Human Computer Interaction Summary HS23

Written by Benjamin Gantenbein
bgantenbein@ethz.ch

1 Introduction

Goal of course is to understand principles of user-centered design and being able to apply these to practice. Learn about the basic notions of Computational Design in HCI context.

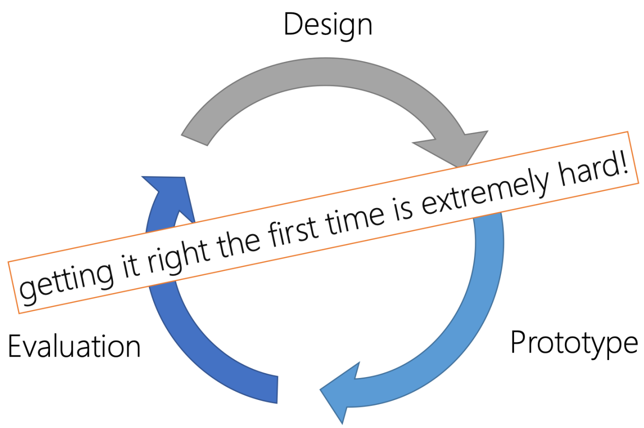
Moore's Law

Computational power grows exponentially. Transistor count doubles every two years. Also with RAM and pixel densities. However: Human capabilities stay stable.

Good System design Accounts for human capabilities, human error and exceptional circumstances.

Human Computer Interaction

Concerned with design, evaluation and implementation of interactive computing systems for human use.



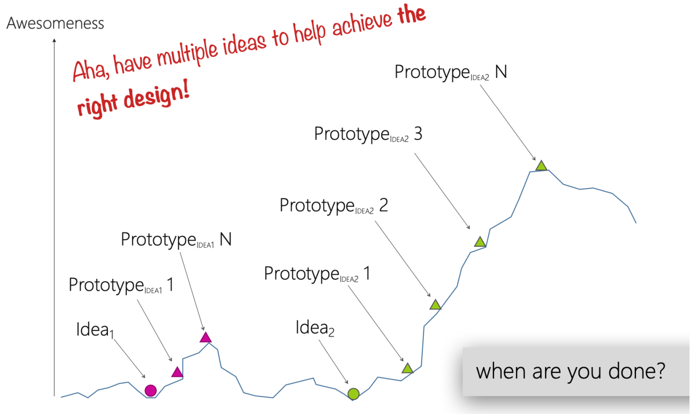
Formative: understand problem and user to inform our design.

Evaluative: understand how well the design works. Also detects mistakes in design.

2 User-centered Design

Design Intention vs User Needs

Prototyping as an interative process



Does the design work properly in the context of use? If not fix the problems and carry out more tests.

Early focus on users and tasks:

Cognitive, behavioral, anthropomorphic AND attitudinal characteristics.

Empirical Measurement:

Observe user's reactions and performance in scenarios, manuals simulations and prototypes, record and analyze.

Root-Cause Analysis

Problems need to be discovered (find the right problem to solve, not any problem to solve) and find the right solution to it.

Need finding

Users rarely know what they want. They cannot imagine what is possible. Instead look at tasks, context:

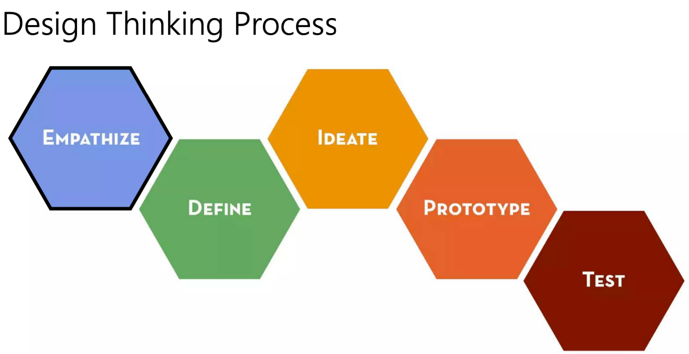
- What information needed?
- Identify collaborators
- Why is task achieved the way it is?
- Identify tasks in existing behavior
- Identify tasks in future scenarios

We ourselves are not representative of the typical user. To learn about customers we conduct interviews, self-reports and logging/analytics. We also observe users performing tasks and understand their cognition.

Understanding the User

Active observation is not knowing yet what we are looking for.

- Immerse
- Observe
- Engage



Goals of Need finding

- Distill useful and actionable insights
- Make meaning from needfinding data
- reframe problem to guide solution search

We start with closed ended questions and move to open ended questions: "What's and why's of feelings". Engage people in their environment. The goal is to find inspiring users, that surprise us and bring us to game-changing ideas.

Needs vs. features vs. requirements

Requirements are goals that the system needs to accomplish. Solutions fulfill requirements. What does the user want to accomplish and how is he doing it? What would they like to be doing? What are they currently disliking? For what is the system usable and what tasks will it support? Answering these questions will make the system more usable.

There are tons of methods to needfinding such as:

- Task Analysis
- Interviews
- Affinity Diagrams
- Cognitive Walkthrough
- Questionnnaires
- Focus Groups
- Diary Studies
- "Speed dating"
- etc.

Interview

Interviewee speaks 90 percent of time and stays on topic. We choose participants to be representative target users, either current or potential future users. We like both experts and typical users. Try to provide an explanation into how users make sense to themselves.

Common pitfalls in interviews

Suggesting answers. Hypthetical questions.

Diary Study

Ask people to record events as they happen. User diary studies for rare events, easily forgotton events and events where the actual frequency is important. Problems with diary studies is that the simple tracking of their behaviors will change their behaviors.

Retrospective Survey

Ask about things that have happened in the past. Use this for critical events (well remembered), recent and memorable events, rare events that had a big impact an are memorable. Do not use them for hard to remember events.

Artifact Analysis

Look at things people leave around to understand a problem they might have. Use this for physical spaces (physical artifacts from workflows), tasks involving artifacts and interactions generating artifacts (emails, social media posts etc.). Only use if there are in fact artifacts and there is no faster way to learn information.

Contextual Inquiry

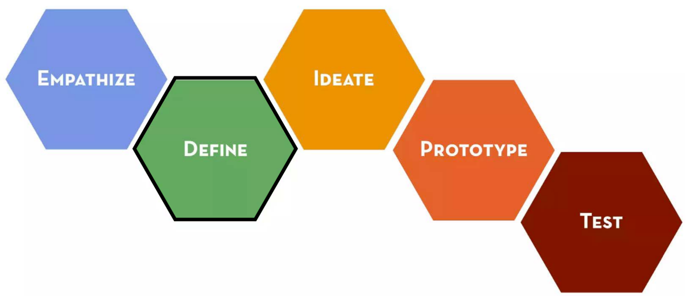
Ethnographics or participatory design, combining aspects of other methods. Interviewing, observing in the context of work. Goal is to discover real requirements of the work. Interview people while they are working and gather real artifacts. User decides the tasks, but you decide the focus.

Key Differences

interviews, surveys, focus groups	Contextual Inquiry
summary data & abstraction	in-situ experience & data
what customers say	what users do
subjective limited reliability of humans what users/customers *think* they want	objective spontaneous, as it happens what users actually need

Result of Need-Finding

We know what works and what does not yet exist. Problems and incomplete parts in process. So we have a long list of problems.



Define

This part is more a focussing part and not flaring. Figure out what is important from collected data. Group info and find relations.

Affinity Diagrams

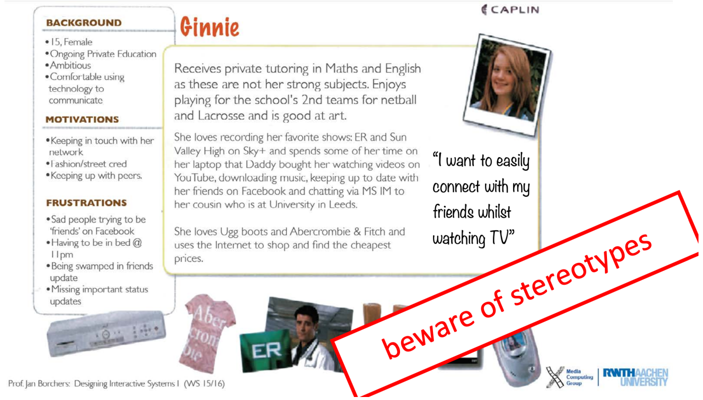
Data with affinity to each other are grouped together to form category. Groups are given labels, can be one or more categories in the end. Identify user, need and insight. Combined to create point of view. Good point of view insires the team, frames the problem in a focussed way. Empowers to make decisions and fuels brainstorming by suggesting "how might we" statements.

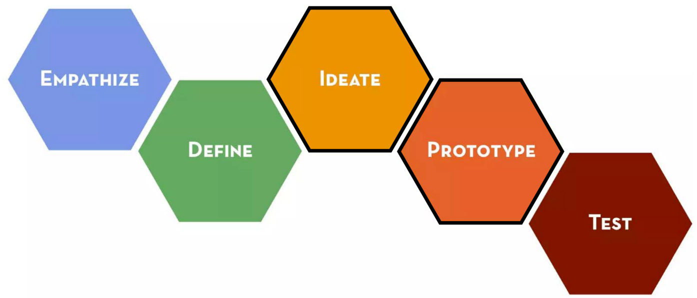
The elastic user

The elastic user can mean everyone and also noone. Vague and unfocused, lack of specifics makes it easy to rationalize any design.

Personas

Personas are precisely described. Act as stand-in for real users. Guide design decision. Fictitious but based on knowledge of real users. Informed from observations. Personas are not elastic, don't make them fit the prototype.





Ideate

Flaring process here, not focussing anymore.

Ideation techniques

- brainstorming
- mind-mapping
- storyboarding
- sketching
- low-fi prototyping

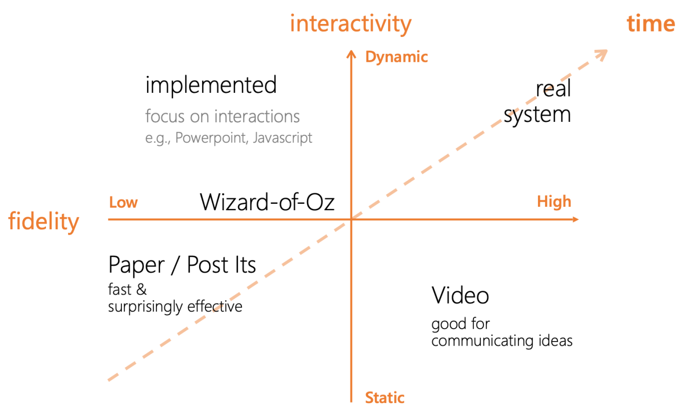
3 Prototyping

Prototypes develop from sketches over time and are more defined in their criteria weights. Make multiple protopytes to evaluate different approaches and check for failure/success. Prototypes help us understand the requirements and specifications of the idea. They answer a specific question.

Vertical vs. horizontal

Vertical provides critical path of one or few features (real feature on that path is completed, goes deep). Horizontal paths provide only overview with little to no functionality.

Fidelity and Interactivity



Paper prototypes

Are rapid and cheap.

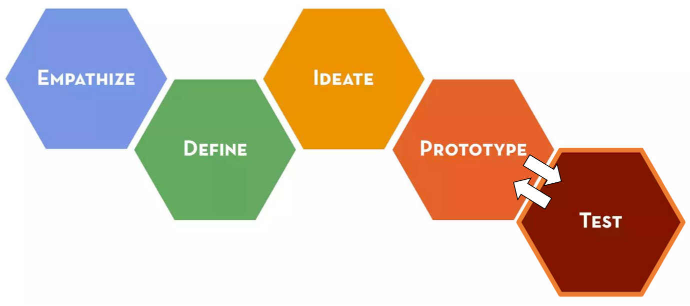
Wizard of Oz

Interprets user input and simulated a system response. Allowes rapid testing of a complex feature before implementing.

MidFi-Prototypes

Physical (paper, cardboard, lego etc.) to software.

- Powerpoint, Keynote
- AdobeXD, Figma



Analytical vs. Empirical

Analytical

Look at inherent attributes of the design, rather then the design in use. Intrinsic characteristics of the design. Examples are usability/UX inspection methods, design walkthroughs, heuristic evaluations.

Empirical

Based on how well a design or design change pays off in terms of real observable usage. Includes quantitative and qualitative data. Examples are usability/UX scores, controlled experiments and case studies.

Formative vs. Sumative

Formative

Helps form the design. Part of iterative process. Evaluations done during testing. Mainly collects qualitative data but also quantitative. Focusses on what is not working.

Summative

Helps sum up the design. Evaluates the success of the finished product, and compares with competitors. Collects quantitative data, and focussed mainly on what is working.

Evaluation criteria for UI design

Usability

Extent to which product can be used by specific users to achieve goals with effectiveness, efficiency and satisfaction. Five quality components of Usability:

- learnability
- efficiency
- memorability
- errors
- satisfaction

User Experience

Totality of the effect(s) felt by a user as a result of interaction with the usage context of a system. device or product.
It includes:

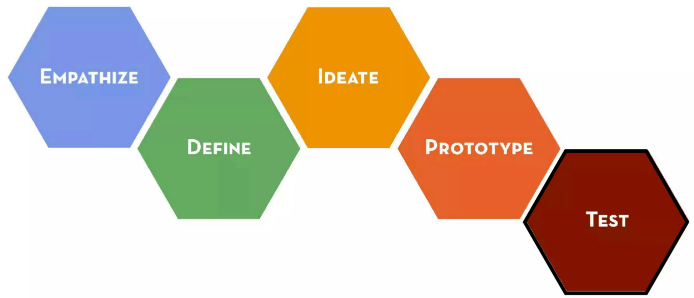
- Usability
- Usefulness
- Emotional impact
- Savoring memory after interaction

It embraces seeing, touching, thinking about the system or product and admiring it and its presentation. Focusses on holistic experience of the user.

Affordances

Actions that the design of an object suggests to the user. Provide strong clues to how objects are to be used without labels, explanations or manuals. Works for both physical objects and software. Up to a certain degree of complexity.

4 Analytical Investigation



Is performed by usability experts and domain experts. They use their knowledge of the users and technology to assess the usability and user experience. Result can be formal or informal reports.

Two types of analytical investigation:

1. Usability and UX inspection (Design, cognitive walkthroughs, heuristic evaluation)
2. Predictive user performance models (GOMS, KLM)

1. Usability and UX inspection

Cognitive Walkthroughs

Evaluate design by experts, with the goal of exploring the design on behalf of the users. Difference to UX inspection: UX inspection is only one aspect of a design presented to experts. Cognitive walkthrough is More focussed on ease-of-learning.

Heuristic Evaluation

Heuristics are design guidelines. They examine the interface, judge it's compliance with recognized usability principles (heuristics). Is cheap, fast and easy to use. Is developed for inexperienced practicioners, experts can be limited through heuristics. Can be done on paper-only prototypes.

1. Briefing to tell evaluators what to do
2. Each evaluator inspects interface alone (at least twice, get feel for flow of interaction and scope of system, also focus on specific interface elements)
3. Evaluators aggregate findings
4. Debriefing session, discussion of possible redesigns for major UX problems, look at positive aspects

Optimally between 3 and 5 evaluators. Limited because it does not encourage to take a rich and comprehensive view of interaction. Its only a rough outline, and expert find problems withs inspection not heuristics. Danger of overestimating heuristics and use it for any evaluation.

Nielsen's Heuristics

1. Visibility of system status

System should always keep users informed about what is going on, through approp. feedback in reasonable time.

2. Match between system and the real world

System should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real world conventions, make info appear in natural and logical order.

3. User control and freedom

Users need a clearly marked emergency exit from unwanted state, if chosen system functions by mistake.

4. Consistency and standards

User should not have to wonder whether different words, situations or actions mean the same thing.

5. Error prevention

Good error messages, but better is design that prevents a problem from occurring in the first place. Eliminate error-prone conditions or check for them and give users a confirmation option before commiting to the action.

6. Recognition rather than recall

Minimize the user's memory load by making opjects, options and actions visible. Instructions for use of the system should be visible or easily retrievable, whenever appropriate.

7. Flexibility and efficiency of use

Accelerators may often speed up the interaction for the expert user, such that the system can support both inexperienced and experienced users.

8. Asthetic ans minimalist design

Dialogs should not contain irrelevant or rarely needed information. Extra infos compete with the relevant units of information.

9. Help users recognize, diagnose and recover from errors

Error messages should be expressed in plain language, precisely indicate the problem and constructively suggest a solution.

10. Help and documentation

Though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Should be easy to reach, list concrete steps and should not be too extensive.

2. Predictive User performance models

Way of evaluating products or design without directly involving users. Estimate of efficiency of systems for different tasks.

We use GOMS to model knowledge about the system and cognitive provesses involved when users interact with systems.

We use KLM to provide numerical predictions to performance and estimate chains of operations.

GOMS model

Goals

The state the user wants to achieve.

Operators

Cognitive processes and pyhsical actions needed to attain the goals (mouse click etc.)

Methods

Procedures for accomplishing the goals, drage mouse over search gield, type in term, press go etc ...

Selection Rules

Decide which method to select when there is more than one.

Goms example:

Goal: delete a word in a sentence

Method for goal of deleting a word using menu option:

- Step 1. Recall that word to be deleted has to be highlighted
- Step 2. Recall that command is 'cut'
- Step 3. Recall that command 'cut' is in edit menu
- Step 4. Accomplish goal of selecting and executing the 'cut' command
- Step 5. Return with goal accomplished

Keystroke Level model (KLM)

Measures and compares execution times.

Operator name	Description	Time (s)
K	Pressing a single key or button Skilled typist (55 wpm) Average typist (40 wpm) User unfamiliar with the keyboard Pressing shift or control key	0.35 (average) 0.22 0.28 1.20 0.08
P	Pointing with a mouse or other device to a target on a display	1.10
P_1	Clicking the mouse or similar device	0.20
H	Homing hands on the keyboard or other device	0.40
D	Draw a line using a mouse	Variable depending on the length of line
M	Mentally prepare to do something, e.g. make a decision	1.35
R(t)	System response time – counted only if it causes the user to wait when carrying out his/her task	*t*

Predictive models strengths and weaknesses

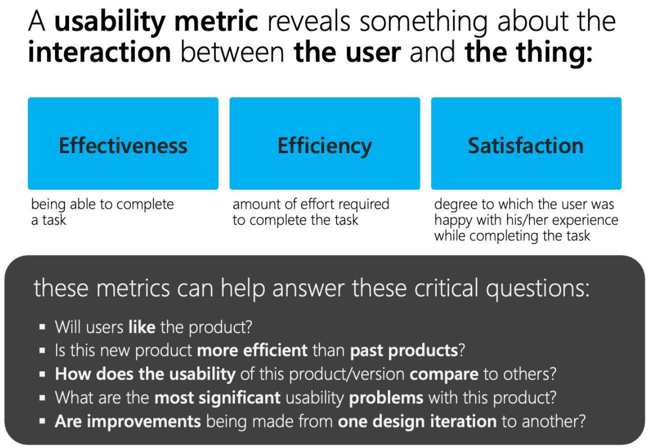
- Relatively easy to perform comparative analysis for different interfaces and prototypes, specifications.
- Can only model high-level tasks, involving small set of high routine low level tasks
- Only valid for predictable/expert behavior (no multitasking, fatigue, learning effects etc)

5 Evaluations and Experimental Design

Formative early in the design process, sanity checks that we're building the right thing. *Summative* to check if we improved upon our last iteration, does it work better than other solutions?

Quantitative Evaluation Methods
Ensure certain level of quality, comparesolutions objectively, attain a scientific statement.

Primary Usability Metrics



Cause and Effect
We want to identify clear causal links. Cause precedes effect, they need to correlate and other explanations have to be ruled out. Isolate causality by controlled experiments. Alter design with suspected cause absent (control) and present (experimental condition). All other conditions should be identical.

Quasiexperimentell — Observational
We observe that independent variable and dependent variable are highly correlated, but did not control for anything (for instance participation in exercices and final exam grade).

Experimental — Controlled
We randomly assign students to exercise and no exercise condition, then we controlled for other variables and results implies causality.

characteristics of Emprical Methods

- Objectivity
- Reproducibility
- Validity (internally and externally)
- Relevance

For instance threat to external validity is over-use of specific participant groups (only psychology or cs students).

The experiment
Indepdendent variables affect the dependent (measured) variables through experiment.
Variables can be catagorical, ordinal (ordered discrete), or cardinal/interval (continuous) data.

Designing an empirical study

1. What is being compared? (which Indepdendent variables)
2. What are they bing comapred in? (dependent variables, metrics)
3. What lse is being varied? (extraneous variables to control/eliminate)
4. Relevance

Look at slide set 5 for various examples.

More complex comparisons
Different experimental designs possible: *Within subjects:* Everyone-does everything. *Between subjects:* Only on condition per group.

Latin Square Counterbalancing

full randomization can lead to huge experiments (e.g., 6! = 720)

Latin square design reduces number of experimental orderings

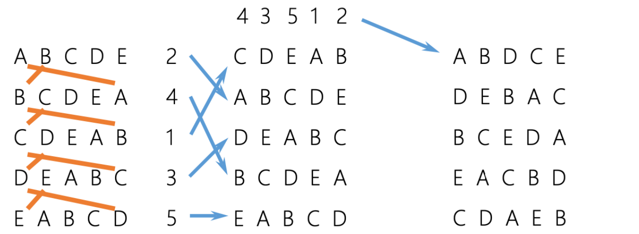
- total number of experimental conditions is the square of the number of treatments
- each treatment appears once and only once in each row and column

A	B	C	D	E
D	E	B	A	C
B	C	E	D	A
E	A	C	B	D
C	D	A	E	B

Latin Square Example for 5

first row in alphabetical order ⇒ A B C D E

subsequent rows – shift letters one position



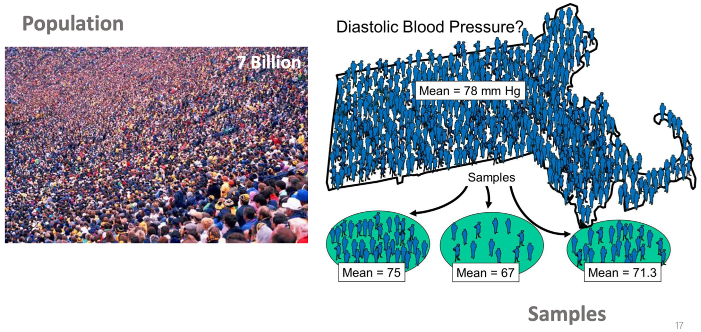
Then: randomize the order of the rows: i.e., 2 4 1 3 5
randomize the order of the columns: i.e., 4 3 5 1 2

6 Statistical Analysis

Data Collection
Important:

- Choose representative sample
- Form hypthesis to make asssumptions testable
- collect data to test Hypothesis
- collect all avaiable data (better too much)

Population vs Sample



Generalizability

Results should be valid for all people. Participants should be representative of population. Look at relevant factors, such as Age, Gender, Occupation.

Hypothesis testing

Effect size is difference in the means of H_A and H_0. It is unknown a-priori and hence we don't know how to show the threshold for acceptance. Therefore instead of showing H_A is true, we show that the obtained data is incompatible with H_0.

Descriptive statistics and validation

How should data be validated?

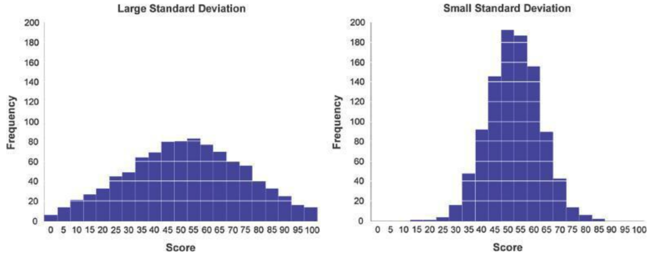
Mean

Least distance to all other data points. Good representation of data points piled together. Bad representation if some data points are extreme values (outliers). Doesn't make sense if we divide through categorical or ordinal data.

Median

is robust against outliers. The number at the middle of the ordered data points. Natural choice to represent ordinal data.

Distribution



Confidence Interval (CI)

Interval in which we are very sure that our true values lie in. We mostly choose 95 percent of values to lie within this interval if often replicated. Confidence interval of mean difference (two samples)

Analysis

Bayesian quantitative approach no covered in this course, also not qualitative analysis methods.

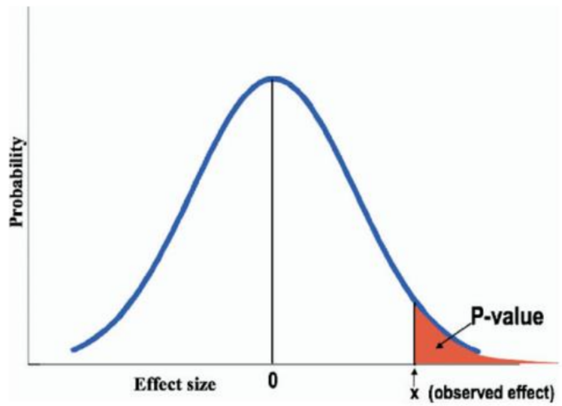
Frequentist Approaches

Hypothesis testing

We assume H_0 to be true. The lower our p-value the less likely that H_0 is true and H_A is true. P-value indicated how compatible the data to which hypothesis is.

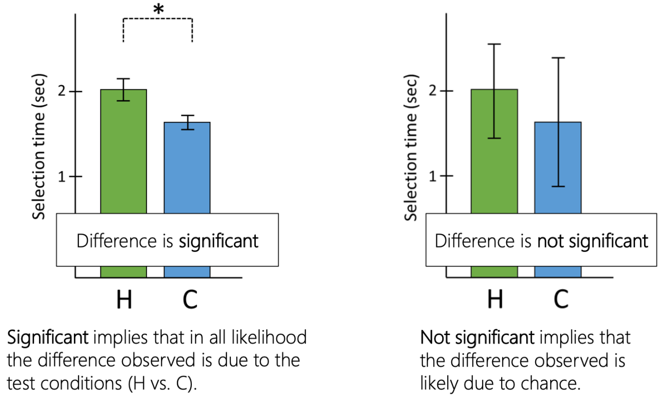
P-Value

P-value is probability of observed data if H_0 were true.



Alpha-Level

Threshold to determine if p-value is lowe enough. Usually $\alpha = 0.05$. In medicine even lower. If p-vlue is larger than α this does not mean that H_0 is true!



Errors

Type I error: Effect was found, but no effect in reality (False-Positive).
Type II error: No effect was found, but effect exists in reality (False-Negative).

Degrees of Freedom

Number of values that are free to cary. Number of observations (n) minus the number of parameter estimates. For one-sample t-test:

$$v = n - 1$$

.

Independent vs dependent samples

Independent: One subject only exposed to one condition (use different subjects for differen conditions). Also referred to as Independent measures or means.
Dependent: Same subject exposed to all conditions (Within subject design). Also referred to as matched pairs or paired samples.

Parametrix vs. non-parametric tests

Non-parametric do not assume specific distribution. Assume equal spread of group samples. Less statistical power. Type II error more likely.

Examples: Chi-Square, Mann Whitnes, Wilcoxons signed rank test, Kruskal Wallis, Friedman

Parametric tests assume gaussian distribution and homoscedasticity (equal variances). More power!
Examples: one-sample t-test, two-sample t-test, paired-sample t-test, one-way/factorial ANOVA, repeated measured ANOVA.

A/B Testing

Common example as in our case: Change one categorical independent variable with two levels (A and B) and measure one interval dependent variable. In our case task execution time.

Independent t-test

Checks if two means are reliably different from each other.
$t =$(variance between groups) /(variance within groups).
Large t means different groups (H_0 refuted).

From t-value to significance

T-values lead us to our p-value over degrees of freedom in standardized tables. T-distribution depends on sample size (degrees of freedom). Its a distribution of t-values of a population where the null hypothesis is true.

ANOVA analysis of Variance

Use this if independent variable /factor has three or more levels. One-way ANOVA is used for data with one factor and multiple levels. Factorial ANOVA is used for data with multiple factors and levels. Does not tell us which levels are different.

Effect size

Statistical significance does not mean that the measured effect is meaningful. So we need a standardized effect size. We use Cohen's d, Pearson's correlation, odds ratio.
Cohens' d is a standardized mean difference between the samples. Depends on the field.

$$\text{Cohen's } d = \frac{\text{mean}(A) - \text{mean}(B)}{\text{mean}([\text{std}(A), \text{std}(B)])}$$

Power analysis

Compute min. number of participants to achieve desired effect. Can be calculated from

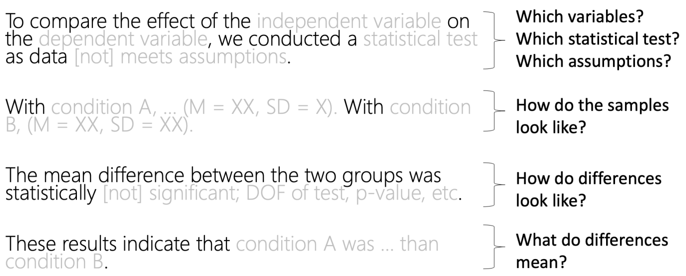
- prob. of finding an effect that is not there ($\alpha = 0.05$)
- prob. of finding an effect that is there ($1 - \beta = 0.8$)
- the desired effect size (HCI d = 0.8)

Software for statistical analysis

- SPSS
- Python
- R
- etc.

Reporting

Writing up the results



7 User Modeling

Human vs Computer

Strengths of human

- intuition
- memorize cohesive information
- signal detection under noise
- recognizing complex signals (speech etc.)
- recognizing complex configurations (scenes etc.)
- adaption to unexpected situations
- learniung aptitude

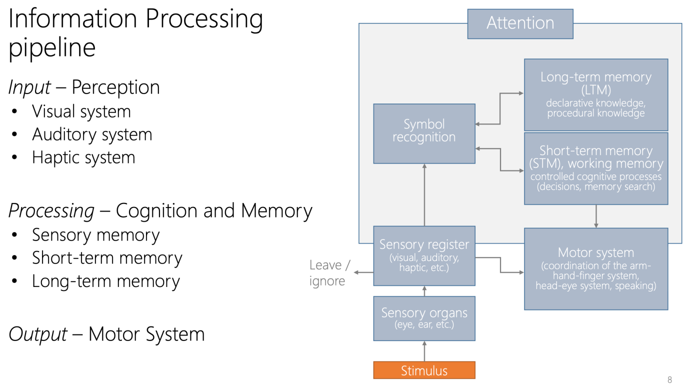
Strengths of computer

- measuring and counting
- storing large amounts of incoherent data
- detecting known signals
- fast and reliable reaction to signals
- reliable fatigue free reaction to known signals
- superiority if problems can be algorithmically formulated

Model Human Processor

We can look at humans as an information processor. Taking the computers output as input and after processing the information outputting the input for the computer.

The humans perception, memory and motor system can be applied to estimate execution time, error rates, training effects for simple stimulation /reaction interactions and system parameters.



We look at three main processors with associated memory.
Perceptual System

Containing sensors and buffers.

Cognitive System

Containing working memory and content symolically coded

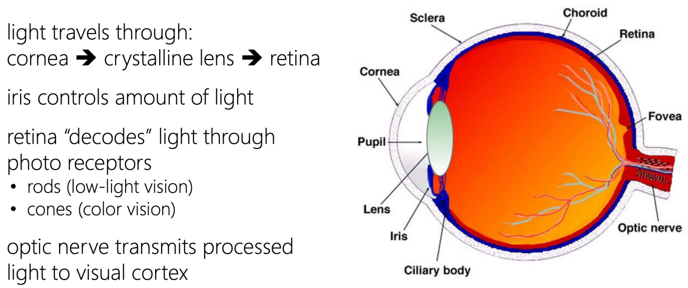
Motor System

Contains movements.

Each processor has associated runtime. Overall runtime is sum of these.

Perception (Visual System)

Anatomy of human eye



Rods are very light sensitive and have a slow response time. Are located in the periphery of the fovea. 120 million per eye, have maximum sensitivity at 500nm.

Cones are fast in responding and concentrated at the fovea. 6 million per eye. Three types for blue (S type, 420nm), green (M type, 534nm) and red (L type, 564nm).

Visual Field

Sharp vision withing 2 degree radius of fovea. Fine detail. Peripheral vision has decreased visual acuity with distance from fovea. Horizontal visual field is 60 degrees nasally and 90 degrees temporally. Vertical visual field is 60 degrees up and 70 degrees down.

Useful field of view is rather small (1-4 degrees for high character density and max 15 degrees for low character density.) We use this in Foveated Rendering to reduce details in the outer parts (for instance with eye tracking).

Eye movement

Saccades are repositioning of fovea. The take around 30ms with an amplitude of max 600degrees per second. Perception is greatly decreased.

Fixations are dwelling on one point. Its between saccades. It takes around 90 percent of the visual time. Take around 150 to 600 ms.

Gaze mvoements are context dependent of foreknowledge, attitute, task and predisposition.

Reading

Reading is a sequential loop of fixation and saccades. In average around 230ms fixation and 30ms saccades. On average around 300 WPM reading speed.

(Visual) Attention

We have great gaps in our perception. Interpretation is much sparser than one might assume. Perception of obkects requires lots of attention. Attention has to be directed. Perceptual processor receives and buffers signals. One buffer per sensory channel. Perception time is around 100ms (ranging 50-200ms).

Bloch's law

$$R = I * t$$

Where R is response, I is intensity and t is exposure time. For t ¡ 100ms, we asusme R constant. As a consequence we have limits on frame-rates (min 10Hz).

Cognitive Processor

Connects perceptual system to motor system. Learning, retrieval of facts, decision making, problem solving etc...

Processing time is around 70ms (25-170). Operates on chunks of information. For instance age, parts of a phone number.

Short term memory

Working memory, responsible for intermediate products of thinking and representations of perceptual system. Holds activated item from long term memory. Capacity is limited to 5-9 units (augmented by LTM). Pure capacity is aroun 2-4 units. Decay rate and capacity can be varied depdending on strategy etc. but decreases strongly with increased items.

Long term memory

Declarative (facts etc.) and procedural (how to do stuff) parts. Practically unlimited capacity with no decay time. Retrievel depends on associations with for instance eyternal stimuli. It is fast-read, slow-write.

Designing for memory

We try to deign for memory through grouping of related functionalities and use familiar structures. We also use recognition instead of recall.

Motor Processor

The average processing time is the sum of time needed for the perceptual, cognitive and motor processor. We differentiate between an open (no perceptual control, motor processor takes around 70ms) and a closed loop (perceptual system controls movement, ca 250ms).

Fitt's Law

Models throughput in aimed movements such as reaching for a control in the cockpit or clicking on icons with a mouse. Is very powerful and widely used. Holds in many circumstances, also under water and intoxicated. Allows for comparison among different experiments.

Originally the task was to touch the centerplate with a pencil, without touching the error plate on the sides. Generally the task is to predict the time to hit a target as a function of distance and size.

Index of difficulty

$$I_D = log_2(2D/W)$$

Index of Performance or Throughput

ID = information (nr of bits) required to specify movement (amplitude within given tolerance)IP is index of performance.

$$IP = ID/MT$$

(is in bits/sec) Depends on input device and limb. Movement time MT

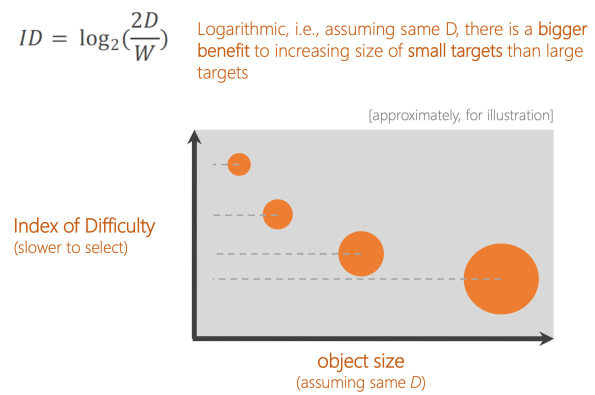
$$MT = a + b * ID$$

$$MT = a + b * log_2(2D/W)$$

In the end we can use the different MTs to estimate a regression on ID and MT (estimate a and b of line).

Fitt's Law implications

We find that doubling the distance adds roughly a constant to execution time (Logarithmic nature of the law). Doubling the target width is roughly equal to halving the distance (implicated by D/W term in the formulation).



Fitt's Law in practice

We can add the last pixel of buttons on the left side, to increase to effective Width of "almost infinite".

Larger fields can be clicked more easily.

Application: Compare input devices

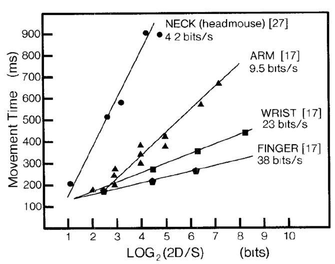
Compare mouse, trackball and stylus in speed. Use pointing and dragging as actions. This corresponds to finding a and b in the formula for MT. The we can compare the index of performance (throughput). We can then use this information to design an "optimal" UI.

Limitations of Fitts' Law

Fits law does not:

- consider body asymmetries (right vs. left hand flexion vs. extension)
- address parallelization strategies (use multiple finger, hands)
- include any cognitive factors (reaction time, visual search time etc.)

Bandwidth



To compare performance of input devices

Card et al. compared mouse, joystick, directional keys, text keys [1978]

comparing mouse, headmouse, fingers:

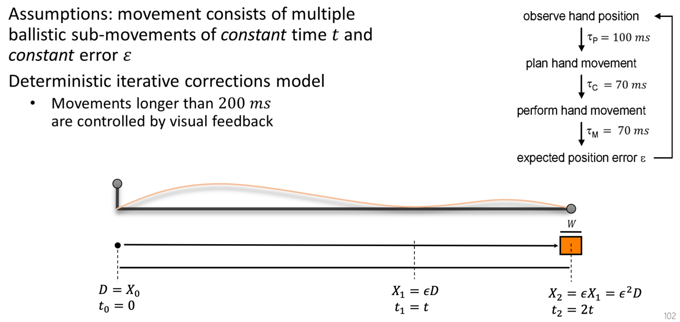
Target	Size S^a (cm)	I_D (bits)	Mouse (ms)	Headmouse (ms)	Fingers (ms)
Paragraph[b]	5.5	1.18	113	280	30
Word[c]	2.3	2.24	220	540	56
Character[d]	0.41	4.59	440	1100	115
Period[e]	0.069	7.14	690	1710	179

Performance of use depends on human (bandwidth of muscle groups), application (precision requirments of the task) and device (effective bandwidth of input device).

From Model Human Processor to Fitt's Law

Visual and Proprioceptive Feedback Loop

FIrst we observe handposition, then we plan the movement, we perform the hand movement and finally asses the error to expected position. This is a loop and gets repeated until desired movement is finished.



We know from MHP that one cycle through all processors is around 300ms. If n is the times we go through loop, the final time then is

$$n \times (\tau_p + \tau_c + \tau_M)$$

After the first cycle we have:

$$X_1 = \epsilon X_0 = \epsilon D$$

In the second cycle:

$$X_2 = \epsilon X_1 = \epsilon(\epsilon X_0) = \epsilon^2 D$$

n^{th} cycle :

$$X_n = \epsilon^n D$$

We stop the movement when:

$$\epsilon^n D \leq \frac{1}{2} W$$

We solve for n:

$$n = -log_2(\frac{2D}{W})/log_2\epsilon$$

We insert into formula for movement time:

$$MT = n \times (\tau_p + \tau_c + \tau_M)$$

$\Rightarrow$

$$MT = \frac{\tau_p + \tau_c + \tau_m}{-\log_2 \varepsilon} \cdot \log_2\left(\frac{2D}{W}\right)$$

$$MT = I_M I_D$$

$$I_M = \text{Index of motion}\left(\frac{\text{sec}}{\text{bits}}\right)$$

8 Visual Perception

Gestalt principles

Gestalt psychology was founded in the 1920s by Max Wertheimer and others. It is about the perception of groups, patterns and objects.

four key principles

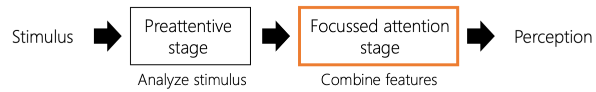
- Emergence
- Multistability
- Eification
- Invariance

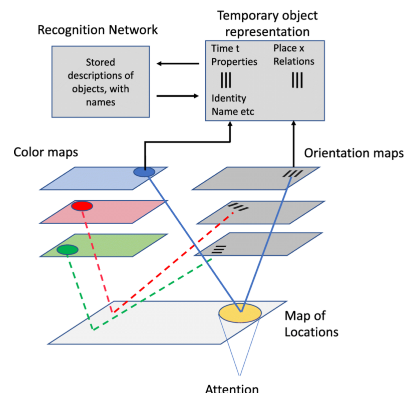
laws of grouping

- Proximity (close objects belong to group)
- Similarity (similar appeareance belong to group)
- Closure (humans prefer to see complete figures)
- Symmetry (Symmetric objects form groups around the center)
- Figure and ground (users tend to seperate images ot fore- and background)
- Continuity (Objects that intersect are perceived as individual objects)
- Past experience (Based on past experience group objects together)

Feature integration theory

Availability of visual information is limited. Full visual acuity in only available in foveal area. Peripheral vision provides limited information. This is why FIT suggests stimuli that are registered eary and automatically.





Theory is based on the process of selective attention. Is useful but certain limited in certain aspects. FIT primarily concers bottom-up activation. Top-down activation through memory and expectation.

Selective Attention

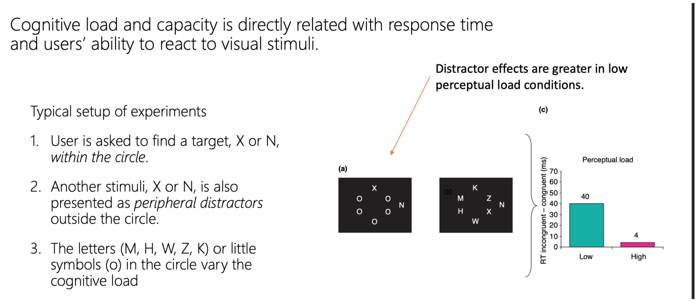
There is an ongoing debate about Early / Late selection. The early selection model has been proposed earlier and mainly relies on the idea of an early "bottleneck" in the attentional process given by perception. It also assumes that focussed attention can prevent distractor processing at an early stage

Late selection was proposed later and assumes unlimited perception and the automatic discrimination of relevant and irrelecant stimuli. It assumes that later processes such as memory or behavior are the processes with selected attention.

Perceptual Load Theory

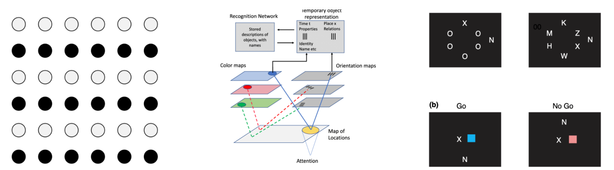
Sees perception as a process with limited capacity. This is in line with the early selection views. PLT also assumes that automatically all stimuli are processed until capacity is filled.

Example experiment on PLT



Takeaways on PLT

Users ability to react to stimuli is related to its context. High load leads to more focus and less distraction. Low load leads to quicker distraction.



Gestalt principles Feature integration theory Perceptual load theory

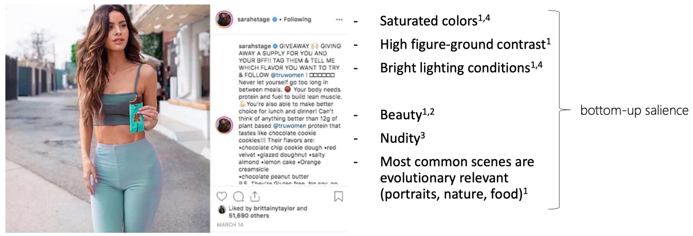
Visual Saliency

The context of obkects popping-out as pre-attentive for selection. Assumes that feature maps are computed in parallel and combined. The computed maps yield a saliency map. Assumes that the "Winner-takes-it-all". Assumes that this sequential processing is based on inihibition of return.

Example is given by an experiment where different skins are compared and snake skin resulted in significantly higher brain activity measured over EEG.

With visual saliency we can predict users' gate and importance. It is mainly bottom-up and therefore feature based. Top-down saliency is challenging, however task and load influence the saliency.

If there is too many competing features, clutter will occur.



→ proficient in catching our visual attention

Visual Search

The process that decides where humans look next.

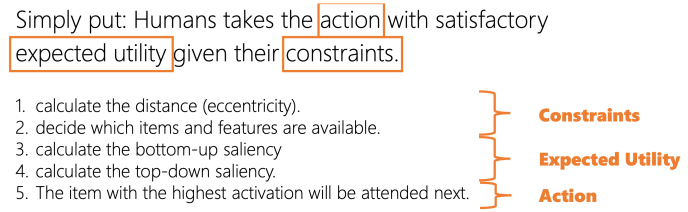
1. Guided Search (rules exist on priorities for items or areas int the scene)
2. Bounded rationality (selection of actions based on expected utility bunder uncertainty)

Guided Search

1. Calculate the distance (eccentricity) to the current fixation location for each item
2. Given exxentricity decide which items and features are available to visual representations
3. Calculate bottom-up saliency for each item
4. Calculate top-down saliency for each item
5. Sum up bottom-up and top-down activations and select the ones with highest activation

Bounded Rationality

Assumes that humans take the action with satisfactory expected utility given their constants.



Both visual saliency and search behavior are important for user interfaces and computational design.

Application in HCI

Aalto interface metrix (AIM)

Combination of empirical models and metrics of user perception and attention. Quantifies user experience, complemets and removes guesswork.

Online UI adaption

Is a tool to measure spatially and semantically relevant labels. Measures pre-attentive object features. It is also possible to distinguish pre-attentive and attentive object features.

There are also many more tools and applications that make use of these concepts.

9 Input

Input Devices

Fitt's law for performance evaluation of input devices, but how to distinguish in functionality and specific metric? Answer is systematization. In general input devices enable to engange in dialogue with a computer or a machine. This dialogue is not in natural language. Dialogue is between fundamentally dissimilar agents-both in perception and processing.

"It's a transducer from the physical properties of the world into logical values of the application."

Fitts' Law in 2D

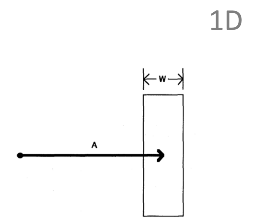


Figure 1. Fitts' law paradigm. The law is inherently one-dimensional since target amplitude (*A*) and width (*W*) are measured along the same axis.

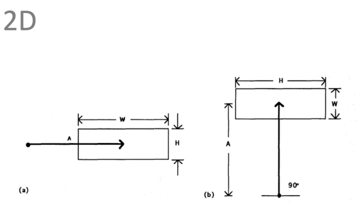


Figure 2. Fitts' law in 2D. The roles of width and height reverse as the approach angle changes from 0° to 90°

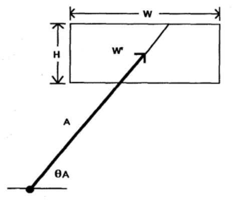


Figure 3. What is target width? Possibilities include *W'* (the width of the target along an approach vector) or the smaller of *W* or *H*.

What is target width for 2D movement?

Variant 1
Length of path part that is inside rectangle, i.e., **W'** ("W-prime") inside rectangle in picture

Variant 2
Smaller value of W and H, i.e., min(W,H)

According to MacKenzie et al., both variants (1,2) are more suitable than using **W** (without prime), but there are no significant differences between the variants (1,2).

Properties

Direct / indirect input

- Direct: Touchscreen, "grasping" virtual 3d objects
- Indirect: Mouse movement translated to cursor position (virtual cursor can directly/indirectly manipulate virtual content)

Absolute / relative

- Absolute: Position of input mapped to position of output (e.g. drawing tablet)
- Relative: Change of input position mapped to change of output position (e.g. mouse)

Position control / rate control

- Manipulate position of something (e.g. mouse cursor) versus its velocity (e.g. thumbstick)

Degrees of freedom

- Examples: only 2D position along surface (2 DoF), 3D position and rotatoin in mid-air (6 DoF), or other combination (3D position + rotation around one axis : 4 DoF)

Isotonic / elastic / isometric

- Movable (isotonic e.g. mouse) vs. movable but goes back to neutral position (elastic, e.g. joystick) vs. immovable (isometric, sense force only e.g. lenovo red dot)

Performance/Bandwidth

The performance depends on the human (bandwidth of muscle groups connected to input device), the device (effective bandwidth of input device) and application (precision requirements of the task).

Effectiveness

- Pointing speed
- Pointing precision
- errors
- Time to learn
- Time to grasp device
- User preference
- Desk footprint
- Cost

Design Space by Card et al

Input device as six-tuple: (M, In, S, R, Out, W)

- M: Manipulation operator
- In: Input Domain
- S: Current state of device
- R: (Resolution) Mapping from input domain to output domain
- Out: Output domain
- W: Additional aspects of ow device works (input lag etc.)

Possible manipulation operators M

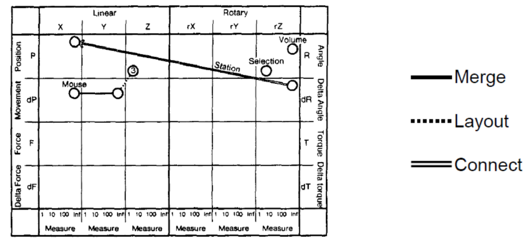
	Linear	Rotary
Position		
Absolute	Position P	Rotation R
Relative	Movement dP	DeltaRotation dR
Force		
Absolute	Force F	Torque T
Relative	DeltaForce dF	DeltaTorque dT

Composition operators

- Merge composition: e.g. sensed X-Y movement of mouse merged into 2D input
- Layout composition: Seperate independent inputs on a device (e.g. independent buttons or wheels on mouse)
- Connect composition: Output of one device/sensor mapped into input of another (e.g. physical mouse is input for virtual screen cursor). In this context virtual cursors count also as input devices.

Card's Graphical representation

Example from Card et al: 3-button mouse and radio station



Input Decoding

Touch Input

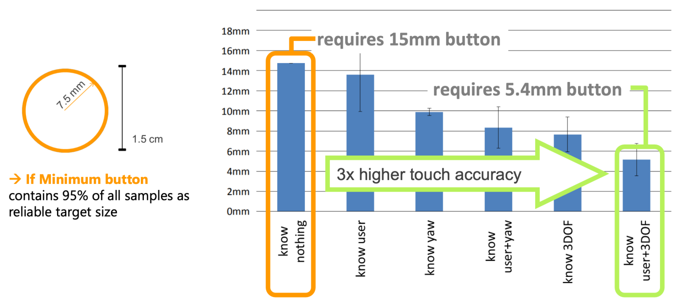
Issues with touch: It's noisy and touch area larger than the target. Also visual occlusions. Mobility increaases accuracy issues.

Task: Hit target on touch screen (repeatedly)

Independent Variables: Yaw, roll, pitch, user (mental model)

We record every trial as a dot at the touch location. Without influence of independent variables should result in circles. If the locations fall into clusters we can compensate if condition is know.

Minimum Touch Input Size

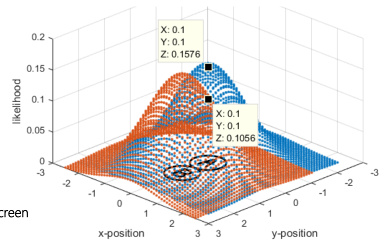


Representing Input

$P(y|x)$ Probability of message y given signal x

$$= \frac{P(y \cap x)}{P(x)} = \frac{P(x|y)P(y)}{P(x)}$$

Conditional probability — Bayes' rule



A concrete example for a single letter input:
$P(y|x)$ is the probability of letter 'a' or 'b' given a screen coordinate in 2D

→ Use for **statistical decoding** of **message sequences**

Sequence decoding

Given an input (touch, key), predict most likely next message.

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$ ← Posterior distribution

Most probable message → $$\hat{y} = \arg\max_y \left[\frac{P(x|y)P(y)}{P(x)}\right]$$

$$\hat{y} = \arg\max_y [P(x|y)P(y)]$$ ← maximize conditional probability of the message given the signal

$$\widehat{hypothesis} = \arg\max_{hypotheses} [likelihood\ model \cdot prior\ model]$$

An example would be to investigate noisy sequences of input and identify the most likely sequence of intended presses. Observation would be the 2D screen coordinates of tap and the observation sequence are the time-ordered observations.

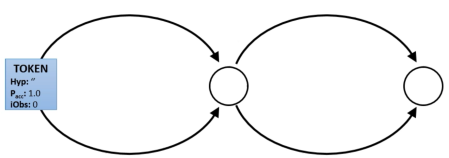
A token in this context would be a datastructure containing the accumulated probability of hypothesis.

$$Acc\_Prop_{n+1} = Acc\_Prob_n * prior * likelihood$$

Simple sequence decoding

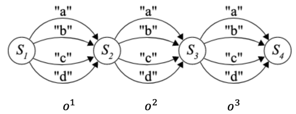
One token per hypothesis per obervation.

- k^i tokens ...



Substitution-only decoder

Input
- set of k symbols that can be recognized (i.e. k letters)
- a series of 2D coordinates (observation sequence) $O = \{o^1, o^2, \ldots, o^n\}$
- language model (probabilities)



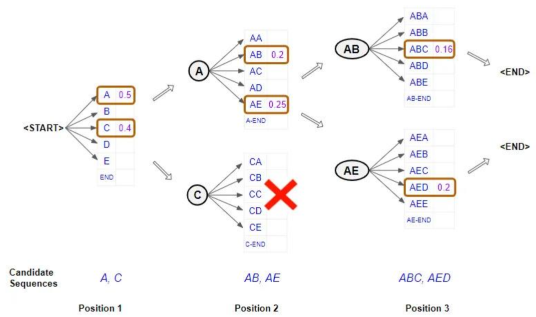
Token propagation rules
- Propagate token for every hypothesis for every observation
- Update token's probability: posterior = likelihood x prior

Bream Pruning

we have an infinite search space but only a few plausible hypothesis. We prevent propagation of unlikely tokens. Rules of beam Pruning:

- Only propagate a token if its probability is among the n-best probs for the observation so far
- If the new prob of the observation is among the n-best then update the list of tokens to propagate

Threshold is also known as beam size.



Language models

Probabilities for the decoder come from Language models. Language models are the probability of individual words or word sequences. A vast amount of letter combinations is unlikely to be written. These models capture valid letter and word sequences and assign them probabilities. These probabilities can be leveraged to infer or predict what users want to write, based on what they have already written.

The simplest models are uni- or bigrams. (Unigram contains one token only and bigrams obviously two).

Trigram Model

The trigram language model is a second-order Markov process. This model defines $P(w|u \cdot v)$ for any $w, u, v \in V$ where V is some vocabulary of words|letters. For a sentence $s = w_1 \cdot w_2 \cdot w_3 \cdot \ldots w_n$ we can compute the probability $P(s)$.

$P(w_1 \cdot w_2 \cdot w_3 \cdot \ldots w_n) \sim=$
$P(w_1)$ *
$P(w_2 \cdot w_1)$ *
$P(w_3 \cdot w_1 \cdot w_2)$ *
$P(w_4 \cdot w_2 \cdot w_3)$ *
.....
$P(w_n \cdot w_{n-2} \cdot w_{n-1})$

The conditional probability P (w | u · v) is typically estimated via some form of counting and smoothing (e.g. linear interpolation of unigrams), bigrams (e.g. u · v) and trigrams (u · v · w).

Smoothing is necessary to deal with sparseness which can occur because many trigrams u · v · w and even bigrams u · v may not occur in the training data.

Time complexity for each word encountered in training is constant, so training is usually fast.

11/16/2023 17

Trigram Model Example

Approximation of maximum-likelihood probability via counting:
$$P(c \mid ab) = \frac{counts(abc)}{counts(ab)}$$

3-grams	# of occurrences
brown fox jumped	125
brown fox walked	45
brown fox snapped	30

$P(jumped \mid brown\ fox) \sim= 125/200 \sim= 0.625$

$P(brown\ fox\ jumped) \sim=$

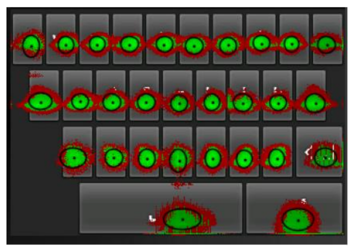
$$P(brown) * P(fox \mid brown) * P(jumped \mid brown\ fox) = \frac{200}{600} * \frac{200}{200} * \frac{125}{200} \sim= 0.208$$

Advanced statistical language models

Recurrent neural network (RNN) and Generative Pretrained Transformers(GPT) are able to capture more complex relationsships for many to many token mappings.

We can then combine probabilities from the touch model with a language model to get the most likely key for a touch.

Real example: bivariate Gaussian keyboard model



[Image: Henze et al. CHI '12.]

$$P(K|T) = \frac{N(T|C_K, \Sigma_K)}{\sum_1^n N(T|C_K^n, \Sigma_K^n)}$$

N(...) is the bivariate Gaussian distribution

T is the touch position

C_K and Σ_K are the mean and covariance of the touch distribution for key K

Application Examples

- Predictive text input
- Text input beyond screens
- Accessible mixed reality

(Check slides if you want to see more on these).

10 Computational Design

The main idea of this chapter is to use models and algorithms to help design and optimize interfaces. The chapter will contain:

- Modeling task as combinatorial optimization problem
- User models as cost / goodness functions in optimization
- The assignment proble and applications in HCI
- Exampels

Example: 50 different items yield 50! options to create menus.

To evaluate a menu we can use:

Time to move to an element i :

$$t_i = a + b\log_2\left(\frac{A_i}{W_i} + 1\right)$$

Average time to operate a menu:

$$T = \sum_i p_i t_i$$

Design as Search
As a goal we want to find the best design decision for given objectives.

Some benefits of using algos in design:

- Efficiently search large solution spaces
- Systematic, rigorous process
- Imporoved quality and reliability
- Guarantees for goodness og outcomes

We want to find optimal $x \in X$ where $X \in \mathbb{K}^n$ which maximizes $f(x)$. So X is the set of all feasible solutions.

Formulate optimization problem

- The design space (variables, contraints)
- Objective functionality
- Way to solve problem (solver)

Design Space

A combination of all design variables forms the design space. Each variable represents an open decision (usually discrete):

- Boolean(e.g. show label)
- Integer(e.g. color)
- Categorical(e.g. type of element)
- Continous(e.g. color value)

As not all combinations yield a feasible design we need to introduce constraints.

Decision variables:

$$x_{ik} = \begin{cases} 1, & \text{if command } i \text{ assigned to slot } k \\ 0, & \text{otherwise} \end{cases}$$

Design space:

$X = \{\mathbf{x} = (x_{ik}) \mid i, k = 1 \dots N, x_{ik} \in \{0, 1\}\}$

Constraints (feasible space):

$\sum_{k=1}^{n} x_{ik} = 1 \quad \forall i = 1 \dots N$, where each command is assigned to one slot.
$\sum_{i=1}^{n} x_{ik} = 1 \quad \forall k = 1 \dots N$, where each slot is assigned to one command.

Objective Function

Assign a score the each solution in the design space. Goal is to find the solution that maximizes or minimizes the objective function. Can be interpreted as quality indicator of UI.

We can find an objective function in different ways.

- Math model
- Machine-learning model
- Simulation-based model
- Look up tables from empirical data
- Heristics, guidelines, best-practices

Example: Minimize average selection time (linear assignment problem):

$$\min \sum_{i=1}^{n} \sum_{k=1}^{n} p_i \cdot t_k \cdot x_{ik}$$

Where p_i is prob of item i, x_{ik} the constraints and t_k the time to move to slot k from the top. We call x_{ik} the decision variables.

Optimization Methods
Mathematical, Exact Methods

Linear or (Mixed-) Integer Programming, Branch and Bound methods

- + Explicit bounds and guarantees optimality
- + Fast standard solvers available
- - Objective function in closed mathematical form
- - Problem formulation might be hard to set up

Heuristic approximation algorithms

Simulated annealing, Genetic algorithms, Biology inspired algos

- + Programmatical description
- + Standard implementations available (Scipy, Opimization Toolbox Matlab etc.)
- + Flexible
- - No bounds or guarantees on global optimum
- - Can have many params

Assignment Problem

Assign items from Set A (e.g. menu items) to items in Set B (e.g. menu slots).

Quadratic Assignment Problem

Similarity between commands: s_{ij}

Distance between menu slots: d_{kl}

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} \underbrace{\boxed{s_{ij} \cdot d_{kl}}}_{\text{The more similar the closer together}} \cdot \underbrace{\boxed{x_{ik} x_{jl}}}_{\text{Pairs of command-slot assignments}}$$

Is an np hard problem. Decision cost: $s_{ij} * d_{kl}$. The second part $x_{ik} * x_{jl}$ is quadratic in the number of decisions.

Example: The letter assignment problem

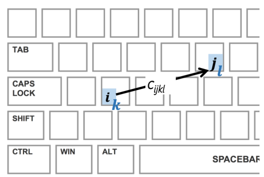
Question: Find best assignment of letters to keys on a smartphone to allow the fastest typing. Apply constraints: Each key assigned to exactly one letter, each letter assigned to exactly one key.

Design Space

n characters – $i, j \in \Sigma$
m keyslots – $k, l \in S$

$x_{ik} = 1$ if character i is assigned to keyslot k,
$x_{ik} = 0$ otherwise

Goal: Find the assignment of characters to keyslots that minimizes the cost of typing (special characters)



The goal was then to minimize the motor performance (average time to move between special characters and letters) and Ergonomics (minimize frequent extreme movements of wrist and fingers for special chars). For Intuitiveness minimize the distance between similar special chars and also their visual similarity. Familiarity refers to redesign similar to known preferences.

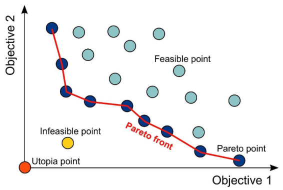
Multi-objective optimization

Goodness of user interface is determined by many aspects.

- Performance
- Ergonomic and Fatigue
- Error prob.
- Mental workload
- learnability
- Accessability
- Subjective experience
- etc.

- Weighted sum approach $w_1\hat{f}_1 + w_2\hat{f}_2 + w_3\hat{f}_3 \dots$
- Pareto optimization
- Hierarchical optimization

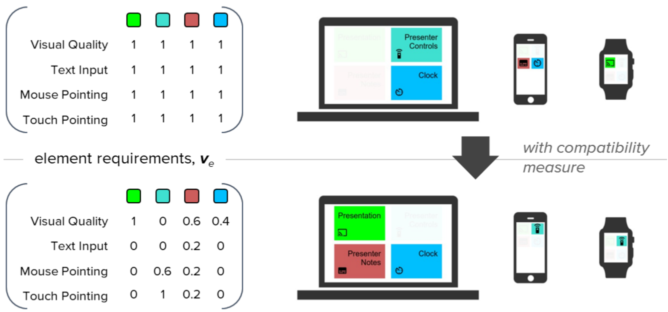
...



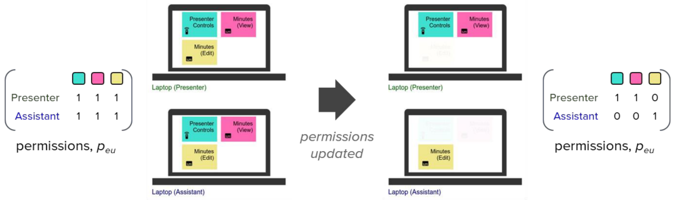
Combinatorial Optimization for User Interface Adaption

Optimize the UI to accomodate changes in environment, cognitive state, abilities, task, technical capabilities etc. Use different objectives such as optmize for quality and completeness.

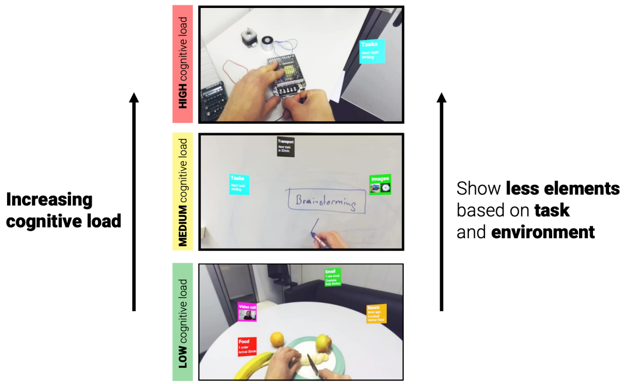
Element device compability



User roles



Cognitive Load



The idea then is to inlude cognitive load, cognitive capacity and cognitive cost to the objective function.

Optimize for environment

The idea is to optimize for differenct environments, e.g. screen placement, number or virtual screens etc. We include interaction modality, occlusion avoidance and temporal consistency in to the objective function.

This can be done through assigning voxel containers to objects and surfaces. The objective would the be to optimally assign elements to containers.

$$x_{e,c} = \begin{cases} 1, & \text{if } e \text{ is assigned to } c \\ 0, & \text{otherwise} \end{cases}$$

$$\text{argmax}_{e,c}\left(\underbrace{w_t \cdot T}_{\text{Temporal consistency}} + \underbrace{w_v \cdot V}_{\text{Occlusion avoidance}} + \underbrace{w_i \cdot I}_{\text{Interaction modality}}\right)$$

Temporal Consistency

$$T = \frac{1}{N_e}\left(-\sum_e\sum_c \|\underbrace{p_c}_{\text{container position}} - \underbrace{p_e^i}_{\text{element input position}}\| x_{e,c}\right)$$

Interaction Modality

$$I = \frac{1}{N_e}\sum_e\sum_c \underbrace{(\underbrace{w_t Q_t^c}_{\text{touch}} + \underbrace{w_{elb}Q_{elb}^c}_{\text{elbow-rested}} + \underbrace{w_m Q_m^c}_{\text{mid-air direct}} + \underbrace{w_d Q_d^c}_{\text{distant}})}_{\text{metrics support placements for respective modality}} x_{e,c}$$

New contexts might introduce physical constraints that may render prior interface layouts unuseable.

11 Haptics

Importance of haptics

Haptics is the study of touch, force and tactile feedback in human-computer-interaction. Haptics feedback is everywhere and could be used for gaming, robotic surgery, education and many more.

- Enhances user experience and engagement
- Enables more intuitive and natural interactions
- Adresses limitations of visual and auditory feedback
- Vital for accessability and inclusion

Interaction benefits

- Increased accuracy and speed
- Reducing errors
- Eyes Free interaction
- Proprioceptive

Tactile

All about vibrations and textures.

- Sense of touch
- Goal: Stimulate skin in a programmable manner to create desired set of sensations
- Tactile feedback is generated by tactile device
- Skin based
- Examples: Vibration, pain, pressure, temperature

- Used in touchscreens, tactile displays and VR

Vibrotactile

- Subset of tactile
- Relies on vibrations to convey information
- Common in mobile devices, notifications and wearables

When designing actuators its important to consider thath different cells are in different parts of hand. They have different receptive fields and frequency ranges.

Limitations of vibrotactile sensation: Broad localization of the sensation, superficial feedback and thus the strength of the vibration may be perceived differentyl based on the area of skin in contact with mounting pressure.

Kinesthetic

Very accurate receptors in muscles, joint and skin. All about movement and force.

Passive Kinesthetic (Force feedback): Perception of resistance or force, it requires human motion/input. Examples are surgical simulators or controllers.
Limitations to passive kinesthetic:

- Hard to design
- Often cumpersome
- User needs to provide inpput
- Limited part of how we perceive the world

Active Kinesthetic

Focuses on the sense of body movement and position. Relevant in motion simulators, exoskeletons and teleoperation systems.
Limitations to passive kinesthetic:

- Hard to design
- Often cumpersome
- Expensive
- Limited part of how we perceive world

Current Limitations of Haptics

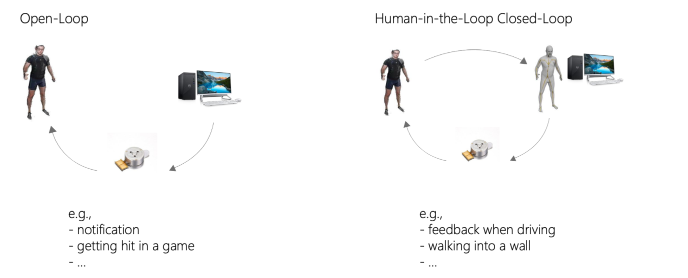
The chain is: System state triggers an actuator, which in turn is perceived by a human.
Kisthetics are currently not common in user electronics.

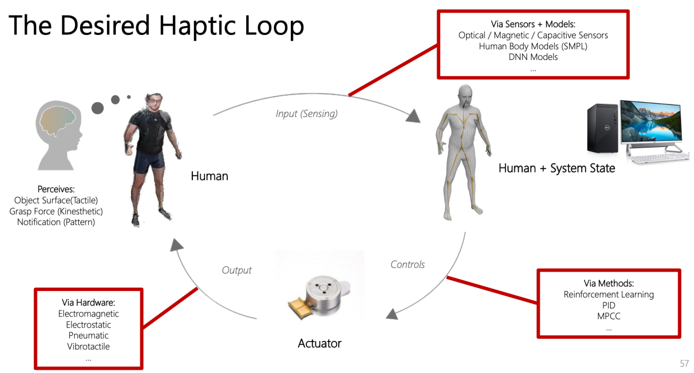
- Kisthetics are currently not common in user electronics.
- Human state is not taken into account
- Current systems only react to system states
- We are limited to vibrotactile feedback
- This leads to underperforming and underwhelmin feedback

How to overcome these limitations?

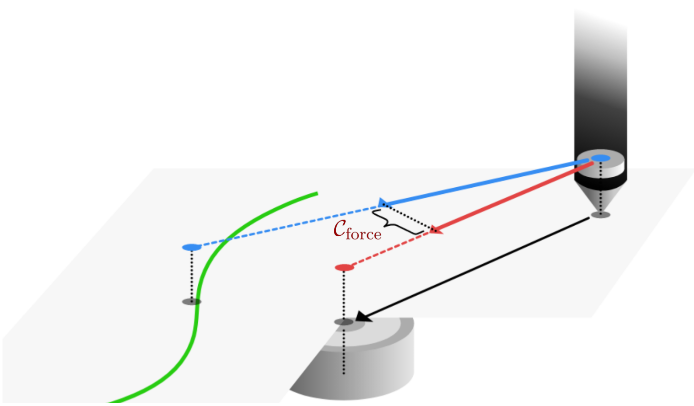
- Enable sensing of human state
- Use human state to control the device intelligently
- Using Closed vs open loop
- Body pose as a first step for truly intelligent haptic feedback systems

Add control:





Example of a closed-form solution to calculate electromagnetic force.



It has been found that more complex shapes beenfit most from feedback. And users are still limited by the speed of the system. But overall drawings are more accurate that way.

12 Computational Rationality

Comp. Rationality converges ideas from AI, robotics, cognitive science and neurosciences.
It refers to computational principles for:

1. Identifying decisions with highest expected utility, while taking into consideration the costs of computation in complex real-world problem, where calculations can only be approximated.
2. Implementing bounded optimality in humans.

Why User Models

- Used as test to help in creation and validation of new HCI theories
- Helps design more robust interaction, with improved safety and accessibility
- Reduce financial, temporal and human costs of usability testing
- Helps take advantage of advances in other engineering disciplines
- Advance next generation of intelligent interactive systems

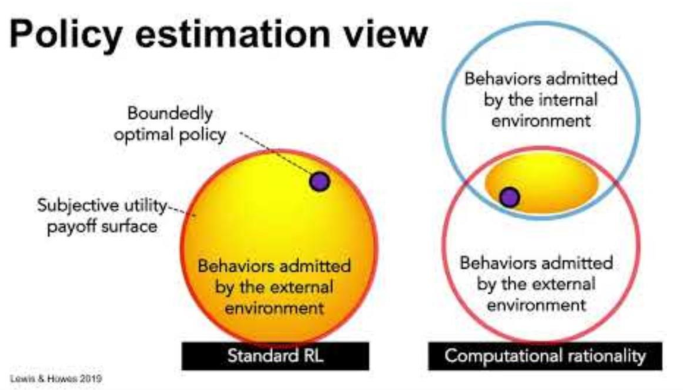
Limitations to Model human processor

- does not learn, adapt, generalize and has othe tradeoffs

Interaction is not equal to emerging behavior. After initial task has failed/succeded, there is no adaptive behavior or reorganization without explicit instruction.

Complexities of real-world tasks

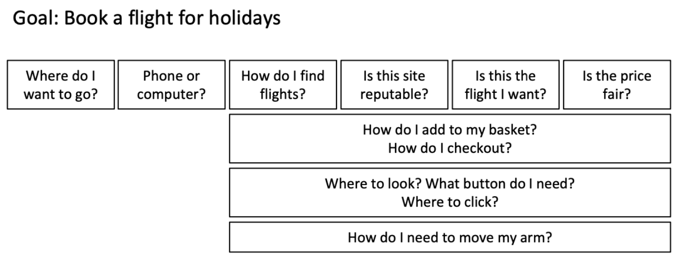
- Generalization: Go from precious episodes to an unseen one
- Latent learning: Adapting to distal changes in environment
- Planning: Sequencing actions while considering long-term effects on reward
- Compositionality: Good solution require putting together partial solutions cleverly
- Exploration/exploitation: Knowing when to learn the structure of task /environment vs when to exploit it
- Uncertainty: Knowledge can be incomplete or incorrect
- Resource limitations: Limited time and capabilities
- Curse of dimensionality: A very large number of possibilities

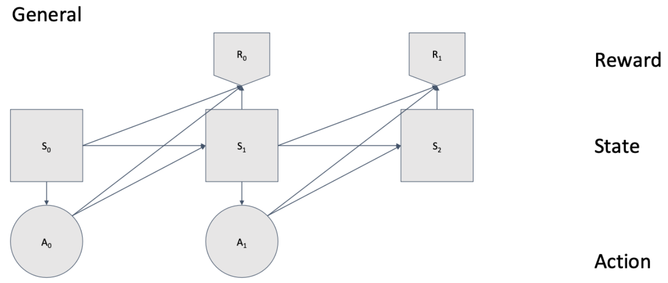


Optimal Policy

Is supposed to look at emerging bahavior component.

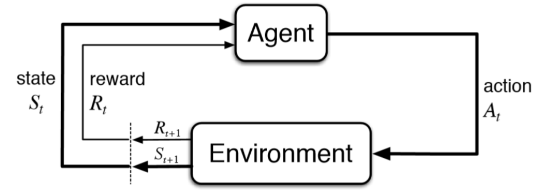
Interactions as a sequential decision-making process





Reinforcement Learning (RL)

RL is an interdisciplinary area of machine learning and optimal control concerned with how an intelligent agent ought to take actions in a dynamic environment to maximize the cumulative reward.

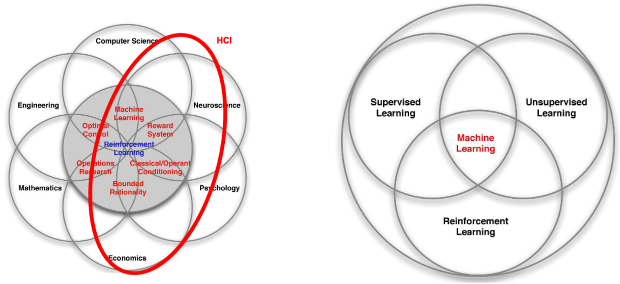


Example for state, action and rewards:

STATE:	ACTION:	REWARD:
Current word Target word Current finger pos/vel/acc Autocomplete	Muscle activation Coordinate Character Acceleration	Task completed successfully Task completion time Effort? Exertion? Effort?

What is the optimal action to take given a state?

Difference from RL to other ML paradigms



- No supervisor, only a reward signal
- feedback is delayed, not instantaneous
- Time really matters (sequential, non i.i.d data)
- Agent's actions affect the subsequent data it receives

Markov Decision Process

Tuple: $\{S, A, T, R, \gamma\}$

- S: A finite set of states (what letters have I typed so far)
- A: A finite set of actions (what letter do I type next, what muscle do I activate)
- T: Transition matrix: $p(s', a, s)$ given state and action, what is the prob of the next state
- R: Reward $R(s', a, ss)$ the immediate reward of taking the action a in state s.
- $\gamma \in [0, 1)$: the discount factor

Optimal action given a state

Policy is the agent's bahavior. It maps a state to an action to maximize the cumulative reward.
The goal is to find an optimal policy: π

Value Function

Value function is a prediction of future reward. Used to evaluate the goodness/badness of states. Therefore select between actions.

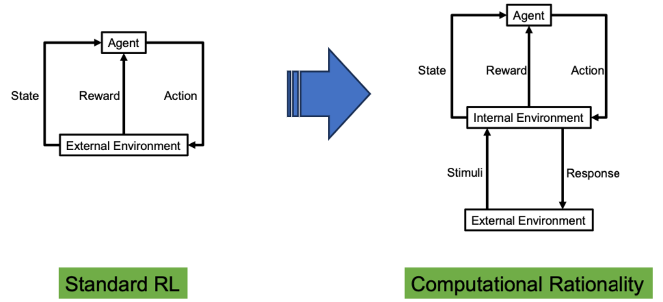
$$V_\pi(s) = E_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ...|s_t = s]$$

On Rewards It is a scalar feedback and describes how well an agent is doing at step t.
The reward hypothesis states that all goals can be described by the maximization of the expected cumulative reward.

Bounded Rationality

Supposed to look at Human-likeness.

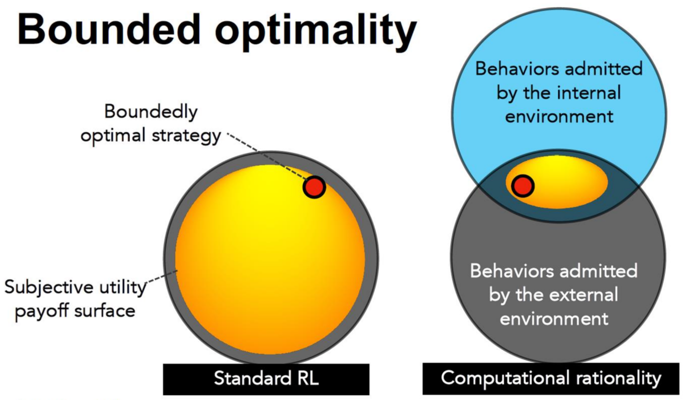
Comparison with Standard RL

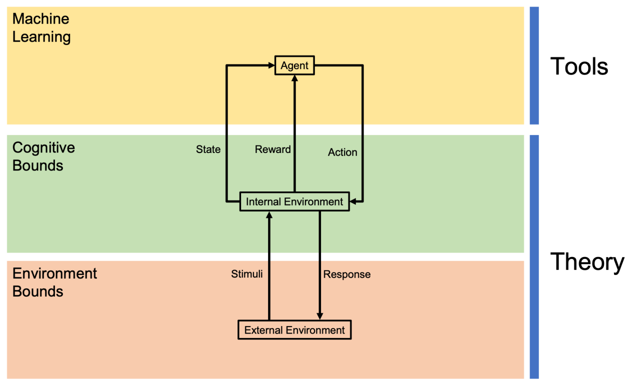


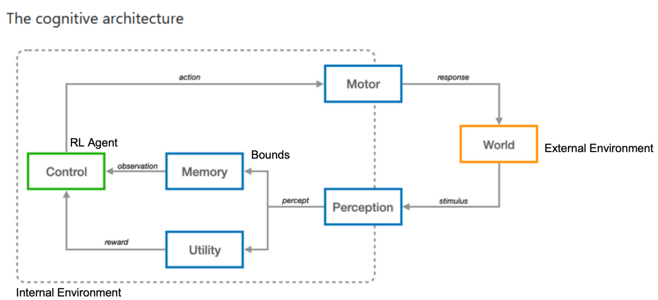
Internal vs External Environmen

- Humans acit in the external environment via their internal environment
- External environment: The world
- Internal environment: Cognition
- Agent: Decison maker

Interaction emerges as adaption within internal and external bounds.







Key cognitive capacities in HCI

- (Suvervisory Control): Adaptively deciding goals, allocating cognitive resources to tasks, and changing course of action when required.
- Memory: Forming, maintaining and accessing beliefs about objects that are not directly perceivable
- Attention: Selectively processing some part of the perceptual failed
- Reasoning: Applying transformation rules to beliefs to form new beliefs
- Gathering information and choosing between options

General props of Human Cognition

- Goal-oriented
- Adapts
- Learns
- Carries out computations on representations
- Is limited
- Requires energy and effort

1. Cognition is goal oriented

How do we choose what stimuli to direct our cognitive resources on. We use cognitive control to decide to which goal we direct thinking and action.

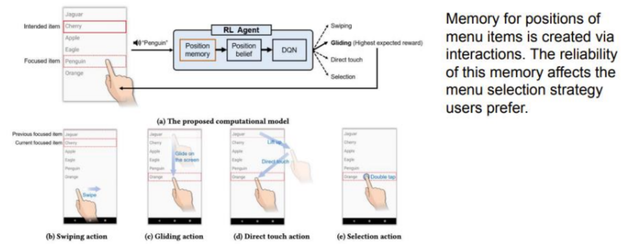
- Setting goals
- Directing resources and Attention
- Multitasking
- Task-switching
- Inhibiting distracting ideas

2. Cognition Adapts

Systems used by people and the way work is carried out changes all the time.
- Cognitive, motor and perceptual processes need to adapt constantly
- Update old beliefs with new ones
- Update old plans and form new ones
- Cognition is not only reacting to environments but also actively finding ways to function better

3. Cognition Learns

Computers are complex but opaque systems. We need internal representations to control them and need to have multiple memory systems to that end. It learns to use external aids.
Cognition also learns how to use external aids such as notes calculators etc. to augment our abilities. This changes the way we use cognition in interaction. Example is GUIs that have lead to forget commands in memory.
Example: Menu interaction of non-sighted users



4. Cognition computes based on internal representations

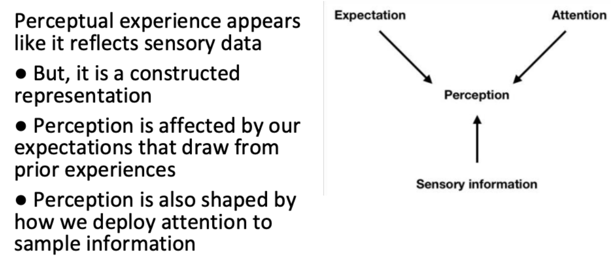
Cognition can reason about things that are not directly perceivable. It uses internal models of reality to reason, formulate goals and plans.
Examples are metaphors to help users understand an UI. Desktop metaphor uses spatial concepts that are rooted in everyday physical experiences.

5. Cognition is limited & costs energy

- Visual attention is spatially limited (periphery vs foveal region)
- Working memory is capacity-limited (only few mental representations active at a time, typically working memory is limited to 2-4 items)
- Forgetting occurs in long-term memory (we cannot remember everything we have experiences and thus forget details)
- Capacity for abstract reasoning and planning is limited (only think few steps ahead)

Perception

Perception is needed to regulate actions:
- UIs communicate their state via perception
- Visual, auditory and tactile perception each have own characteristics and roles in HCI



Elementary perceptual Tasks

- Discrimination: telling whether a difference occurs in sensory data
- Detection: telling whether an event of interest occurs or not
- Recognition: Categorizing a stimulus as something
- Estimation: estimate property of an object of event in the environment
- Search: localizing an object of interest

Gaze-based Interaction

Proble of selecting items on a computer with eye movements and fixations. These eyemovements obey Fitt's Law. Movement time of the eyes is proportional to distance and inversely proportional to size of the target.

- **World** On each trial, a circular target with a diameter "width" and a location (x,y) appears in the World. Target locations are randomly sampled from a bivariate Gaussian distribution centered at (0,0).
- **Perception** The location of target location stimuli are corrupted by Gaussian noise in human vision. The standard deviation of noise increases linearly with eccentricity from the fovea.
- **Memory** Sequences of stimuli are optimally integrated.
- **Utility** Rewards are received for getting closer to the target.
- **Motor** Intended eye movements (oculomotor actions) are corrupted by signal dependent Gaussian noise to generate responses.
- **Control** The Control module observes the memory and receives a reward. It learns a policy for making eye movements to the target that maximizes the reward.

Comparison with Cognitive Architectures

Cognitive Architectures	Computational Rationality
An information processor	An agent
Manually specified rule set	Policy is optimized (e.g., RL)
Bounded by cognition	Bounded by cognition
Architecture	Decision-problem