

Analytische Evaluation: Modellbasierte Techniken

Ilhan Aslan, Chi Tai Dang, Björn Bittner, Katrin Janowski,
Elisabeth André



Human Centered Multimedia

Institute of Computer Science

Augsburg University

Universitätsstr. 6a

86159 Augsburg, Germany

- Gegeben:
 - Entwurf für ein Interface
 - Aufgabe, die der Nutzer mit dem Interface bearbeitet
- Frage:

Wie kann man vorhersagen, ob ein bestimmter Interface-Entwurf dem Usability-Kriterium „**Effizienz**“ genügt?
- Idee:

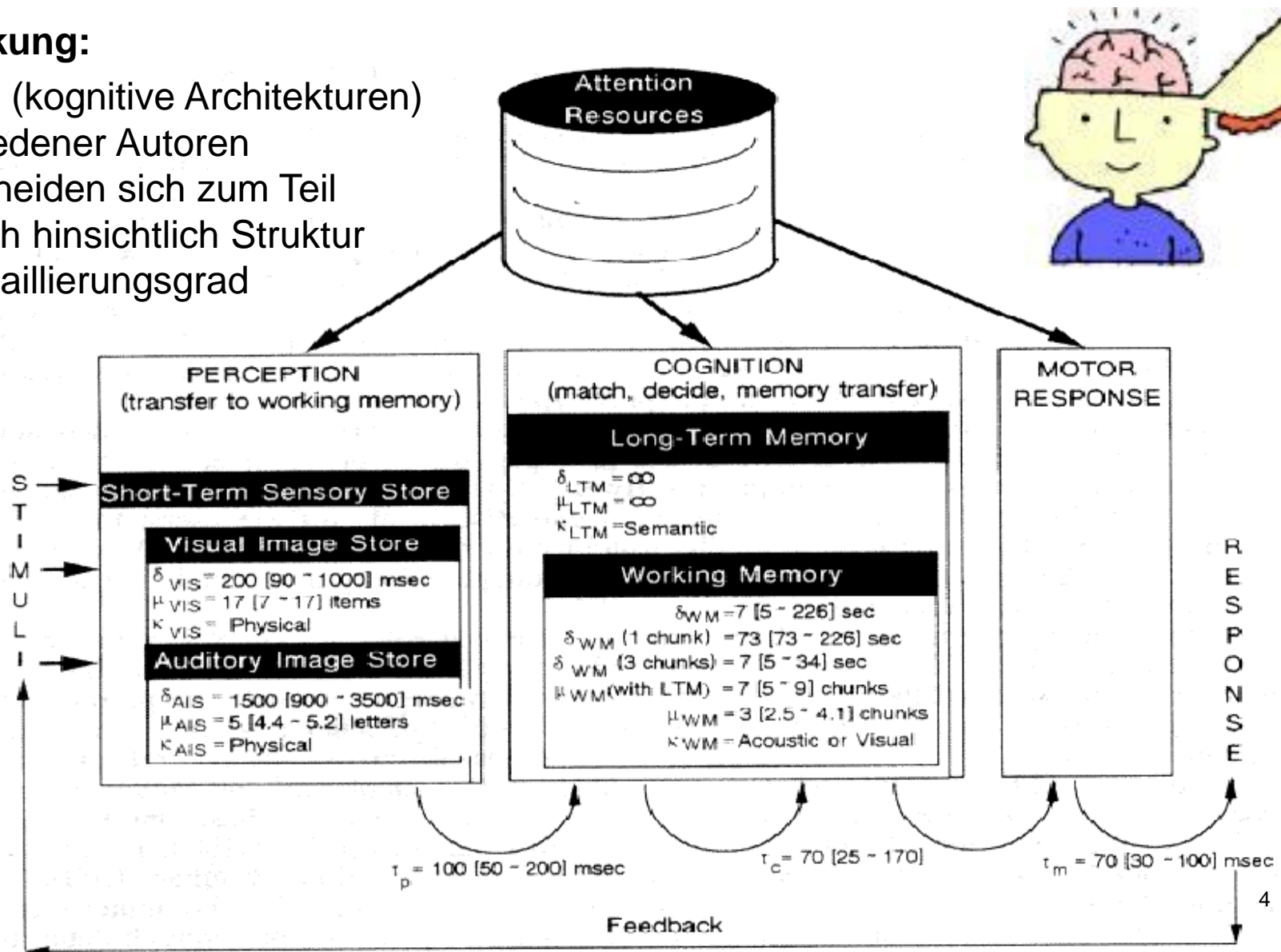
Man verwendet ein sog. **Benutzermode**ll, in dem Wissen über menschliche Fähigkeiten zusammengefasst ist.

Menschliche Fähigkeiten:

- perzeptive Fähigkeiten
(z.B. Dauer für Wahrnehmung visueller, akustischer oder haptischer Reize)
- kognitive Fähigkeiten
(z.B. Dauer für Treffen einer Auswahl, Erinnern, Erkennen einer Analogie)
- motorische Fähigkeiten
(z.B. Dauer für Drücken einer Taste, Bewegung der Maus auf eine bestimmte Koordinate)

Anmerkung:

Modelle (kognitive Architekturen) verschiedener Autoren unterscheiden sich zum Teil erheblich hinsichtlich Struktur und Detaillierungsgrad



- Vorteile:
 - relativ früh im Designprozess anwendbar
 - keine Nutzer benötigt
- Nachteile:
 - sehr aufwendig
 - begrenzter Anwendungsbereich

	TECHNIQUE	EFFICIENCY	EFFECTIVENESS	EVIDENCE
Technical	HTA	<ul style="list-style-type: none"> ▪ Decomposes complex tasks into subtasks ▪ Complex activities demand extensive hierarchy construction/charting 	<ul style="list-style-type: none"> ▪ Improves problem diagnosis and useful for concurrent operations ▪ Does not account for system dynamics 	MacLean et al., 1991 Annet and Stanton, 2000 Hollan et al., 2000 Shepherd 2001
	GOMS	<ul style="list-style-type: none"> ▪ Requires detailed analysis of keystroke level interaction 	<ul style="list-style-type: none"> ▪ Improves productivity ▪ Not applicable to broader problems ▪ Ignores contextual factors 	Card et al., 1983 Preece et al., 1994 John and Kieras, 1996
Conceptual	CTA	<ul style="list-style-type: none"> ▪ Defines a coherent knowledge representation for the domain being studied ▪ Requires deep engagement with a particular knowledge domain 	<ul style="list-style-type: none"> ▪ Increases the understanding of cognitive aspects of the task ▪ Captures task expertise ▪ Fails to fully incorporate learning, contextual and historical factors 	Barnard and May, 2000 Chipman et al., 2000 Dubois and Shalin, 2000
Work-Process	Activity Theory	<ul style="list-style-type: none"> ▪ Analyzes the activity, not the task, implying a potentially great increase in scope and complexity ▪ Requires near-ethnographic knowledge of culture 	<ul style="list-style-type: none"> ▪ Accounts for learning effects ▪ Extends scope of technology ▪ Requires a high level of abstraction ▪ No disciplined set of methods ▪ Difficult to apply systematically 	Kuutti, 1996 Hollan et al., 2000

Table 1. Efficiency, effectiveness and empirical evidence in task analysis research.

Goals

- Was will der Benutzer erreichen?

Operators

- Welche Aktionen stehen dem Benutzer zur Verfügung?

Methods

- Mit welchen Aktionsfolgen kann ein bestimmtes Ziel erreicht werden?

Selection Rules

- Welche Methode wird zur Erreichung eines Ziels eingesetzt?

Beispiel:

- Goal
 - Fahrt zum Flughafen
- Methods (Alternativen)
 - Taxi, Fahrdienst oder Zug
- Operators
 - Taxi: Nimm das Handy, Wähle die Nummer des Taxidienstes...
- Selection Rules
 1. Zug ist die billigste Methode
 2. Fahrdienst ist schneller und bequemer
 3. Taxi kann spontan gerufen werden und ist schneller

Wieso GOMS?

- **Abschätzung** der **Effizienz** existierender Systeme und Prototypen
- **Prüfung** der **Konsistenz** der Methoden (ähnliche Ziele mit ähnlichen Methoden erreichbar)
- **Sicherstellen**, dass **häufige Ziele** mit **schnellen Methoden** erreicht werden können
- **Entscheidungshilfe** zwischen **verschiedenen Designs**

➤ **Fazit: quantitative Evaluationstechnik**

Vorgehen:

1. Taskbeschreibung in einem Szenario
 - a) Definition des übergeordneten Ziels (=Goal) der Nutzer
 - b) Auflistung der Methoden, mit denen das Ziel erreicht werden kann
 - c) Aufsplittern der Methoden in Sub-Goals
 - d) Auflistung der Methoden, mit denen die Sub-Goals erreicht werden können.
 - e) Rekursive Iteration bis die Operatoren erreicht wurden.
2. Evaluation anhand der Modelldaten
3. Anpassung der Nutzerschnittstelle anhand der Ergebnisse
4. Iteration

Beispiel:

Goal: Schließe ein Fenster in Windows

[Select

Nutze-Menu-Methode

Bewege-Maus-zum-Datei-Menü

Öffne-Menü durch Klicken

Bewege-Maus-zur-Close-Operation

Klicke-Maus-über-der-Close-Operation

Nutze-Key-Shortcut-Methode

Halte-ALT-Taste-Gedrückt

Drücke-F4-Taste

Nutze-Close-Button-Methode

Bewege-Maus-zum-Close-Button

Klicke-auf-Close-Button

Operatoren
einer Methode

Alternative Methoden

Selection- Rules (Beispiele)

Regel 1: Wähle „Nutze-Key-Shortcut-Methode“, wenn bekannt.

Regel 2: Wähle ansonsten die „Nutze-Close-Button-Methode“.

]

Wie gut sind GOMS Analysen?

Predicted execution times

Task	Original interface (seconds)	Revised interface (seconds)	Percent change
1	152.5	94.5	-38
2	199.5	111.5	-44
3	185.2	102.6	-45
4	130.6	87.4	-33
5	139.8	97.8	-30
6	191.0	108.6	-43
7	180.5	105.7	-41
Mean	168.4	101.2	-40

Observed execution times

Task	Original interface (seconds)	Revised interface (seconds)	Percent change
1	71.0	40.6	-43
2	82.6	53.3	-35
3	70.8	40.1	-43
4	55.3	36.4	-34
5	57.4	39.6	-31
6	82.4	51.1	-38
7	74.1	41.6	-44
Mean	70.5	43.2	-39

- Keine exakte Vorhersage der Dauer
- Relativer Unterschied zwischen Methoden wird gut vorhergesagt.

- Vorteile
 - Liefert quantitative Messdaten für die Effizienz
 - Hilft Ergebnisse von Experimenten zu erklären
 - Bei Änderungen des Interface leicht anzupassen
- Nachteile
 - Schwerer als andere Methoden (z.B. Guidelines, Heuristische Evaluation)
 - Benötigt viel Zeit, Einsatz und Fähigkeiten
 - Funktioniert nur für zielgerichtete Tasks
 - Beachtet die anderen Aspekte der Usability (Effektivität, Zufriedenheit) nicht

NGOMSL (Natural GOMS Language) von D. Kieras

- **Idee:** Formale Sprache zur Beschreibung von Operatorabfolgen. (Analog zu höheren Programmiersprachen)
- **Vorteil:** Schränkt Freiheiten beim Modellieren ein, um nicht plausible Operatorsequenzen auszuschließen.
- **Beispiel:**

Method for goal: cut text

Step 1. Accomplish goal: select text.

> Step 2. Retain that the command is CUT, and accomplish goal: issue a command.

Step 3. Return with goal accomplished.

Method for goal: paste text

Step 1. Accomplish goal: select insertion point.

> Step 2. Retain that the command is PASTE, and accomplish goal: issue a command.

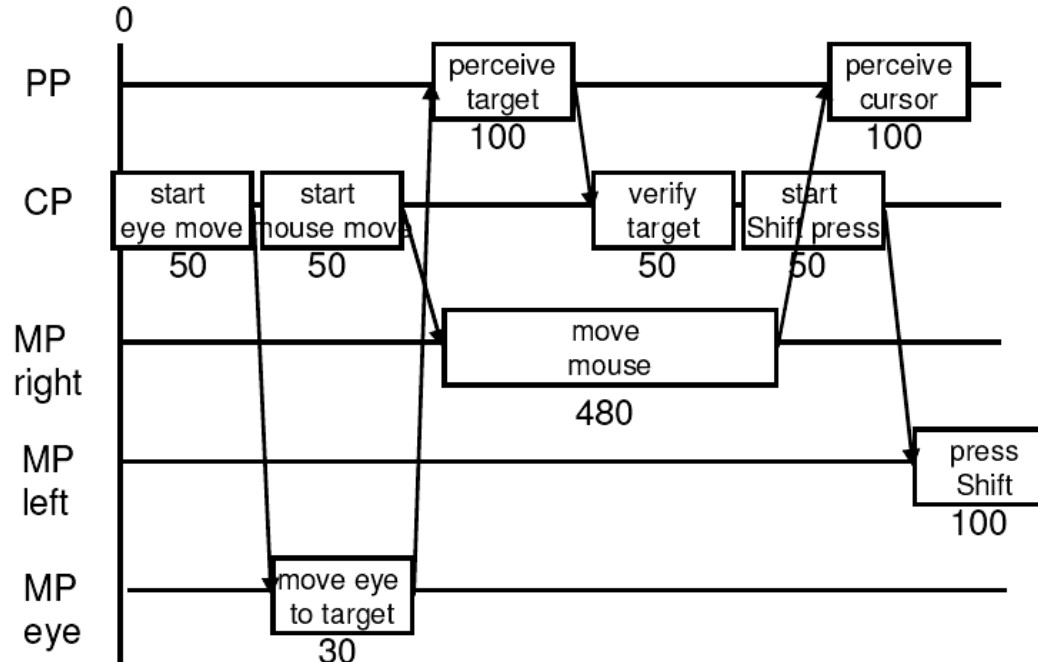
Step 3. Return with goal accomplished.

Problem: Nebenläufige Aufgaben

- **Beobachtung:**
 - Benutzer führen mehrere Aktivitäten gleichzeitig aus.
- **Problem:**
 - GOMS-Analysen werden unter der Annahme durchgeführt, dass der Benutzer eine Aktion nach der anderen durchführt.
 - Annahmen über Aufmerksamkeit und Inhalt im Kurzzeitgedächtnis sind nicht mehr haltbar.
- **Idee:**
 - Erweitere GOMS so, dass auch parallel ausführbare Aktionen modelliert werden können, z.B. Lesen und dabei Maus bewegen.

CPM-GOMS (Cognitive Perceptual Motor GOMS)

- Modellierung parallel ausführbarer Aktionen möglich (z.B. Lesen und dabei Maus bewegen).
- **“kritischer Pfad”**
 - Sequenz mit dem **längsten Pfad** durch den Chart
 - **Abschätzung** der gesamt benötigten Zeit für den Task.



- **Fallbeispiel:**

- Einführung neuer Computer-Arbeitsplätze für den Auskunftsdienst bei der US-Telefongesellschaft NYNEX
- **GOMS-Analyse** versprach eine Effizienzsteigerung von **20%**
- Nach Einführung des Systems jedoch nur Verbesserung von **4%**
- D.h. die Investition hatte sich nicht wie gewünscht ausgezahlt.
- **CPM-GOMS-Analyse** zeigte, dass die vermeintlichen Verbesserungen **nicht** auf dem „**kritischen Pfad**“ lagen und somit **keine Zeiteinsparung** zu erwarten war.

- **Fazit:**

- CPM-GOMS ist genauer als GOMS.
- Optimierungsversuche bei „kritischen Pfaden“ ansetzen.

Modellierung nebenläufiger Handlungen

- CPM-GOMS Analyse setzt genaues Wissen darüber voraus
 - was ein Nutzer gerade **wahrnimmt**
(Aktivität des perzeptuellen Prozessors)
 - wie groß die **geistige Belastung** zu einem bestimmten Zeitpunkt ist
(Aktivität des kognitiven Prozessors)
 - welche **motorischen Aktionen** zu einem bestimmten Zeitpunkt
ausgeführt werden (Aktivität des motorischen Prozessors)
- Problem:
 - situativer Kontext der Aufgabenausführung meist nicht bekannt
 - Analysen enthalten viele Annahmen und Unsicherheiten

- Vereinfachte Version von GOMS
 - Nur Operatoren
 - Keine Ziele
 - Keine Methoden
 - Keine Auswahlregeln
- Sagt voraus wie lange es dauert einen Task auszuführen

- Tastaturbasiertes, physikalisches Modell
- Zerlegt die Ausführung eines Tasks in primitive Operatoren:
 - **Motorische Operatoren**
 - **K** Drücken einer Taste auf der Tastatur
 - **B** Drücken der Maustaste
 - **P** Zeigen mit der Maus
 - **D** Zeichnen
 - **H** Bewegung der Hand von Maus zur Tastatur bzw. umgekehrt
 - **Mentaler Operator**
 - **M** Mentale Vorbereitung
 - **Systemreaktionsoperator**
 - **S** Systemreaktion
(Zeit in der keine Aktion des Nutzers möglich ist)

Interface-Operatoren (Zeiten basieren auf empirischen Studien)

Operator	Bemerkungen	Zeit (in s)
K	Drücken einer Taste	
	ausgebildete Datentypistin	0.12
	erfahrener Benutzer	0.2
	kompletter Anfänger	1.20
B	Drücken einer Maustaste	
	Hoch oder runter	0.10
	Klick	0.20
P	Zeigen mit der Maus	
	Fitts Law	$0.05 + 0.15 * \log_2(D/S + 1)$
	Durchschnittliche Bewegung	1.10

Interface-Operatoren (Zeiten basieren auf empirischen Studien)

Operator	Bemerkungen	Zeit(s)
H	Handbewegung zur Tastatur und von der Tastatur weg	0.40
D	Zeichnen	domänenabhängig
M	mentale Vorbereitung	1.35
S	Rückmeldung vom System → Messung	-

Beispiel: Verwendung des Memorizing-Operators M

- Aufgabe: Eintippen des Wortes „HALLO“
- GOMS-Modellierung:

$$\begin{array}{ccccccccc} M & K & M & K & M & K & M & K & M & K \\ H & A & L & L & O & & & & & \end{array} = 5 \cdot 1,35 + 5 \cdot 0,2 \text{ s} = 7,75 \text{ s}$$

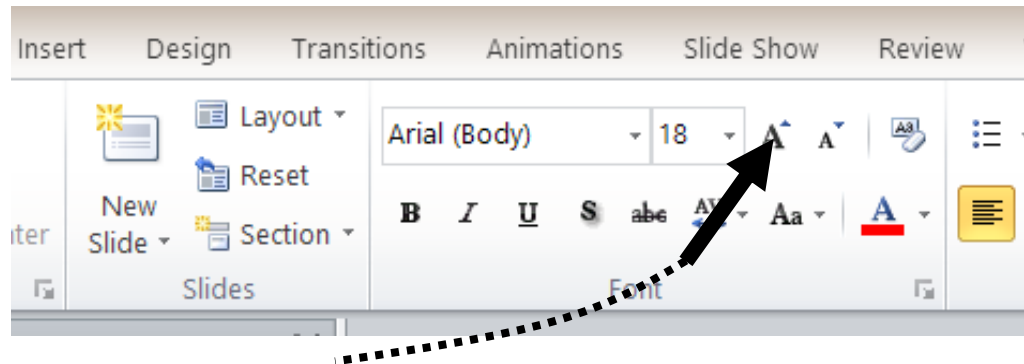
- Problem: Nutzer denkt nicht vor jedem Tastendruck 1.35s nach, sondern tippt nach kurzem Nachdenken das gesamte Wort ein.
- Nach Anwendung der Selektionsregel:

$$\begin{array}{ccccccc} M & K & K & K & K & K \\ H & A & L & L & O & \end{array} = 1,35 + 5 \cdot 0,2 \text{ s} = 2,35 \text{ s}$$

- **Zweck:** Regeln geben an, unter welchen Umständen welche Operatoren zum Zuge kommen.
- **Beispiel:** Verwendung des Memorizing Operators M

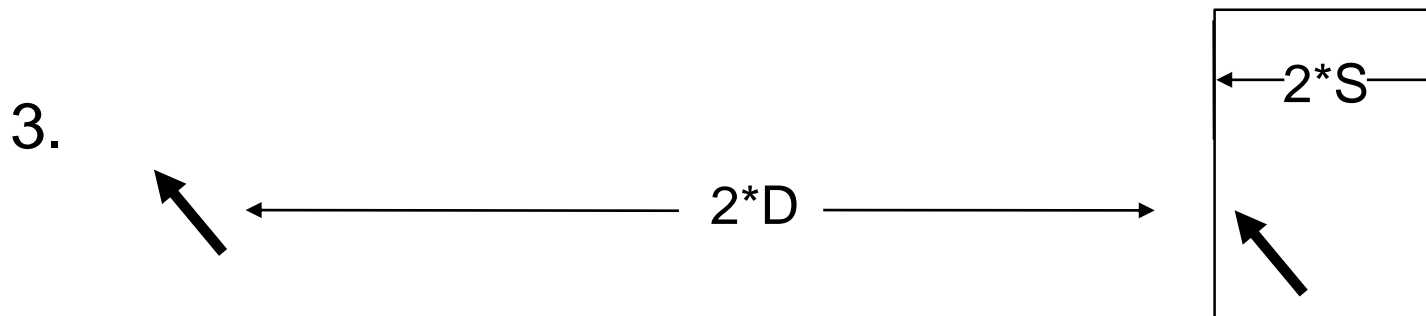
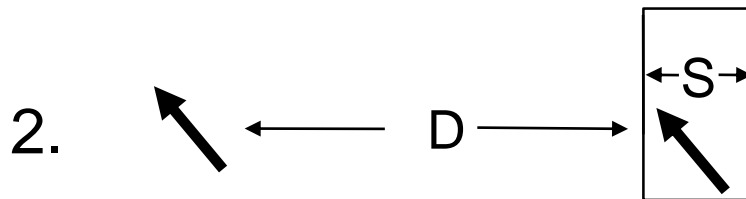
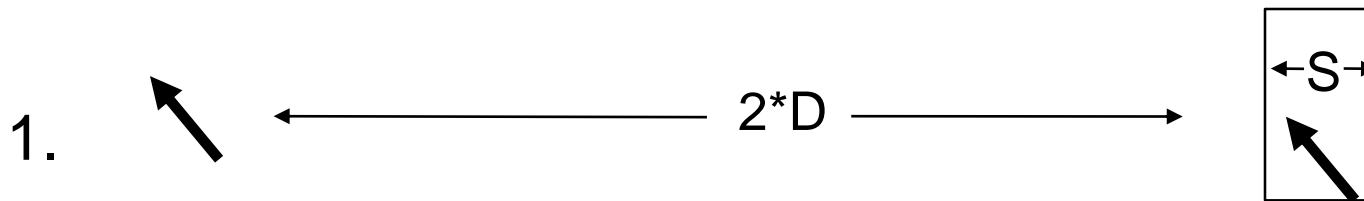
- R0: M wird vor allen Tastenanschlägen (Operator K) eingefügt
- R1: Wird eine Zeichenkette mit n Ks eingegeben, so kann man die Sequenz M K M K ... M K vereinfachen zu M K K ... K
- R2: Ist K ein überflüssiger Begrenzer einer Eingabe (z.B. Return-Taste), so entfällt das M vor K
- R3: M wird vor jedem Zeigeoperator P eingefügt, der zur Auswahl eines Befehls dient, jedoch nicht vor einem Operator P, der auf Argumente des Befehls zeigt.
- R4: Wird ein P ausgeführt, um auf ein Texteingabefeld zu gelangen, so vereinfacht sich die Sequenz M P M K zu M P K

- **Frage:** Wie kann man die Zeit abschätzen, die ein Benutzer benötigt, um mit einem Zeigegerät (Maus, Stift, Hand usw.) von einer Ausgangsposition auf ein Objekt zu zeigen?



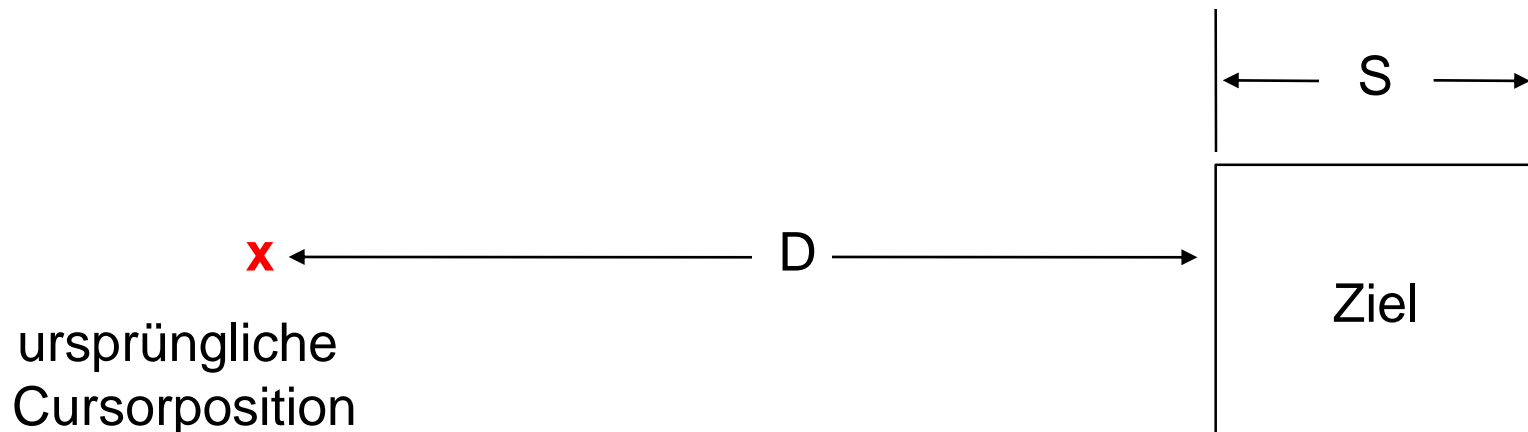
- Randbedingungen
 - Man betrachtet nur die am schnellsten ausführbare Geste (d.h. keine mutwilligen Schnörkel)

- Welche Aufgabe zur Positionierung der Maus geht am schnellsten?



Allgemeine Formulierung



- „Die Zeit, die ein Cursor für die Bewegung zu einem Ziel benötigt, hängt von der Größe des Zielobjekts S und der Entfernung D ab.“



- Anwendbar für Interaktionen mit Maus, mit Touch mittels Finger oder Stift und im geringen Maße mit Joysticks

$$Mt = a + b \log_2(\underbrace{D/S + 1}_{ID})$$

ID: Index of Difficulty

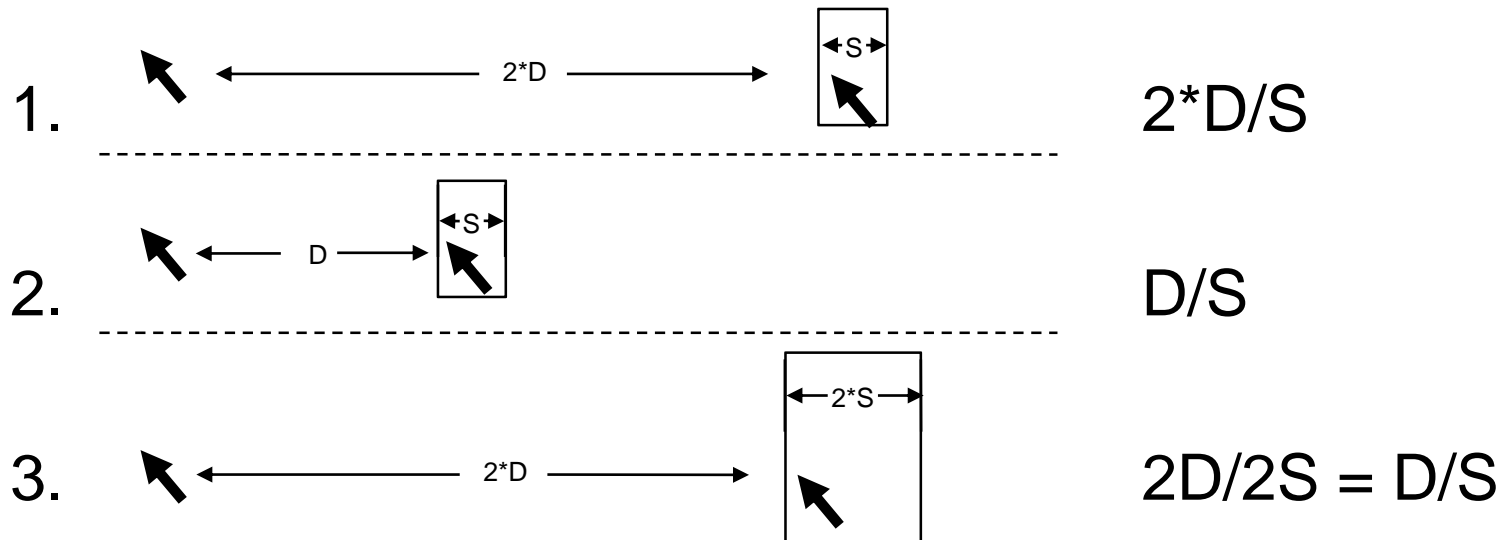
- Mt = Zeit, um Ziel zu berühren
- D = Entfernung zum Ziel
- S = Größe des Ziels
 - Eindimensionaler Fall (Quadrat): 
 - S = Seitenlänge
 - Zweidimensionaler Fall (Rechteck): 
 - S = **kleineres** der horizontalen und vertikalen Maße des Ziels (Mackenzie 1995)
- a, b = Konstanten, die vom Zeigehilfsmittel abhängen

Fitts' Law wurde in zahlreichen Untersuchungen bestätigt.

Für unterschiedliche Zeigemittel wurden unterschiedliche Werte für a und b (bzw. für den Schwierigkeitsgrad IP) gefunden.

Device	Study	<i>IP (bits/s)</i>
Hand	Fitts (1954)	10.6
Mouse	Card, English, & Burr (1978)	10.4
Joystick	Card, English, & Burr (1978)	5.0
Trackball	Epps (1986)	2.9
Touchpad	Epps (1986)	1.6
Eyetracker	Ware & Mikaelian (1987)	13.7

- Welche Aufgabe zur Positionierung der Maus geht am schnellsten? Fitts' Law: $Mt = a + b \log_2(D/S + 1)$



Antwort: 2 und 3 gehen am schnellsten

Beispiel: Schließen eines Fensters unter Windows
(Ausgangsposition: Hand auf Maus)

- **Nutze-Key-Shortcut-Methode**

M H[zur Tastatur] K[Alt Taste] K[F4 Taste]

Zeit: 1.35s + 0.4s + 0.2s + 0.2s = **2.15 s**

- **Nutze-Close-Button-Methode**

M P[zum Icon] B[Klick]

Zeit: 1.35s + 1.1s + 0.2s = **2.65 s**

- **Nutze-Menu-Methode**

M P[zur Leiste] B[Klick] M P[zur Option] B[Klick]

Zeit: 1.35s + 1.1s + 0.2s + 1.35s + ??? + 0.2s = ??? s

P[zur Option]

Fitts' Law: $Mt = a + b \log_2(D/S + 1)$

- a (230ms) und b (166ms) für Maus-Bewegung gegeben
- Höhe des Menüs: **80 mm (=D)**
- Höhe der Option: **5 mm (=S)**
- Zeit für Cursorbewegung:

$$Mt = 230 \text{ ms} + 166 \text{ ms} * \log_2(80/5 + 1) = \mathbf{1.62 \text{ s}}$$

Beispiel: Schließen eines Fensters unter Windows:

- **Nutze-Key-Shortcut-Methode**

M H[zur Tastatur] K[Alt Taste] K[F4 Taste]
Zeit: 1.35s + 0.4s + 0.2s + 0.2s = **2.15 s**

- **Nutze-Close-Button-Methode**

M P[zum Icon] B[Klick]
Zeit: 1.35s + 1.1s + 0.2s = **2.65 s**

- **Nutze-Menu-Methode**

M P[zur Leiste] B[Klick] M P[zur Option] B[Klick]
Zeit: 1.35s + 1.1s + 0.2s + 1.35s + 1.62s + 0.2s = **5.82 s**

Wieso macht die Nutzung der Ecken Sinn?

Zwischen Apps wechseln

Bewegen Sie Ihren Mauszeiger ganz in die Ecke, und klicken Sie, um zur zuletzt verwendeten App zurückzukehren.

Suche, Einstellungen und andere Charms

Bewegen Sie Ihren Mauszeiger ganz in die Ecke, bewegen Sie ihn am Rand nach unten, und klicken Sie.

„Start“ befindet sich hier

Bewegen Sie Ihren Mauszeiger ganz in die Ecke, und klicken Sie.

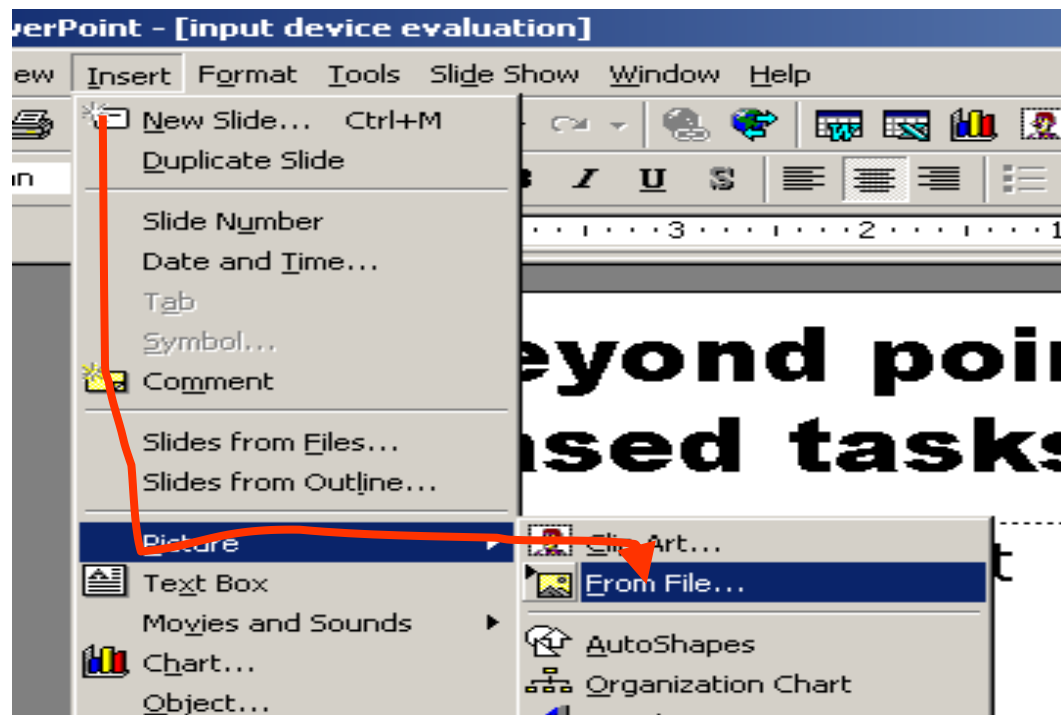


Fitts' Law:
$$Mt = a + b \log_2(D/S + 1)$$

Problem: Komplexe Zeigeoperationen

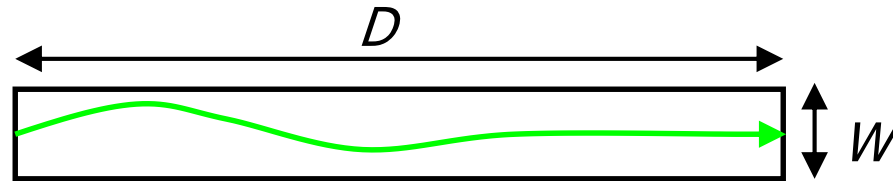
In grafischen Oberflächen gibt es auch schwierigere Zeigeoperationen.

Beispiel: Abfahren einer bestimmten Kurve

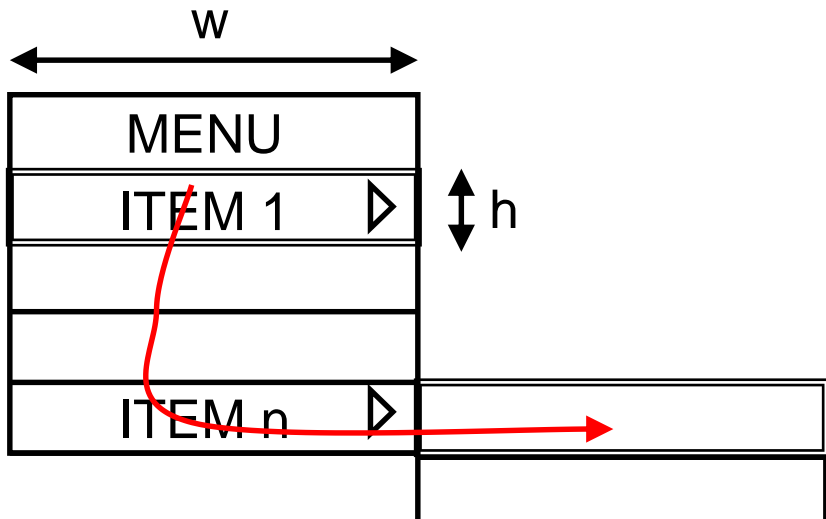


- Tunnel mit konstanter Breite (MT = Movement Time):

$$MT = a + b \cdot \frac{D}{W}$$



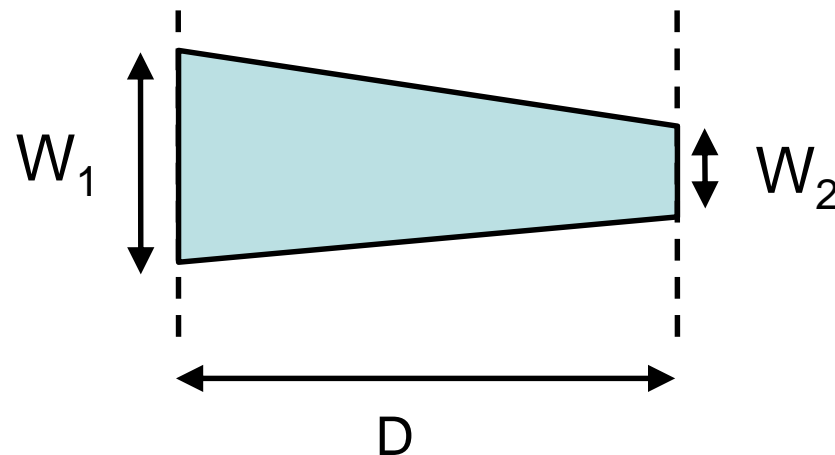
- Zeitaufwand bei Menüauswahl



$$MT = \underbrace{a + b \cdot \frac{n \cdot h}{w}}_{\text{Vertikaler Pfad}} + \underbrace{a + b \cdot \frac{w}{h}}_{\text{Horizontaler Pfad}}$$

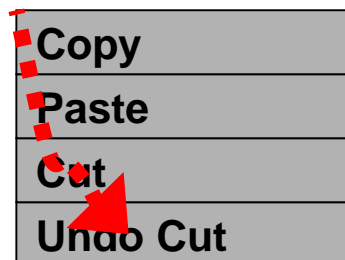
- Sich verengender Tunnel der Länge D (ID = Index of Difficulty)

$$ID_{NT} = \frac{D}{W_2 - W_1} \cdot \ln \frac{W_2}{W_1}$$



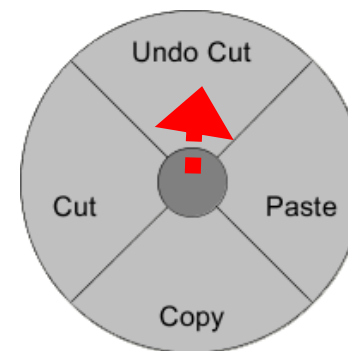
- GOMS Analyse
 - Mit Fitts' Law lassen sich die Ausführungszeiten der Zeigeoperationen wesentlich spezifischer für ein bestimmtes Interface angeben.
 - genauere Abschätzung der Gesamteffizienz
- Menüdesign
 - Beispiel: Kontextmenü

Listen-Menü



vs.

Pie-Menü



Anwendung von Fitts' Law - Großbildschirme



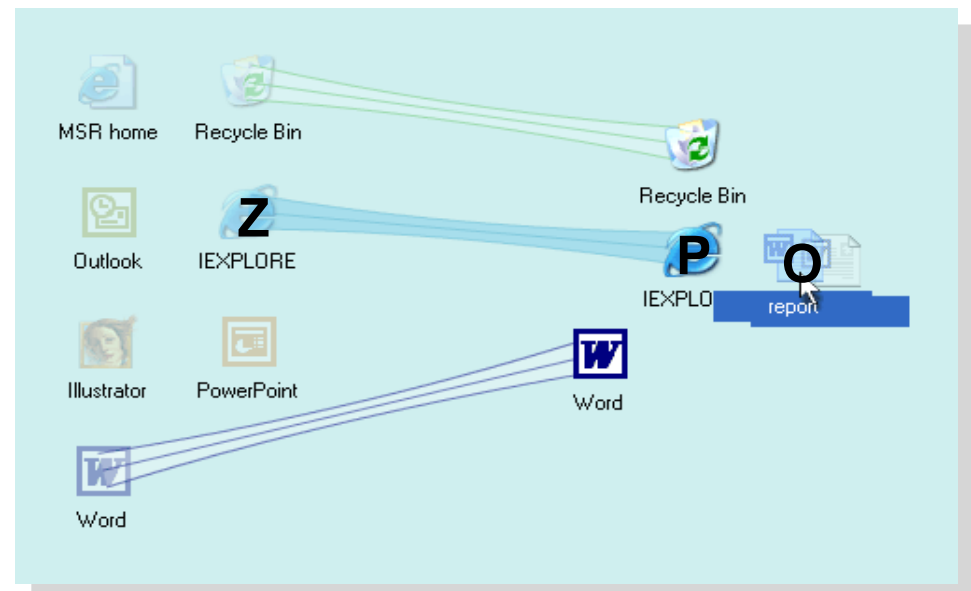
Drag-and-Pop Technik (Baudisch, Microsoft Research)

- Idee: Anstatt ein **Objekt O** auf ein **Ziel Z** zu ziehen, werden bei der Selektion von O **wahrscheinliche Ziele** in der Nachbarschaft **temporär dubliziert (Proxy P)**. Dann wird O auf P gezogen.

➤ kürzerer Weg

- Demo:

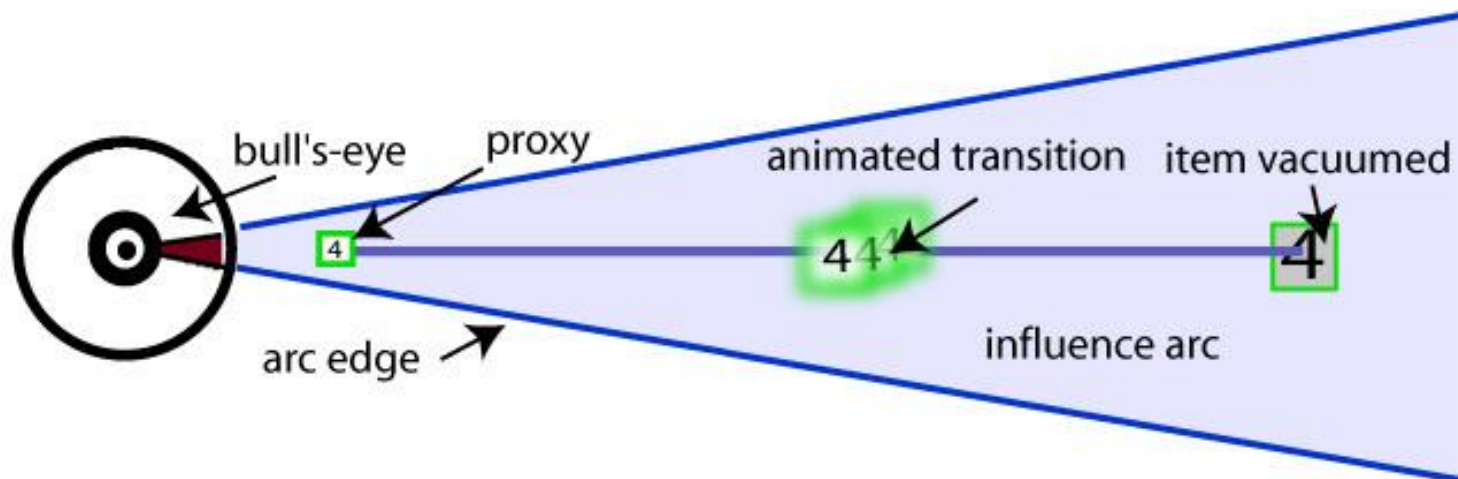
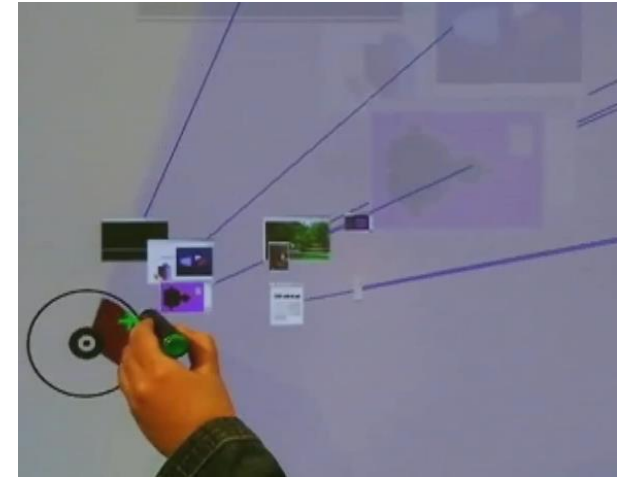
<http://www.patrickbaudisch.com/projects/dragandpop/index.html>



Vacuum-Technik

(Bezerianos and Balakrishnan, 2005)

- Idee: ähnlich wie bei Drag-and-Pop wird mit Proxies gearbeitet.
- Nutzer legt jedoch Richtung fest, aus der Proxies “angesaugt” werden.



Quelle: Homepage von R. Balakrishnan <http://www.dgp.toronto.edu/~ravin/>

- **Beobachtung:**
Bei Zeigeaufgaben spielt die Übung eine große Rolle. Geübte Nutzer einer Maus sind wesentlich schneller, als ungeübte.
- **Power Law of Practice**
Versucht, den Geschwindigkeitsgewinn durch Training in einer Berechnungsregel zu formulieren.

$$T_n = T_1 * n^{-0.4}$$

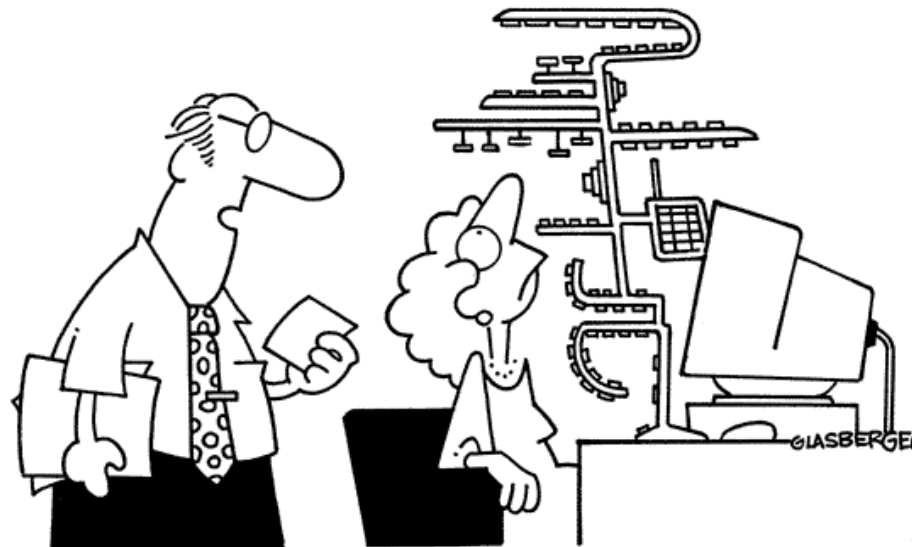
(mit empirisch ermitteltem Exponent 0.4)

- T_n = Ausführungszeit für die n-te Runde
- n = n-te Trainingsrunde
- T_1 = Ausführungszeit der ersten Runde

Anmerkungen

- Anwendbar für sensorische und motorische Aufgaben, die sich gut trainieren lassen.
- Ungeeignet zur Quantifizierung des Zeitbedarfs kognitiver Aufgaben

Copyright 2002 by Randy Glasbergen.
www.glasbergen.com



"It's an ergonomic keyboard. Once you learn how to use it, it will increase your speed by six percent!"

- **Motivation:**

- Aufgabe: Auswahl unter n gleichwahrscheinlichen Optionen
- Umso schwieriger, desto mehr Optionen man hat.
- Schwierigkeit spiegelt sich in der benötigten Entscheidungszeit wider.

- **Hicks Law:**

- Entscheidungszeit T verhält sich proportional zum 2er-Logarithmus der Anzahl der Optionen

$$T = a + b \log_2(n + 1)$$

(a und b sind Konstanten der Regressionsgeraden)

- Bei unterschiedlichen Wahrscheinlichkeit für Optionen gilt:

$$T = a + b * \sum p(i) \log_2(1 / p(i) + 1)$$

Aufgabe: Auswahl eines Designs für ein Menü mit 8 Optionen

Variante 1

Opt 1
Opt 2
Opt 3
Opt 4
Opt 5
Opt 6
Opt 7
Opt 8

$$a + b * \log_2(9) \approx \mathbf{a + 3b}$$

Variante 2

Opt 1
Opt 2
Opt 3
Opt 4

Opt 5
Opt 6
Opt 7
Opt 8

$$\begin{aligned} & a + b * \log_2(5) + a + b * \log_2(5) \\ &= 2(a + b * \log_2(5)) \approx \mathbf{2a + 4b} \end{aligned}$$

- Hicks Law: Variante 1 schneller
- Jedoch auch andere Faktoren wichtig:
z.B. verfügbarer Platz, funktionale Gruppierungen, Ästhetik

- Vorteile:
 - Ein ausgearbeitetes Modell kann zur Bewertung sehr unterschiedlicher Entwürfe verwendet werden.
 - Vergleichsweise objektiver Effizienzvergleich von Interfaces
 - keine aufwändigen Tests mit Benutzern nötig
 - Analytischer Ansatz zwingt den Designer, sich detaillierte Gedanken über die vom Nutzer tatsächlich auszuführenden Operationen zu machen.
- Einschränkungen:
 - nur anwendbar bei detaillierter Kenntnis des Arbeitsablaufs
 - aufwändig bei heterogener Nutzerzielgruppe