Karlsruher Institut für Technologie

ZUSAMMENFASSUNG

Mensch Maschine Interaktion

Institut für Telematik – Lehrstuhl für Pervasive Computing Systems

Sophie v. SCHMETTOW

SoSe 16

September 23, 2016 in Karlsruhe

Contents

Disclaimer

Dieses Dokument wurde im Rahmen des Master-Studiums für das Modul „Mensch-Maschine-Interaktion“ erstellt. Es stellt eine Zusammenfassung dar und dient zur Vorbereitung auf die mündliche Prüfung. Neben den Materialien der Vorlesung fließen auch weitere Quellen ein, um den behandelten Stoff auszuarbeiten. Auf den Verweis von Quellen wird verzichtet, da die Erstellung keinerlei wissenschaftlichen Zweck verfolgt und nur für den privaten Gebrauch bestimmt ist.

1 Introduction

1.1 Definitions

- **Pervasive Computing**
  - Pervasive Computing ≈ Ubiquitous Computing
  - pervasive = *alles durchdringend*
  - IBM-Definition: „Convenient access, through a new class of appliances, to relevant information with the ability to easily take action on it when and where you need“
  - Beispiel: Smartphones → relativ neue Technologie, daher viel höhrere Dynamik in der Entwicklung als beim Desktop.
  - Implizite Interaktionen: Gerät reagiert implizit auf User-Aktivität (insbesondere bei Smart Homes), jüngere lernen das besser (je älter man wird, desto mehr muss man anhand von Bezügen zu bereits Gelerntem/Metaphern lernen)
- **Human Adopted Production**
  - Comber and Maltby (1997) found that both overly simple and overly complex computing interfaces were low in usability
  - Wie kann man den User „bei Laune halten“?
    → Die aktuell erfahrene Komplexität muss im Mittelmaß sein
    → Lernsysteme greifen per GSR Stress Level des Users ab und tunen dementsprechend das Lernsystem
    → kognitive Paramater sind sehr individuell

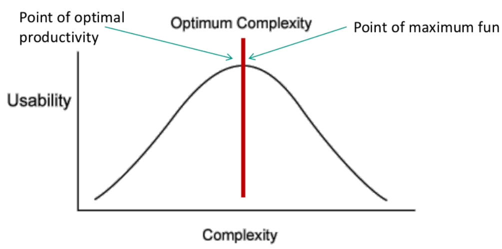


Figure 1: Human Adopted Production

- **Human-Computer-Interaction**
  - Important part of Pervasive Computing, e.g. Mobile Phone interfaces
  - Pervasive Computing Technology produces now most/all innovative human machine interfaces, e.g. augmented reality glasses
  - everything you see, touch, feel, do... to exchange information with a technical system
  - Wikipedia: “Human-Computer interaction (HCI) is the study of interaction between people (users) and computers. It is an interdisciplinary subject, relating computer science with many other fields of study and research. Interaction between users and computers occurs at the user interface (or simply interface), which includes both hardware (i.e. peripherals

and other hardware) and software (for example determining which, and how information is presented to the user on a screen)."

- SIG CHI: "Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them."

1.2 Applications

- Reprogramming Senses: Proximity Helmet (mit Arduino) → wird nach einer Weile zum Gefühl, so kann man neue Sinne erschaffen
- Augmented Reality, Mobile Computing User Interface Software Design
- Wearable Thermography: detektiert Überbelastung bei Sport...
- Ubiquitous Sensing: Lobster House, Büro welches sich nach Sonne dreht, jeder Quadratmeter individuell klimatisierbar

1.3 Interactive Systems Design

An interactive systems designer needs:

- Knowledge about people: Sociology, anthropology, psychology, culture
- Knowledge about technologies: Software, systems, electronics, communications, materials, databases, etc.
- Knowledge about activities and contexts: Communities of practice, information systems, organizations, knowledge management
- Knowledge about design: Fashion, interior, information design, architecture, product design

Design: The creative process of specifying something new and the representations that are produced along the way (e.g. site map, blueprints, sketches, etc.). It typically involves much iteration – both problem and solution evolve during design. Design is a spectrum of activities:

- Engineering design – using scientific principles: Architect designs buildings, urban planner designs roads, etc.
- Artistic design: Creative design where imagination is key
- Design as craft: Design is a 'conversation with materials', e.g. pottery designer works with clay, clothes designer works with fabrics, interior designer works with furniture, paints, lighting, etc.

1.4 PACT

PACT is a framework for designing interactions. People undertake activities, in contexts using technologies.
People:

- Physical differences: Height, weight, different capabilities in sight, hearing, touch,...
- Psychological differences: Different ways of working; different memory abilities (short term and long term) and spatial ability; different amounts of attention at different times; differences in perception and attention and – crucially – different 'mental models'

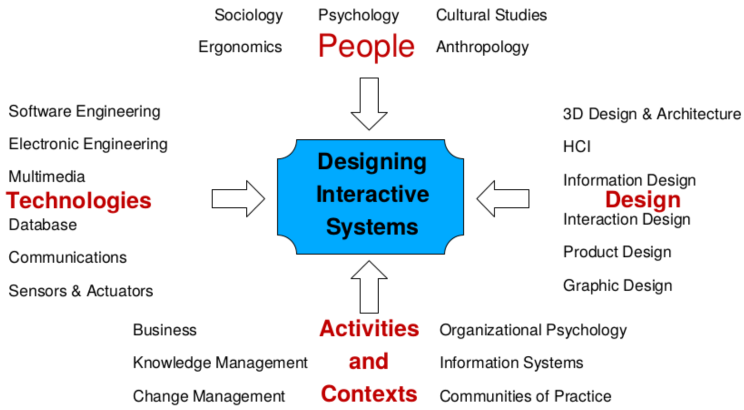


Figure 2: PACT method

- Mental Model: A mental representation of a real-world object or process, a set of associations with it. Describes the ways in which we think about things - about how we conceptualize things.
  → A key aspect of the design of technologies is to provide people with a clear model, so that they will develop a clear mental model but of course that depends on what they know already, their background, experiences, etc. etc.

- Usage differences: Experts versus novices, discretionary users of technologies, differences in designing for a heterogeneous group or a homogeneous group

Activities:

- Temporal aspects: How regular or infrequent are the activities? Continuous set of actions, or can be interrupted? Response time[1] from the system?
- Co-operation and Complexity: Working with others or not?
- Safety critical: What problems happen if something goes wrong?
- Content: What information and media are we dealing with?

Contexts:

- Physical environment is one sort of context: e.g. ATM or ticket machine versus computer at home
- Social context is important: Help from others, acceptability of certain designs
- Organizational context: Power structure, changes in life style, etc.

Technologies:

- Input: Some methods are needed to enter commands (tell the system what we want it to do). We also need to be able to navigate through the commands and the content of the system. We need to enter data or other content into the system. The medium needs to be suited for different contexts/activities.
- Output: The system has to tell us what is happening, i.e. provide feedback. It has to be able to display the content to us.
- Communication: Between person and technology and between devices → Bandwidth, speed

[1] 100ms for hand-eye coordination activity, 1 second for cause-effect activity, over 5 seconds and people quickly get frustrated.

- Content: Functional systems versus systems more focused on content

When doing a PACT analysis, **scenarios** are good for **evaluation and envisionment**. Scenarios are stories about people undertaking activities using technologies in contexts. You start to develop conceptual scenarios that cover the main activities that the technology has to support. Then you develop concrete versions of these for specific designs of the technology. E.g. a conceptual scenario might say 'Pete logs onto the computer' and a concrete version might be 'Pete clicks on the *log on* icon'.
A **Persona** is a profile of an archetypal person in the domain. Personas are synthesized from knowledge of real people in the domain. They need to have goals – describe what they are trying to achieve. Like scenarios, conceptual personas are abstract types – students, lecturers, etc. For design it is best to develop a few concrete personas who have hard characteristics such as age, interests, a name, etc. Try to bring the character alive – perhaps including pictures.

2 Evaluation/Observation

There are two basic classes / times for studies and evaluation (and many variants):

- **Formative**: At the beginning to inform about context and to study possible options
- **Summative**: To judge on the impact of a HCI design. A summative evaluation of a design might be a formative one for the next step.

Evaluation: Iterative design & evaluation is a continuous process that examines:

- Why to evaluate: to check users' requirements and that they can use the product and they like it.
- What to evaluate: a conceptual model, early prototypes of a new system and later, more complete prototypes.
- Where to evaluate: in natural and laboratory settings.
- When to evaluate: Formative: throughout design; Summative: finished products can be evaluated to collect information to inform new products.

Usability: The extent to which a product can be used by specified users to achieve specified goals with **effectiveness, efficiency and satisfaction** in a specified context of use (Figure 3).
Over-arching usability principles: **Learnability, effectiveness, accommodation** for its intended user population.
Evaluation classes:

- **Setting**:
  - Controlled settings: Setting conditions are controlled; non-controllable conditions are measured; e.g. Lab experiments, living labs
  - Natural settings: Study in "everyday" and natural conditions that cannot be controlled; some, but not all non-controllable conditions can be measured; e.g. Field studies, in-the-wild studies
- **Evaluation Time**:
  - Inspective: Inspection / Evaluation while run of an experiment or while use
  - Retrospective: Evaluation after run of the experiment or after use
  - Short term
  - Long term
- **Partner**:

Usability objective	Effectiveness measures	Efficiency measures	Satisfaction measures
Overall usability	• Percentage of tasks successfully completed • Percentage of users successfully completing tasks	• Time to complete a task • Time spent on non-productive actions	• Rating scale for satisfaction • Frequency of use if this is voluntary (after system is implemented)
Meets needs of trained or experienced users	• Percentage of advanced tasks completed • Percentage of relevant functions used	• Time taken to complete tasks relative to minimum realistic time	• Rating scale for satisfaction with advanced features
Meets needs for walk up and use	• Percentage of tasks completed successfully at first attempt	• Time taken on first attempt to complete task • Time spent on help functions	• Rate of voluntary use (after system is implemented)
Meets needs for infrequent or intermittent use	• Percentage of tasks completed successfully after a specified period of non-use	• Time spent re-learning functionsNumber of persistent errors	• Frequency of reuse (after system is implemented)
Learnability	• Number of functions learned • Percentage of users who manage to learn to a pre-specified criterion	• Time spent on help functions • Time to learn to criterion	• Rating scale for ease of learning

Figure 3: What do we want to know – Common usability metrics

  - The user: Gives direct feedback e.g. for use; Best for gaining new insight into context (If its an experiment: called “subject”)
  - The expert: Allows for best practice information; Reported expert experience may require many users / test subjects to be collected

- **Result type**:
  - Subjective: Results cannot be directly compared between subjects
  - Objective Results can be directly compared between subjects (e.g. using statistics)
  - Quantitative: Results are numbers
  - Qualitative: Results are text

Example Methods:

- Inspective, End User Focused:
  - Subjective Short Session: Think Aloud
  - Objective Short Session: Measurements of Workload, Interaction Process (e.g. Video Recording in User Setting / Usability Lab; Eye Gaze which is a quantitative/objective measure)
  - Subjective Long Session: Diary (Logging, asking questions from time to time e.g. through pop ups)
- Inspective, Expert Focused: Cognitive Walk Through
- Retrospective, End User Focused: Interviewing / Questionnaire

Data Gathering – Key Issues

1. Setting goals: Decide how to analyze data once collected
2. Identifying participants: Decide who to gather data from
3. Relationship with participants: Clear and professional; Informed consent when appropriate
4. Triangulation: Look at data from more than one perspective; Collect more than one type of data, e.g. qualitative from experiments and qualitative from interviews.
5. Pilot studies: Small trial of main study

2.1 Data Gathering – Interviews & Questionnaires

Types of interviews:

- **structured interviews**: pre-developed questions, strictly following the wording (e.g. introduction, purpose of the questionnaire); easy to carry out – but limited to the question set; more precise to evaluate
- **semi-structured interviews**: structured part + "open" questions ("Tell me your ideas on ...")
- **unstructured interviews**: used e.g. when little background information is available; minimizes the influence of the questioner

Questionnaires:

- Questions have orders (positive or negative, can be mixed)
- Version needs to be adapted to population.
- Instructions of use need to be clear.
- No excess whitespace, not too long, no compound sentences or jargon.
- frame: consent form, (monetary) compensation, time constraints, number of participants, introduction to scenario, scale type

Question and response format:

- Yes' and 'No' checkboxes
- Checkboxes that offer many options
- Rating scales: Likert scales (5-point scale, *strongly agree, agree, neutral, disagree, strongly disagree* → quantitative result), semantic scales
- Open-ended responses

Conducting the interview:

- contextual factors (light, noise...) should be kept constant, interviewer should provide a neutral attitude, interviewee might try to say what he thinks you want to hear → questions to cross-check answers needed

1. Introduction: introduce yourself, explain the goals of the interview, reassure about the ethical issues, ask to record, present the informed consent form.
2. Warm-up: make first questions easy and non-threatening.
3. Main body: present questions in a logical order.

4. A cool-off period: include a few easy questions to defuse tension at the end.
5. Closure: thank interviewee, signal the end, e.g. switch recorder off.

- documentation: notes, video taping, audio recording, photographs (always use visual impression)
- interview can be enriched by props, e.g. prototype scenario
- encouraging good responses: clarify purpose of study, promise anonymity, ensure questionnaire is well designed, Follow-up with emails/phone calls/letters, provide an incentive → 40% response rate is good, 20% is often acceptable

Standard Questionnaires in HCI:

- System Usability Scale (SUS)
- NASA Task Load Index (TLX)
- Questionnaire for User Interface Satisfaction (QUIS)
- Computer System Usability Questionnaire (CSUQ)

System Usability Scale (John Brooke, 1986 for Digital Equipment)

- Quick and dirty subjective measurement of usability in terms of Effectiveness, Efficiency and Satisfaction
- Used for hardware, software, mobile devices, websites, applications
- Evaluation: Positive questions get a 0-4 rating, negative ones a 4-0 rating, points are added and multiplied by 2.5 – good result if over 68 (max: 100)
- Benefits: Very easy scale (Likert); Useful in small sample sizes with o.k. results; Validity o.k. (you see differences in bad and good design)
- Restrictions: Score 0-100 (confusing, not a percentage); Not diagnostic, just to classify (no conclusion on specific problems possible)

NASA Task Load Index: Measures along 5 dimensions

- Mental demand: How mentally demanding was the task?
- Physical demand: How physically demanding was the task?
- Temporal demand: How hurried or rushed was the pace of the task?
- Performance: How successful were you in accomplishing what you were asked to do?
- Effort: How hard did you have to work to accomplish your level of performance?
- Frustration: How insecure, discouraged, irritated, stressed and annoyed were you?

It consist of two steps:

- Step 1: The dimensions are scored on a 100 point scale, in increments of 5 (20 scale)
- Step 2: User is then asked to perform 15 pair-wise comparisons to assess the weight for each of the measures (Asks the following question: "Which factor is the more important contributor to work load for the task?")

→ Second step reduces subjectivity greatly

Scoring procedure: For each dimension, sum the number of times it was selected as the leading pair wise factor. Divide this sum by 15 (the number of pairwise comparisons). Multiply this result by the score for each dimension. Sum the resulting dimension scores
Result: 0-100 score, weighted for dimensions considered important by the participant.
The optimum workload has to be determined case by case. Overload leads to poor performance due to stress. Underload leads to poor performance due to boredom.

Online Questionnaires:

- Advantages: Relatively easy and quick to distribute, Responses are usually received quickly, No copying and postage costs, Data can be collected in database for analysis, Time required for data analysis is reduced, Errors can be corrected easily
- Problems: Sampling is problematic if population size is unknown, individuals cannot really be prevented from responding more than once, individuals have also been known to change questions in email questionnaires

Problems with interviews and questionnaires:

- retrospective, user may not remember (correctly) how he did everything
- some aspects of interaction cannot be described verbally
- user may try to answer how he feels is expected → observation shows how he actually does it, however: obtrusiveness (user is disturbed by the investigator's presence, time needed to decrease this effect)

2.2 Data Gathering – Observation

Types of observation:

- Direct observation in controlled environments
- Direct observation in the field: Structuring frameworks, Degree of participation (insider or outsider), Ethnography
- Indirect observation: tracking users' activities: Diaries, Experience Sampling Method (ESM), Interaction logging, Video and photographs collected remotely by drones or other equipment, web analytics

→ Video, audio, photos, notes are used to capture data in both types of observations

Structuring frameworks
Three easy-to-remember parts:

- The person: Who?
- The place: Where?
- The thing: What?

A more detailed framework (Robson, 2014):

- Space: What is the physical space like and how is it laid out?
- Actors: What are the names and relevant details of the people involved?
- Activities: What are the actors doing and why?
- Objects: What physical objects are present, such as furniture

- Acts: What are specific individual actions?
- Events: Is what you observe part of a special event?
- Time: What is the sequence of events?
- Goals: What are the actors trying to accomplish?
- Feelings: What is the mood of the group and of individuals?

Materials that might be collected in observations:

- Activity or job descriptions.
- Rules and procedures that govern particular activities.
- Descriptions of activities observed.
- Recordings of the talk taking place between parties.
- Informal interviews with participants explaining the detail of observed activities.
- Diagrams of the physical layout, including the position of artifacts.
- Other information collected when observing activities: Photographs of artifacts (documents, diagrams, forms, computers, etc.), Videos of artifacts, Descriptions of artifacts, Workflow diagrams showing the sequential order of tasks, Process maps showing connections between activities

2.2.1 Observation in a controlled environment

Think Aloud While using an application, a user is constantly explaining what he is thinking what he is doing. So-called cooperative usability evaluation (users work as co-evaluators), maximizing data from a simple testing session.
→ The quality of the evaluation depends on

- (as always) selection of test candidates: non-expert users preferred, all should have comparable visualization skills
- appropriate preparation of the candidate
- appropriate setting so that a natural usage can be guaranteed: quiet and comfortable, no interference from examiner, warm-up phase, as expectation free as possible

Think Aloud Session (c.f. table in Benyon p.280):

- preparation: explain system and setting (how test works, that system is tested and not the participant (!), that they should talk continuously) and expectation (that he should verbalize what he is doing and thinking about, just what comes to mind)
- Recording device and experimenter should stay out of sight
- Encourage participant to keep talking
- Record video or audio, transcript

→ Qualitative, mostly subjective

→ Generalizations difficult

→ Psychological theories needed for interpretation

→ Helpful for intermediate formative design improvements

2.2.2 Observation in the Field

Laboratory-based usability testing and field studies can complement each other. One may use a field study to evaluate initial ideas and get early feedback, adapt the design accordingly and then perform a usability test do check specific design features. Then again a field study to see what happens when used in natural environment. Then some final design changes. **Ethnography**: Ethnography is an interpretivist technique that includes participant observation and interviews.
The goal is to experience the participant and his context. Ethnographers immerse themselves in the culture that they study. A researcher's degree of participation can vary along a scale from 'outside' to 'inside'. Analyzing video and data logs can be time-consuming. Collections of comments, incidents, and artifacts are made. The co-operation of people being observed is required. Data analysis is continuous. Questions get refined as understanding grows. Reports usually contain examples.
Online-Ethnography (Netnography): Observing interactions in the information world. Interaction online differs from face-to-face! Virtual worlds have a persistence that physical worlds do not have. Ethical considerations and presentation of results are different.

Living Labs

Ubicomp studies: Types:

- Studies of current behaviour – What people doing now: better understanding of current use of technology, implications for future technology; open-ended questions; results=stories; starts from hours to weeks
- Proof-of-concept studies – Does my technology function in the real world: Validation of a (new) technology, often several days long
- Experience studies – How does using my prototype change people's behaviour or allow them to do new things: using the technology for a longer time (often several weeks), Wizard-of-Oz type studies (subjects interact with a computer system that subjects believe to be autonomous, but which is actually being operated or partially operated by an unseen human being)

Study Design

- start with a concrete research goal question that should be answered while study
- Study design document:
  - Research question/hypothesis
  - Detailed participant profile
  - What will your participants do during the study: free or specific problem, do they use their own devices etc.
  - Detailed timeline description/How long will the study be
  - What data will you collect: quantitative (to *compare* results) and qualitative (to better *understand* results) data maximizes insight
  - Analysis method
  - Method of drawing conclusion/validating hypothesis

2.2.3 Indirect observation methods

Can be used in controlled and uncontrolled environments. Tracking users' activities using Diaries, Interaction logs, Web analytics. Need to be specific for specific application areas and settings.

Logging: Select appropriate data to log at the right frequency, list specific questions that you expect to answer from the log data (best conduct small pilot study beforehand) e.g. Web analytics: Typically focus on the number of web visitors and page views.

Experience Sampling Methodology: Uses short questionnaires that participant fill out at various points throughout the day (either randomly distributed, scheduled or at an event).

Diaries: Participant has freedom when to enter information, but how to remind them to complete the diary?

2.2.4 General Considerations

How long should the study be? Depends on:

- type of study: current behaviour, proof-of-concept, experience
- novelty of system
- effort from participants: measurement time needs to be restricted if it requires a lot
- frequency of use: high frequency reduces required measurement time

Selecting Participants (should obviously not be from the designer group):

- Representation of Participants to the intended user group (e.g. the population, the worker,...)
- Grouping of Participants: one or multiple groups? Group selection based on
    - self-reported expertise (e.g. novice, intermediate, expert)
    - frequency of use (e.g. number of application use per day)
    - amount of experience (e.g. with working in a application domain, in days, month ..)
    - demographics (gender, age, location, etc.)
    - different activities the participants have to perform (e.g. each group only performs one type of work for a program)
- Data Sampling Strategy:
    - Random Sampling: Everyone has equal probability to be selected as participant based on a list
    - Systematic Sampling: Based on predefined criteria, e.g. every 10th person entering the ECE Centre
    - Stratified Sampling: Additionally, it is important to select people reflecting the distribution in your intended user group. So you care e.g. that your final set contains 50% male and 50% female
    - Samples of convenience: Volunteer based. Must be adjusted to the wanted user group.

Sample size depends on acceptable error. 3-4 participants are enough to identify major problems, according to Nielsen on average 5 are enough in an early stage design (formative evaluation) to identify 75-80% of usability problems. Best perform a pre-test where participants have to first detect known usability issues. The average percentage p of found usability issues over all the participants can be plugged into the formula $1-(1-p)^n$ to calculate the average amount of found issues for n participants used.

Study design:

- Within subjects: One participant gets several task options, Results of task options are compared between the participant

- Between subjects: Any subject gets the same task, compare how different users perform the task. Often users are grouped in categories beforehand (novice, expert...) Then user group performance is compared.
- Mixed design studies are possible

Order of test tasks needs to be balanced across participants (at least of the tasks are somewhat related) → impossible if tasks depend on each other.

Interpreting the data:

- Reliability: does the method produce the same results on separate occasions?
- Validity: does the method measure what it is intended to measure? Internal Validity (certainty that you know the cause) vs. External Validity (result can be generalized)
- Ecological validity: does the environment of the evaluation distort the results? Is the result transferable to a general environment?
- Biases: Are there biases that distort the results?
- Scope: How generalizable are the results?

2.3 Inspections – Evaluation without User

Types of Inspections:

- Experts use their knowledge of users & technology to review software usability. Expert critiques can be formal or informal.
- Heuristic evaluation is a review guided by a set of heuristics.
- Walkthroughs involve stepping through a pre-planned scenario noting potential problems.

Heuristic Evaluation (Jacob Nielsen, 1990s)

- Visibility of system status.
- Match between system and real world.
- User control and freedom.
- Consistency and standards.
- Error prevention.
- Recognition rather than recall.
- Flexibility and efficiency of use.
- Aesthetic and minimalist design.
- Help users recognize, diagnose, recover from errors.
- Help and documentation.

Heuristics for websites focus on key criteria (Budd, 2007): Clarity, Minimize unnecessary complexity & cognitive load, Provide users with context, Promote positive & pleasurable user experience.

Stages of heuristic evaluations:

1. Briefing session to tell experts what to do.

2. Evaluation period of 1-2 hours in which: Each expert works separately; one pass to get a feel for the product; a second pass to focus on specific features.

3. Debriefing session in which experts work together to prioritize problems.

Advantages: Few ethical & practical issues to consider because users not involved.
Disadvantages: Can be difficult & expensive to find experts as they need to have knowledge of application domain and users. Important problems may get missed; Many trivial problems are often identified; experts have biases.

Cognitive Walkthrough:
Focus on ease of learning. Designer presents an aspect of the design & usage scenarios. Expert is told the assumptions about user population, context of use, task details. One or more experts walk through the design prototype with the scenario. Experts are guided by 3 questions:

- Will the correct action be sufficiently evident to the user?
- Will the user notice that the correct action is available?
- Will the user associate and interpret the response from the action correctly?

Variation – Pluralistic Walkthrough: Experts work separately, then discuss. Walkthroughs are focused so are suitable for evaluating small parts of a product.

Predicive models:
Provide a way of evaluating products or designs without directly involving users and are less expensive than user testing. Their usefulness limited to systems with predictable tasks - e.g. telephone answering systems, mobiles, cell and smart phones. They are based on expert error-free behavior. Fitts' law predicts that the time required to rapidly move to a target area is a function of the ratio between the distance to the target and the width of the target. Thus, it applies to clearly defined tasks with limited key presses, such as data entry and smart phone use.

Workload Assessment: Criteria for Creating an Measure of Mental Workload:

- Sensitivity: index must be sensitive to changes in task difficulty or resource demand
- Selectivity: index should NOT be sensitive to changes unrelated to resource demands
- Diagnosticity: index should indicate not just that workload is varying but the cause of variation
- (Un)Obtrusiveness: an index should not interfere with or contaminate the primary task being assessed
- Reliability (Reproducibility): index should produce the same estimate for a given task and operator
- Bandwidth: the index should respond to high-frequency (quick) changes in workload

Workload can be assessed via performance metrics (time, speed, strength) or physiologic metrics (heart rate, respiration, oxygen uptake). Tool: NASA TLX

3 Human Information Processing (HIP)

3.1 Senses

A sense is a system that consists of a sensory cell type (or group of cell types) that respond to a specific kind of physical energy, and that correspond to a defined region (or group of regions) within the brain where the signals are received and interpreted.
Six external senses: sight, hearing, touch, smell, taste, balance
Three internal senses: thermoception, nociception (for pain), proprioception

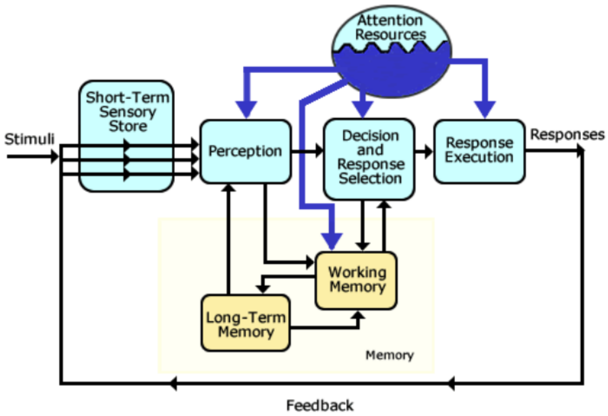


Figure 4: HIP – Overview

3.2 Short-Term Sensory Storage (STSS)

Sensory Memory is the retention, for brief periods of time, of the effects of sensory stimulation.
→ Collecting information for processing, holding it briefly while initial processing is going on, filling in the blanks when stimulation is intermittent.
STSS does not need explicit attention, it works pre-attentive and is not part of the conscious memory.
Note that STSS != short term memory.
Two most important types of short-term sensory memory:

- echoic memory (lasts $\approx 2 - 10s$ after stimulus) → e.g. "What did you say?"
- iconic memory (lasts $\approx 0.5 - 1.0s$ after stimulus) → e.g. grid of letters

3.3 Perception

Human capability of information processing is limited! It is not only of interest, how much information is presented to a human, but

- how much information is transmitted from stimulus to response
- capacity of the information channel
- how rapidly information is transmitted (bandwidth)

Raw sensory data must be interpreted and given meaning. This process is called perception. It generally happens automatically (little attention needed) and fast (in contrast to cognitive processes).

- driven by sensory inputs → bottom-up
- driven by long-term memory → top-down

3.4 Attention

Attention is the cognitive process of selectively concentrating on one aspect of the environment while ignoring other things. It is the crucial element for cognitive processes as learning or task execution. Types of attention:

- selective attention: willingly select focus
- focused attention: respond to external events
- divided attention: simultaneous focusing on different events

Allocation Model of Attention (Kahneman, 1973): Limited amount of "processing power" at our disposal. Task execution depends on how much of our attention "capacity" we can spare on it.

Controlled and Automatic Attention (Schneider and Shiffrin, 1977):

- Controlled processing makes heavy demands on attentional resources, is slow and limited in capacity, and involves consciously directing attention towards a task.
- Automatic processing makes no demands on attentional resources, is fast, unaffected by capacity limitations, unavoidable and difficult to modify, and is not subject to conscious awareness

Selective and Focused Attention: Concentration on one stimulus source needed for e.g. perception or learning (visual sampling or scanning). Being too selective is referred to as "cognitive tunneling".
Negative examples: selecting cues that stand out rather than useful ones (e.g. during a presentation/argumentation).
Humans have tendency to be distracted →important e.g. for directing attention to warning/error messages
Differences in errors between selective attention and focused attention: intentional selection of the wrong source vs. unintentional external influences.

Divided Attention: C.f. attention models. Limited attention capacity of humans, important e.g. for layout of instruments in a cockpit (see Gestalt laws). Differences in errors between focused attention and divided attention: some of our attention is directed to stimuli we do not wish to process vs. our limit to attend to all stimuli we wish to process.
→ Only a certain maximum of attention capacity can be divided to tasks!
Attention can be directed at a particular task and/or divided between a number of different tasks. Practice reduces the amount of attention required by a particular task (e.g. typing blindly on a keyboard while reading a text). Attention and awareness are closely linked.

3.5 Vigilance

Vigilance is an aspect of attention which refers to detecting a rare event or signal in a desert of inactivity or noise. It means to detect signals over a long period of time which are intermittent, unpredictable, and infrequent, but it is known that they happen.
Examples of vigilance tasks: security inspector x-raying luggage, quality control in production
→ vigilance level (steady-state level performance)
→ vigilance decrement

Paradigms:

- free-response paradigm: a target event occurs at any time, non-events are not defined; example: power plant monitor supervision
- inspection paradigm: events occur at fairly regular intervals; some are events, most are non-targets; example: quality control (most items are ok, only some have defects)
- successive vigilance paradigm: target stimulus has to be remembered; successive events have to be compared to the target stimulus; example: detect if a color is darker than the initial target
- simultaneous vigilance paradigm: all events/information needed for discrimination are present at the same time; example: compare many types of garment to a standard piece of fabric
- sensory vigilance paradigm: signals represent changes in the auditory or visual intensity; example: color changes
- cognitive vigilance paradigm: signals represent "information" (symbolic or alphanumeric stimuli); example: proofreading a manuscript

3.6 Workload and Measurement – Signal Decision Theory

The SDT model assumes that there are two stages of information processing in the task of detection: Sensory evidence is aggregated concerning the presence or absence of the signal. A decision is made about whether this evidence indicates a signal or not.
Signal detection & neural activity: External stimuli generate neural activity. On average, there will be more neural evidence when a signal is present than when it is absent.

Types of noise:

- internal (in our brain, e.g. firing rate of neurons varies for the same stimulus)
- external (in the data) internal response (in the observer's brain)

The probability distributions for internal response can be different for noise and for signal! Internal response as neural activity resulting from a stimulus: Optimal beta (bias) should be set where the

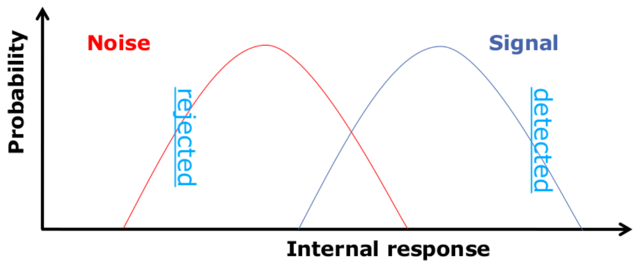


Figure 5

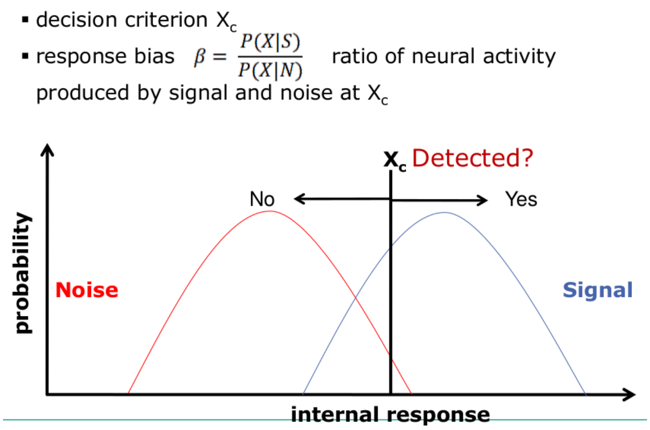


Figure 6

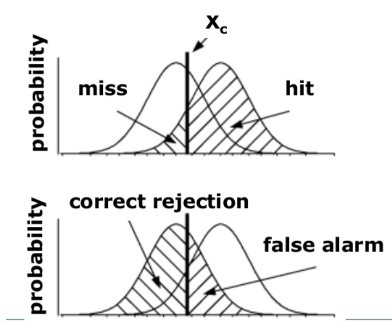


Figure 7: signal classifications revisited: hit, miss, false alarm and correct rejection

probability of signal and noise are equal. β_{opt} tells us where the bias should be, while β tells us where

the bias really is. $\beta_{opt} = \frac{P(N)}{P(S)}$
Usually, humans adapt "their" beta not enough (too conservative or too risky):

→ Over-estimation of rare events

→ Under-estimation of frequent events

→ "Sluggish beta" refers to highly focused attention processes, rare attention or sub-conscious processes underestimate rare events.

Sensitivity (d') is the resolution of the detection mechanism. It refers to the separation of noise and signal distributions along the X-axis and corresponds to the separation of the means of the two distributions.

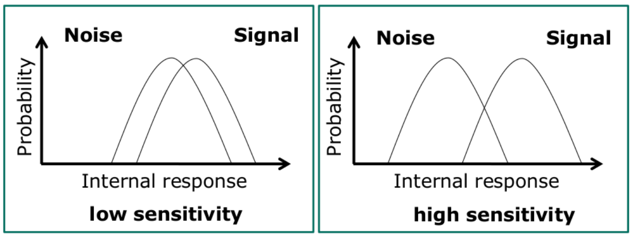


Figure 8: Sensitivity

3.7 Memory

Human memory consists of two major components: Working memory (formerly: short-term memory) and long-term memory. Memory "processes" include recall, recognition, chunking and rehearsal. Human memory is multi-modal! The brain makes up 2% of a person's weight. It consumes 20% of the body's energy, even at rest. Brain energy consumption is 10 times the rate of the rest of the body per gram of tissue. Average power consumption of a typical adult lays at 100 Watts brain consumes 20% (20W) of this.

Working Memory:

- Memory storage for up to 30 seconds, after that short period, the contents decay or are displaced.
- rapid access: ≈ 70ms access time
- storage size: 3-4 "chunks" (NOT: 7+–2 elements!)
- it is easy to overwrite the contents (intentionally and unintentionally!)
- can store two different types of data at the same time: visual information (visuo-spatial sketchpad) & verbal information (articulatory loop)
- to maintain the contents of working memory: rehearsal is needed
- to persistently store information from working memory: move it to long-term memory

Types of WM:

- Verbal: Phonological Store
  → Retrieval: Articulatory loop
  → "Programmed" and associated by words written, spoken (the latter are simpler to remember)
- Visual; Visuospatial Sketchpad, stored often in form of images
- Operate in parallel, but there is only a restricted number of attention levels

Long-Term Memory:

- "unlimited" in capacity
- lasts from a few minutes to life time
- slow access: 100ms seconds access time
- multi-modal memory: smell as strong trigger to long-term memory (sound as most efficient cue for working memory)
- three types (could be clustered otherwise, depending on psychological model): episodic – serial memory of events, procedural – knowledge of how to do things, semantic – structured memory of facts, concepts, skills
- semantic LTM (→ relationships between bits of information, inference through inheritance) derived from episodic LTM
- Forgetting: decay (information is lost gradually, but very slowly), interference (retroactive interference: new information replaces old, proactive inhibition: old may interfere with new); affected by emotion (can subconsciously 'choose' to forget)

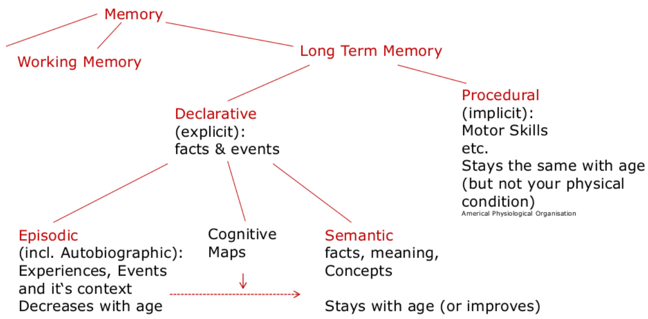


Figure 9

Long Term Memory Processes:

- Encoding is the process by which information is stored in memory.
- Retrieval is the means by which memories are recovered from long-term storage.
- Forgetting is the name of a number of different possible processes by which we fail to recover information.
- recall ("Erinnerung"): active memory search to retrieve a particular piece of information
- recognition ("Wiedererkennung"): searching the memory and deciding whether the retrieved piece of information matches a given information; recognition is generally easier and quicker than recall!

→ Icons often use metaphor to recall an associated object or activity[2]

- rehearsal: There is no direct link from perception to LTM! The process of repeating information in working memory. This facilitates the short-term recall of information and its transfer to long-term memory. The amount stored is proportional to the number of rehearsals!
- chunking: grouping of items into more meaningful units

[2] Horton's checklists for icons: Understandable, Familiar, Unambiguous, Memorable, Informative (concept important?), few (¡20), distinct, attractive, legible (tested in all view contexts), compact (means minimal), coherent (can be separated from other icons and the background), extensible (means size scalable)

3.8 Memory Design Guidelines

[3]

M1: Organize information into a small number of "chunks".

M2: Try to create short linear sequences of tasks.

M3: Use persistence, so do not flash important information onto the screen for brief time periods.

M4: Do not "overwrite" the contents of working memory by giving additional tasks to the user.

M5: Organize data fields to match user expectations or to organize user input (e.g. the automatic formatting of phone numbers)

M6: Provide reminders or warnings of the state the user has reached in an operation.

M7: Provide ongoing feedback on what is happening and/or what has just happened.

M8: The user interface should behave in consistent ways at all the times for all screens.

M9: Terminology, icons and the use of color should be consistent between screens.

3.9 Decision Making

Semantic coding: without memory, there is no "thinking" or decision-making! After the perception of the stimulus, a response needs to be selected.
Automatic vs. controlled decisions:

- automatic($\rightarrow$ fast): little or no attention required, learned reflexes or behavior, a long-term memory procedure is executed nearly automatically in response to the stimulus
- controlled ($\rightarrow$ slow): attention required, typically conscious of thoughts, interaction with WM and LTM systems

3.10 Response and Feedback

After a decision has been made, it has to be executed by complex motor movements.
Feedback-loop: we observe the consequences of our actions, producing closed-loop feedback $\rightarrow$ the model is circular rather than linear (Figure 4)!

3.11 HIP by Numbers

CMN-Model (Card, Moran & Newell, 1983):
Three interacting subsystems, each with processor & memory and described by parameters; serial & parallel processing

1. Perceptual processor: receives sensory input (audio & visual), codes info symbolically, output into audio & visual image storage (WM buffers)
2. Cognitive processor: input from sensory buffers, can access LTM to determine response according to previously stored info, output response into WM
3. Motor processor: input response from WM, carry out response

Subsystem interactions in terms of input/output. Processing:

[3] Guideline: specific and practical rules for solving problems of UI design;
Principles: help for analyzing and comparing design alternatives;
Theories and Models: describe objects and actions with consistent terminology and give a comprehensible explanation of connections

- serial action: e.g. pressing key in response to light
- parallel perception: e.g. driving, reading signs & hearing

Parameters (based on empirical data):
Processors have: cycle time (τ)
Memories have: storage capacity (μ), decay time of an item (δ), info code type (κ, physical, acoustic, visual & semantic)

- Processor cycle time: $\tau_{per} = 100ms$, $\tau_{cog} = 70ms$, $\tau_{mot} = 70ms$ → total time: $\tau_{per} + \tau_{cog} + \tau_{mot}$
- Visual image store: μ_{per} =17 letters (visual) / 5 letters (auditive), μ_{WM} =3-7 chunks
- Decay time (half-life index): $\delta_{per} = 200ms$ (visual), $\delta_{per} = 1500ms$ (auditive); $\delta_{WM} = 7s$ (for 3 chunks), $\delta_{WM} = 73s$ (for 1 chunk)
- Info code type: κ = physical → physical properties of visual stimulus (e.g. intensity, color, curvature, length)

Principle of Operation – Power Law of Practice: task time on the n^th trial $T_n = T_1 n^{-0.4}$ → you get faster the more times you do it, applies to skilled behavior (perceptual & motor) but not to knowledge acquisition
⇒ These numbers are used later for the interaction design models

4 Perception

4.1 Visual Perception: Illusions and Gestalt Theory

Visual perception is the process of extracting meaning from sensory information. It is concerned with recognition and understanding. Vision is an easier process concerned with detecting color, shapes or edges of objects. Vision does not necessary require an understanding of the world surrounding us.
The Gestaltists, a group psychologists, identified a number of properties that can be regarded as innate to all humans.

Gestalt Laws: methods that the brain uses to simplify recognition by ordering them

1. Proximity: Objects that are close in space or time tend to be perceived together. This can be used e.g. for UI arrangement of buttons or information.
2. Common fate: Principle – Objects that "move" together are seen as related
3. Prägnanz – Law of Good Gestalt: Complex an unknown figures are automatically separated into known simple forms to make sense of them
4. Closure: Tendency to see things as complete objects even though there may be gaps in the shape of the objects. Closed figures are perceived more easily than incomplete or open figures.
5. Continuity: Tend to perceive smooth, continuous patterns instead of disjoint, interrupted patterns.
6. Similarity: Similar figures tend to be grouped together.
7. Part-whole relationships
8. Symmetry
9. Area Principle (also called the smallness principle): Objects with small area tend to be seen as the figure, not the ground.

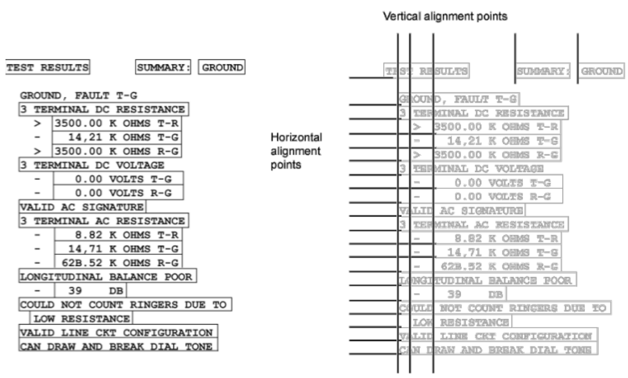
10. Surroundedness Principle: An area that is surrounded will be seen as the figure and the area that surrounds will be seen as the ground.

Other Principles of Perception:

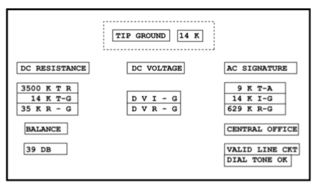
- Stimulus Intensity: We respond first to the intensity of a stimulus and only then do we begin to process its meaning.
- Proportion: can be used to represent logical hierarchies (e.g. font size of headings, subheadings...)
  → Golden Ratio: $\frac{a+b}{a} = \frac{a}{b}$The golden ratio expresses the relationship between two aspects of a form such as height to width and must equal 0.618; is seen as aesthetic
  → Fibonacci: A sequence of numbers in which each number is the sum of the two preceding numbers. The relationship between the numbers in the Fibonacci series is similar to the golden ratio.
- Screen Complexity (Tullis, 1984): can be used to calculate the relative complexity, and therefore the difficulty, of a design. This measure of complexity uses information theory (Shannon & Weaver, 1949) $C = -N \sum_{n=1}^{m} p_n \log_2 p_n$ with
  - C = complexity of the system in bits
  - N = total number of events (widths or heights)
  - m = number of event classes (number of unique widths or heights)
  - p_n = probability of occurrence of the n^{th} event class (based on the frequency of events within that class)

  Calculating screen complexity:
  1. Place a rectangle around every screen element
  2. Count the number of elements and the number of columns (vertical alignment points)
  3. Count the number of elements and the number of rows (horizontal alignment points)

  → lowering these numbers reduces visual complexity (law of common fate, closure)



(a) Complexity Analysis



(b) Improved Interface

Figure 10: Example

→ Comber and Maltby (1997): both overly simple and overly complex screens low in usability (in terms of Effectiveness, Learnability, Attitude)

Depth Perception: Types of depth cues:

- Primary depth cues (relevant e.g. to immersive virtual reality systems)
  - retinal disparity: As our eyes are approximately 7cm apart, each retina receives a slightly different image of the world. This is processed by the brain and interpreted as distance information.

- Stereopsis: Process by which the different images of the world received by each eye are combined to produce a single three-dimensional experience.
- Accommodation: A muscular process by which we change the shape of the lens in our eyes in order to create a sharply focused image. The information from the muscles is unconsciously used for depth information.
- Convergence: Over distances of 2-7m we move our eyes more and more inwards to focus on an object at these distances. This process is used to provide additional depth information.

- Secondary depth cues (more relevant e.g. to non-immersive applications such as games)
  - Light and shade: An object with light and shadow improves the depth perception.
  - Linear perspective
  - Height in the horizontal plane
  - Motion parallax: Near objects are perceived to flash by quickly, objects further away as slower.
  - Overlap: e.g. overlapping windows in a GUI
  - Relative size: see sun + cloud
  - Texture gradient: textured surfaces appear closer

4.2 Usability: Goals – Principles – Guidelines

- Usability Goal – Easy to use: most people do not want to struggle with tools are interested in completing their tasks.
- Design Principle – Simplicity: Simple things require little effort and can often be accomplished without much thought. If interaction designs are guided by the principle of simplicity, they will be easier to use.
- Project Guideline – Dialog Boxes: All dialogue boxes should present only the basic functions that are most often used and that other, less used functions can be accessed using an expandable dialogue with a link for "More Options".

4.3 Colors

Additive (displays) and subtractive (printed) color mixing.

Limitations:

- Blue Color is on the edge of the spectrum → Avoid using blue color for small and tiny elements.
- The ability to distinguish color is directly related to the size of an object!
- Shapes are identified by their edges. Edges are faster identified by lines than by colors.
- Better color perception in the middle of the focus.
- About 8% of male (0.4% of female) have deficiencies to perceive colors correctly. Most common deficiency is detection of green
- Colors may cause problems leading to divided attention (as shown in Stroop test)
- Colors may cause emotional response: varies depending on culture, age, sex → Industries, professional communities, corporate have their own colour connotations

Colors for Design – Color Coding – Optimal Colors

- For Clarification, Relation, Differentiation: e.g. Subway Maps

- Support finding: e.g. important element in a text
- Comprehension, Retention and Recall: e.g. Levels of danger, different measurements in a scatter plot etc.
- Color coding can improve recall, search-and-locate tasks, decision judgements → overall performance but does not replace good structure!
- Avoid any incompatible color combination, e.g. bright red on blue (background to foreground contrast)
- Color presentation and perception change with medium and display technology; light in the workspace may influence color perception
- Thorell & Smith: red, blue, green and yellow are most beneficial in learning environments
- Using a color code: Max. 4 main colors, each having max. 4 variants; structure the content and identify the content that should be supported with a color

4.4 Hearing

Loudness (dB): dB is a logarithmic scale; frequency range of hearing 20Hz-20kHz

Four Stages of Auditory Perception:

- Transduction: Translation of sound vibrations into neural impulses
- Auditory grouping: Segregation into separate streams + Integration of sound in coherent streams (based on similar harmonic, frequency, location etc.)
- Scene Analysis
- Interpretation

Audio User Interfaces: Vision and Hearing go together, but audio can give impression from area that are beyond vision (audio can tell your eyes where to look). However, people become habituated to continuous sounds and ignore important sound based information. Continuous sounds requires resources from the human, even if it is a continuous sound, leading to additional stress. Audio can be perceived faster than visual cues. Audio is transitory! Vision is often not!

(Non-)Speech: We can speak faster than we can write, but spoken interaction contains more redundancy in terms of content. Speech requires knowledge of language. We can read faster than we can listen. It is often therefore more efficient to read than to listen. Listening is a liner task. Reading can jump! Auditory feedback can be non-speech. Beyond simple forms, sounds must be learned or familiar. They can be annoying and ambiguous and there are not so many expressions possible.
→ Use of Sound:

- Redundant Coding: E.g. the “click” sound when pressing a GUI button
  → redundant coding aids recalling an interaction by giving additional association
  → can increase efficiency for tasks, as experts can react faster (on sound only)
  → people with deficits (e.g. color blindness) may benefit from redundant coding
  → Can be used for emotion (e.g. the “miauw” sound of my camera)
  → Most important: Positive / Negative-Feedback
- Speech applications

Use Case: Nomadic Radio

5 Design Analysis

5.1 Foundation Laws

Physical Models – Fitts' Law:
Physical models can predict efficiency based on the physical aspects of a design. They calculate the time it takes to perform actions such as targeting a screen object and clicking on it. Fitts' law can be used to determine the size and location of a screen object. It states that the time it takes to hit a target is a function of the size of the target and the distance to that target:

- Index of Difficulty (ID): Quantifies the difficulty of a (motor pointing) task based on width and distance

  $$ID = \log_2(A/W + 1)$$

  A=amplitude, W=width

- Movement Time (MT): Quantifies the time it takes to complete a task based on the difficulty of the task (ID) and two empirically derived coefficients that are sensitive to the specific experimental conditions

  $$MT = a + b\log_2(A/W + 1) = a + bID$$

  (assuming linear movement speed).[4] $a\,[sec]\,, b\,[sec/bits]$ constants dependent on the input device → Fitts' law (1954) predicts that the time to acquire a target is in logarithmic relation the target size.

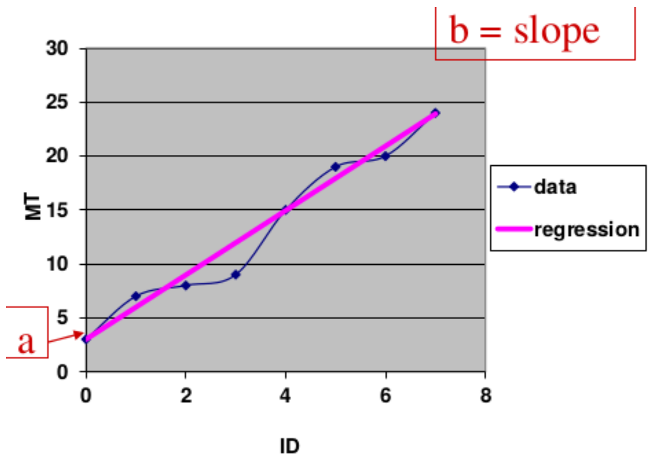


Figure 11: Linear Regression Model

- Index of Performance (IP), also called throughput (TP): Based on the relationship between the time it takes to perform a task and the relative difficulty of the task

In practice A corresponds to the distance from starting position, W to the size of target along the line of motion. Common parameter values (e.g. for a mouse) are $a = 50ms, b = 150ms/bit$. Fitt's Law is a predictive model of time to point at an object. Implications for interaction design in GUIs:

- Doubling the distance increases time but does not double it.
- Increasing target size enables more rapid pointing.

[4]Original form: $MT = a + b\log_2(2A/W)$

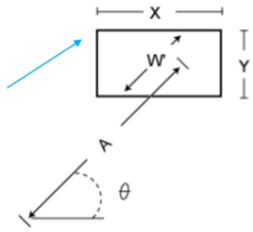


Figure 12: Fitts' law in practice.

- MT for "gross-movement tasks": Formula from above, where $a=$ time to start and stop in seconds for a device and b = inherent speed of the device (e.g. a mouse)
- Movement Time (MT) for "precision pointing tasks": $PPMT = a + b\log_2(A/W+1) + clog_2(d/W)$, where c = added constant, dependent on the user's context and d = distance between hand location and spot where the user first touched the screen
- GUIs with a Mouse Interface:
  - Overly elongated objects hold no advantage (W/H ratios of 3 and higher)
  - Objects should be elongated along the most common trajectory path (widgets normally approached from the side should use W, those approached from the bottom or top should use H).
  - Objects should not be offset from the screen edge
  - Fitts law does not address touch specific problems, e.g. fat finger
- Fitts' Law is not 1:1 applicable to touch interfaces:
  - Fat finger problem
  - Fingers are not moving directly from target to target but make a 3D movement
  - Fingers may directly reach target or need to be stretched or bended or a user may be even forced to change to 2-handed input
  - Screen size and hand size have a major impact on performance

Modelling Structure – Hick's Law:
Hick's Law states that the time T needed to make a decision (e.g. a selection) is proportional to the log number of alternatives given.

$$T = b \cdot H$$

where H is the information-theoretic entropy of a decision:

$$H = \begin{cases} \log_2(n+1), & n = \text{ alternatives of equal probability} \\ \sum p_i \log_2(\frac{1}{p_i+1}), & p_i = \text{ probability of alternative } i \end{cases}$$

Note: Hick's Law does not apply if it requires linear search (e.g. a randomly ordered list of commands in a menu). It applies if the user can search by subdivision.
The total decision time in practice amounts to

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

where n is the number of choices. The coefficients are empirically determined from experimental design. Raskin (2000) suggests that $a = 50ms$ and $b = 150ms/bit$ are sufficient place holders for "back-of-the-envelope" approximations.

Power Law of Practice:
Users get better every time they use a system. Neglecting disturbance and decay effects

$$Time = B \cdot N^{-\alpha}$$

where N = Trial Number, B, α = constants

5.2 Applied Design Guidelines & Principles: Mental Models – Mapping – Affordances

Mental Models:
People use a mental model to have a basic understanding of what is going on. The mental model is often unsharp, e.g. you know your car speeds up when pressing the gas pedal, but the concept behind is unsharp. Unsharp models might be sufficient, but in certain situations they might not be enough to explain the system. Interaction that is both predictive and explanatory can be understood with the concept of mental models. Mental models are

- Unscientific: they are based on guesswork
- Partial: They do not describe the whole system
- Unstable: They evolve an adapt to context and experience
- Inconsistent: Some parts of the model may be incompatible with other parts of the system
- Personal: They are specific to individuals

However, they can be used to foster understanding and building a correct model and thus increase usability. E.g. Error Avoidance Design Guidelines exploit natural mappings between intentions and possible actions, between actions and their effects on the system state and what is perceivable, as well as between the system state and the needs, intentions and expectations of the user.

Principles of Good Design (Don Norman):
An interface should include good mappings that show the relationship between stages. There should be a clear correlation between control element and action. A "good" mapping should be understandable, consistent, recognizable or quickly learnable and natural (consistent to the user's knowledge, world and domain knowledge). For example for cooking plates it would be helpful to arrange the buttons corresponding to the plates exactly like the plates themselves.

Constraints lead humans to build correct mental models. They guide the user to the next appropriate action or decision and also minimize the chance to make errors.

- physical constraints (dial vs. button)
- semantic constraints (assumption that create something meaningful)
- cultural constraints (borders provided by cultural conventions, e.g. traffic signs, colours, ..)
- logical constraints (restrictions due to reasoning)

It is good (for safety) to add some redundancy, as constraints can only work at their own level (e.g. having plugs color- and shape-coded in safety-critical situation).

Affordances (Gibson):
To build a mental concept of a system we need to interpret symbols and components. This can be seen as the functionality of the device and what we actually want to do
→ Semantic and Articulator Distance
Don Norman developed this concept further picking up the idea of affordances. It refers to the perceived and actual properties of an (everyday) object, which have to be matched:

– Make usable properties visible
– Use natural associations
– Give Feedback
Objects and environments imply their usage through Gestalt, e.g. door handle animates to press, cup to drink → Partially learned!

Normans Evaluation/Execution Action Cycle:

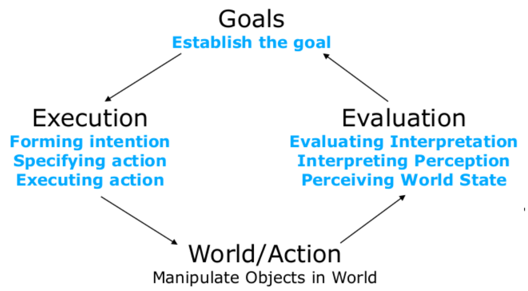


Figure 13: Evaluation/Execution Action Cycle

- Gulf of execution: Mismatch between the user's intentions and the allowable actions. Problems:
  - Forming intention → Insufficient knowledge of concept
  - Specifying action → Insufficient knowledge of usage
  - Executing action → Difficult access to function

  The difference between the intentions and the allowable actions is the Gulf of Execution. How directly can the actions be accomplished? Do the actions that can be taken in the system match the actions intended by the person?
  → Good design minimizes the Gulf of Execution

- Gulf of evaluation: Mismatch between the system's representation and the users' expectations. Problems:
  - Evaluating Interpretation → Comparing goal and state
  - Interpreting Perception → Interpretation of state
  - Perceiving World State → perception of State

  You may check how easy you are able to determine the function, determine what actions are possible, determine mapping from intention to physical movement, perform the action, determine whether the system in the desired state, determine mapping from system state to interpretation and determine what state the system is in. The Gulf of Evaluation reflects the amount of effort needed to interpret the state of the system and how well this can be compared to the intentions. Is the information about state of the system easily accessible? Is it represented to ease matching with intensions?
  → Good design minimizes the Gulf of Evaluation

Example: The user wants a document written on the system in paper (the goal).
Gulf of Execution: What actions are permitted by the system to achieve this goal?
Gulf of Evaluation: Is the process observable? Are intermediate steps visible?
Implications on design:

- Principles of good design (Norman):
  - Stage and action alternatives should be always visible
  - Good conceptual model with a consistent system image
  - Interface should include good mappings that show the relationship between stages

  - Continuous feedback to the user

- Critical points/failures:
  - Inadequate goal formed by the user
  - User does not find the correct interface / interaction object
  - User many not be able to specify / execute the desired action
  - Inappropriate / mismatching feedback

5.3 Task Analysis

A **goal** is a state of the application domain that a work system wishes to achieve. Goals are specified at particular levels of abstraction.
A **task** is a goal together with some ordered set of actions. Or more elaborated: A task is a structured set of activities required, used or believed to be necessary by an agent to achieve a goal using a particular technology. A task will often consist of subtasks where a subtask is a task at a more detailed level of abstraction. The structure of an activity may include selecting between alternative actions, performing some actions a number of times and sequencing of actions.
An action is a task which has no problem solving associated with it and which does not include any control structure. Actions and task will be different for different people.
Task analysis is the study of how work is achieved by tasks. It is about understanding people and how they carry out their work (part of human-centred design):

- identify a set of methods people use to carry out their work
- identify if they reach the goal
- calculate how long it takes to reach the goal

Two main categories of task analysis methods concerned with the logic of tasks (sequence of steps to achieve a goal) and the cognitive[5] aspects (understanding which cognitive processes the work system will have to undertake to achieve a goal).
Different Task Analysis methods can be sorted into four **dimensions**: notation, usability for communication, usability for modelling tasks, adaptability to new types of systems/aims/requirements.
Task analysis is an integral part of System Development and undertaken several times, during analysis (independent from technology, understanding the "essentials", current way of doing things), design (cognitive load minimization) and evaluation (dependent on technology, degree of achievement of work).

In the following, two models will be discussed: Hierarchical Task Analysis (HTA), concerned with the logic of the task, and Goal-based Task Analysis (GOMS), concerned with the cognitive analysis of the task. Both models rely on Cognitive System models (see previous chapters).

Hierarchical Task Analysis (HTA): Graphical representation of a task structure in structure chart notation: sequence of tasks, subtasks and actions as hierarchy, actions can be repeated (iteration), alternative actions possible (selection), annotations to indicate plans (optional). This is not trivial: acquisition of task and subtask descriptions, hierarchical modelling, iterative process.

Goal-based Task Analysis – GOMS: Goals, Operators, Methods, Selection Rules
An application of Human Information Processing and Task Analysis/Decomposition focusing on Cognitive Load Analysis. Introduced by Card, Moran & Newell (1983) they feature an explicit task structure:

- Hierarchy of goals and sub-goals

[5]Cognition: thinking, solving problems, learning, memory → HIP!
Representations of things that people are assumed to have in their heads → mental models

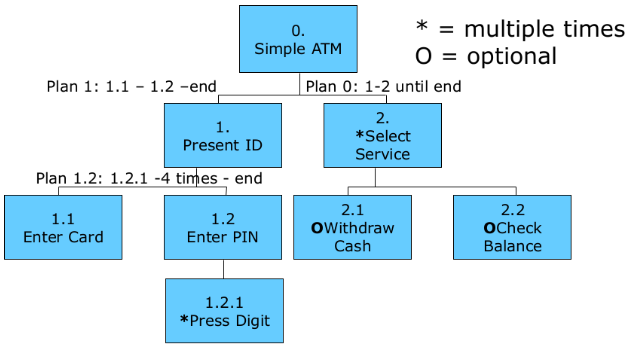


Figure 14: HTA – Example

- Procedure: define goals and refine them
- Outcome are several sheets of GOMS descriptions, starting with the topmost description
- Methods in the topmost description are subsequently refined in the "lower" GOMS descriptions

CNM-GOMS – Engineering model of user interaction:

- Goals – user's intentions (tasks): e.g. write a text, something the user perceives as a task
- Operators – actions to complete task: Basic, low level action to perform or basic mental state; cognitive, perceptual & motor (HIP); e.g. press X, read dialog; time of activity can be given
- Methods (Subgoals) – sequences of actions (operators): may be multiple methods for accomplishing same goal, e.g. shortcut key or menu selection
- Selections – rules for choosing appropriate method: method predicted based on context; selection rules are separately described

CPM-GOMS represent Cognitive, Perceptual, Motor operators. It uses Program Evaluation Review Technique (PERT) charts which map task durations using the critical path method (CPM). It is based directly on the Model Human Processor which assumes that perceptual, cognitive, and motor processors function in parallel.

Keystroke Level Model (KLM): The KLM is a practical design tool that can capture and

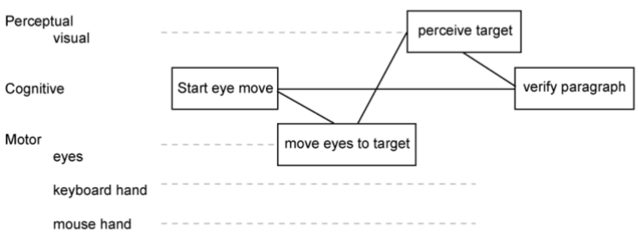


Figure 15: PERT chart – Example

calculate the physical actions a user will have to carry out to complete specific tasks. It can be used to determine the most efficient method and its suitability for specific contexts.

- Given:
  - A task (possibly involving several subtasks)
  - The command language of a system
  - The motor skill parameter of the user

- The response time parameters

- Predict: The time an expert user will take to execute the task using the system, provided that he or she uses the method without error.

The KLM is comprised of Operators, Encoding methods, and Heuristics for the placement of mental (M) operators. Operators include *Press a key or button* (K), *Point with mouse* (P), *Home hands to keyboard or peripheral device* (H), *Draw line segments* (D), *Mental preparation* (M), *System response* (R)

6 Interaction Design

6.1 Scenario-Based Design

Scenarios are a foundation for the design of interactive systems. Scenario "engineering" involves requirements work, prototyping, envisionment, evaluation, conceptual design and physical design. **Design rationale**: making the reasons for design decisions explicit. Methods for capturing and rep-

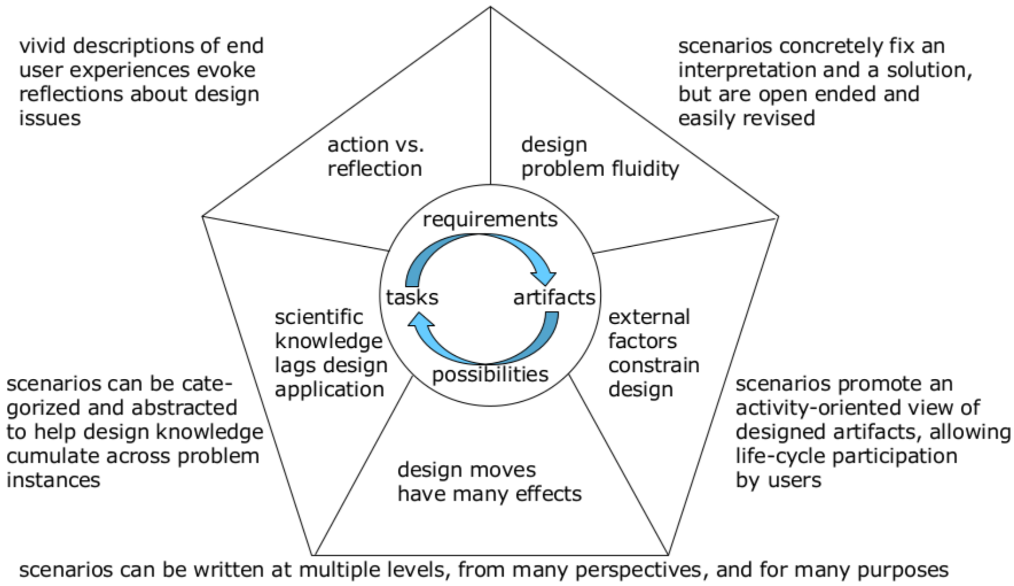


Figure 16: Challenges and Approaches in Scenario-based Design

resenting design rationale:
IBIS – issues, positions, arguments and relations QOC – questions, options and criteria (graphical links)

User Stories: reveal ideas, anecdotes, knowledge of people, real-world experiences, activities and context. They are used to identify the problem, the stakeholders and their constraints.

Conceptual Scenarios: more abstract than user stories, derived by combining several user stories. The are used for generation of ideas and understanding requirements but do not include technology and do not specify how functions are provided.

Concrete Scenarios: derived and generated by conceptual scenarios; each conceptual scenario may generate many concrete scenarios. They suggest a particular user interface design and help allocating functions between devices and people. They form a good start for prototyping and evaluation.

Use Cases: may result from many concrete scenarios and describe the interaction between devices and people. They abstract from concrete scenarios. Each use case covers many slight variations. Allocation of tasks and functions is needed for a use case. The sum of all use cases specifies the system

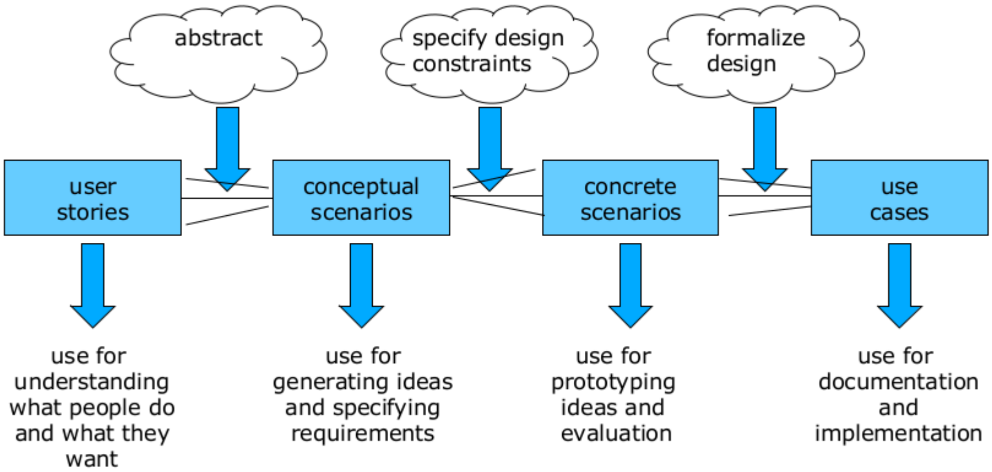


Figure 17: Scenarios in the Design Process

design. Use cases may consist of diagrams, pseudo code, etc.

Requirements and Problems: By analysis and abstractions, difficulties become visible. In gen-

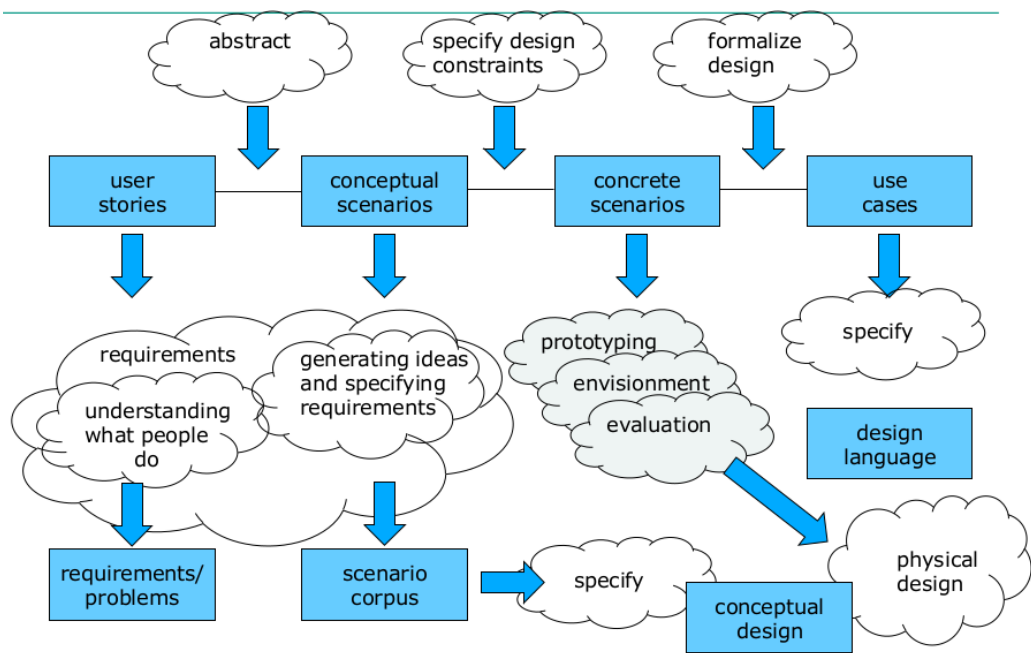


Figure 18: Overall Scenario-Based Design

eral these steps allow designers to collect experience in the formative process, in which requirements (desirable qualities of the system) can be derived. Prioritization is necessary since not all properties will be realizable.

Scenario Corpus: One wants to create a representative set of scenarios that cover a wide range of user stories. Some may be more specific, some may be more general but one takes a high-level, abstract view of the most activities in order to uncover the dimensions of the systems and the involved domains. If you don't do this, the outcome is most likely determined by technical features rather than by user requirements.

Conceptual Model/Design: An object or data model includes scenario and analysis of scenarios and describes the main objects, attributes and relationships among them. It ensures an accurate mental model of the interactive system. Here an analysis of cognitive processes, e.g. cognitive load, conceptualization is required. For example the desktop is a metaphor (mental model) for your desk.

Design Language: A standard set of patterns for interaction. General Design Language Concepts may be in place before concrete design application (e.g. Company Design Guidelines). Interaction patterns include physical attributes, conceptual actions and objects, key elements of design as well as principles and rules. Common language reduces the number of elements for the involved designers.

6.2 Requirements

Requirement analysis means understanding what people do, what people might want to do, problems with existing systems, and how people do what they do. The goal is to create "better" interactive systems for people. According to Robertson and Robertson (1999) a requirement is "... something the product must do or a quality that the product must have". Requirement work is the transformation of observation and information. It requires iterations on analysis, design, and evaluation and results in a requirements specification (formal and precise documentation of the requirements). User stories are a good start for user requirement analysis. The design process helps identifying requirements. Requirement Analysis work is very similar to software engineering! Two types:

- functional requirements: specify what the system must do; e.g. phone function button must be accessible while connected to the internet
- non-functional requirements: specify what qualities the system must have, concerned with how the functionality operates; e.g. an elderly person must be able to use the phone

For both requirement types, the technology is not yet specified, i.e. there is no specification of the implementation how-to, only usage how-to!
Not all requirements have equal impact on the interactive systems. Since resources are limited one needs prioritization and good balance.
MoSoCoW rules for prioritizing requirements:

- **m**ust have: fundamental requirements without which the system will be unworkable and useless, effectively the minimum usable set
- **s**hould have: would be essential if more time were available, but the system will be useful and usable without them
- **c**ould have: of lesser importance, therefore can more easily left out of the current development
- **w**ant to have: will not have this time round, can wait till a later development

One may use a requirement template for functional and non-functional requirements, for project (time, man power) and design (technology) constaints: One can define four requirements activities, which all have the same goal:

- requirements gathering, which suggests requirements are lying around waiting to be picked up / explored with little interaction between designer and user. Cons: unstructured
- requirements generation, which suggests a more creative designer activity, but tends to de-emphasize links to users' current practice. Pros: creative. Cons: not user centered
- requirements elicitation, suggests some user-designer interaction. Pros: user-designer collaboration. Cons: May stuck to user's creativity
- requirements engineering, used in software engineering, suggests a very formal approach. Pros: goal oriented. Cons: formalizing to early, kills creativity

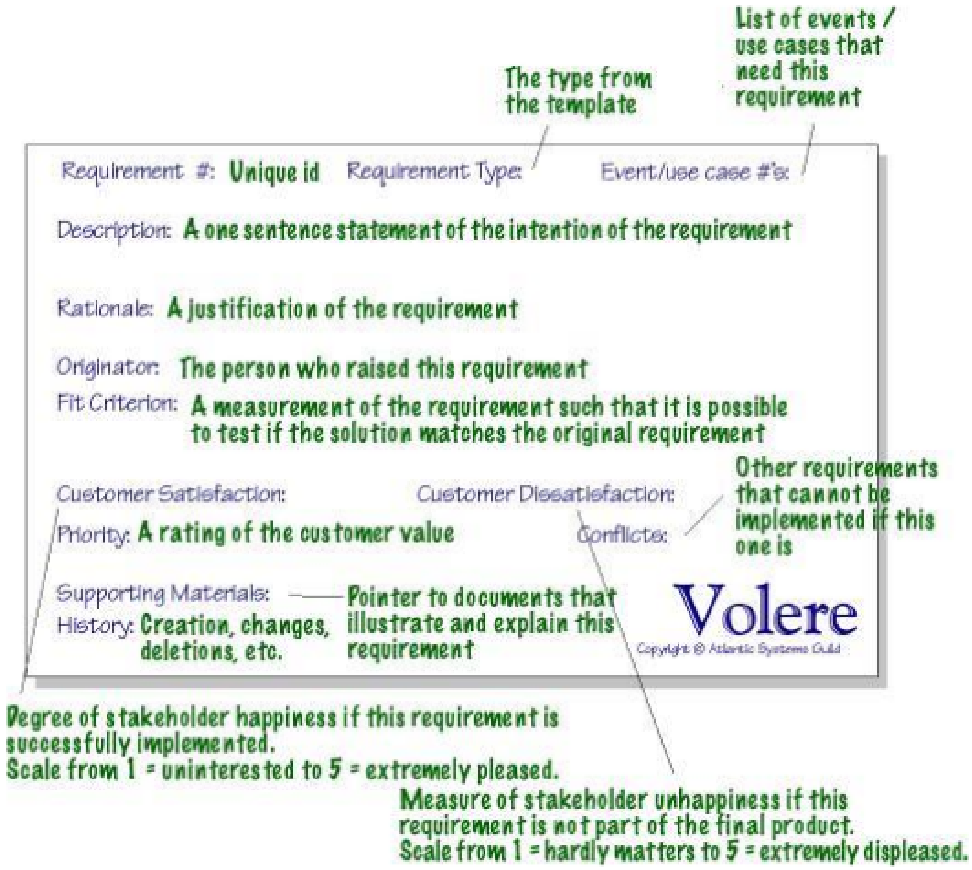


Figure 19: Using a requirement template.

There are numerous techniques for requirements elicitation/gathering: interviews, observations, documentation (e.g. video-taping), focus groups, workshops... (c.f. Chapter Observation). No matter which techniques are used, requirements elicitation is a user-centered activity!
Interviews, questionnaires and observation will identify artifacts used during the task. Most tasks require artifacts to complete. Artifacts provide additional insight in the task. Examples are documents, forms, printouts, tools...

6.3 Envisionment

Envisionment is concerned with making ideas visible by externalizing thoughts. The goal is to represent design work to the designers and stakeholders using stories and scenarios, presentations, sketches, formal models, software and hardware prototypes, cardboard models, mockups, ...This is fundamental to user-centered design and enables designers to see things from other people's perspective and to communicate ideas to people. It serves for generation, communication and evaluation of ideas but not all representations are suitable for all setups!
Exploring design concepts:

- how do you do it? → how do we affect the world (manipulate, sit, etc.); examples: handles (continuous) vs. buttons (discrete)
- how do you feel? → how does the user make sense of the world, sensory qualities that shape media; satisfaction, affect, enjoyment, engagement, involvement
  - hot media (photo): exact, authoritative, extend a single sense
  - cool media (cartoon): fuzzy and incomplete
- how do you know? → how do people learn and plan; how designers want users to think; example: paths vs. maps (step by step vs. understanding alternatives)

Sketches: sketching (quickly drawing ideas and thoughts) is an important ability of a designer and used for exploration and discussion. Many ideas have been sketched out on "the back of an envelope" or on a napkin. Shneiderman's principle for sketches: "overview first, zoom and filter, then details on demand" (similar for Web-Site design!)

Snapshots: show key moments in an interaction, often refined sketches; exploring the impact of a certain style or design snapshots by sketches, storyboards or using software (e.g. Macromedia Flash, HTML, ...)

Storyboards: Technique from film making. Cartoon-like structure presenting key moments from the interactive experience: This allows people to get an idea of the flow and is an economical way to present design (6-8 scenes on one paper). There are three main types of storyboards:

- traditional storyboarding: as in filmmaking; scenes with notes to overcome static nature of scenes on paper; annotations of interactive behavior; good for non-multimedia user interfaces
- scored storyboards: good for applications with motion graphics; annotation and notes about the type, colors, images, sounds
- text-only storyboards: used for applications with complex sequences; specifies what images appear, what text comes with them, and any additional media, ...

Mood boards gather visual stimuli that capture something about the feeling of the design. They are widely used in advertising and interior design and may feature photographs, colors, textures, shapes, headlines from newspapers or magazines, quotations from people, pieces of fabric, ...

Cultural probes (Gaver, 1999) allow involvement and creativity and are about getting to know your target group. You prepare a set of common items (introduction how to use and what to do with the material, e.g. booklet, maps, postcards, ...; suitable prototypes). You give the items to the participants and let them use it over a certain period of time. Not all items might be used as envisioned!

Navigation Maps: Navigation is a key feature for many (complex) systems. Find out, how the user moves through the site or applications focussing on how users will experience the site. This is used to uncover functional complexity of the system and to reduce recognized complexity by better grouping navigational elements. Each page in the site or location in the application is represented with a box or heading; each page should "flow" from there. Navigation maps can be redrawn many times during the project lifecycle – poor navigation is a main reason to turn off customers e.g. from a website.

Scenarios in Envisionment: Scenarios, in their most concrete form, can be used to envision or evaluate specific interactions.
Concrete scenario → prototyping → envisonment scenarios in the figure
The aim is to complement scenarios with some of the more visual envisioning techniques. Scenarios in envisionment should include real data and materials to allow direct involvement and you should think hard about underlying assumptions. You should also include good characterizations and develop a good number of personas. This allows "talking" about the characters. Also provide rich contextual background grounding in real life forces to think about practicality and acceptability. It is impossible to write scenarios for all variations and users, but the produced scenarios should cover interactions which are typical for a number of similar use situations, important design issues which are particularly important for the focus of the project, areas where requirements are unclear, and any aspects which are safety-critical. Scenarios should be structured (e.g. title, activities, rationale). When writing, thinking about or using scenarios, issues will arise about the design space. Only when people "see" a concrete representation, they are able to comment meaningfully. This is achieved by e.g. notes on the storyboards, or end notes to the scenario. Often, requirements will have to be changed. You should list positive and negative features for a concrete design (capturing rationale). Scenarios are effective to bring open issues to the surface.

An Outline Envisionment Process:

1) Review the requirements and conceptual scenarios

2a) Develop representations of the design ideas. These should include: concrete scenarios, storyboards, snapshots and sketches, ..

2b) Alternatives: If your product is a new one, experiment with different metaphors and design concepts through your representations

2c) Users: Your intended users should be involved in exploring design ideas throughout wherever possible

3) Decisions: Resources permitting, explore and document detailed design decisions using a method such as Questions, Options, Criteria (QOC)

4) Reconsider requirements in the light of the developing design, and carry out supplementary analysis where gaps in your background information have been uncovered.

6.4 Prototyping

Prototyping is an effective way for communicating ideas and design and complements/precedes techniques for envisionment. It is important to select the appropriate techniques. There are different levels of prototypes. Prototyping allows for participatory designs. Prototypes are a concrete but partial representation or implementation of a system. So far, screen sketches where discussed. These are, however, not interactive. Prototyping is a participatory design activity. You could either have deep/vertical prototypes or shallow/horizontal prototypes.

High Fidelity Prototypes are similar in look and feel to the anticipated final system. They are produced in the software that will be used for implementation or using a "simulation" package (e.g. using graphical toolkit: Flash, etc.) and useful for a detailed evaluation of the main design elements (content, visuals, interactivity, functionality and media). Hi-fi prototypes are also used in crucial stages of the project, e.g. client acceptance tests. They are used when the ideas, requirements and features of the system become stable. However, people might believe that the final system will exactly like the prototype! Also, a hi-fi prototype suggests that the system can be implemented/made to work! Hi-fi prototypes are costly!

Low Fidelity Prototypes are more focused on the underlying ideas, such as content, form and structure, key functionality requirements and navigational structure. They are designed to be produced quickly and thrown away after use and capture very early design thinking and should aid, not hinder, the process of generating and evaluating many possible design solutions. They often come in the form of paper prototypes (printout, paper, pen, post-its), series of screen shots. Paper prototypes are often an ideal means of prototyping interactive, screen-based systems! Practical issues with designing paper prototypes are:

- robustness: can be handled by many "users" without breaking – unlike a "soldered together piece of electronics"...
- scope: focus on broad issues and key elements; too detailed is hard to understand
- instructions: trade-off between adding enough detail to allow the user alone to explore and between adding too much detail (so the designer needs to walk the user through)
- flexibility: have adjustable parts (e.g. lay over other printouts e.g. on a menu, text box, ...) for visualizing interactive parts

Lo-fi prototypes usually do not "work" – the designer is needed to play the computer (Wizard of Oz). Videotaping the "interaction" if there is much feedback. User guidance to help people access the prototype, e.g. "setting the scene" for a new UI for a web shop: "You are interested in buying the shirt shown on this screen but want to know more about the material – show me what you would do now." Alternatively, if the system is too fragile to be exposed to users, the designers might act under the user's guidance. Much cheaper than Hi-fi, much cheaper than re-design!

Trade Offs: LoFi or HiFi Prototypes? What and how to prototype? → think in terms of PACT. Who is the prototype aimed at? What is the designer trying to achieve with the prototype? What stage of the project are things at and what is the context for use of the prototype? What technologies are appropriate?
According to Rosson and Carrol (2002) High-quality graphics and animation can be used to create convincing and exciting prototypes but may also lead to premature commitment to some design decision? Detailed special-purpose prototypes help to answer specific questions about a design, but building a meaningful prototype for each issue is expensive. Realistic prototypes increase the validity of user test data, but may postpone testing, or require construction of throw-away prototypes. Iterative refinement of an implementation enables continual testing and feedback, but may discourage consideration of radical transformations. Test users just get tired/used to the system.
Prototyping is necessary during the whole design process, but seldom done. Prototyping is much cheaper / leads to faster solution than re-implementation or project failure. "requirements animation": illustrating requirements by use of prototypes. Rapid prototyping for exploring the design space. Throw-away prototypes as part of the rapid prototyping activity to test/verify assumptions, but: "You will throw away your first few designs, so you might as well plan to throw them away in the first place".

Prototype Tools:

- for requirements animation might include: PowerPoint, paper (and paper overlays), Visual Basic
- for throw away/rapid prototyping: Adobe Flash, Web Tools
- for incremental/evolutionary prototyping: development environment, object-oriented languages
- for physical user interfaces: clay, wood, ..., 3D printers

7 Interface and Interaction

7.1 WIMP Interface

Contemporary GUIs are sometimes called "WIMP" (Windows, Icons, Menus, Pointers) interfaces. Multiple windows can pose management difficulties. There are two types of window managers: The operating system software and the user who must minimize, maximize, resize, access, and organize windows. Studies have shown that the advantages offered by windowing systems can be negated by excess window manipulation requirements. Window Managers are responsible for a common look-and-feel. This look-and-feel is either embedded into the operating system (Windows, MacOS) or kept separate (Unix). Most windowing systems use standardized windows that look similar and behave consistently.

How is a information presented to the user in term of Form, Structure, and Content/Description? The decision determines which information/action is easier to access and which is more difficult to access. There are various was to arrange windows: Tiled windows afford drag-and-drop methods. Overlapping windows use screen real estate efficiently, but they can become overwhelming. Cascading windows use screen real estate efficiently and can be used to create visual organization. Maximized windows are visually less complex, but they require easy navigation methods to get from window to window. Tabs increase the size of the dialogue by stacking layers on top of each other. Stacked tabs move around to accommodate the different levels and they destroy location consistency.

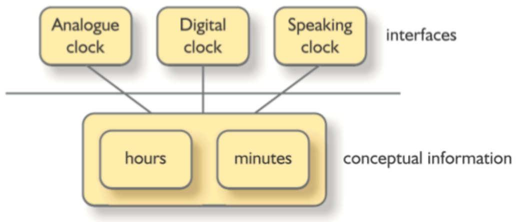


Figure 20: The same conceptual information can be presented via different physical interfaces.

7.2 Text and Reading

The Reading Process can be monitored in term of saccades (quick, jerky eye movements, about 8-10 letters) and fixations (intermittent pauses on areas of interest). Visual and cognitive processing occurs during fixation but not during saccades. If text is difficult to comprehend, if it includes long or unfamiliar words, fixations increase in duration. Reading is a 2-step process:
1) distinguish letter or word shapes (only experienced readers recognize word shapes)
2) Associate meaning

Some known human issues concerning text/reading:

- Readers do not have much trouble reading texts where the letters of the individual words are jumbled, as long as the first and last letter of each word remain in place.
- Single letters are simpler identifiable if they are in uppercase
- We read extended text passages more quickly in lowercase than uppercase
- Lowercase presentation is more common. Lowercase words have more distinctive shapes.
- Reading a novel is a continuous process. For many other texts we use scanning
- Reading from screens or paper is different: We often rely on our spatial memory when we search for information (Web-Problem!). On paper, the ability to annotate aids comprehension.

In interaction design, text is either used in two ways:

- Commentary – Text that informs: The most common form is help text.
  - Contextual help provides immediate assistance to users without requiring them to leave the context in which they are working, such as pop-up menus.
  - Procedural help provides the steps necessary for carrying out a task.
  - Reference help serves as an online reference book.
  - Conceptual help provides background information, feature overviews, or processes
- Instrumental – Text that does work: Controls (Buttons, Checkboxes, Radio Buttons, Icons, Hyperlinks) have labels which form one entity with their respective function. E.g. Hypertext links must give unambiguous indications of the target destination.

Other Design Issues are Legibility (Age, context determine size, contrast etc.), Readability (comprehension is affected by Line length, Line spacing, Formatting, Margin width, Scrolling and also by grammatical issues, such as semantics and syntax) and **Physical Factors**:

- Font size: 9-12pts are equally readable (Dix et al.); several factors affect font size according to Horton:
  - Reading Distance: Greater distances require larger text.
  - Screen Resolution: Smaller text requires greater resolution to keep the characters clear and legible.

  - Text/Background Contrast: Negative contrast is optimal (black type on a white background).
  - Visual Acuity of User: Age, Abilities etc.
  - Type of Reading: Text can be scanned, read word by word, or read character by character

- Line length: affects reading performance but not comprehension. Lines of greater length are read more quickly. People prefer medium line lengths.

- Margin width: Shorter lines with large margins increase reading performance (Maximal use of white space)

- Vertical line spacing: The spacing between lines of text (single spacing, double spacing, etc.) is called leading. Double spacing has been shown to improve reading speed. However, it might necessitate a smaller font size to increase the amount of visible information per screen.

- Alignment: For optimal reading of lengthy texts, right and center alignments should be avoided. Text should also be considered a graphical component of a page.

- Contrast: Contrast sensitivity decreases significantly with age. Because black and white have the highest contrast the addition of any color will reduce the contrast. Luminance contrast is more significant than color contrast.

- Scrolling versus paging: Which is better depends on application.

Fonts: Serif fonts guide the reader. Thus they are suitable for long text lines e.g. in books. Non-serif fonts are less complex to recognize (less elements), thus they are more suitable for cases where fast recognition is important, e.g. presentation slides. Cursive text is similar to hand written text and requires high-resolution screens. One further distinguished between variable-width and fixed-width fonts. In the latter, each character takes up the same horizontal width (→ more whitespace in between letters). There are good (most of them expensive) and cheap fonts (E.g. Arial is a cheap version of Helvetica). Font creation and readability of fonts is a science of its own.

7.3 Information

The Information Architecture structures the presented information, thus influencing the conceptualization of the user. Ontologies define the concept of information. How we group things together/ classify them is the concern of taxonomy. Information should be presented consistently (i.e. using one taxonomy), but we know that in practice several taxonomies can coexist.
Two general ways for structuring:

- Coarse-grained: The general information structure provides many objects to be selected. This reduces the number of steps until you reach the object in question. Example: One HTML document containing several sections

- Fine-grained: There are fine-grained objects, that are reached through a sophisticated, often multi-step process (e.g. through several menus). Example: One HTML document per section

How to build information structure:

- Volatility: Select an ontology that stays stable, e.g. so you do not have to change your menu logic

- Size: Decide if moving in an object is more suitable than moving between objects, e.g. scrolling vs. forward-next clicking (experts prefer scrolling)

- Conceptual / physical mapping: Information must be suitable to all input/output devices (→ 20 Menu buttons on mobile phone?)

- Topology: Most important also for Web, determines how easy it is to move through information space (e.g. a Web-Site). Distance is the number of clicks needed for navigation. Direction plays a role for navigation, e.g. an Audio Interface has often only forward navigation

In information design it is also important to use one consistent classification scheme, e.g. Alphabetical, Chronological, Geographical (e.g. travel agent: “Germany”), Topic (e.g. Cars, Repairs), Task-based (“buy a car”,”repair a car”), Audience (Car Buyers), Semantic (Functions, Objects etc. separated)

7.4 Interaction

There are many interaction styles:

- Command Line: You need to recall information, not recognize it. But it is also simpler to remember words than graphical interface elements. We need a very good concept of use. Steep learning curve, thus better for expert users. Repetition of tasks is simpler. Complexity can be handled easy through concatenation etc. In general faster for complex tasks. But higher error rate, higher cognitive load.

- Menu-Based Interface: Function can be recognized, position/structure recalled. Menus may be graphical or textual. Much better for seldom used functions, self explanatory. Range of possible options shown → constraints. Allows a user to build a concept of the system. Menu Types include Single, Sequential, Hierarchical and Network / Meshed menus.

- Form Fill-In: Special for gathering strings of information. Linear, not related to navigation. It is important that the users know how long the form will be, and where he is. Errors are problematic, annoy the user. Unambiguous labeling is important to preserve data integrity. Input formats may vary (e.g. how to enter a date?)

- Wizards/Question and Answer: Form with flow for beginners. Often represents “most used” design flow. Present only very restricted information at one time. Very restricted in control flow, thus inappropriate for any input where a great variety of control flows exist. BUT: Today omnipresent in many applications, because most users are beginners at some point (e.g. installation time) for programs. Or always, because of infrequent use.

- Direct Manipulation/Metaphors (Ben Sheiderman, 1982): Continuous representations of objects and actions of interest with meaningful visual metaphors. Simple physical actions to perform manipulations. Rapid, incremental, reversible actions whose effects on objects are visible immediately. Metaphors are most critical: They allow a user to understand a concept without or with reduced learning, often using real-world associations. However, they are not always consistent and simplifying. Also, different people have various concepts for the same objects. Touch interaction is also a form of direct manipulation.

- Web Navigation

- Three-Dimensional Environments

- Zoomable Interface

- Natural Language

Object-Action model: The user first selects an object and then selects the action to be performed on the selected object (e.g. MAC user interface).
Action-Object model: The user first selects an action to be performed and then selects the objects on which this action will be performed (e.g. typical Windows usage).