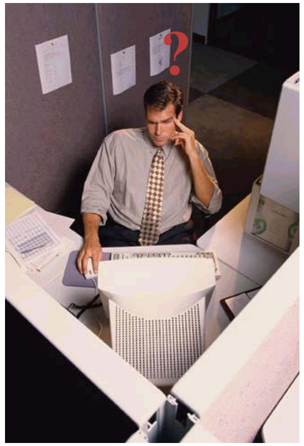
# Human-Computer Interaction

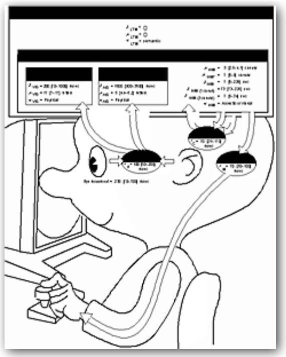
**Session 4:**
Psychological basis -- Reasoning and Acting



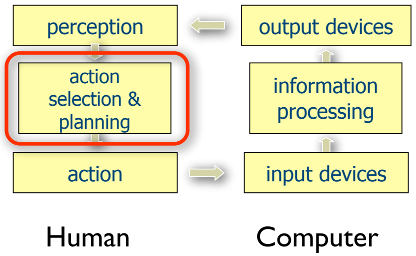
## „What am I going to do next?“





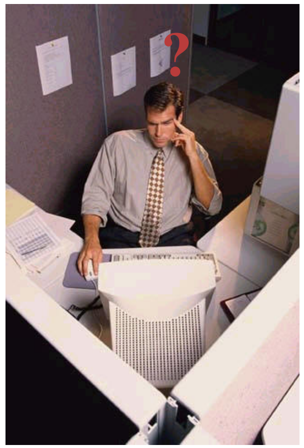


HCI = two information processors coupled in goal-directed action.

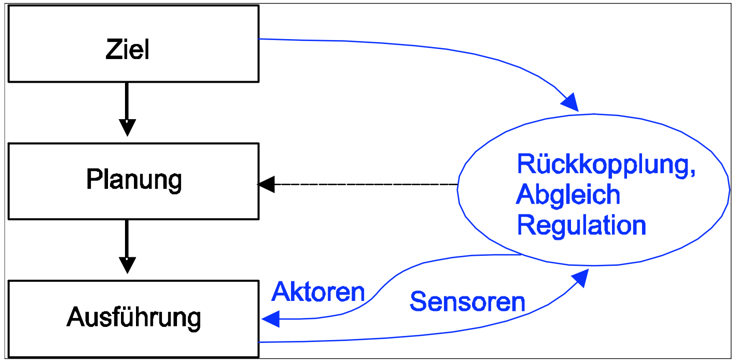





## „Ok, I did this and got that.
## What am I going to do next?“



## How users act




Fortlaufender Prozess der „Handlungsregulation“



## Levels of Action (Hacker 1986)

- Conscious action: aware and controlled regulation
  - control based on declarative knowledge in working memory
  - only one cognitive action at a time, possibly task switching
- Routinized action: flexible patterns & simple regulation
  - few routinized actions in parallel possible
  - only little attention needed (e.g. ironing while watching TV)
- Automatized action: automatic sensorimotor regulation
  - in prallel, no conscious control needed, no distraction
  - based on implicit memory
  - *Example*: blind typing, car-driving



## Conscious Action

Controlled regulation by means of...

- reasoning (deductive, abductive, inductive)
- deliberation & problem-solving
- (re-)planing & acting & monitoring



## Deductive Reasoning

- derive logically necessary conclusion from premises
  
  e.g. If it is Friday, then she will go to work
  → It is Friday, therefore she will go to work.
- not necessarily true (in the real world):
  
  e.g. If it is raining, then the ground is dry
  → It is raining, therefore the ground is dry
- truth and logical validity can clash
  
  e.g. Some people are babies. Some babies cry.
  → Some people cry

People are aware of these shortcomings: make „uncertain“ conclusions, bring world knowledge to bear

# Inductive Reasoning

- generalize from cases seen to the general case
  - e.g. all elephants we have seen have trunks, therefore all elephants have trunks
- inherently unreliable
  - can only prove false not true (you never know)
  - but useful
- humans have a confirmation bias
  - tend to neglect negative evidence
  - attempt to make a claim (inductive) and to confirm it, while forgetting that it is also important to test and falsify it



# Wason's cards

| 7 | E | 4 | K |
|---|---|---|---|

If a card has a vowel on one side, it has an even number on the other

Which cards do you <u>at least</u> need to turn over to prove or disprove this?



# Wason's cards - another version

- Cards have an age on one side and a beverage on the other
- *„If you are drinking alcohol then you must be over 18"*

| 16 | beer | 18 | coke |
|---|---|---|---|

Which cards do you <u>at least</u> need to turn over to prove or disprove this?



# Abductive reasoning

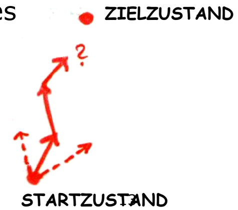
- reasoning from effect/symptom to possible causes

  Sam drives fast when he is drunk.

  ->If I see Sam driving fast, I assume he is drunk.
- primary way to form hypotheses (diagnoses) and explanations about the world
- unreliable, can lead to false explanations
- need to be combined with hypothesis testing

# Problem-solving

- process of finding a solution to an unfamiliar task using domain knowledge
- problem space theory
  - problem space out of problem states
  - search generates and tests states, using legal operators
  - heuristics helps to select operators and assess states
  - operates within human information processing system, i.e., suffers from STM limits, retrieval difficulties, etc.
- different problem solving (=search) strategies
  - forward search (start --> goal)
  - backward search (goal <-- start)
  - means-end analysis (mixture), psychologically most plausible




# Problem-solving

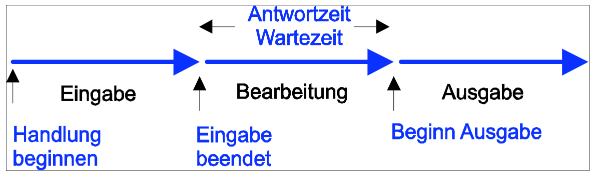
- users will tend to apply associations & analogical mapping
  - use knowledge of similar problems from similar domains for problem in new domain
  - difficult if domains are semantically different, sometimes overlooked
- can a user be skilled at problem-solving?
  - skilled cognitive processing characterized by proper chunking
  - optimizes working memory *AND* problem-solving
    - chess masters *plan* not single moves but „manoeuvers“
  - conceptual grouping of operator applications that solve sub-problems



# Conscious vs. sensorimotor action


- what dominates an interaction?
  - *Example*: fully aware action selection → user “thinks” most of the time instead of acting (e.g., stock exchange)
- graphical interfaces dominated by sensorimotor actions
  - continuously invoke small, physical interaction between human and maschine
    - doing rather than thinking
    - recognition rather than recall
  - time and effectiveness heavily influenced by speed and accuracy of sensorimotor user actions as well as respone times (latency) of the system

# System response time - latency




Waiting for completion of output is perceived as part of the respone time of the system (increasing felt latency)

- trick: „progress bar“

# Latency - reference values

If the response times of an interactive system is roughly

- ...up to 1 second: immediate (instantaneous)
- ...up to 5 seconds: slow
- ...up to 10 seconds: strongly delayed
- ...more than 10 seconds: no answer expected anymore, user is annoyed and drops out (e.g. changes website)

- Values differ between use cases, user groups, previous experiences of the user




Volume 23, Issue 5, Pages 385-564 (September 2011)

**40 years of searching for the *best* computer system response time**
*Pages 555-564*
Jim Dabrowski, Ethan V. Munson
Show preview | Related articles | Related reference work articles

**Highlights**

► We survey over 40 publications in the area of computer system response time (SRT). ► We show that consistent results are only found for low-level tasks. ► We show that previous models by Shneiderman and Seow have limited utility. ► We propose that control tasks and conversational tasks are fundamentally different for SRT.



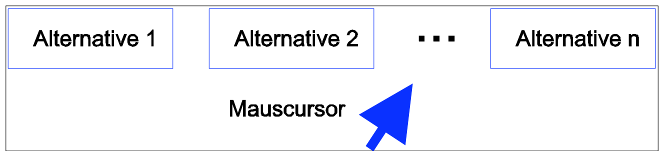
# User response time

Response time = reaction time + movement time

- both depend on age, fitness, etc.
- reaction time also depends on stimulus type
  - visual ~ 200ms
  - auditory ~ 150 ms
  - pain ~ 700ms
  - combined ~ quickest response!
- reduced reaction time decreases accuracy in the unskilled operator (not in the skilled operator)



# Reaction time: Hick's Law




Time to select a target [ms] = a + b * ld (n + 1)

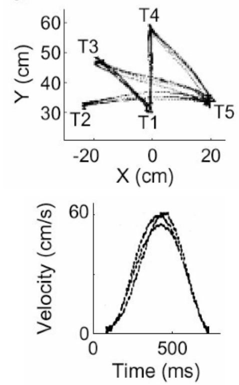
- presumes that alternatives are recognized as such
- n = number of alternative targets (distractors)
- a, b constants; reduced by learning
- if alternatives are picked with different probability:
  time [ms] = a + b * Summe( p(i) * ld (1/p(i) + 1) )
  - with p(i) = probability of selecting target i

# Hick's Law – conclusions

- selecting among complex alternatives takes longer than among simple alternatives
- selecting from a large number of alternatives that are present at the same time is faster than selecting from a nested structure with fewer alternatives each
  - *Example*: 1 menu with 8 entries vs. 2 menus with 4 entries each ld(8+1)=3,17 < 2 ld(4+1)=4,64
  - conforms studies on menu structures
- subject to limitations due to screen size, capacity of STM, ...



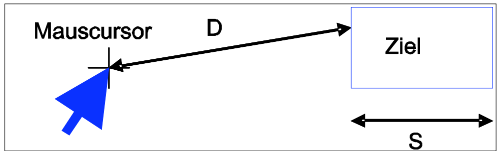
# Movement time




- goal-directed hand movement
  - linear or curved segments
  - bell-shaped velocity profiles
  - in 3D constant plane of movement
- movement time depends on difficulty of the movement
  - distance to target (D)
  - size of target (S)



# Fitts's Law




**Paul M. Fitts**
(1912–1965)

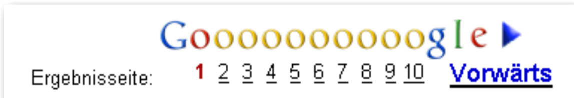
time to position [ms] = a + b * ld (D/S + 1)

- presumes the target is recognized as such
- a, b constants
  - empirically determined, common values:
  - a = 50 (for constant search time)
  - b = 150 (for scaling)



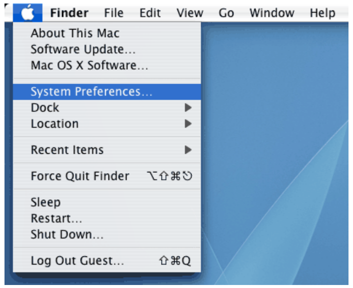
# Fitts's Law – conclusions

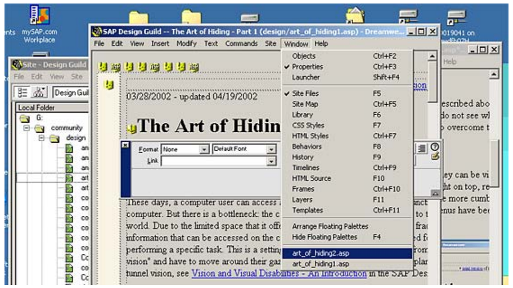
- targets need to be recognized and found and then need to to be hit, don't visualize them too small
- in a continuous, coherent activity, don't put the targets that need to be hit too far away from each other
  - users should not have to hit distant targets shortly one after the other
  - what belongs together should be placed together
- Place targets that are often needed and looked for, always at the same locations



## Fitts‘s Law - example

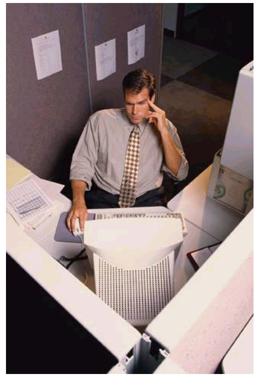
- why are Mac menus selected faster that Windows menus?
  - always only one visible
  - always at the same position at the top part of the screen, reduced search time, can‘t be missed (no over-shooting)



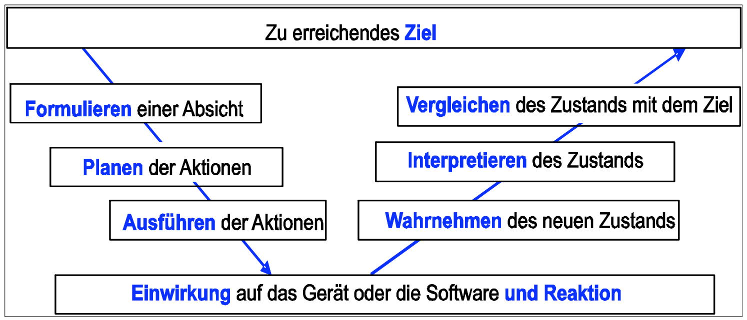


**How do all these cognitive levels of action control and regulation go together when humans interact with a machine?**





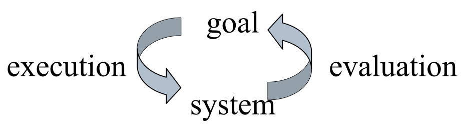
## Stages of Action (D. Norman, 1988)




(Damm 2005)



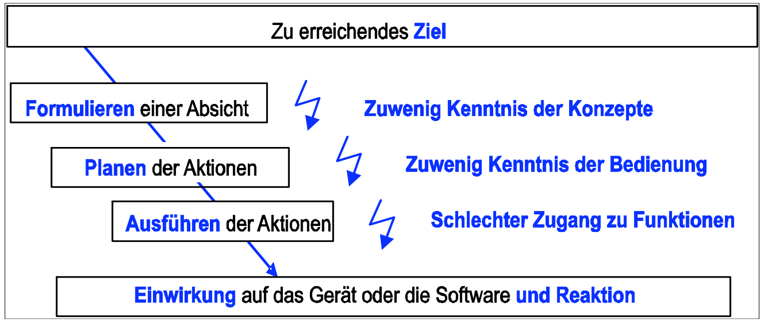
## Example




You are sitting reading as evening falls

| | |
|---|---|
| Goal | - need more light |
| Intention | - switch on desk lamp or ask for it or... |
| Actions | - reach over, press lamp switch |
| Result | - light is either on or off |
| Interpret | - light is off? Maybe bulb has blown |
| → goals | - change bulb |
| Evaluate | - light is on? Is it enough? |
| → goals | - switch on main ceiling light too |

# Problem 1: „Gulf of Execution"



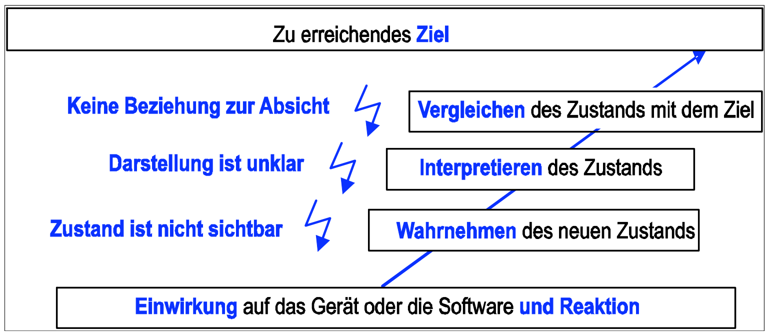



# Example


Intention –
I don't want to see this warning anymore, and I don't want cookies to be stored at all!

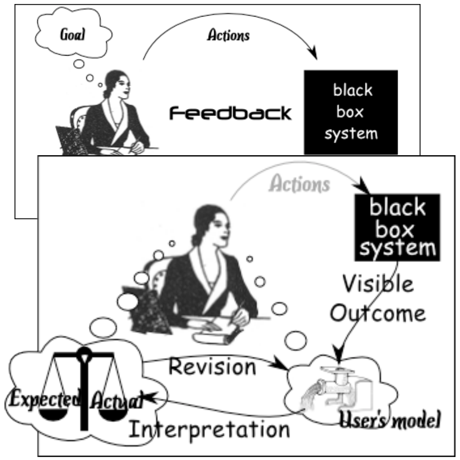
No suitable action offered for *both* goals of the user

# Problem 2: „Gulf of Evaluation"






# Problem 2: „Gulf of Evaluation"




Processing feedback of the system happens in the context of the user's mental model

- interpretation: model provides new state of the system as explanation
- revision: model does not - is adapted to accommodate new information

# Mental models

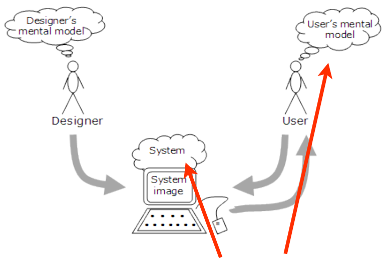
- "*In interacting with the environment, with others, and with the artifacts of technology, people form internal, mental models of themselves and of the things with which they are interacting. These models provide predictive and explanatory power for understanding the interaction.*"
  -Norman (in Gentner & Stevens, 1983)

- first used by Craik (1943), rennaisance in 80's in Cognitive Science and then in HCI (Johnson-Laird, Gentner & Stevens)
  - structural models: set of beliefs about how a system works
  - functional models (a.k.a. task-action mapping models): procedural knowledge about how to use the system



# Mental models

- for HCI practitioners: a set of beliefs about how a system works, humans interact with systems based on these beliefs (Norman, 1988)
- models involved in HCI (cf. Norman 1988)
  - System model: The actual way that a system works from the programmer's perspective
  - User's Mental Model: The way the user perceives that the system works
  - Designer's model: The way the designer represents the system to the user, creating a „system image"




Mismatches lead to interaction problems



# Conclusions for HCI

From the `stages of action´ point of view, in an ideal system:

- the reaction of the system and its state are recognizable at all times and easily interpretable
- a displayed or indicated new state is easily comparable with the goals of the user
- transforming goals into intentions and operating actions is as easy as possible
- possible actions are determinable in each situation
- actions can be executed easily and robustly



# Conclusions for HCI (cont'd)

From the `stages of action´ point of view, in an ideal system

there is an underlying consistent, conceptual system model, and this model is easily recognizable or deducible from the designer's model such that the user can build and maintain an appropriate mental model

➔ make interface design either consistent with people's natural mental models about computers, the environment, and everyday objects, or provide cues that help users create new, accurate mental models

# Use metaphors

- relate computing to other real-world activity
- enable analogical mapping, evoke a mental model of the system's structure and functions
- must be consistent and tap on user's actual experiences

Facilitates learning and retention of the interface


L.Allison 1995

# Interaction metaphors around

- Desktop metaphor: currently predominant
- Book metaphor: for big documents, e.g. hypertext
- Filing cabinets: for online documentation, system settings, etc.
- Office metaphor: for collections of programs/tools
- Library metaphor: for large collections of documents
- Building metaphors, etc.: for virtual worlds
- Agent metaphor: for autonomy and intelligence
- Humanoid metaphor: for natural communication
- ...

- Composite metaphors: e.g. office + file cabinet + desktop

# Use affordances

*"The affordances of the environment are what it offers the animal, what it provides or furnishes, either for good or ill"* (Gibson, p.127)

*"refers to the properties of objects -- what sorts of operations and manipulations can be done to a particular object"*
(D. A. Norman 1988)

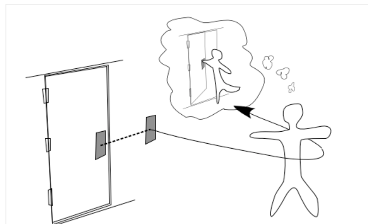
A „door handle" affords pushing or pulling, a „chair" affords support

# Perceived affordances

- perceived affordances: the extent to which users perceive an object's affordance (by its design)
- enable intuitive use

mug handle

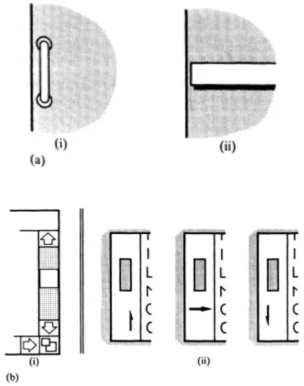
'affords' grasping

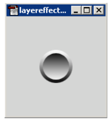


panel "Affords" person to push

# Perceived affordances

- physical objects
  - by their shape and size suggest actions (pick up, twist, throw…)
- virtual screen objects
  - button–like object 'affords' mouse click
  - physical-like objects suggest use
- `enculturated´ affordances
  - icons 'afford' clicking
  - … or even double clicking, not at all like real buttons!



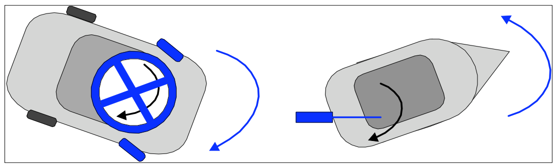




# When an affordance is not enough…




# Mapping

- of the afforded function of an interaction element onto the controlled function/effects
  - natural mapping: direct transformation (e.g., steering wheel movement → car movement)
  - problems when mapping between affordance and effect are indirect (e.g. mouse) or even counter-intuitive (e.g. rudder)





# Next session

How to build interfaces and systems for human users?

- User interface styles and technology