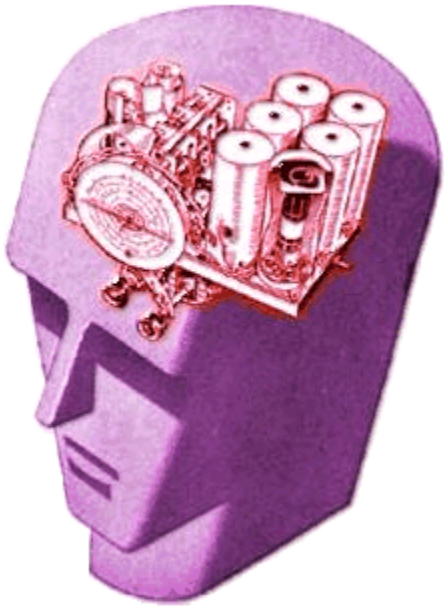
Kognitive Modelle



Claas Oppitz, Jan-Martin Pulwitt

RWTH AACHEN
RHEINISCH-WESTFÄLISCHE TECHNISCHE HOCHSCHULE AACHEN

Media Computing Group

Kognitive Modelle

- stammen aus der Psychologie
- werden für HCI verwendet
- beschreiben Interaktion mit der Benutzerschnittstelle
  - z.B. Benutzereingaben

Der Nutzen?

- kognitiven Aufwand quantifizieren
- zeitliche Abschätzungen

→ Effektivität von Benutzerschnittstellen bewerten

- **vor** der Programmierung

→ Geld, Zeit und Nerven sparen

Gliederung

- Hierarchische Modelle
  Aufgaben- und Ziel-Modellierung
- Linguistische Modelle
  linguistische Modellierung der Interaktion
- Modelle der Physis
  Modelle zur Beschreibung physischer Handlungen
- Kognitive Architekturen
  Grundannahmen zur kognitiven Funktionsweise

GOMS

- **G**oals, **O**perators, **M**ethods, **S**election

- Card, Moran, Newell (1983)

- Leistung bemessen

  Zeit zur Ausführung einer Aufgabe

GOMS: Funktionsweise

- **G**oal

  Was der Benutzer erreichen will

- **O**perators

  Basisaktionen

- **M**ethods

  Möglichkeiten, eine Aktion auszuführen

- **S**election

  Konkrete Auswahl der Methode

GOMS: Ziele

- Hauptziel
- Unterziel
  - evaluierbarer Teilerfolg
  - Ausgangspunkt bei Fehlern

→ Ziel und Unterziel Hierarchie

Verschachtelungstiefe:

Beanspruchung des Kurzzeitgedächtnisses

GOMS: Operatoren

Beispiele: Textbox lesen

x-Taste drücken

Konkretisierung variiert, z.B.

- Menüpunkt auswählen
- Maus über Schaltfläche bewegen und linke Maustaste drücken

GOMS: Methoden

verschiedene Umsetzungen einer Operation

Bsp: Schaltfläche oder Tastenkürzel

GOMS: Auswahl

Festlegung der Methode

Festlegung von Auswahlregeln

Beispiel

Wenn möglich Tastenkürzel

GOMS: Beispiel

Ziel: Fenster schließen

. [Wahl Ziel: Benutze Schließen-Methode

. . Bewege Mauszeiger auf Titelleiste

. . Pop-auf-Menu

. . Klicke über Schließen Option

Ziel: Benutze Alt-F4 Methode

. . Drücke Alt-F4]

GOMS: Fazit

Sehr rege benutzt

-> NGOMSL, CPM-GOMS, KL-GOMS

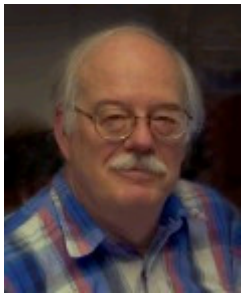
GOMS spart Geld!

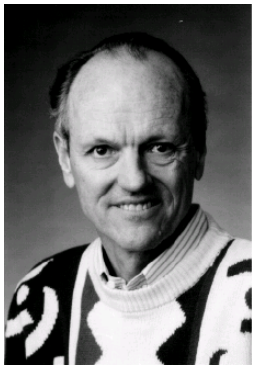
Geht von Expertenbenutzer aus

kein Augenmerk auf Training oder Transfer

Cognitive Complexity Theory (CCT)

- David Kieras und Peter Polson (1985)
- Ziel: Präzisere und umfangreichere Vorhersage

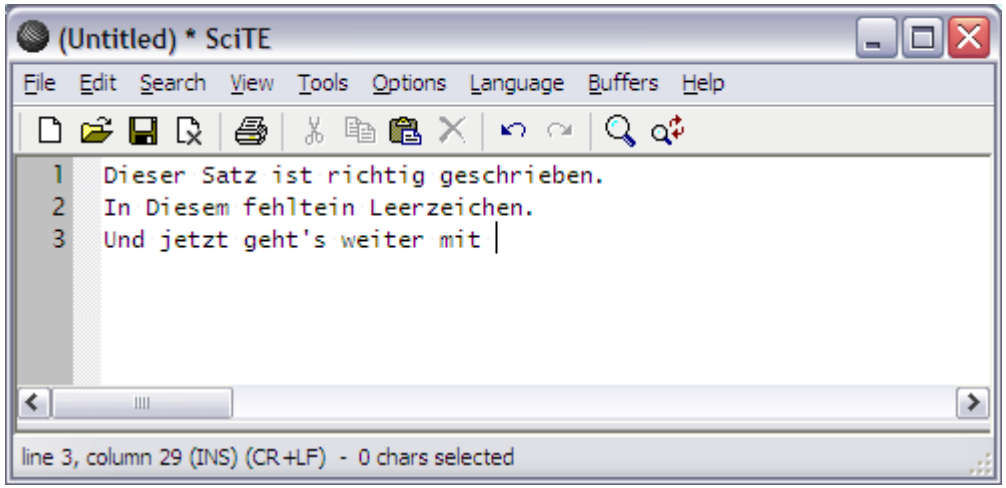




CCT: Funktionsweise

- Ziel- und Unterziel-Hierarchie wie bei GOMS
- Handlungen durch Produktionsregeln modelliert
- Produktionsregeln entsprechen IF-Klauseln
  - **if** Bedingung **then** Aktion

CCT Beispiel: Leerzeichen einfügen



Modellstatus:

(*GOAL* Text schreiben)

(*TEXT* Aufgabe: Leerzeichen einfügen)

(*TEXT* Aufgabe ist bei Zeile 2, Zeichen 16)

(*CURSOR* 3 29)

CCT Beispiel: Leerzeichen einfügen

Modellstatus:

(*GOAL* Text schreiben)
(*TEXT* Aufgabe: Leerzeichen einfügen)
(*TEXT* Aufgabe ist bei Zeile 2, Zeichen 16)
(*CURSOR* 3 29)

- Produktionsregeln durchlaufen
- „passende“ ausführen

Produktionsregeln:

…
(ZIELSETZUNG_LEERZEICHEN_EINFÜGEN
IF (*AND* (*TEST-GOAL* Text schreiben)
(*TEST-TEXT* Aufgabe: Leerzeichen einfügen)
(*NOT* (*TEST-GOAL* Leerzeichen einfügen))
(*NOT* (*TEST-NOTE* bearbeite
“Leerzeichen einfügen”))))
THEN (…)
…

CCT Beispiel: Leerzeichen einfügen

Produktionsregeln:

…

(ZIELSETZUNG_LEERZEICHEN_EINFÜGEN

IF (…)

THEN ((*ADD-GOAL* Leerzeichen einfügen)

(*ADD-NOTE* bearbeite “Leerzeichen einfügen”)

(*LOOK-TEXT* Aufgabe ist bei Zeile *%LINE*, Zeichen *%COLUMN*)))

…

Neuer Modellstatus:

(*GOAL* Text schreiben)

(*TEXT* Aufgabe: Leerzeichen einfügen)

(*TEXT* Aufgabe ist bei Zeile 2, Zeichen 16)

(*NOTE* bearbeite “Leerzeichen einfügen”)

(*GOAL* Leerzeichen einfügen)

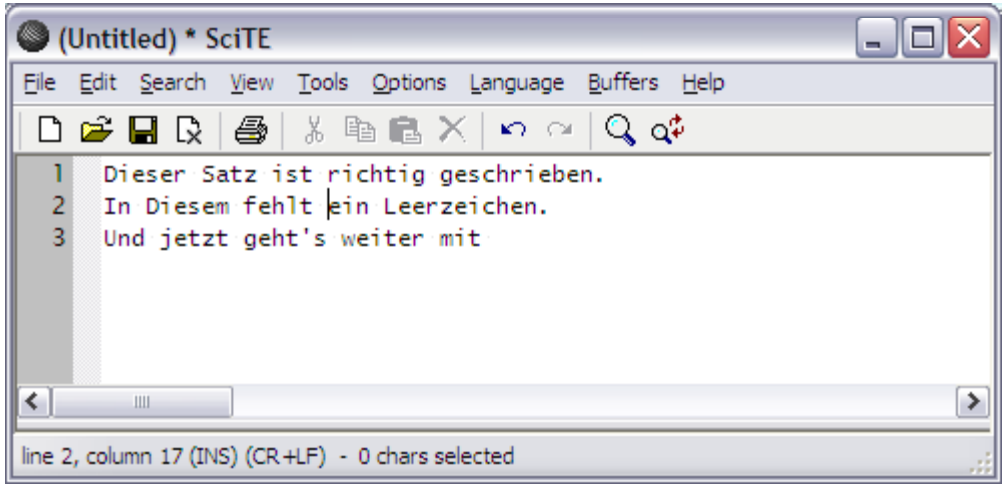
(*LINE* 2)

(*COL* 16)

(*CURSOR* 3 29)

CCT Beispiel: Leerzeichen einfügen

- Nächste Schritte:
  BRINGE_CURSOR_AN_POSITION
  FUEGE_LEERZEICHEIN_EIN

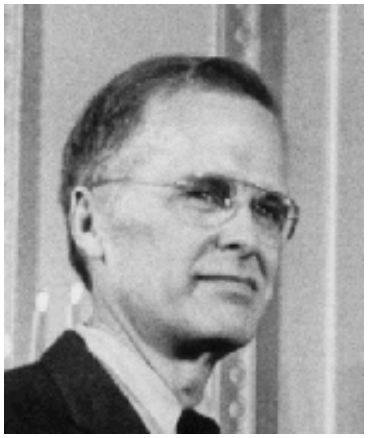


CCT: Nutzen

- Lernaufwand abschätzen
- kann Fehlersituationen simulieren
- gleichzeitige Handlungsstränge modellierbar
- Unstimmigkeit mit Benutzerschnittstellen-Modell kann Probleme aufdecken

Linguistische Modelle

Backus-Naur-Form



- John Backus, Peter Naur (1959)
- Phyllis Reisner (HCI)

- Dialog Grammatik beschreiben
  - Syntax ohne Semantik

BNF - Aufbau

- Herunterbrechen von Aktionen bis zur Ebene von TERMINALEN (= Benutzeraktionen)

- Form

Name ::= ausdruck

ausdruck kann „UND“ – „+“

und/oder „ODER“ – „|“ enthalten

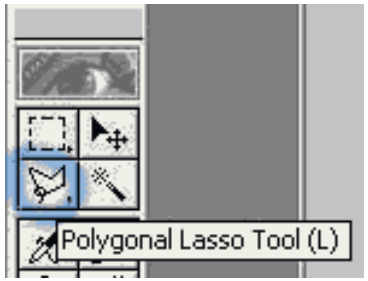
BNF - Beispiel

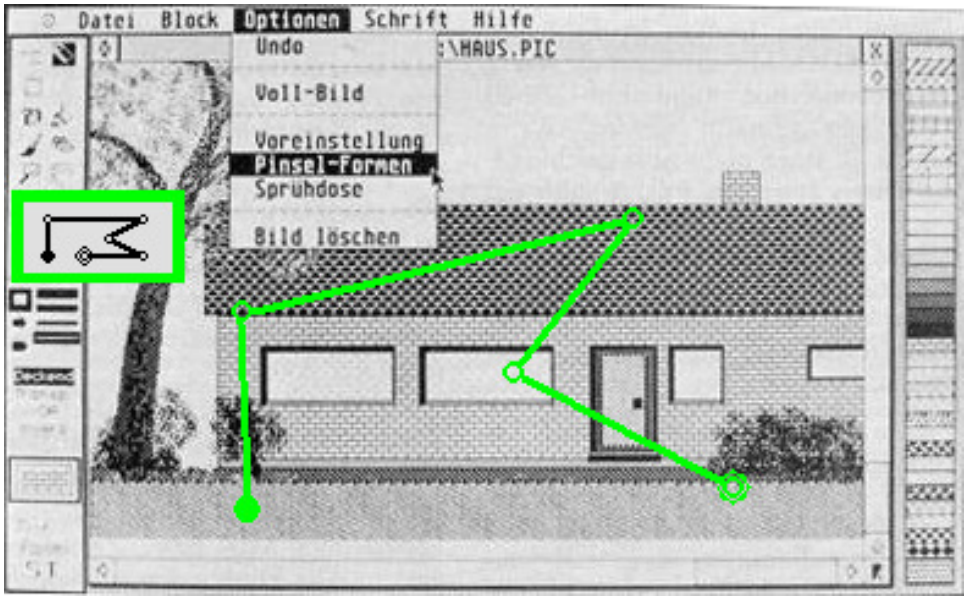
Linien-Zeichnen-Funktion eines Grafikprogramms

Start:Mausklick

Punkt:Mausklick

Endpunkt:Doppelklick





BNF - Beispiel

zeichne-linie ::= wähle-linie + wähle-punkte + letzter-punkt

wähle-linie ::= positioniere-maus + KLICK-MAUS

wähle-punkte ::= wähle-einen | wähle-einen + wähle-punkte

wähle-einen ::= positioniere-maus + KLICK-MAUS

letzter-punkt ::= positioniere-maus + DOPPELKLICK-MAUS

positioniere-maus ::= leer | BEWEGE-MAUS + positioniere-maus

BNF - Analysemöglichkeiten

- Anzahl der Regeln = Komplexität des Systems

- Aber Verschachtelungstiefe willkürlich

→ Anzahl von Operatoren (= + |)

- Anzahl benötigter Basisaktionen

BNF - Erweiterung

Nur Benutzer Aktionen

- Keine Wahrnehmung (und deren Probleme)

Reisners Ansatz: „Information-suchen Aktionen“

Task Action Grammar (TAG)

- Stephen Payne, Thomas Green, 1986
- Erweiterung der BNF
- Ziel: kognitiven Aufwand präziser schätzen

→ Konsistenz berücksichtigen

→ Benutzerwissen einbeziehen

TAG: Funktionsweise

- Grundlage ist BNF

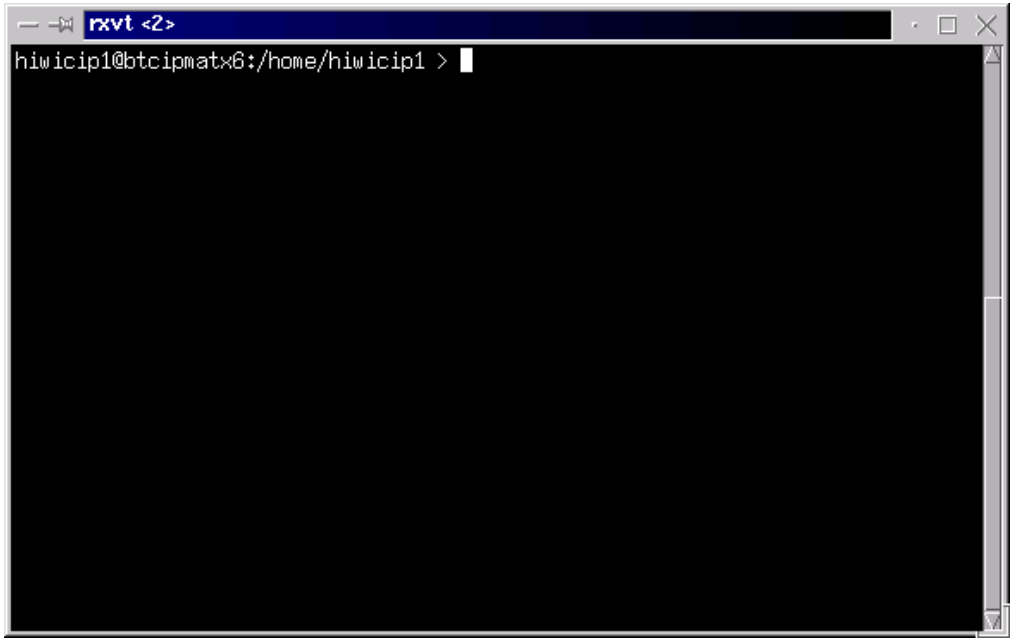
- Beschreibung des Benutzerwissens

→ zusätzliche Schlüsselwörter

- Konsistenz modellieren

→ Erweiterung der Nichtterminale um Parameter

TAG Beispiel: Konsistenz der Shell



Shell-Aktionen *Kopieren, Verschieben* und *Verknüpfen*

TAG Beispiel: Konsistenz der Shell

- Beschreibung von *Kopieren, Verschieben* und *Verknüpfen* **in BNF**:

Kopieren ::= cp + *Dateiname* + *Dateiname* | cp + *Dateinamen* + *Verzeichnis*
Verschieben ::= mv + *Dateiname* + *Dateiname* | mv + *Dateinamen* + *Verzeichnis*
Verknüpfen **::= ln +** ***Dateiname*** **+** ***Dateiname*** **|** **ln +** ***Dateinamen*** **+** ***Verzeichnis***

- Problem: inkonsistente Syntax **in BNF** gleichwertig

Kopieren ::= cp + *Dateiname* + *Dateiname* | cp + *Dateinamen*+ *Verzeichnis*
Verschieben ::= mv + *Dateiname* + *Dateiname* | mv + *Dateinamen* + *Verzeichnis*
Verknüpfen **::= ln +** ***Dateiname*** **+** ***Dateiname*** **|** **ln +** ***Verzeichnis*** **+** ***Dateinamen***

TAG Beispiel: Konsistenz der Shell

BNF:

Kopieren, Verschieben, Verknüpfen

TAG:

*Datei-Operation[**Kopieren**],*
*Datei-Operation[**Verschieben**],*
*Datei-Operation[**Verknüpfen**]*

Definition von **Parametern** im Detail:

Datei-Operation[**Aktion**] ::=*Befehl*[**Aktion**] + *Dateiname* + *Dateiname*
| *Befehl*[**Aktion**] + *Dateinamen* + *Verzeichnis*

Befehl[**Aktion= Kopieren**] ::= cp
Befehl[**Aktion = Verschieben**] ::= mv
Befehl[**Aktion = Verknüpfen**] ::= ln

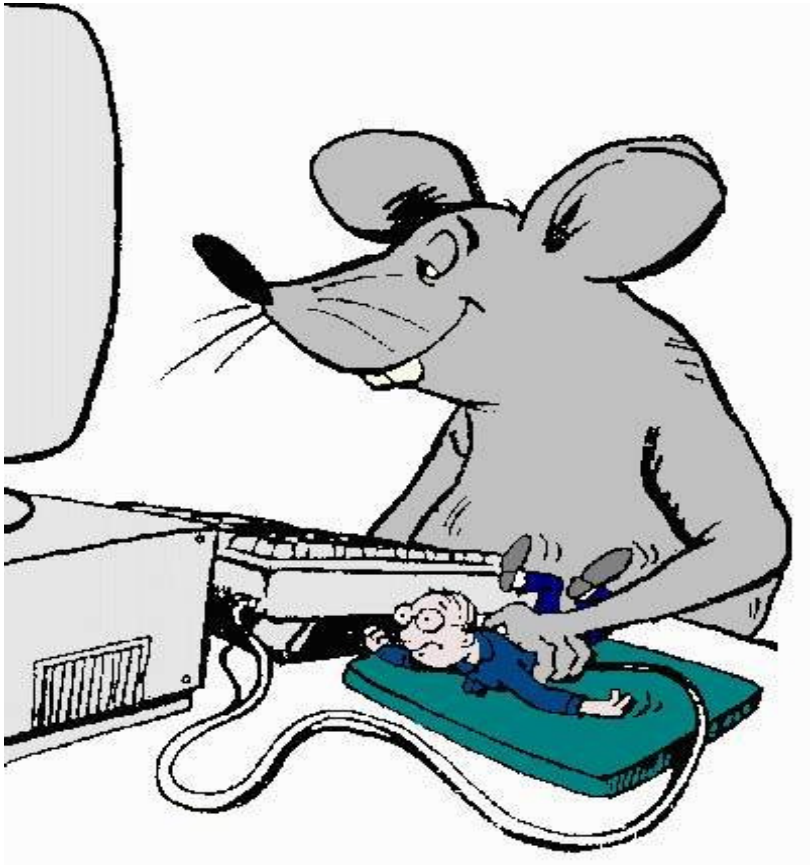
TAG: Nutzen und Schwierigkeiten

- Konsistenz wird Berücksichtigt
- Vorwissen wird Berücksichtigt

→ Lernaufwand abschätzen

- schwer auf grafische Applikationen anwendbar

Modelle auf körperlicher Ebene



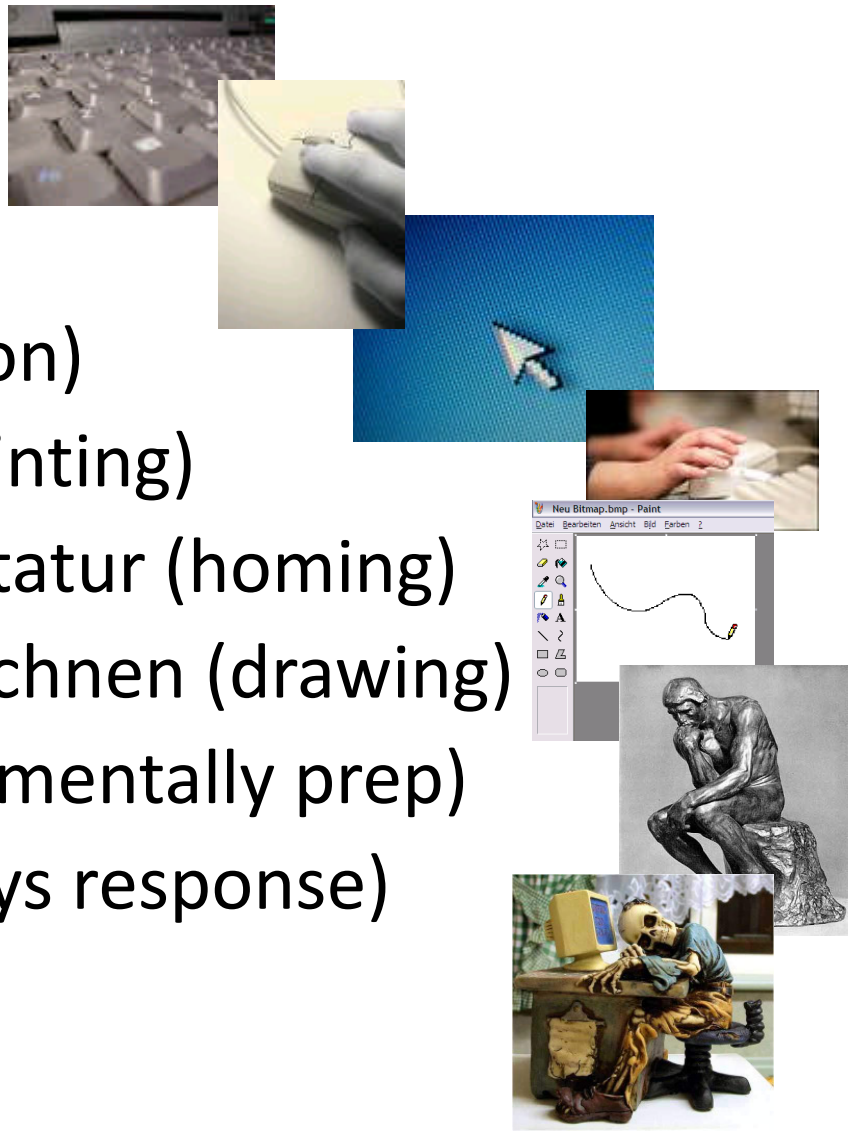
Keystroke-Level Model

Wie sehr einfaches GOMS

Zeitliche Komplexität durch Einteilung in und Summierung von Operationstypen

- 5 physisch motorische
- 1 mentaler
- 1 System Antwort

KLM - Elemente



K	Tastendruck (keystroke)
B	Maustastendruck (button)
P	Mauszeiger auf Ziel (pointing)
H	Hand zu Maus oder Tastatur (homing)
D	mit der Maus Linien zeichnen (drawing)
M	Mentale Vorbereitung (mentally prep)
R	Antwort des Systems (sys response)

KLM - Beispiel

Beispiel Zeichen in Texteditor korrigieren

Hand an Maus	H (Maus)
Maus an Zeichen und Klicken	PB (Links)
Hand an Tastatur	H (Tastatur)
Lösche Zeichen	MK (Löschen)
Gib Korrektur ein	K (Zeichen)
Repositioniere Schreibmarke	H (Maus)MPB

KLM - Beispiel

K	B	P	H	D	M	R
0.12	0.20	1.10	0.40	-	1.35	-
2	2	2	3	0	2	0
0.24	0.40	2.20	1.20	0	2.70	0

= 6,74 s

P: Durchschnittswert, K: Gute Tippleistung (90 Wörter/min)

Drei-Zustands-Modell

- Bezogen auf Zeigegeräte
  - Maus
  - Tablet-PC Stift
- 3 Status
  - Position bekannt, Button nicht gedrückt
  - Position bekannt und Button gedrückt
  - Keine Verbindung
- Ausführungszeiten hängen vom Status ab

Kognitive Architekturen

- Basis der kognitiven Modelle
- Annahmen über kognitive Vorgänge
- Für GOMS und CCT z.B. angenommen:
  - Langzeit und Kurzzeit-Gedächtnis
  - Aufteilung von Problemen in Teilprobleme

Problem Space Model

- Analyse erzeugt Erkenntnis über nötiges Wissen und Können potentieller Benutzer

- Führt zu **P**rogrammed **U**ser **M**odel

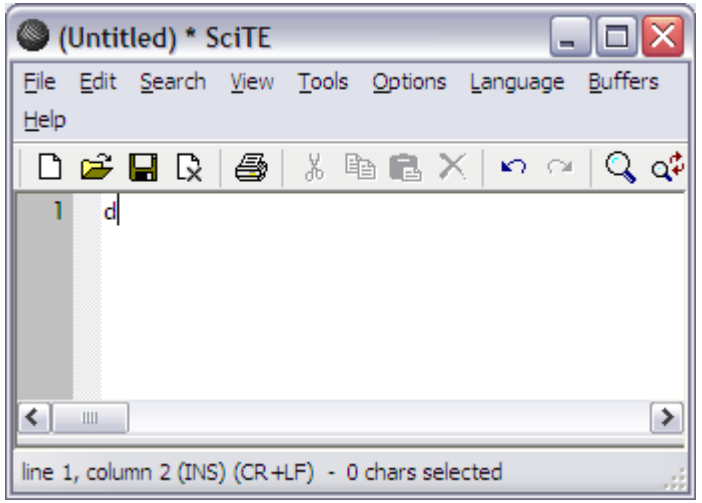
  kann PUM wegen Kenntnismangel Aufgabe nicht ausführen

  = System selbst überarbeiten

PSM - Elemente

- Kenntnis-Ebenen System

- Agent
- Operationen
- Umwelt



PSM - Funktionsweise

Problem – Suche in einem Set von Status

Weg zwischen 2 status – mögliche Aktionen

Ziel – gewünschter Status

4 Aktionstypen

Zielformulierung, Operationswahl,

~-anwendung, Prüfung auf Zielerfüllung

PSM - Aktivitätskreisel

1. Zielformulierungsprozess definiert Satz von gewollten Status und ersten Status für Aufgabe
2. Operationsauswahlprozess schlägt zielnähernde Operation vor
3. Operationanwendungsprozess führt Operation aus und verändert Status (und Umwelt)
4. Zielerfüllungsprozess deaktiviert Agent sobald aktueller Status angestrebter Status ist

Problemraum Modell

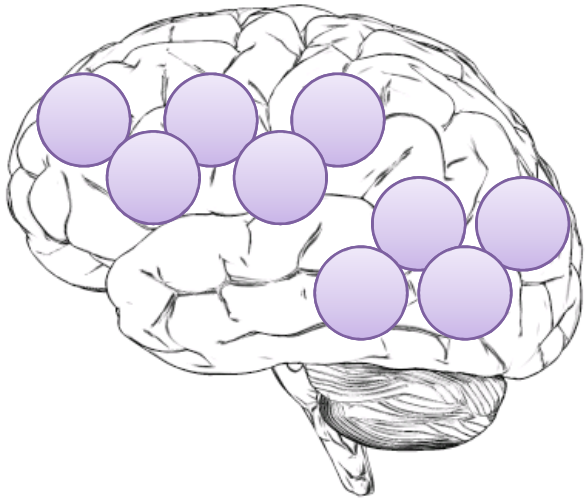
Nicht implementierbar – zu komplex

Führt aber zu SOAR

(Laird, Newell, Rosenbloom; 1990)

Interaktive kognitive Teilsysteme

- 9 kognitive Teilsysteme
- Ein System pro Sinn
- Vier zur internen Wissensverarbeitung



→ Genauere vorhersagen über Wahrnehmung

Bedeutung kognitiver Architekturen

- Rechtfertigen Existenz verschiedener Modelle
- Motivieren Ergänzung bestehender Modelle

Nochmal wichtige Abkürzungen

- GOMS

  Goals, Operators, Methods, Selection

- CCT

  Cognitive Complexity Theory

- KLM

  Keystroke Level Model

Was hängenbleiben sollte.

- Schätzung der Effektivität von Benutzerschnittstellen **vor** der Programmierung
- Standard-Ansätze: **GOMS**, **KLM**
- **CCT** Modelle erlauben sehr ausführliche Analyse (aber aufwendig)