



## 3 Interaktion

### 3.1 Objektdesign und die Folgen

## Affordances



- Modell von Donald Norman (1988) nach Gibson (1977)
- Affordances sind Aktionen, die das Design eines Objektes dem Benutzer suggeriert
- „...the term *affordance* refers to the perceived and actual properties of the thing, primarily those fundamental properties that determine just how the thing could possibly be used ...“

[Norman88, S.9]

## Zweck des Affordances-Modells



- Durch die Produktgestaltung intuitive Benutzbarkeit ermöglichen
- Beschriftungen einsparen
  - Ein Design mit Beschriftungen ist oft ein schlechtes
  - Dies gilt grundsätzlich auch für Software-Benutzerschnittstellen
  - Einschränkungen entstehen bei komplexen, abstrakten Funktionen
    - keine einfachen „physischen“ Affordances möglich



Folie 50

## Beispiel 1: Stühle



- Die Form eines Stuhls suggeriert, sich darauf zu setzen
- Auch andere Aktionen sind möglich
  - hochheben, sich daraufstellen etc.
  - Aber: Sie sind nicht die durch das Design am naheliegendsten
- ➔ Ein einfacher Stuhl ist intuitiv „benutzbar“
- Keine Beschriftung nötig („Hier setzen“)



Folie 51

## Beispiel 2: Türen

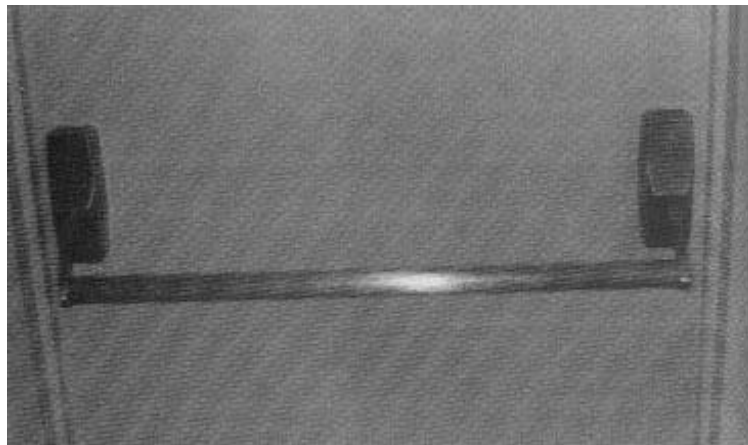


- Kleine Türgriffe suggerieren Aufziehen
- Breite, flache „Druckflächen“ suggerieren Aufdrücken
  - Beispiel: US-Hotel-Fluchttüren
- „Designertüren“ verbergen oft bewußt Hinweise auf die Funktion
  - auf welcher Seite liegen die Angeln, wo muss ich drücken/ziehen? — sinnvoll?
- 1-Wort-Handbuch notwendig?
  - „Ziehen“/„Drücken“



Folie 52

## Beispiel Fluchttür in USA

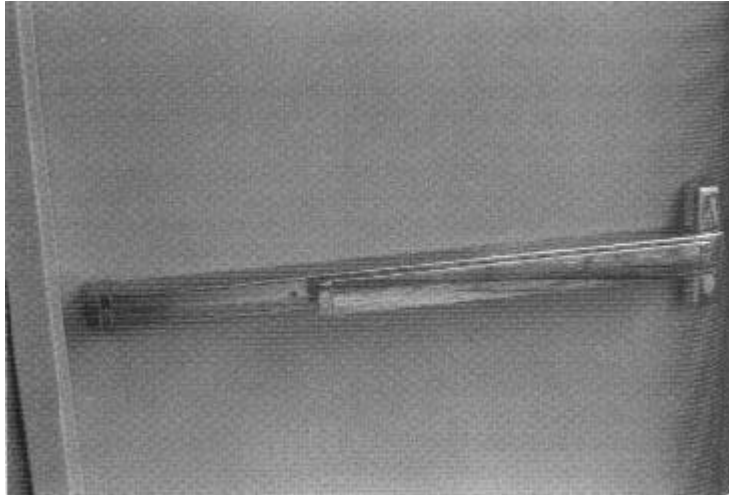


Tür, die nicht zeigt, auf welcher Seite man ziehen oder drücken muss



Folie 53

## Beispiel Fluchttür in USA

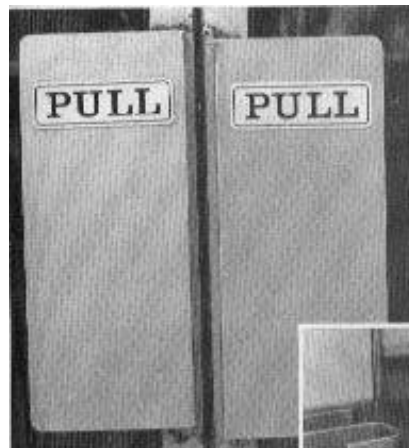


Bessere Fluchttür: „Rechts drücken“ wird klar suggeriert



Folie 54

## Beispiel Eingangstür in USA



Schlecht: Design suggeriert Drücken —  
Beschriftung erforderlich



Folie 55

## Beispiel 3: Wartehäuschen



- British-Rail-Wartehäuschen aus Glas wurden ständig zerschlagen
  - Glas suggeriert Zerbrechlichkeit
- Nach Ersatz durch (gleich stabiles) Sperrholz wurden sie nicht mehr zerschlagen
  - Holz suggeriert Stabilität / Stützfunktion
- Aber jetzt wurden sie bemalt
  - flaches, glattes Holz suggeriert „Schreibfläche“



Folie 56

## Abbildungen (Mappings)



- Verbindung zwischen Bedienelement und realer Auswirkung
- Gute Abbildungen sind *natürlich*:
  - Sie nutzen physikalische Analogien
  - Sie nutzen kulturelle Standards
- ➔ Sie erlauben sofortiges Verständnis
- ➔ Sie sind leicht ins Gedächtnis zurückzurufen
- ➔ Zweck:  
Sie ermöglichen einfachere Bedienung



Folie 57

## Beispiele natürlicher Abbildungen



- Räumliche Analogien
  - Hebebühne: Hebel hoch bewirkt, dass sich Bühne hochbewegt
  - Knöpfe für Lampen auf einem Panel so wie die Lampen im Raum anordnen
  - Lenkradoberkante nach rechts drehen, um nach rechts abzubiegen
- Frage:
  - Ist die Tafelsteuerung in H2O eine natürliche Abbildung?



Folie 58

## Beispiele natürlicher Abbildungen



- Biologisch-kulturelle Analogien
  - Thermometer:  
Steigender Wert = „mehr“, fallender = „weniger“
  - Auch für Menge, Lautstärke, Linienlänge, Helligkeit
    - „additive Dimensionen“
  - Aber:  
nicht für Farbe, Tonhöhe, Ort, Geschmacksrichtung
    - „substitutive Dimensionen“
  - Reihenfolge von oben nach unten
  - von links nach rechts
    - vor allem westliche Welt



Folie 59

## Constraints



- Constraints schränken die Benutzung von Objekten ein
- „Gegenteil“ von Affordances
  - Constraints ergänzen sie
- Zweck:
  - falsche Benutzung vermeiden
  - zu merkende Informationen minimieren



Folie 60

## Physikalische Constraints



- Begrenzen die Anzahl möglicher Operationen
- Begrenzung erfolgt durch Formgebung
  - Beispiel: Schlüssel mit Bart passt nicht in Sicherheitsschloß
- Effizienter und nützlicher, wenn vorab deutlich erkennbar
  - Beispiel: Autoschlüssel sollte in beiden Richtungen ins Schloß passen, aber dann auch schließen



Folie 61

## Semantische Constraints



- Nutzen unser allgemeines Wissen über die Welt und speziell die Bedeutung der Situation
- Beispiel:
  - Fahrerfigur im Modellbausatz muss vorwärts sitzen
- Mächtiges Mittel für intuitive Bedienung
- Aber: Nur allgemeingültige „Regeln“ verwenden!



Folie 62

## Kulturelle Beschränkungen



- Stützen sich auf allgemein anerkannte kulturelle Konventionen
  - Bsp. 1: Beschriftungen sind zum Lesen da, stehen also normalerweise nicht auf dem Kopf
  - Bsp. 2: Rot = Stop
- Problem:  
Gelten nur für bestimmten Kulturkreis
  - zu Bsp. 1: Chinesische Beschriftung gibt uns keinen Hinweis darauf, wo an einem Gerät „oben“ ist
  - zu Bsp. 2: chinesische Ampeln mit Grün = Stop (?)



Folie 63

## Logische Beschränkungen



- Verwenden logische  
Schlußfolgerungen, um bestimmte  
Lösungen auszuschließen
  - Beispiel Bausatz:  
alle Teile sind aufzubrauchen
- Natürliche Abbildungen verwenden oft  
logische Constraints
  - Beispiel:  
Linker Schalter = linke Lampe ist  
„natürlich“ („logisch“)



Folie 64

## Zusammenfassung



- *Affordances*: durch Gestaltung  
suggerierte Aktionen zur Benutzung  
eines Objekts
- Natürliche *Abbildungen*  
(Bedienelement–Welt) erhöhen  
Benutzbarkeit & Merkbarkeit
- *Constraints* (physikalisch, semantisch,  
kulturell, logisch) verhindern  
Bedienfehler



Folie 65



## 3 Interaktion

### 3.2 Normans Aktionszyklus

## Einführung

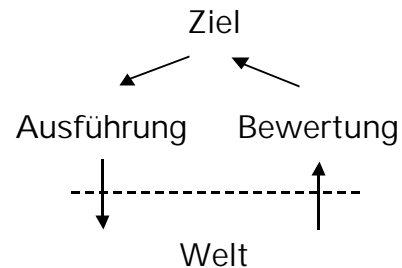


- Zentrale Frage
  - Was macht Dinge schwierig zu benutzen?
- Für die Antwort muss man zuerst fragen: Was passiert, wenn jemand ein Objekt benutzt?
- ➔ Wir brauchen ein Modell dieses Vorgangs
  - Der *Aktionszyklus* ist so ein Modell

## Erster Ansatz



- Um etwas zu tun, muss man
  - Ein *Ziel* formulieren (was will man tun?)
  - Eine Aktion *ausführen* (die Welt verändern...)
  - Das Ergebnis *bewerten* (entspricht das Ergebnis dem Ziel?)



Folie 68

## Die Zielformulierung



- *Ziele* sind oft sehr vage, problemorientiert formuliert
  - „Ich brauche mehr Licht“
- Sie müssen in zielorientierte *Intentionen* übersetzt werden
  - „Den Lichtschalter betätigen“
- Aus diesen müssen dann konkrete *Aktionssequenzen* abgeleitet werden
  - „Umdrehen, Arm ausstrecken, Finger auf Schalter“

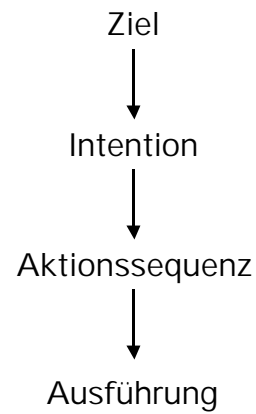


Folie 69

## Die Zielformulierung



- Alternative Intentionen und Aktionssequenzen für das gleiche Ziel sind möglich
  - „Machst Du mal das Licht an, Liebling?“ ...

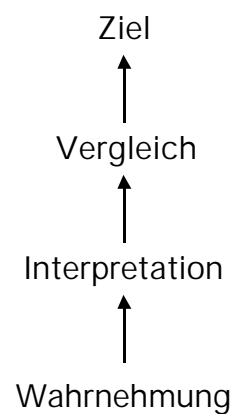


Folie 70

## Die Bewertung

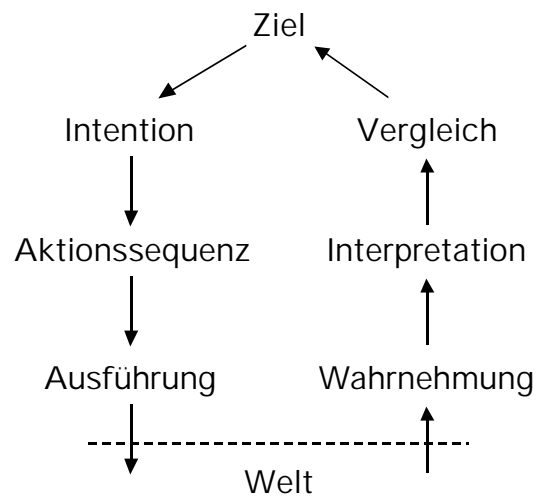


- Bewertung erfordert ebenfalls mehrere Stufen:
  - *Wahrnehmen* des Ergebnisses
  - *Interpretieren* des Ergebnisses
  - *Vergleich* mit dem ursprünglichen Ziel



Folie 71

## Ergebnis: Der Aktionszyklus



Folie 72

## Ergänzungen



- Stufen i.a. nicht klar trennbar
- Komplexe Aufgaben beinhalten Folgen/Hierarchien von Zielen
  - Feedback-Schleife
- Ziele werden vergessen, verworfen, verändert
- Viele Aktionen sind opportunistisch, nicht geplant
  - Gespräch durch Aufeinandertreffen, Deadline-driven work
- Zyklus kann an jeder Stelle beginnen!

Folie 73

## Bruchstellen in der Ausführung



- Auch eigentlich einfache Aktionen können schwierig erscheinen
- Ein Grund:
  - Es ist nicht zu erkennen, wie das System funktioniert bzw. „was zu tun ist“
- Beispiel: Einschweißverpackungen...
- Relation zwischen Intentionen und Ausführung der Aktionssequenzen unklar
- Woran liegt's? — Abbildungen (Mappings)!



Folie 74

## Bruchstellen in der Ausführung



- Bruchstellen entstehen durch Differenzen zwischen
  - *beabsichtigten* Aktionen des Benutzers und
  - *angebotenen* Aktionen (→Affordances) des Systems
- Ideal: System lässt Benutzer beabsichtigte Aktionen direkt ausführen, ohne extra Aufwand
- Beispiel: Filmeinfädeln
  - Alte Projektoren: unklar, mühsam
  - Neue Projektoren: automatisches Einfädeln
  - Videorecorder: Einfädeln unsichtbar gemacht



Folie 75

## Bruchstellen in der Bewertung



- Oft ist unklar, ob die Aktion erfolgreich war oder welche Wirkung sie hatte
- Das Problem: mangelndes Feedback
- Ideal: Systemstatus einfach wahrzunehmen und zu interpretieren, entspricht dem geistigen Modell des Benutzers
- Beispiele:
  - blinkende Lampe am Drucker ...
    - Rechnet er noch, oder hat er sich aufgehängt?
  - Tonfolge beim Verbindungsaufbau des Telefons



Folie 76

## Aktionszyklus als Designhilfe



- Liefert grundlegende Checkliste von Fragen, um Bruchstellen auszuschließen:
- "Wie einfach ist es, ..."
  - die *Funktion* des Systems zu erkennen? (Ziel)
  - die ausführbaren *Aktionen* zu bestimmen? (Intentionen)
  - deren *Abbildung* auf Aktionen zu ermitteln? (Aktionssequenz)
  - die Aktionssequenz *auszuführen*? (Ausführung)
  - den *Systemzustand* zu erkennen? (Wahrnehmung)
  - diesen auf eine *Interpretation* abzubilden? (Interpretation)
  - zu *entscheiden*, ob der gewünschte Zustand erreicht wurde? (Vergleich)



Folie 77

## Ergebnis: erste Designprinzipien



- Sichtbarkeit
  - Zustand & Aktionen erkennbar
- Gutes konzeptionelles Modell
  - konsistente Präsentation von Operationen und Ergebnissen
  - kohärentes „Bild“ des Systems im Kopf des Benutzers
- Gute (natürliche) Abbildungen zwischen
  - Aktionen und Ergebnissen
  - Bedienelementen und Auswirkungen
  - Systemzustand und Sichtbarem
- Gutes Feedback
  - über Auswirkungen, komplett und kontinuierlich



Folie 78

## Zusammenfassung



- *Aktionszyklus:*  
mentales Modell der Interaktion
- Ziel – Ausführung – Bewertung
- Bruchstellen sind Bedienprobleme  
(Checkliste)



Folie 79



## 3 Interaktion

### 3.3 Internes und externes Wissen

#### Viel Wissen ist „in der Welt“



- Beispiele, was nicht „im Kopf“ ist:
  - Buchstabenreihenfolge auf Tastaturen
  - -" - auf Telefonen mit Buchstabenaufdruck
- Trotz unpräzisen Wissens ist präzises Verhalten möglich
- Wie ist das möglich?

## Gründe



- Verhalten wird durch Kombination von internem Wissen („im Kopf“) mit externem Wissen („in der Welt“) bestimmt
- Hohe Wissenspräzision ist nicht erforderlich
  - Wissen muss nur genau genug sein, um die korrekte Verhaltensweise unter den möglichen herausfinden zu können
  - Beispiel: Münzexperiment



Folie 82

## Gründe



- Natürliche Constraints wirken
  - Physikalische Begrenzungen limitieren die möglichen Aktionen (was kann wie bewegt/zusammengefügt/manipuliert werden)
- Kulturelle Constraints wirken
  - gesellschaftliche Regeln werden einmal erlernt und sind dann in vielen Situationen anwendbar
  - Beispiel: Was macht man, wenn man ein Restaurant betritt
    - aber: Unterschied D-USA



Folie 83

## Beispiele für Constraints



- Reisende Künstler konnten Gedichte mit Tausenden von Zeilen aufsagen
- Wie? Reim ist Constraint!
- Walkman reparieren:
  - 1000e von theoretischen Möglichkeiten, aber nur ein paar Dutzend durch physikalische Constraints möglich
- Constraints begrenzen zu lernende Menge



Folie 84

## Folgen



- Der Mensch kann die Menge / Genauigkeit / Tiefe zu merkender Information minimieren
  - Dies kann sogar geistige Behinderungen oder fehlende Fähigkeiten (z.B. Analphabetismus) überspielen helfen



Folie 85

## Beispiel „Tippen“



- Anfänger nutzt nur externes Wissen (Tastenbeschriftung)
- Fortgeschrittener nutzt teilweise externes (periphere Sicht, Tasten fühlen) und teilweise internes Wissen (Position wichtiger Tasten „auswendig“ kennen)
- Experte internalisiert Wissen komplett, dadurch kein Blickkontakt zur Tastatur mehr nötig
  - Tradeoff Kosten/Nutzen



Folie 86

## Beispiel „Stadtplan“



- Niemand hat einen exakten Stadtplan im Kopf
- Trotzdem kann man sich in einer Stadt bewegen
- Schilder und Constraints (Hausnummern etc.) liefern externes Wissen
- Man kann oft nicht einmal die exakte Route aus dem Kopf beschreiben



Folie 87

## Wissensformen



- Deklaratives Wissen („was“)
  - Fakten („München liegt südöstlich von Ulm“)
  - Regeln („An roten Ampeln halten“)
  - leicht aufzuschreiben und zu *lehren* (nicht lernen!)
- Prozedurales Wissen („wie“)
  - Wie spielt man ein Stück auf einem Instrument
  - Wie spricht man „Fischers Fritze“ aus
  - schwierig/unmöglich aufzuschreiben, unbewusst
  - schwierig zu lehren, am besten Demo/Übung
- Design kann *deklaratives* Wissen einfach liefern



Folie 88

## Was kann man sich merken?



- Beliebige Dinge ohne Zusammenhang:  
wenig
  - Zum Starten drücke Alt-Ctrl-Del
  - eigentlich gar nicht lernbar (nur durch Assoziationen, Eselsbrücken)
  - Beispiel: Pi auf 1000 Stellen (Mark Dettinger)
  - Wenn Kochrezept versagt, ist man aufgeschmissen
- Dinge im Zusammenhang:  
mehr (Assoziationen)
  - Beispiel Desktopmetapher



Folie 89

## Was kann man sich merken?



- Erklärte Dinge:  
Sehr viel (wieder herleitbar)
  - sehr wertvoll in unerwarteten Situationen, wo „Kochrezept“ versagt
  - Modell im Kopf entsteht



Folie 90

## Gedächtnisformen



- „38294“ laut lesen, kurz danach wiederholen: einfach
  - Kurzzeitgedächtnis
- Was vor 3 Tagen zu Abend gegessen? schwierig
  - Assoziationen finden, um aus Langzeitgedächtnis hervorzuholen
- Anschließend 5 Ziffern nochmal sagen: schwierig
  - aus Kurzzeitgedächtnis verdrängt



Folie 91

## Tägliche Quälerei



- Passwörter für 5 verschiedene Systeme
- 2 Kreditkartennummern
- 2 Bankkontonummern + BLZs
- Nummernschild
- Telefonnummern, Adressen, Geburtstage, Alter von Bekannten
- Kleidungsgrößen,...
- ➔ Ab aus dem Kopf in die Welt!



Folie 92

## Probleme externen Wissens



- Grundsätzlich:  
Sehr gut (nichts zu merken)
- Aber:  
Nur da, wenn man drauf „sieht“
- „Aus den Augen, aus dem Sinn“
- Problem vor allem bei nicht immens persönlich wichtigen Dingen
- Lösung: Sich erinnern lassen
  - Beispiel: Terminkalender vs. Organizer
  - Signal + Nachricht



Folie 93

## Tradeoff



- Externes Wissen
  - Erreichbarkeit: sobald sichtbar
  - Lernen nicht erforderlich
  - Effizienz niedrig (Interpretieren)
  - Benutzbarkeit bei Erstkontakt hoch
  - Ästhetik: kann bei vielen Infos Problem werden
- Internes Wissen
  - schlechter erreichbar, zu erlernen, schwierig für Anfänger, aber effizient und unsichtbar



Folie 94

## Abbildungen



- Natürliche Abbildungen ersparen das Auswendiglernen (und Labels)
- Beispiel: Anordnung von Herdplatten und Schaltern
  - Bis heute oftmals unbefriedigend gelöst!
- Trick:  
Produkt / Design / Prototyp nicht nur ansehen, sondern ausprobieren



Folie 95

## Zusammenfassung „Wissen“



- Gedächtnis und Formen von zu merkender Information
- Vor- und Nachteile internen und externen Wissens



Folie 96



## 3 Interaktion

### 3.4 GOMS Prinzip und Bestandteile

## Eine Geschichte



- Aufgabe für Jakob Nielsen  
(Sun Distinguished Engineer) 1995:  
3 neue Buttons auf Sun-Homepage?
- Zeit: < 24h.
- Ergebnis: Zusätzlicher, unbenutzter  
Button kostet Besucher 1/2 Mio \$/a.
- ➔ 2 Buttons wieder gestrichen
- Verwendete Methode: GOMS



Folie 98

## Engineering Models in HCI



- GOMS: Beispiel für "Engineering Model" im  
HCI-Bereich "Kognitive Modellierung"
- Ziel solcher Modelle (im Vergleich zu  
psychologischen Ansätzen):
  - Nützliche a-priori-Abschätzungen liefern
  - Einsetzbar durch  
Software Engineering-Forscher & -Praktiker  
(Psychologisches Wissen "eingebaut")
  - Abdecken vieler relevanter HCI-Design-Fragen  
(Wahrnehmung + Motorik, aber  
aufgabenumfassend)
  - 80% Exaktheit bei 20% Aufwand  
(Prototyp = 100%)



Folie 99

## GOMS im Überblick



- Methode zur Beschreibung von Aufgaben (Tasks) und des Wissens des Menschen, wie man sie bearbeitet
- Ziele:  
Bearbeitungs- und Lernzeiten abschätzen
- GOMS-Modellbestandteile:  
Goals, Operators, Methods & Selection Rules
- Stuart Card, Thomas Moran und Allen Newell: "The Psychology of Human-Computer Interaction" (1983, wegweisend)



Folie 100

## Ziele (Goals)



- Ziele des Benutzers: Was will er durch die Benutzung des Systems erreichen?
- Routineaufgaben
- Nicht zu  
abstrakt/kreativ/problemlösend
- Hierarchie von Unterzielen
  - Bsp. Manuskript editieren
- Im Jargon der Domäne formuliert



Folie 101

## Operatoren



- Kleinste, atomare Aktionen des Benutzers
- Beispiele:
  - Ursprünglich: Textkommando + Parameter, Meldung lesen
  - Mit WIMPs auch: Menüauswahl, Drag&Drop, ...
  - Inzwischen: Gesten, Sprachkommandos, ...
- Auf verschiedenen Abstraktionsebenen definierbar, aber Tendenz zu konkreten Aktionen
- Kontextfrei konstante/parametrisierbare Dauer



Folie 102

## Methoden



- Gelernte Folgen von Unterzielen und Operatoren, um ein Ziel zu erreichen
- Beispiel: Methode "Lösche-Absatz-Drag":
  - Mit Maus Cursor an Absatzbeginn platzieren
  - Mausknopf gedrückt halten
  - Maus bis zum Ende des Absatzes ziehen
  - Mausknopf loslassen
  - Delete-Taste drücken  
(alles Operatoren)



Folie 103

## Selektionsregeln



- I.a: mehrere Methoden für das Erreichen eines Ziels vorhanden
  - Bsp. "Lösche-Absatz-Del" (zeichenweise löschen)
- ➔ Persönliche, individuelle *Selektionsregeln* des Benutzers entscheiden auf Basis anderer Faktoren, welche Methode er verwendet
- Beispiel: Verwende "Lösche-Absatz-Del" bei Absätzen < 5 Buchstaben



Folie 104

## Form eines GOMS-Modells



- Programmform:
  - Ähnlich Computerprogramm
  - Beschreibt, wie Klasse von Aufgaben gelöst wird
  - Aufgabenvariablen als Parameter
  - Methoden als Prozeduren
  - Beispiel: Lösche-Absatz-Drag (startpos,endpos)
  - Vorteil: Umfassend, maschinenausführbar
  - Nachteil: Unübersichtlich, aufwendig
- Sequenzform:  
einfacher, aber unparametrisiert
  - Ähnlich speziellem Programmdurchlauf, nicht allgemein



Folie 105



## 3 Interaktion

### 3.5 GOMS Varianten und Beispiele

#### KLM



- Keystroke-Level Model
- Einfache Vorversion des Original-GOMS
- Ebenfalls von Card/Moran/Newell, 1980
- Sequenzform auf „Tastenniveau“ ohne Goals/Methoden/Selektionsregeln
- Operatoren: Nur Key, Point, HomeHands, MentalPrepare, ResponseSystem
- Dazu 5 heuristische Regeln für Platzierung von M-Operatoren („Denkpausen“)

## KLM: Beispiel



- Text mit Menümethode bewegen:
    - Vorbereiten (Regel 0) M 1.35s
    - Cursor auf Bereichsanfang (kein M nach Regel 1) P 1.10s
    - Mausklick (kein M nach Regel 0) K 0.20s
    - Cursor auf Bereichsende (kein M nach Regel 1) P 1.10s
    - ...
- Summe: 14.38s



Folie 108

## KLM: Bewertung



- Am schnellsten und einfachsten anwendbares (und begründbares) Modell
- Nützlich für Vergleichsschätzungen von Bedienzeiten und zum Beurteilen des Einflusses externer Parameter (Tippgeschwindigkeit etc.)
- Nachteil: Sequenzform schlecht automatisierbar (keine Parameter)



Folie 109

## CMN-GOMS



- Originalmodell von Card, Moran, Newell 1983
- Unterschiede zum KLM:
  - Hierarchische Ziele, Methoden und Selektionsregeln sowie „Verify“-Operationen statt des einfachen M-Operators
  - ➔ Dadurch detailliertere Modellierung
  - Programmform, dadurch allgemein und ausführbar



Folie 110

## CMN-GOMS: Beispiel



- GOAL: EDIT-MANUSCRIPT
  - GOAL: EDIT-UNIT-TASK (\* wdh. bis keine UNIT-TASK mehr)
    - GOAL: ACQUIRE UNIT-TASK (\* wenn nicht im Kopf \*)
      - GOAL: TURN-PAGE (\* wenn am Seitenende \*)
      - GOAL: GET-FROM-MANUSCRIPT
    - GOAL: EXECUTE-UNIT-TASK (\* wenn eine gefunden \*)
      - GOAL: MODIFY-TEXT
        - select:
          - GOAL: MOVE-TEXT\* (\*wenn Text zu verschieben ist\*)
          - GOAL: DELETE-PHRASE (\* wenn Text zu löschen ist \*)
          - GOAL: INSERT-WORD (\* wenn Wort einzufügen ist \*)
          - [...]
        - VERIFY-EDIT



Folie 111

## CMN-GOMS: Beispiel



- GOAL: MOVE-TEXT (\* Ausformulierung \*)
  - GOAL: CUT-TEXT
    - GOAL: HIGHLIGHT-TEXT
      - Select: (\* je nach Länge des Bereichs \*)
        - GOAL: HIGHLIGHT-WORD [...] 1.10s
        - GOAL: HIGHLIGHT-ARBITRARY-TEXT
        - MOVE-CURSOR-TO-BEGINNING 1.10s
        - CLICK-MOUSE-BUTTON 0.20s
        - MOVE-CURSOR-TO-END 1.10s
        - SHIFT-CLICK-MOUSE-BUTTON 0.48s
        - VERIFY-HIGHLIGHT 1.35s
      - GOAL: ISSUE-CUT-COMMAND
        - MOVE-CURSOR-TO-EDIT-MENU 1.10s
        - PRESS-MOUSE-BUTTON 0.10s
        - MOVE-MOUSE-TO-CUT-ITEM 1.10s
        - VERIFY-HIGHLIGHT 1.35s
        - RELEASE-MOUSE-BUTTON 0.10s
    - GOAL: PASTE-TEXT [...]

... Summe: 14.38s



Folie 112

## CMN-GOMS: Bewertung



- Detaillierter und realistischer als das KLM
- CMN-GOMS-Modelle durch hierarchische Programmform besser automatisierbar
- Aber:  
Genauere Anleitung zur Erstellung fehlte noch



Folie 113

## NGOMSL



- Natural GOMS Language (David Kieras, 1988)
- Ausformulierter als das Original
- Basiert auf Cognitive Complexity Theory (CCT)
- Programmiersprachen-ähnliche Darstellung der Komponenten
- Dazu: Grobe Regeln für viele Faktoren
  - 4 Ebenen bis Tastenlevel, 5 Schritte pro Methode, Zielsetzung & -abschluss, Gedächtnisanforderungen



Folie 114

## NGOMSL-Kochrezept



- Oberste Ziele wählen
  - Bsp. Dokument editieren
- Rekursiv Methoden pro Ziel skizzieren (BFS)
  - "Wie würde der Benutzer beschreiben, wie man das macht, erst mal ohne die genauen Details?"
  - Neue Unterziele festhalten
  - Shortcuts für später aufheben
- Konsistenz (gleiche Methoden,...) und Auswirkung der Modellannahmen prüfen



Folie 115

## NGOMSL: Beispiel



- Ziel: Text editieren (Ebene 1)
- Methode für Ziel: Text verschieben (Ebene 2)
  - Erfülle Ziel: Text ausschneiden
  - Erfülle Ziel: Text einfügen
  - Zurück, Ziel erfüllt
- Methode für Ziel: Text ausschneiden (Ebene 3)
  - Erfülle Ziel: Text markieren
  - An Kommando „Cut“ erinnern
  - Erfülle Ziel: Kommando absetzen
  - Zurück, Ziel erfüllt



Folie 116

## NGOMSL: Beispiel



- Methode für Ziel: Text einfügen (Ebene 3)  
[...]
- Selektionsregelmenge für Ziel:  
Text markieren
  - Wenn Text= Wort, dann erfülle Ziel:  
Wort markieren
  - Ansonsten: Erfülle Ziel:  
Beliebigen Text markieren
  - Zurück, Ziel erfüllt
- Methode für Ziel: Wort markieren (Ebene 4)  
[...]



Folie 117

## NGOMSL: Beispiel



- Methode für Ziel: Belieb. Text markieren (Ebene 4)
    - Bestimme Position des Textanfangs      1    1.20s
    - Bewege Cursor auf Textanfang            1    1.10s
    - Mausklick                                      1    0.20s
    - ...
- ... Summe: 16.38s



Folie 118

## NGOMSL & Lernaufwand



- Lernaufwand für Methoden (prozedurales Wissen) grob abschätzbar
  - Lernzeit =  
Reine Lernzeit + Ausführungszeit im Training
  - Reine Lernzeit =  
Methoden-Lernzeit + Langzeitgedächtnis-Lernzeit
  - Methoden-Lernzeit =  $17s * |\text{NGOMSL-Vokabular}|$
  - Gedächtnis-Lernzeit =  
 $7s * \#$  zu erlernende "chunks" (gemäß CMN)
  - Im Beispiel: 800s zum Erlernen der Prozeduren
  - Wichtig: Gilt nicht bei neuen *Operatoren* (Gesten,...)!



Folie 119

## CPM-GOMS

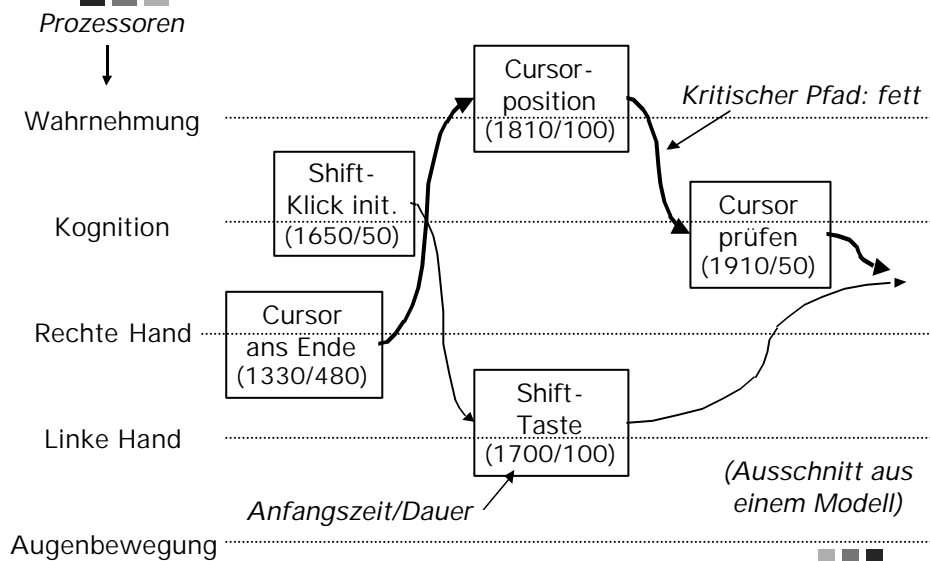


- Gray, John, Atwood 1993
- Idee:  
Mache CMN-GOMS-Modell realistischer durch Parallelisierung von Zielen & Operatoren
- Sequenzform
- Beinhaltet Operatoren-Kombinations-Templates für Wahrnehmung (Perception), Erkennung (Cognition) und Motorik
- Verwendet PERT Chart (Critical Path Method)



Folie 120

## CPM-GOMS: Beispiel



Folie 121

## CPM-GOMS: Bewertung



- Liefert deutlich kürzere Zeitschätzungen
- Realistisch für hochtrainierte Experten-Benutzer
- Für hochparallelisierbare, multimodale Tätigkeiten (Hotline, ...)
- Nachteile:
  - Wegen Sequenzform schlecht automatisierbar
  - Erstellen der Templates für Parallelverarbeitung aufwendig



Folie 122

## EPIC



- Executive Process-Interactive Control
- David Kieras et al. [ToCHI 9/97]
- Paralleles NGOMSL-Folgemodell
- Besonders für multimodale, hochgradig reaktive Aufgaben (Flugzeugcockpit etc.)
- Verhaltensmodell variabel
  - von naiv-hierarchisch-sequentiell bis geübt-hochoptimiert-parallel
  - entspricht verschiedenen Praxisphasen



Folie 123

## EPIC: Bewertung



- Getestet mit Telephone Operator Scenario
- Liefert bessere Schätzungen als CPM-GOMS, Modellkonstruktion einfacher (generativ)
- Interessant: Bester „Fit“ mit nicht-optimalem Modell (entspricht Kompromiss zwischen zusätzlichem Lernaufwand und Effizienzgewinn)
- State of the Art für „Engineering Cognitive Models“



Folie 124



## 3 Interaktion

### 3.5 GOMS Einsetzbarkeit

## Was kann GOMS liefern?



- Ausführungszeiten für Aufgaben bei geübten Benutzern vorhersagen
  - Verknüpfung geschätzter Zeiten pro Operator
  - auch für parallel bearbeitete Aufgaben
- Lernzeit für das Ausführen von Aufgaben vorhersagen (Methoden-Lernzeit)
- Gedächtnisbasierte Fehlbedienungs-wahrscheinlichkeiten vorhersagen (wo, wie oft)



Folie 126

## Was lässt sich daraus ableiten?



- Trainingskosten
- Kosten täglicher Benutzung (Effizienz)
- Fehler(behebungs)kosten
- ➔ Verknüpfbar mit Anschaffungs-, Änderungs-, Herstellungs-, Wartungskosten
- ➔ Aussagen in der richtigen Sprache
  - ➔ Letztlich in Geld messbar



Folie 127

## Anwendungsbereiche



- Passive Einbenutzeranwendungen
  - Benutzer initiiert Aktion – System handelt – Feedback
  - Beispiele: Betriebssystem, Tabellenkalkulation, Texteditor, DTP, CAD, Informationsbrowser, WWW-Seiten, Flugcomputer, Oszilloskope, progr. Fernseher
- Aktive Einbenutzeranwendungen
  - System liefert ständig neue Situationen
  - Beispiele: Radar, Spiele



Folie 128

## Anwendungsbereiche



- Ersatzbenutzersituationen
  - Benutzer löst Probleme eines Clients (Hotline)
- Auch kreative, Nicht-Standard-Aufgaben
  - Routineanteil erleichtern (Programmieren, Design)



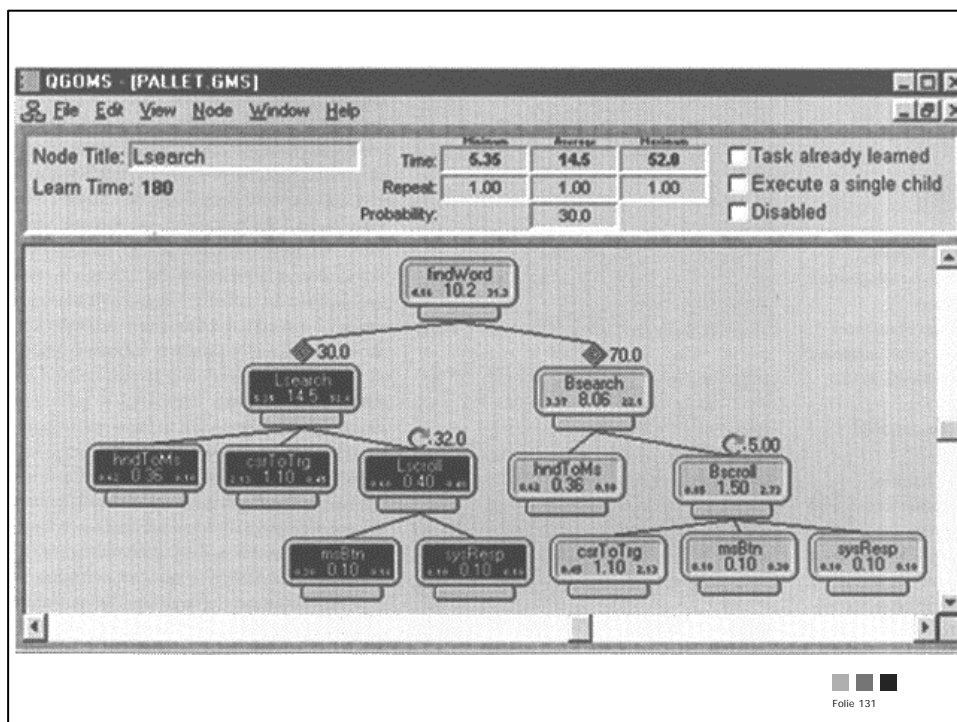
Folie 129

## Tools: QGOMS



- Quick (and Dirty) GOMS
  - David Beard, 1996
- Tool zur schnellen graphischen Erstellung von Ziel- und Operatorhierarchien
- Templates für gängige Operationen
- Automatische Berechnung der Bearbeitungszeit
- <http://cob.isu.edu/qgoms/>

Folie 130



Folie 131

## Einsatzbeispiele Quantitativ



- Alternative Systemangebote modellieren
  - Beschaffungskosten/Nutzungseffizienz analysieren
  - Expert-User-Vorhersagen & \$-Angaben oft nur mit GOMS möglich
  - Bsp.: Neue NYNEX Operator Workstations \$2M/a teurer [Gray93]
- Redesign-Auswirkungen abschätzen
  - Kurzfristige Trainingskosten / langfristige Arbeitersparnis analysieren
  - Bsp. Kartendigitalisiersystem [Gong93]: GOMS = 12% der Projektzeit, dadurch 50% kürzere Einlernzeit, 40% kürzere Experten-Bedenzeiten



Folie 132

## Einsatzbeispiele Qualitativ



- Dokumentation
  - GOMS beschreibt Wissen, um Aufgaben zu erledigen
  - ➔ Guter Ausgangspunkt für Online-Hilfe, Tutorials, Schulungen ("task-oriented documentation")



Folie 133

## Wann nicht GOMS?



- Kiosksysteme ohne Mehrfachbenutzer
  - Neuartige Systeme ohne Benutzungsprofil
  - Systeme mit Schwerpunkt des Erlernens auf der Operatorebene (Gestenerkennung, VR)
  - Generelle Gefahr: mechanistische Sichtweise des Menschen
- ➔ GOMS nicht immer, aber oft sinnvoll



Folie 134

## Zusammenfassung GOMS



- GOMS liefert a-priori-Abschätzungen für Ausführungszeiten (und Lernzeiten) von Routinetätigkeiten des Benutzers
- Aufgaben hierarchisch durch Ziele, Operatoren, Methoden und Selektionsregeln modelliert
- Evolutionäre Familie von Methoden
  - KLM, CMN, NGOMSL, CPM-GOMS, EPIC
- Tools existieren
  - z.B. QGOMS
- Bewährte Ergänzung/Alternative zu Prototyping
  - für Beschaffung, (Re-)Design-Tradeoffs, Doku



Folie 135

## GOMS-Literatur



- B. John: *Why GOMS? interactions* 2(4), 1995, 80-89. {Kurzer Einstieg+ Überblick}
- S. Card, T. Moran, A. Newell: The keystroke-level model for user performance time with interactive systems. *CACM* 23(7), 1980, 396-410. {KLM}
- S. Card, T. Moran, A. Newell: *The psychology of human-computer interaction*. Erlbaum, Hillsdale, NJ, 1983. {CMN-GOMS}
- D. Kieras: Towards a practical GOMS model methodology for user interface design. In M. Helander (ed.): *Handbook of human-computer interaction*, North-Holland, Amsterdam, 1988, 135-157. {NGOMSL, aktualisiert in 2nd ed. 1998}



Folie 136

## GOMS-Literatur



- W. Gray, B. John, M. Atwood: Project Ernestine: A validation of GOMS for prediction and explanation of real-world task performance. *Human-Computer Interaction* 8(3), 1993, 207-209. {CPM-GOMS}
- D. Kieras, S. Wood, D. Meyer: Predictive engineering models based on the EPIC architecture for a multimodal high-performance human-computer interaction task. *Trans. Comput.-Human Interact.* 4(3), 1997, 230-275. {EPIC}
- D. Beard et al.: Quick GOMS: A visual software engineering tool for simple rapid time-motion modeling. *interactions* 4(3), 1997, 31-36. {Quick GOMS}



Folie 137