Summary

Updated: 9. Juli 2019

Jörg Cassens

Contextual Design of Interactive Systems

Universität Hildesheim
medieninformatik
IMAI – Institut für Mathematik und Angewandte Informatik

1 Introduction

Exam Dates

- Contextual Design of Interactive Systems
  - First exam: Tuesday, 16.07., 14:00-16:00, A 009 Spl.
  - Second exam: Tuesday, 24.09., 10:00-12:00, A 009 Spl.
  - Following exam: summer term 2020
- Duration of the exam: 90 minutes
- No books, scripts, etc. allowed
- Registration for the exam at least one week before
- Applications for disadvantage compensation (Nachteilsausgleich) at least one week before

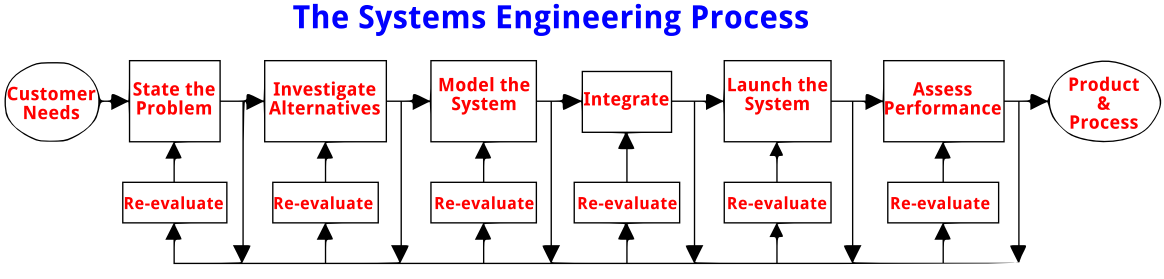
2 Overview

Design

Human-oriented design activities include:

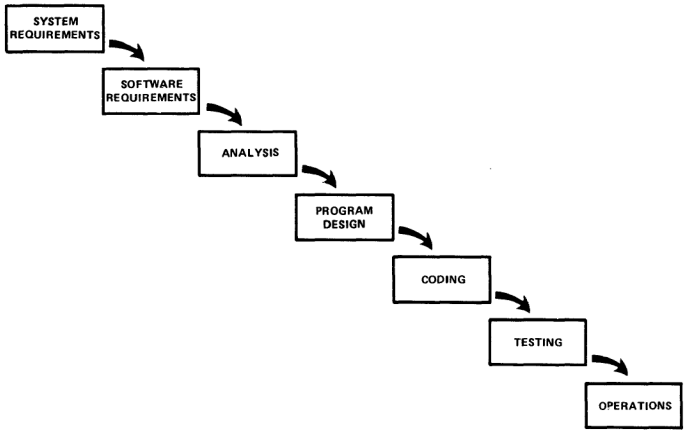
- Understanding the desires, needs, motivations, and contexts of people using products
- Understanding business, technical, and domain opportunities, requirements, and constraints
- Using this knowledge as a foundation for plans to create products whose form, content, and behavior are useful, usable, and desirable, as well as economically viable and technically feasible

Systems Engineering



According to Bahill and Gissing (1998)

Waterfall



"In my experience, . . . the simpler model has never worked on large software development efforts . . . " – Royce (1970)

Results

- What is interesting is more the aspects than the structure
  - Analysis – Understanding the world
  - Concept – Designing a solution
  - Implementation – Realization of said solution
  - Evaluation – Test of said solution
- We will need to cover these aspects

User-Centred Design

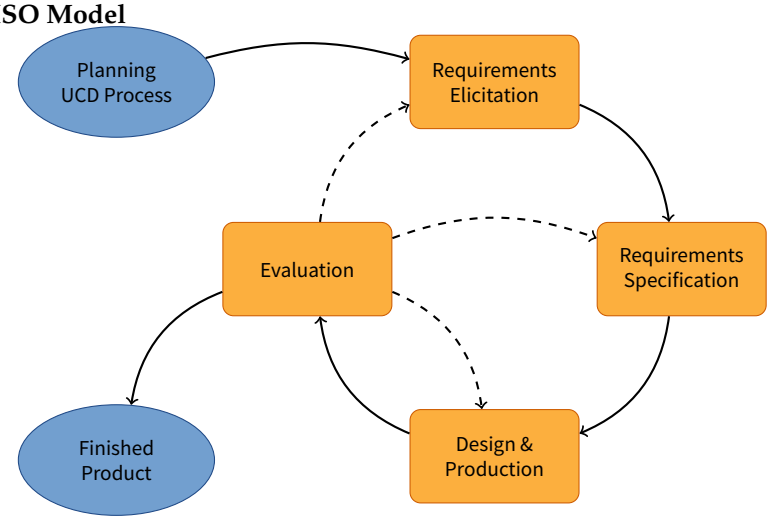
A User-Centred Design Process

- is a method, that includes relevant human factors
- caters for conscious and accountable decisions
- helps set the focus on important questions and requirements
- supports evaluation and testing of assumptions

Process as well as product are based on goals, activities, tasks, capabilities, needs and context of users. Therefore, user participation plays a role early in the process.

To be able to measure success, requirements need to be translated into quantifiable and measurable criteria

ISO Model



Processes and Steps

- The ISO-Standard gives only a broad definition of human-centred processes
- A couple of different approaches can be classified as being instantiations of such an abstract definition
- We will often find the following aspects
  - Analysis (Requirements Elicitation)
    - Description of context
    - Description of user
    - Activity analysis
    - Artefact analysis
  - Concept (Requirements Specification)
    - Activity design
    - Information design
    - Interaction design
  - Prototypical Implementation (Design & Production)
  - Evaluation (Evaluation)

Context- and Institutional Analysis

- To begin with, we need to describe the context of use of the product
  - Production, safety-critical, entertainment
  - Market analysis – expectations of users
- Description of the spatio-temporal setting for using the system
  - Is it to be used outside? At what time?
- Description of the institutional context of use
  - for business software, how is it used in the business

User Analysis

- Description of the target audience of the system
  - Physical and cognitive abilities
  - Cultural and social factors
- Different methods available
  - User classes – the potential users of the system are categorized into different classes, using a range of criteria
    - Experts, casual users
    - Roles they have using the system
  - Personas – concrete, but fictitious description of person dealing with the system
    - Primary, secondary, negative

Task Analysis

- Different techniques can be used, such as interviews or ethnological studies
- First we need to know, how the users work today
  - And that is not how they (or their boss) thinks how they do it
- Individual activities, tasks and operations can be identified and e.g. hierarchically organized
- Different models exist, for example the Hierarchical Task Analysis (HTA)
- Example: In a bank, the tasks REVIEW-ACCOUNTS can be divided into:
  - RETRIEVE-ACCOUNT-LIST
  - FIND-RECENT-ACTIVITY
  - REVIEW-ACTIVE-ACCOUNTS

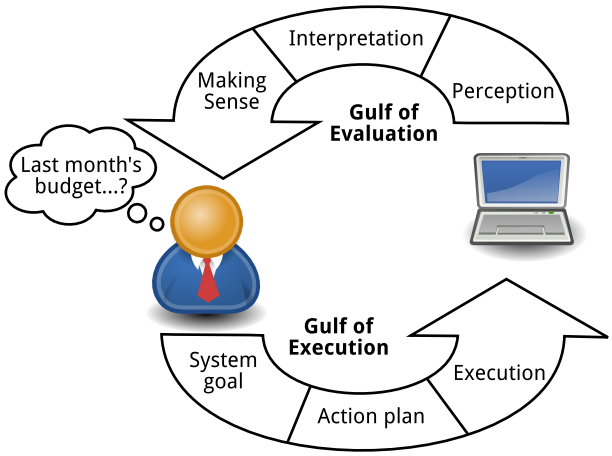
Artefact Analysis

- Description of artefacts used
- What are the "things" that are being used? And how?
- What is an artefact? A culturally defined object.
  - Office furniture
  - Pens, staplers, hole punchers
  - Protocols
  - Forms
  - Files
  - Computer (Hard- and Software)

Activity Design

- First step: what activities are to be supported by the new system?
- Some process models have this as an explicit step (e.g. Scenario-Based Development)
- Challenges and opportunities of current situation are transformed into system behaviour
  - Keep opportunities and address challenges
- Goal: specification of what can be done with the system
  - What information is made available?
  - What operations are possible?
  - What kind of results is getting delivered?
- Activity design defines the opportunities, but the experience is constructed through the interface
- The activity design has to be transformed into a design that supports bi-directional interaction of human and computer

Gulfs



Information Design

- The **objects** and **actions** possible in a system are **represented** and **arranged** in a way that facilitates **perception** and **understanding**
- Includes the design of
  - Application screens
  - Web pages
  - Menus

  - Dialogs
  - Icons
- Other modalities
  - Sound
    - Speech synthesis
  - Tactile
    - Force feedback game controls
  - Visual
    - 3D-displays (geowall)
- Addresses the Gulf of Evaluation

Interaction Design

- Goal: specify the mechanisms for accessing and manipulating task information
- **Information design** focuses on determining which task objects and actions to show and how to represent them
- **Interaction design** tries to make sure that people can **do the right things at the right time**
- Broad scope:
  - Selecting and opening a spreadsheet
  - Pressing and holding a mouse button while dragging it
  - Specifying a range of cells
- Addresses the Gulf of Execution

Prototypical Implementation

- Instead of "Do it right the first time" we will develop iterative prototypes
- A prototype is a concrete but partial implementation of a system design
- Constructed and evaluated to guide redesign and refinement
- Created to explore many questions during system design
  - System reliability
  - Bandwidth consumption
  - Hardware compatibility
- User interface prototype
  - Built to explore usability issues
- User interface prototypes can be built early on in the design process
  - Paper prototype
- Late prototypes will probably be very close to the actual system (depending on process model)

Evaluation

- Necessary to assess progress and satisfaction of requirements
  - Formative and summative
- Helps understanding the activity and resulting requirements
- This leads to enhanced specifications and implementations
- Evaluation can start early in the process
  - Not only "finished" Software can be evaluated
  - Evaluation of User-Interface-Specifications
- Evaluations help to detect deficits in the design early on and to correct mistakes

3 Research Methods

Definition

- DIN EN ISO 9241 Teil 110
- Anforderungen an die Gebrauchstauglichkeit – Leitsätze
  - "Das Ausmaß, in dem ein Produkt durch bestimmte Benutzer in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele
    - effektiv,
    - effizient und
    - mit Zufriedenheit

    zu erreichen."

Definition (contd.)

- **Nutzungskontext**: "Die Benutzer, Arbeitsaufgaben, Arbeitsmittel (Hardware, Software und Materialien) sowie die physische und soziale Umgebung, in der das Produkt genutzt wird."
- **Effektivität**: "Die Genauigkeit und Vollständigkeit, mit der Benutzer ein bestimmtes Ziel erreichen."
- **Effizienz**: "Der im Verhältnis zu Genauigkeit und Vollständigkeit eingesetzte Aufwand, mit dem Benutzer ein bestimmtes Ziel erreichen."
- **Zufriedenstellung**: "Freiheit von Beeinträchtigungen und positive Einstellungen gegenüber der Nutzung des Produkts."

Formativ und Summativ

- **Formative Evaluation**
  - Während des Prozesses, "wo stehen wir?"
  - Hinweise auf konkrete Mängel bzw. Produktmerkmale, die Mängel verursachen
  - Ausgangspunkt für konstruktive Verbesserungsvorschläge
  - während der Entwicklung von Produkten
  - als Grundlage für Neuentwürfe
- **Summative Evaluation**
  - Am Ende, "sind die Anforderungen erfüllt?"
  - Abschließende Beurteilung
  - Produktvergleich
  - Überprüfung der Einhaltung von Kriterien
  - Zumutbarkeit von Softwaresystemen
  - Zertifizierung

Analytisch und Empirisch

"If you want to evaluate a tool, say an axe, you might study the design of the bit, the weight distribution, the steel alloy used, the grade of hickory in the handle, etc., or you might study the kind and speed of the cuts it makes in the hands of a good axeman." (Scriven, 1966)

- Die Evaluation der Charakteristik der Axt ist eine **analytische** Evaluation
  - Experten bewerten das System
  - Checklisten, Cognitive Walkthroughs
- Die Evaluation der Benutzung durch den Handwerker ist **empirisch**
  - Nutzer benutzen das System
  - Benutzertests, Benutzerbefragungen
- Diese Unterscheidung ist orthogonal zur Aufteilung formativ/summativ

Qualitativ und Quantitativ

- **Qualitativ**
  - Konkretes Feedback, aber nicht in Form von Meßgrößen
    - Kommentare, Eindrücke, subjektive Bewertungen in Benutzerbefragungen
    - Detaillierte Ergebnisse einiger weniger Benutzertests
- **Quantitativ**
  - Erhebung von Meßgrößen in kontrollierten Umgebungen
    - Messung von Fehlerraten, Dauer der Interaktion, Anzahl der Interaktionsschritte
    - In erster Linie zur Messung der Effizienz des Systems

Experten-Evaluation

- 1-3 Usability-Experten sehen sich das System an und suchen gezielt nach Usability-Problemen
  - benutzen Wissen und Erfahrung und/oder heuristische Richtlinien
  - Dabei immer die vorgesehene Aufgabe und die vermuteten Benutzereigenschaften in den Vordergrund stellen
- Alternative: Die Entwickler betrachten das System fortlaufend anhand von Checklisten und Styleguides
  - Aber: Regelwerke sind nie vollständig
  - Entwickler zumeist nicht geschult
  - Externe Kontrolle besser (Voreingenommenheit)

8 goldene Regeln von Shneiderman

- Konsistenz
  - Verwende Styleguides und weitere schriftliche Konventionen
- Berücksichtige unterschiedliche Erfahrungen
  - Eine Benutzungsschnittstelle sollte jeder NutzerIn möglichst eine passende Interaktionsform anbieten
    - Anfänger: über Menüs
    - Abkürzungen für erfahrene Benutzer
- Rückmeldungen auf Aktionen des Benutzers
  - Aktion bei der Software angekommen
    - Insbesondere, wenn die Aktion spät ein Ergebnis liefert
    - Akustisch, visuell, taktil
- Abgeschlossene Operationen
  - Schritte einer Operation im Zusammenhang darstellen
- Fehler verhindern
  - Darstellung eindeutig
  - Auswahlalternativen anbieten
- Einfache Rücksetzmöglichkeiten (undo)
  - Selbstsicherheit des Benutzers steigt stark an
  - Exploratives Lernen
- Benutzer bestimmt den Kontrollfluss
  - Gefühl, die Anwendung steuern, kontrollieren zu können
    - "I am in control"
- Geringe Belastung des Kurzzeitgedächtnisses
  - Aufbau von Menüs besser breit statt tief

Shneiderman and Plaisant (2005)

Usability-Heuristiken von Nielsen

- Einfache und natürliche Dialoge
  - dem Lösungsweg der Aufgabe angepaßt
  - an den Erwartungen des Benutzers orientiert
- Ausdrucksweisen des Anwenders
  - Fachsprache des Anwendungsgebiets
- Minimale mentale Belastung des Benutzers
  - "Don't make me think" (Steve Krug)
- Konsistenz
  - Darstellung
  - Dialoge folgen immer der gleichen Logik
- Rückmeldungen
  - über Annahme der Aktion
  - insbesondere, wenn die Aktion länger dauert
- Klare Auswege
  - bei falscher Navigation
  - bei falscher Aktion
- Abkürzungen
  - Standardwerte, History-Funktionen
  - für geübte Benutzer: shortcuts
- Gute Fehlermeldungen
  - konstruktive Rückmeldungen
- Fehlervermeidung
  - besser als Fehlerbehandlung
- Hilfe und Dokumentation
  - bei Anwendung und Einarbeitung unterstützen.
  - vollständig und übersichtlich
  - korrekt und auf dem aktuellen Stand

Nielsen (1994)

Usability Test

Quelle: M. Herzceg

- Variante: retrospektives Lautes Denken
  - Video-Aufzeichnung der Interaktion (VAI)
  - Video-Konfrontation mit VAI mit der Aufforderung laut zu denken:
    - Was haben Sie in dem Moment gedacht?
    - Warum haben Sie Interaktion x ausgeführt?

Strukturiertes Interview

- direkte Befragung durch einen Experten
  - strukturiert anhand von Leitfaden
- Vorteile
  - direkter Eingriff bei Mißverständnissen und Unklarheiten möglich
  - qualitativ hochwertige Daten
- Nachteile
  - hoher sozialer Druck
  - hohe soziale Kompetenz der Befragten und Befragenden notwendig
  - Große Fehlermöglichkeit durch (unfreiwillig) suggestive Fragen
  - sehr aufwendig

Feldstudien

- grundsätzliche Methoden
  1. naturalistische Beobachtungen
  2. teilnehmende Beobachtungen
     - Befragung im Kontext (Contextual Inquiry)
  3. Artefact Walkthroughs
- Die Methoden unterscheiden sich in
  - Echtzeitinformation oder Retrospektion
  - Umfang der Interaktion mit den Benutzern
  - Fokus auf den Kontext und die Tätigkeit *vs.* Fokus auf die Interaktion mit Technologie

Effekte bei Feldstudien I

- **Hawthorne-Effekt**:
  - Seit den 1920er-Jahren bekanntes Phänomen
  - Die Teilnehmer an einer Studie ändern ihr Verhalten wenn Sie unter Beobachtung stehen
    - Alle Arbeiter auf der Baustelle tragen ihren Schutzhelm solange der Beobachter dabei ist
- **Novelty-Effekt**:
  - Die stärkste psychologische Antwort auf eine (potentiell gefährliche) Situation gibt es bei den ersten Begegnungen mit der Situation
    - Wird eine neue Technologie eingeführt so kann sich die Leistung allein aufgrund der Tatsache steigern daß es eine neue Technologie ist

Effekte bei Feldstudien II

- **Observer-Expectancy-Effekt**:
  - Die unbewußte Beeinflussung des Tests durch die Erwartungen des Beobachters
    - Der "Kluge Hans", ein Pferd welches anscheinend zählen konnte, aber allein auf die Reaktionen des menschlichen Publikums reagierte
- **Subject-Expectancy-Effekt**:
  - Der umgekehrte Fall
  - Das Testsubjekt hat eine Erwartung, die sein Verhalten beeinflußt
    - Placebo-Effekt

Kognitive Dissonanz

- **Loss Aversion Bias:**
  - Die Tendenz, eher einen Verlust zu vermeiden, als Gewinne zu erzielen (Teil des Status Quo Bias)
- **Negativity Bias:**
  - Negativen Informationen und Erfahrungen wird stärkeres Gewicht beigemessen als positiven
- **Selective Perception Bias:**
  - Erwartungen beeinflussen die Wahrnehmung
- **Confirmation Bias:**
  - Die Tendenz, Informationen so auszuwählen, daß die eigenen Erwartungen bestätigt werden
- **Reactance Bias:**
  - Das Gegenteil von dem tun zu wollen was jemand anders von einem erwartet um das Gefühl zu haben, frei entschieden zu haben

Quantitativ-Analytisch: GOMS

GOMS (Goals, Operators, Methods, Selection Rules)

- Reduzieren der Interaktion auf elementare Aktionen (Operators)
- Elemente:
  - **Goals:** Was will der Benutzer erreichen
  - **Operators:** Aktionen die ausgeführt werden um das Ziel zu erreichen
  - **Methods:** Sequenz aus Operationen zur Erreichung des Ziels
  - **Selection Rules:** Auswahl aus Optionen
- Dekomposition in Teilziele
- Motivation:
  - Frühe Entscheidungen
  - Keine teuren Prototypen
  - Klare Metriken

4 GDD

Goals & Activities

- Goals are not the same as tasks or activities
- Goal expectation of an end condition
- Activities and tasks are intermediate steps
- Donald Norman describes his extended hierarchy based on Activity Theory
  - **Activity:** Coordinated, integrated set of tasks. For example, staying at a hotel.
  - **Tasks:** An individual task is for example to check into the hotel.
  - **Actions:** Tasks consist of collections of actions. An action is performed consciously, the hotel check-in, for example, consists of actions like presenting the reservation, confirmation of room types, and handover of keys.
  - **Operations:** Actions consist themselves of collections of non-conscious operations. Writing your name on a sheet of paper or taking the keys are operations.
- Activity-Centred Design focuses on the activity

Assignment 2.1: D. Norman

- Required reading for week 1
  - Norman, Donald A. "Human-centered design considered harmful." interactions 12, no. 4 (2005): 14-19.
- The text will be discussed in the tutorial 16.04.2019
- Course readings can be downloaded in the learnweb
- Every text has a wiki-page in the learnweb
  - Use it to describe the text
  - Use it to link the text to the course
- Results of the discussion may also be written up

Goals

- Activities useful in breaking down the "what" of user behaviors, but it really does not address the first question any designer should ask: *Why* is a user performing an activity, task, action, or operation in the first place?
- Goals motivate people to perform activities; understanding goals allows you to understand your users' expectations and aspirations, which in turn can help you decide which activities are truly relevant to your design
- Asking, "What are the user's goals?" lets you understand the meaning of activities to your users
- Activities and tasks are much more transient, because they are based almost entirely on whatever technology is at hand
- In out example, the goal of with the hotel stay is for it to be comfortable, safe and affordable

Good Design

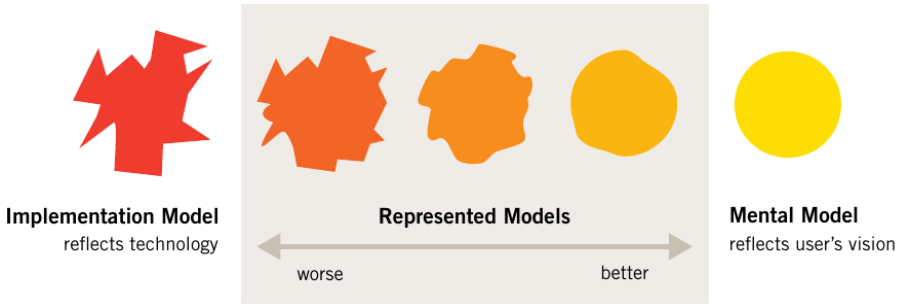
What is good design?
Good design makes helps users to effectively accomplish tasks related to their goals, fulfil them efficiently and be satisfied in doing so

- Software that enables users to perform their tasks without addressing their goals rarely helps them be truly effective
- If the task is to enter 5,000 names and addresses into a database, a smoothly functioning data-entry application will not satisfy the user nearly as much as an automated system that extracts the names from the invoicing system
- The user's job is to focus on her tasks, the designer's job is to look beyond that

Models in Human-Computer Interaction

- Model of a system describes how it works
  - its constituent parts and how they work together to do what the system does
- We are here concerned with three models:
  - The **system model** (sometimes called implementation model) is how the system actually works.
  - The **interface model** (or represented model) is the model that the system presents to the user.
  - The **user model** (or conceptual model) is how the user thinks the system works.
- There are more models
  - The model the developers have about how they think the user model is like
  - The model the system has about the user (inscribed, in terms of Actor Network Theory)

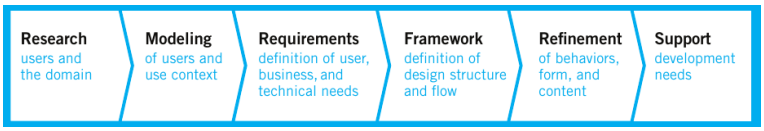
Choice of Represented Model



A comparison of the implementation model, mental model, and represented model.
(Cooper et al., 2014)

Overview

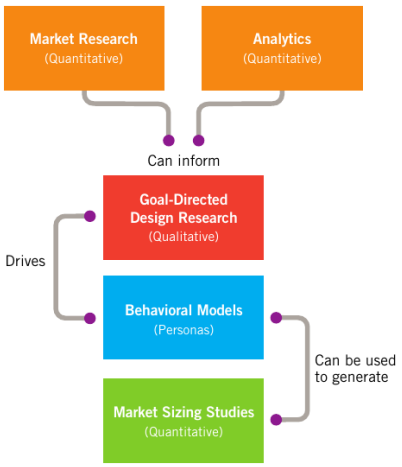
- Goal-Directed Design combines techniques of ethnography, stakeholder interviews, market research, detailed user models, scenario-based design, and a core set of interaction principles and patterns
- It provides solutions that meet users' needs and goals while also addressing business/organizational and technical imperatives
- Can be roughly divided into six phases:
  - Research, Modeling, Requirements Definition, Framework Definition, Refinement, and Support



(Cooper et al., 2014)

5 Research

Qualitative and Quantitative Research



(Cooper et al., 2014)

Qualitative Methods used

- Kickoff meeting
- Literature review
- Product/prototype and competitive audits
- Stakeholder interviews
- Subject matter expert (SME) interviews
- User and customer interviews
- User observation/ethnographic field studies

(Cooper et al., 2014).

User Interviews: What to achieve

- The context of how the product (or analogous system, if no current product exists) fits into their lives or work flow
- Domain knowledge from a user perspective: What do users need to know to do their jobs?
- Current tasks and activities: both those the current product is required to accomplish and those it doesn't support
- Goals and motivations for using their product
- Mental model: how users think about their jobs and activities, as well as what expectations users have about the product
- Problems and frustrations with current products (or an analogous system)

Combining Observations and Interviews

- Combination of observation and one-on-one interviews is the most effective and efficient tool for gathering qualitative data about users and their goals
- The technique of ethnographic interviews is a combination of immersive observation and directed interview techniques
- Holtzblatt and Beyer (2016) pioneered an ethnographic interviewing technique they call contextual inquiry
- Contextual inquiry methods closely parallel the methods described here

Contextual Inquiry

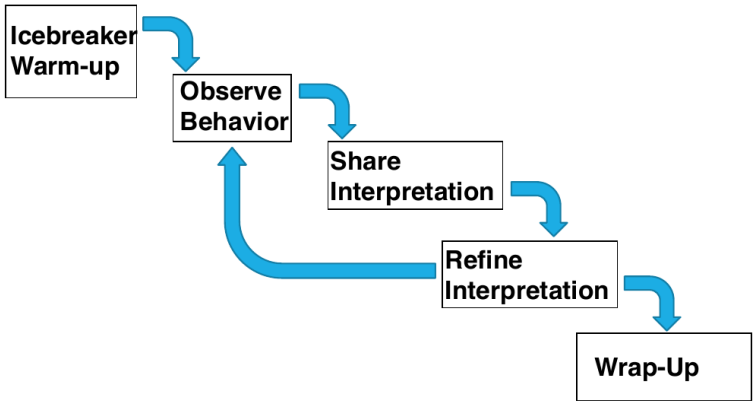
- Master-apprentice model of learning: observing and asking questions as if the user is the master craftsman, and the interviewer the apprentice
- Four basic principles of engaging in ethnographic interviews:
  - Context
    - Important to interact with and observe the user in her normal work environment, or whatever physical context is appropriate for the product
  - Partnership
    - The interview and observation should take the tone of a collaborative exploration with the user, alternating between observation of work and discussion of its structure and details
  - Interpretation

    - Reading between the lines of facts gathered about users' behaviors, their environment, and what they say
  - Focus
    - Rather than coming to interviews with a set questionnaire or letting the interview wander aimlessly, direct the interview so as to capture data relevant to design issues

Master/Apprentice

- The "master/apprentice" relationship is at the heart of contextual inquiry
- In a master/apprentice relationship:
  - The master is doing stuff
  - The master explains what they're doing
  - The apprentice asks clarification questions
  - The master answers
- Limits of the metaphor
  - The goal is not to learn to do the task
  - Instead, the goal is to learn how the participant does the task in order to learn how to support it
  - And for the researcher to enlist the participant's active assistance in understanding the task

Stages of Contextual Inquiry



Dell (2018)

Basic methods

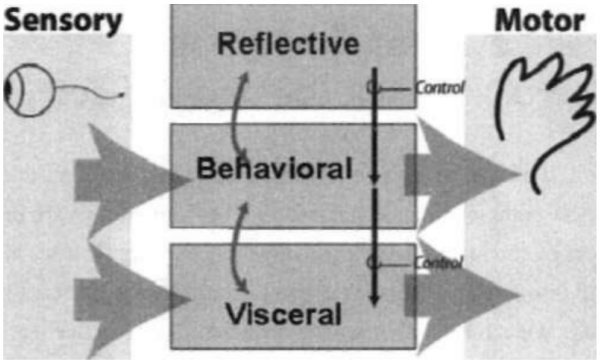
- Interview where the interaction happens.
- Avoid a fixed set of questions.
- Assume the role of an apprentice, not an expert.
- Use open-ended and closed-ended questions to direct the discussion.
- Focus on goals first and tasks second.
- Avoid making the user a designer.
- Avoid discussing technology.
- Encourage storytelling.
- Ask for a show-and-tell.
- Avoid leading questions.

Assignment 5.2: Contextual Inquiry

- Form pairs of 2 who do not know each other well
- Use a system you are acquainted with and research your use of it
  - You actually have to do the task
  - You would actually do the task on campus on the device you're using
- The other person conducts a contextual inquiry on their task:
  - Focus. Decide what to pay attention to.
  - Partnership. You act as an interested learner, they act like a knowledgable expert.
  - Perform the inquiry. Ask probing questions. Have them teach you. Don't generate questions in advance; think of them as you observe. Focus questions on what you see happening in context.
- Document your findings
- Present your findings in the course
  - Due date: 07.05.2019

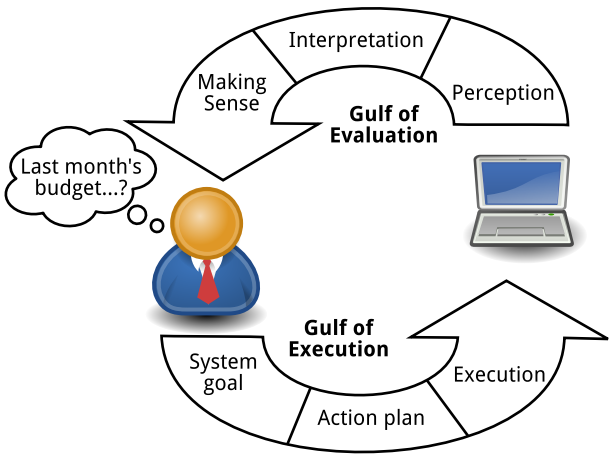
6 Modelling Users

Three levels of processing



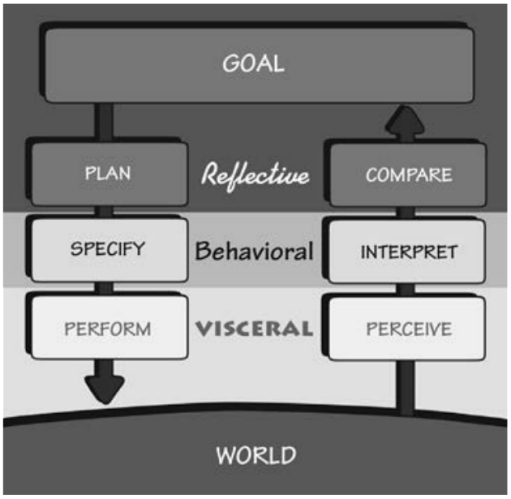
(Norman, 2004)

Gulfs



Adapted from Norman (2013)

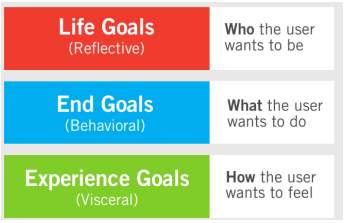
Foundations of the Action Cycle



Visceral response is at the lowest level: the control of simple muscles and sensing the state of the world and body.
The **behavioural** level is about expectations, so it is sensitive to the expectations of the action sequence and then the interpretations of the feedback.
The **reflective** level is a part of the goal- and plan-setting activity as well as affected by the comparison of expectations with what has actually happened.

Three types of user Goals

- Norman presents a three-level theory of cognitive processing and discusses its potential importance to design
- Cooper et al. (2014) claim that three types of user goals correspond to Norman's visceral, behavioural, and reflective processing levels
  - Experience goals
  - End goals
  - Life goals



Assignment 3.1: Pruitt & Grudin, Chapman & Milham

- Required reading for week 2
  - Pruitt, John, and Jonathan Grudin. "Personas: practice and theory." In Proceedings of the 2003 conference on Designing for user experiences, ACM, 2003.
  - Chapman, Christopher N., and Russell P. Milham. "The personas' new clothes: methodological and practical arguments against a popular method." In Proceedings of the human factors and ergonomics society annual meeting, vol. 50, no. 5, pp. 634-636. Sage Publications: Los Angeles, CA, 2006.
- The texts will be discussed in the tutorial 30.04.2019
- Course readings can be downloaded in the learnweb
- Every text has a wiki-page in the learnweb
  - Use it to describe the text
  - Use it to link the text to the course
- Results of the discussion may also be written up

What are personas?

> A persona is a fictional character that is meant to represent a group of users that share common goals, attitudes and behaviours when interacting with a particular product or service. (Dell, 2018)

See also our required reading about personas.

All Users

(Cooper et al., 2014)

Specific Users

(Cooper et al., 2014)

Why Personas are effective

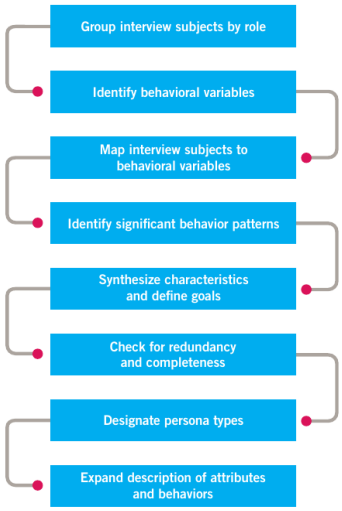
- **Empathy** – We are engaged by fictional characters all the time in movies and books.
- **Focus** – Personas provide a precise way of thinking about…
  - how users behave
  - their motivations
  - how they think
  - what they wish to accomplish (goals)
  - why they want to do what they do
- **Communication** – Provides a way of conveying a broad range of quantitative and qualitative data
- **Assumptions** – about users made explicit

Strengths of personas as a design tool

- **Determine** what a product should do and how it should behave
- **Communicate** with stakeholders, developers, and other designers
- Build **consensus and commitment** to the design through a common language
- **Measure** the design's effectiveness

- **Contribute** to other product-related efforts such as marketing and sales plans
- It addresses common problem in user-centred design
  - Elastic user, self-referential design, edge cases

1. Group interview subjects by role.
2. Identify behavioural variables.
3. Map interview subjects to behavioural variables.
4. Identify significant behaviour patterns.
5. Synthesize characteristics and define goals.
6. Check for completeness and redundancy.
7. Designate persona types.
8. Expand the description of attributes and behaviours.



Step 7: Designate persona types

- By now, your personas should feel very much like a set of real people you know
- What we must do then is prioritize our personas to determine which should be the primary design target
- Goal is to find a single persona from the set whose needs and goals can be completely and happily satisfied by a single interface
  - Primary
  - Secondary
  - Supplemental
  - Customer
  - Served
  - Negative

Step 7.1: Primary persona

- Main target of design
- A product can have only one primary persona per "interface", but it is possible for some products (especially enterprise products) to have multiple distinct interfaces, each targeted at a distinct primary persona
- In some cases, two separate interfaces might be two separate applications that act on the same data; in other cases, the two interfaces might simply be two different sets of functionality served to two different users
- A primary persona will not be satisfied by a design targeted at any other persona in the set
- However, if the primary persona is the target, all other personas will not, at least, be dissatisfied
- Focus the design for each interface on a single primary persona
- Choosing the primary persona is a process of elimination: You must test each persona by comparing its goals against goals of the others

Step 7.2: Secondary persona

- Is mostly satisfied with the primary persona's interface
- However, it has specific additional needs that can be accommodated without upsetting the product's ability to serve the primary persona
- We do not always have a secondary persona
- More than three or four secondary personas can be a sign that the proposed product's scope may be too large and unfocused
- As you work through solutions, your approach should be to first design for the primary, and then to adjust the design to accommodate the secondary

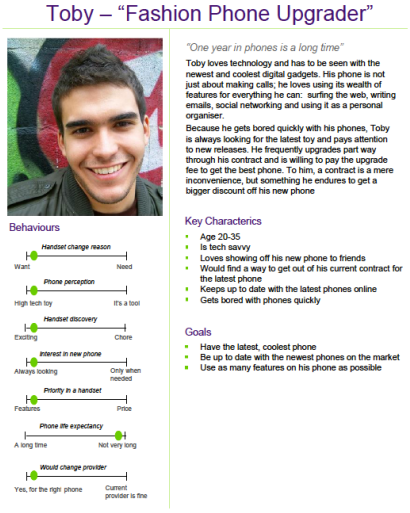
Step 7.3: Supplemental persona

- User personas that are not primary or secondary are supplemental personas
- Their needs are completely represented by a combination of primary and secondary personas and are completely satisfied by the solution we devise for one of our primaries
- Any number of supplemental personas can be associated with an interface
- Often political personas — the ones added to the cast to address stakeholder assumptions — become supplemental personas

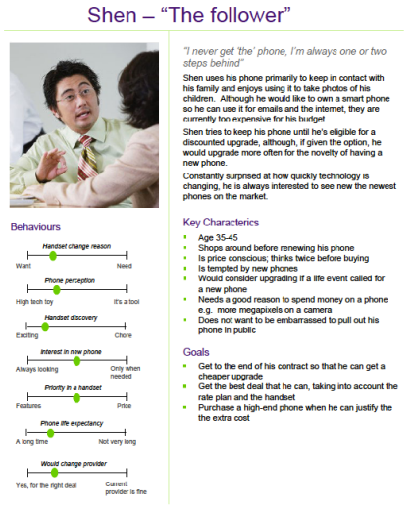
Step 7.4: Customer persona

- Customer personas address the needs of customers, not end users
- Typically, customer personas are treated like secondary personas
- However, in some enterprise environments, some customer personas may be primary personas for their own administrative interface

Examples



Dell (2018)



Dell (2018)

Assignment 6.2: Persona Construction

- Form groups of 4 out of the pairs from Assignment 5.2
- Imagine you are re-designing the system used in the previous assignment
- Create at least two different personas
  - For example a primary and a negative

- Focus on
  - Characteristics
  - Experiences
  - Motivations
  - Goals
- Feel free to use a templates
- Present your findings in the course
  - Due date: 21.05.2019

7 Scenarios

Assignment 6.1: Go & Carroll

- Required reading for week 3
  - Go, Kentaro, and John M. Carroll. "The blind men and the elephant: Views of scenario-based system design." interactions 11, no. 6 (2004): 44-53.
- The texts will be discussed in the tutorial 14.05.2019
- Course readings can be downloaded in the learnweb
- Every text has a wiki-page in the learnweb
  - Use it to describe the text
  - Use it to link the text to the course
- Results of the discussion may also be written up

User Stories: Components

Components
As a [role],
I can [functionality]
so that [rationale]

- Role
  - Persona;
  - important and specific class of user
- Functionality
  - Activity, action, task
- Rationale
  - Reason, motivation
  - The rationale demonstrates the value to the eventual user / owner and determines its priority and the effort to be expended.

User Stories: Example

Example 1. As a music fan,
I can establish an account,
so that I can legally stream and download music.

- Role is important to the eventual owner
- Rationale demonstrates value to the eventual owner
- Functionality enables the value to be achieved

Writing Stories

- User-centred stories
- In agile approaches, often relegated to customer
  - Written in language of business to allow prioritization
  - Customer is primary product visionary
- Writing good stories needs practice & insight
  - Customer are often not the right stakeholder to write the stories
- Good stories can INVEST
  - Independent
  - Negotiable
  - Valuable to users or customers
  - Estimable
  - Small
  - Testable

Example: Scenario
Suppose an accountant wishes to open a folder displayed on his screen in order to open and read a memo. However, the folder is covered by a budget spreadsheet that he also needs to see while reading the memo. The spreadsheet is so large that it nearly fills the display. The accountant pauses for several seconds, then resizes the spreadsheet, moves it partially out of the display, opens the folder, opens the memo, resizes and repositions the memo, and continues working. (Rosson and Carroll, 2002)

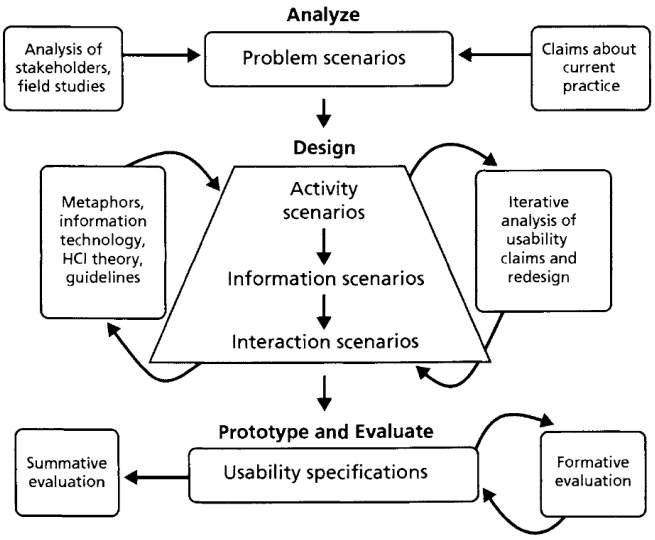
- This is about as routine a work scenario as one could imagine
- Yet even this story conveys important information about window management and application switching:
  - People need to coordinate information sources, to compare, copy, and integrate data from multiple applications; computer displays inevitably get cluttered; and people must find and rearrange windows in these displays.

Scenarios vs. User Stories

- Both methods of describing the user's interaction with a system
- They serve very different functions
  - Scenarios are an iterative means of defining a product's behaviour from the standpoint of specific users (personas)
    - Does not only include the system's functionality
    - Priority of functions and how those functions are expressed in terms of what the user sees and how she interacts with the system
  - User stories are exhaustive descriptions of the system's (functional) requirements, focusing on low-level user action and accompanying system response
    - Use cases permit a complete cataloguing of user tasks for different classes of users
    - say little or nothing about how these tasks are presented to the user or how they should be prioritized in the interface

Characteristic Elements

- Rosson and Carroll (2002) suggest that interaction scenarios should have the following components
  - Setting
  - Actors
  - Task goals
  - Plans
  - Evaluation
  - Actions
  - Events



(Rosson and Carroll, 2002)

Claims Analysis

- In SBD, the analysis and refinement of scenarios is stimulated by claims, statements that list important features of a situation and their impacts on users' experiences
- In requirements analysis, these features are elements in the current situation; as the scenario content shifts from analysis to design, the claims call out features of the proposed solution
- Claims are related to the general notion of tradeoffs in design, because they always analyse both positive and negative usability impacts
- The analysis of claims organizes and documents the "what- if" discussions the design team carries out when considering and prioritizing alternatives

Claims

- Claims elaborate a set of scenarios, explaining how and why a particular feature is having a range of impacts on the personas
- A claims analysis documents why one or more scenarios were written, by isolating the most important features of the narratives
- The claims extend the scenarios, pointing to possible effects a feature might have in other scenarios (i.e., without writing out a new scenario)
- Claims analysis promotes a balanced view of a situation
- Each feature is analyzed to consider both positive impacts (prefaced with plus signs) and negative impacts (prefaced with minus signs)
- The claims motivate design reasoning–designers will try to increase positive impacts while decreasing negative impacts

Persona-based scenarios

- Persona-based scenarios are concise narrative descriptions of one or more personas using a product or service to achieve specific goals
- They allow us to start our designs from a story describing an ideal experience from the persona's perspective, focusing on people and how they think and behave, rather than on technology or business goals
- Scenarios can capture the non-verbal dialogue between the user and a product, environment, or system over time, as well as the structure and behaviour of interactive functions
- Goals serve as a *filter for tasks* and as a *guide for structuring* the display of information and controls during the iterative process of constructing the scenarios

Three Types

- Goal-Directed Design employs three types of persona-based scenarios through the design process, each with a successively more interface-specific focus
  - Context scenario
  - Key path scenario
  - Validation scenario
- *Context scenarios* are used to explore, at a high level, how the product can best serve the needs of the personas
- They are created before any design sketching is performed
  - Written from the persona's perspective, focusing on human activities, perceptions, and desires
- Once functional and data elements are defined, a context scenario is revised into a *key path scenario*
  - More specifically describing user interactions
- *Validation scenarios* test the design solution in a variety of situations
  - Tend to be less detailed and typically take the form of a number of what-if questions about the proposed solutions

Requirements Definition Process

(Cooper et al., 2014)

Context Scenarios

- Context scenarios address questions such as the following:
  - In what setting(s) will the product be used?
  - Will it be used for extended amounts of time?
  - Is the persona frequently interrupted?
  - Do several people use a single workstation or device?
  - With what other products will it be used?
  - What primary activities does the persona need to perform to meet her goals?
  - What is the expected end result of using the product?

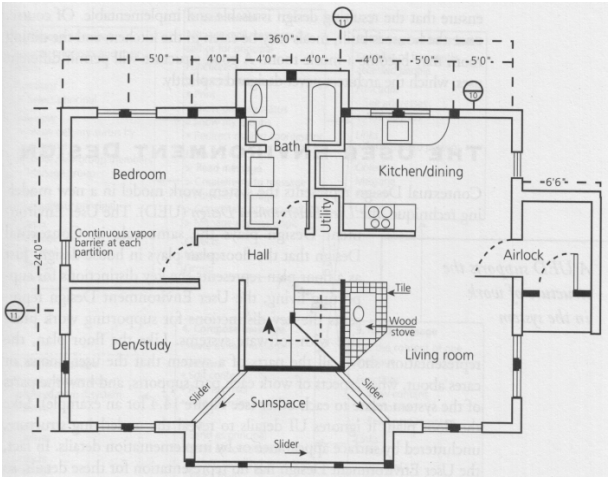
  - How much complexity is permissible, based on persona skill and use frequency?
- Context scenarios should *not* represent current product behaviours
- They should represent the new world of Goal-Directed products
- Don't worry yet about exactly how things will get accomplished

Assignment 7.2: Scenario

- Form groups of 4 out of the pairs from Assignment 6.2
- Create a context scenario for each of your personas
- Make sure you cover the typical aspects
  - Setting
  - Actors (Personas)
  - Task goals
  - Plans
  - Evaluation
  - Actions
  - Events
- Present your findings in the course

8 Design

Floor Plan

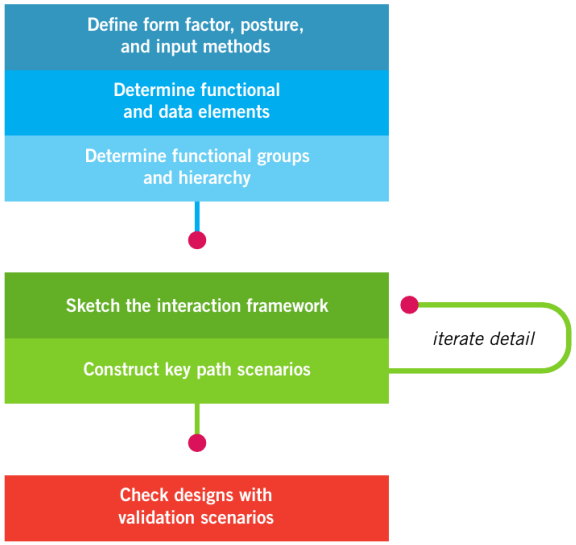


Beyer and Holtzblatt (1997)

Form and Behaviour

- Form and behaviour must be designed in concert; the Design Framework is made up of
  - Interaction framework
    - use scenarios and requirements to create rough sketches of screens and behaviours that make up the interaction
  - Visual design framework
    - use visual language studies to develop a visual design framework, commonly expressed as a detailed rendering of a single screen archetype
  - Industrial design framework
    - Industrial designers execute form language studies to work toward a rough physical model and industrial design framework
  - Service frameworks
    - Service designers build models of the information exchange for each touch point in a service framework

Interaction Framework



1. Define form factor, posture, and input methods.
2. Define functional and data elements.
3. Determine functional groups and hierarchy.
4. Sketch the interaction framework.
5. Construct key path scenarios.
6. Check designs with validation scenarios.

Step 4: Sketch the interaction framework

- We are ready to sketch the interface
- This visualization of the interface should be simple at first
- Start for example by subdividing each view into rough rectangular areas corresponding to panes, control components (such as toolbars), and other top-level containers
- Label the rectangles, and illustrate and describe how one grouping or element affects others
- Draw arrows from one set of rectangles to others to represent flows or state changes

Step 5: Construct key path scenarios

- A key path scenario describes how the persona interacts with the product, using the vocabulary of the interaction framework
- depict the primary pathways through the interface that the persona takes with the greatest frequency
  - In an e-mail application, key path activities include viewing and composing mail, not configuring a new mail server
- Typically evolve from the context scenarios, but here we specifically describe the persona's interaction with the various functional and data elements
- As we add more and more detail to the interaction framework, we iterate the key path scenarios

Context vs. Key Path Scenarios
Unlike the goal-oriented context scenarios, key path scenarios are more task-oriented, focusing on task details broadly described and hinted at in the context scenarios

Step 5.1: Storyboarding

- By using a sequence of low-fidelity sketches accompanied by the narrative of the key path scenario, you can richly portray how a proposed design solution helps personas accomplish their goals
- Storyboarding is borrowed from film-making and cartooning, where a similar process is used to plan and evaluate ideas without having to deal with the cost and labour of shooting actual film
- Each interaction between the user and the product can be portrayed on one or more frames or slides
- Advancing through them provides a reality check of the interactions' coherence and flow

Defining the visual design framework

- This process follows a trajectory similar to the interaction framework, in that the solution is first considered at a high level and then narrows to an increasingly granular focus
- The visual design framework typically follows this process:
  1. Develop experience attributes.
  2. Develop visual language studies.
  3. Apply the chosen visual style to the screen archetype.

Industrial Design Framework

- The industrial design framework typically follows this process:
  1. Collaborate with interaction designers about form factor and input methods.
  2. Develop rough prototypes.
  3. Develop form language studies.

Defining the service design framework

- Because service design often affects organizations' business models, the service design framework may be conducted before other areas of design
- The service design framework typically follows this process:
  1. Describe customer journeys.
  2. Create a service blueprint.
  3. Create experience prototypes.

Usability test

- Given sufficient time, more formal usability testing has some advantages
- Because the findings are often quantitative, usability research is especially useful in comparing specific design variants
- Feedback gathered from usability testing is most useful when you need to validate or refine particular interaction mechanisms or the form and expression of specific design elements
- **Naming** — Do section/button labels make sense? Do certain words resonate better than others?
- **Organization** — Is information grouped into meaningful categories? Are items located in the places customers might look for them?
- First-time use and **discoverability** — Are common items easy for new users to find? Are instructions clear? Are instructions necessary?
- **Effectiveness** — Can customers efficiently complete specific tasks? Are they making missteps? Where? How often?

Don't Listen to Users – Experience Sampling

- Traditional usability tests often focus on first-time use
- A systematic way of having participants provide samples of their ongoing behaviour
- Participants record the behaviour of interest (e.g., activity, location, mood, thoughts)
- Dependent on either
  - **Signal:** signalled with a beeper, cell phone call, or similar at random times within a fixed time period
  - **Interval:** pre-set intervals for reporting events
  - **Event:** whenever a key event occurs

- Advantages and disadvantages
  - reports are personal and subjective
  - data are similar to those obtained by diaries, but less dependent on memory
  - less intrusive than direct observation
  - data tend to not have the richness of ethnography

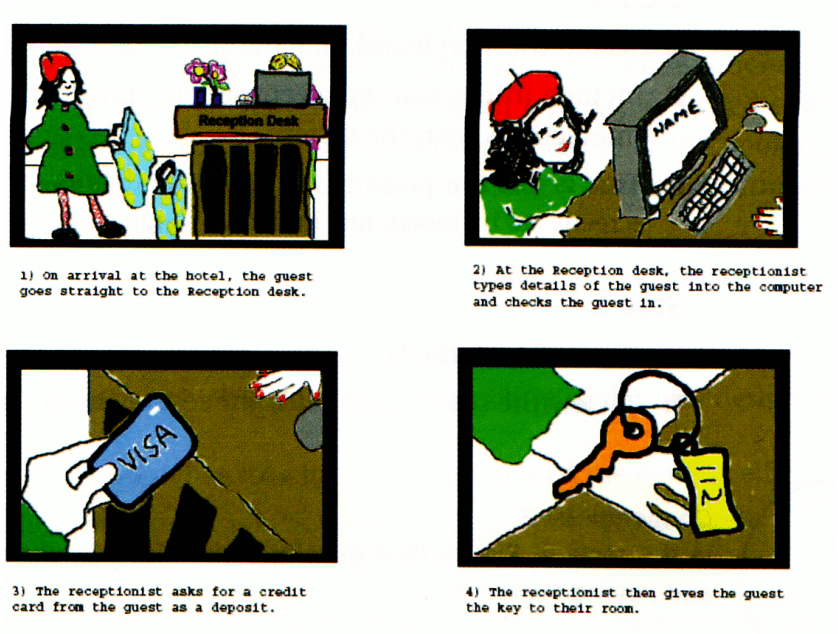
Storyboarding

- Storyboards are illustrations that represent a story
- Images are arranged together to visualize the story
- Invented by Walt Disney in the 1920s
- Visual storytelling with rough sketches/cartoons/comics
- A great way to bring a story to life!

Storyboards can be used to:

- Describe a user's current situation (pre or post design).
- Describe a user's hypothetical experience using a new technology/design.

Storytelling



Storyboards tell a story (Benyon et al., 2005)

Process

1. *First!* Figure out the story you want to tell!
   - Iterative process with lots of drafts
   - Do a lot of brainstorming
2. Define a specific scenario
   - Set the stage: Who? What? Where? Why? When?
3. Break it into segments
   a) Start with simple text (captions) and arrows
   b) Add emotions
   c) THEN sketch visuals on paper
4. Generate more polished versions only when you have refined/ finalized the story!

Elements of a Storyboard

- Five key elements:
  1. Level of detail
  2. Inclusion of text
  3. Inclusion of people (personas!) and emotions
  4. Number of frames
  5. Portrayal of time

Assignment 8.2: Storyboards

- Consider the following design concepts
  - A navigation system that helps long-distance cyclists find restaurants and other services
  - Alternatively, use the situation you have been analysed so far
- Create storyboard solution(s):
  - Write a short narrative scenario.
  - Create a storyboard for this design concept (stick figures are fine).
    - Start with text and arrows
    - Add emotions
    - Then draw pictures to create the storyboard
- Keep in mind:
  - Use of personas and emotions, Passage of time
  - Usage of text captions, Level of detail
  - Number of frames (4-6 per storyboard)
- Present your findings in the course

9 Prototyping

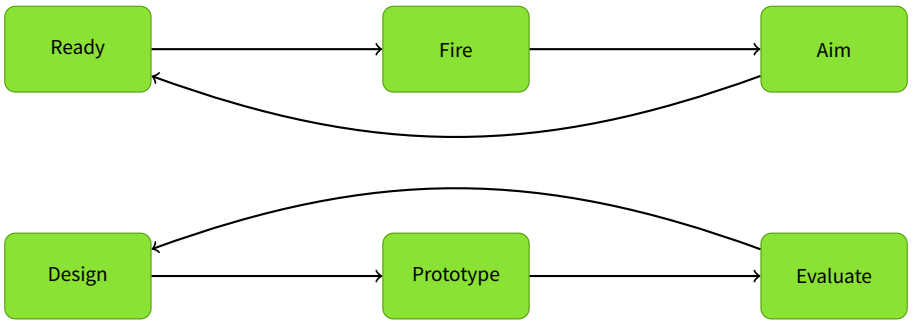
Assignment 8.1: Sauer & Sonderegger

- Required reading for week 4
  - Sauer, Juergen, and Andreas Sonderegger. "The influence of prototype fidelity and aesthetics of design in usability tests: Effects on user behaviour, subjective evaluation and emotion." Applied ergonomics 40, no. 4 (2009): 670-677.
- The texts will be discussed in the tutorial 28.05.2019
- Course readings can be downloaded in the learnweb
- Every text has a wiki-page in the learnweb
  - Use it to describe the text
  - Use it to link the text to the course
- Results of the discussion may also be written up

Prototyping

- Problem:
  - We can't evaluate a design until it's built
- But...
  - After building, changes to the design are difficult
- Solution
  - Prototype!
- Simulate the design in low-cost manner
- Make it fast. Make it cheap.
- Facilitate iterative design and evaluation
  - Your first idea is rarely your best!
- Promote feedback
- Allow lots of flexibility for radically different designs
  - Don't kill crazy ideas!

Prototyping



(Hix and Hartson, 1993)

Prototyp-Types

- Storyboard
  - Sketches or screen shots illustrating key points in a usage narrative
- Paper Prototype
  - Fabricated devices with simulated controls or display elements
- Wizard of Oz
  - Workstation connected to invisible human assistant who simulates input, output, and processing functionality not yet available
- Video Prototype
  - Video recording of persons enacting one or more envisioned tasks
- Computer Animation
  - Screen transitions that illustrate a series of input and output events
- Scenario Machine
  - Interactive system implementing a specific scenario's event stream
- Rapid prototype
  - Interactive system created with special purpose prototyping tools
- Partial Working System
  - Executable version of a system with a subset of intended functionality

Throw-Away vs. Evolution

- **Throw-away** prototyping means that a series of prototypes are constructed and then discarded
  - Typically higher and higher fidelity versions are developed
- **Evolutionary** prototype means that the same prototype evolves into higher and higher fidelity and eventually becomes the system
- Throw-away prototyping can be difficult for design teams or management to accept because it seems as if the time spent developing the prototype is a waste
- Evolutionary prototyping is possible with software development, but there is a danger
- An early high-fidelity prototype might have been built to illustrate a design idea, but not designed to be extended
- Nevertheless evolutionary prototyping can be useful for eliciting more and more subtle design aspects

Low-Fidelity Prototypen

- Designskizze
  - Frühe Skizzen auf Papier oder Tafel
  - Vermitteln Eindruck von Designideen
  - Diskussion der Skizzen kann gegenseitiges Verständnis von Gestalter und Anwender fördern
  - Fehler können frühzeitig erkannt werden
  - Beispiele wären ein Storyboard oder Comic
- Papierprototypen
  - Bildschirminhalte werden durch Papierstreifen und Post-Its simuliert
  - Austauschen und Manipulation dieser lassen das Verhalten der Benutzungsschnittstelle erkennen
- Diese Low-Fidelity-Prototypen können mit minimalen Kosten erstellt werden
- Der offensichtlich unfertige Charakter lädt zur Kritik und Manipulation ein
- Man kann grundlegende Beurteilungen konzeptioneller Modelle erhalten

Medium-Fidelity Prototypen

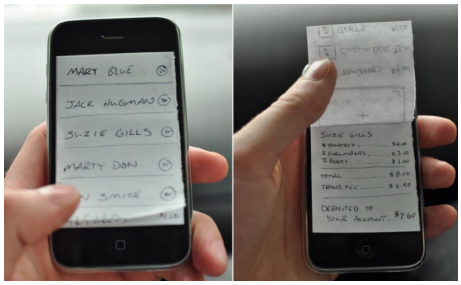
- Wizard of Oz
  - Sollen Nutzer einen realistischen Eindruck eines Systems bekommen, das noch nicht implementiert ist, bieten sich Simulationen an
  - Dem Benutzer wird die Benutzungsschnittstelle präsentiert
  - Ein menschlicher Operator (Wizard) beobachtet den Benutzer und seine Umgebung und steuert die Funktionen des Systems
  - Besonders nützlich, wenn vor der Implementierung Interaktionsdaten benötigt werden
    - Ambiente System
    - Sprachsteuerung
- Mock-Up
  - Typischerweise mit Rapid Prototyping Werkzeugen erstellt
    - Webseiten mit Screenshots
  - Simuliert immer noch die funktionalen Teile des Systems
  - Erlaubt typische Interaktionssequenzen

High-Fidelity Prototypes

- Simulation
  - In particular object-oriented simulations
  - Agents with particular goals, believes, intentions interact via simulated sensors with the real software
  - Data and/or modelling necessary
- Proof-of-concepts
  - Later versions of throw-away prototype
  - Rudimentary and/or incomplete
- Implemented Application
  - Later versions of evolutionary prototypes
  - Large-scale implementations
  - Long running systems
  - Suitable for a field-study

"Mixed"-Fidelity

- Easy access to cameras makes it easy to blur the lines between lo-fi and hi-fi prototypes
- Photos of hand-drawn prototypes can easily be captured and displayed on real screens
- Sequences of photos can also be animated to simulate interaction

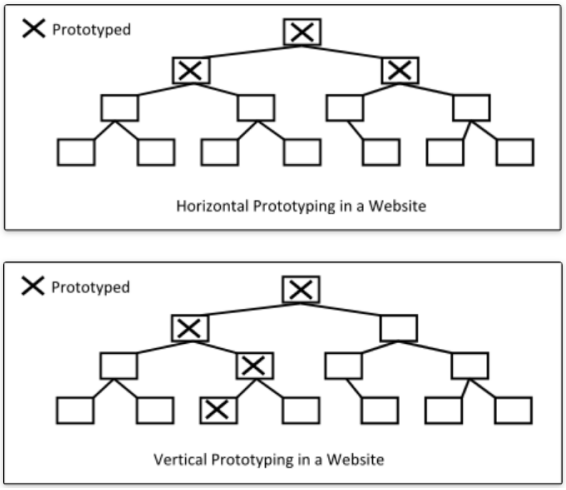


Dell (2018)

Breadth or Depth?

- Partially working systems
  - Horizontal prototype
    - all the intended functionality, but only at the top level
    - Example: initiate a shopping spree, but cannot actually order
    - Good for testing high level goals and action plans
  - Vertical prototype
    - only one or two tasks are implemented in detail
    - Example: shop til you drop, but cannot see shipping information
    - Good when only few tasks are seen as particularly complex or important

Horizontal vs. Vertical Prototype



McCracken et al. (2004)

Papierprototypen: Methode

- mit einer Papierattrappe der Benutzerschnittstelle versuchen die Benutzer "echte" Arbeitsaufgaben mit "echten" Daten zu erfüllen
- Testen und Verändern der Benutzerschnittstelle
  - Benutzer als Co-Designer
- Struktur und Funktion testen, nicht Layout und Icons
- Interview nach den Richtlinien des Contextual Inquiry Interviews
- (mindestens) 2 Personen, einer "spielt" das System, der andere macht Notizen
  - Das System erklärt sich nicht

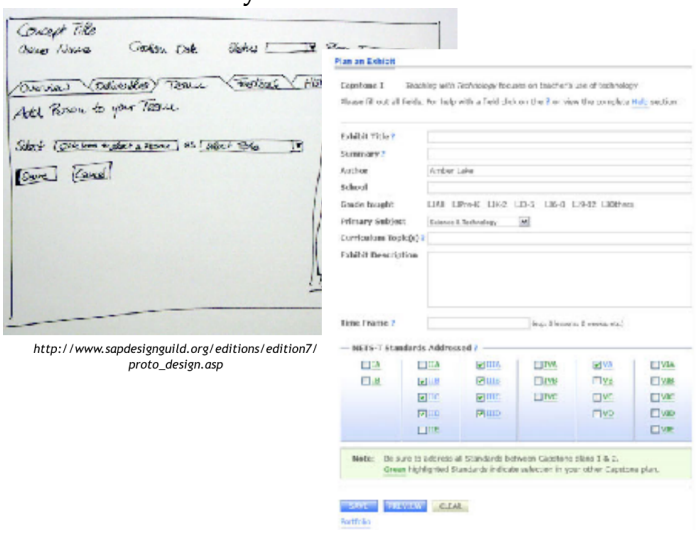
Material

- Paper:
  - heavy stock unlined
  - light stock unlined
  - sticky notes, different colors
  - Acetate sheets
- Markers:
  - color pens
  - color board markers
  - color pencils
- Adhesives:
  - scotch tape
  - glue sticks
  - soft adhesive (like what is found on sticky notes)
- Scissors:
  - Scissors
  - eXacto knives
- Everything else you can think of

Assignment 9.1: Paper Prototype

- Your task is to design a privacy controller app.
  - An app that magically helps you control and keep track of the privacy settings of all other apps on your device. You should be able to have custom privacy settings for different apps, but it should still be easy to use.
- Form groups of 4-6 people
  - Pick a couple of concrete tasks to focus on (your choice)
  - Create a paper prototype for those tasks
  - Work quickly! Set a deadline.
  - Evaluate your paper prototype with another team
  - Take turns in using each other's prototypes
  - Write down the results from testing your prototype
  - Iterate!
- Present your findings in the course

Dimensions of Fidelity



http://www.sapdesignguild.org/editions/edition7/proto_design.asp

- Fidelity can be broken down into four basic dimensions:
  - Breadth
  - Depth
  - Look
  - Interaction

Dell (2018)

Breadth

The "breadth" of a prototype refers to how much of the product's functionality is represented in the prototype

- A very narrow prototype only represents a single feature
- A broad prototype represents all intended functionality
- Prototypes should generally be as broad as needed to cover basic or most important tasks, but not much more

Depth

The "depth" of a prototype refers to how much of the prototype is functional, and how robust it is

- A very shallow prototype has no backend at all and is hard-coded to respond as though the user had provided ideal input
- A deep prototype has some logic and error-handling capabilities
- At first glance, depth may seem unimportant, but it affects the amount of exploration a user can do
- Thus depth can actually have a profound influence on user testing!

Look

The "look" refers to how accurately a prototype represents the product's intended appearance, including fonts, colors, and graphics

- "Look" is probably what most people think of when they think of prototype fidelity
- It's generally a good idea to hold off on something that has a high fidelity look until later in the design process
- People are less likely to point out flaws and mistakes
- People can easily fixate on the "little" things
- You are less likely to throw it out and start again

Interaction

"Interaction" refers to how the prototype handles input and output

- Interaction can often be simulated
- For example, you might create a digital prototype for an iPad application which runs on your desktop and responds to traditional a traditional mouse and keyboard
- You might use hyperlinks or animation to simulate clicking interaction (e.g., in Powerpoint)

10 Models & Metaphors

Interaction Styles

- We look at the following Interaction Styles
  - Command language/command line
  - Menus & forms
  - Direct manipulation
    - Touch and Mouse
- Also interesting, but outside the scope today
  - Other forms of graphical interaction
  - 3D-Gestures
  - Natural Language Interfaces
  - Explicit *vs.* implicit interaction
  - Behavioural Interfaces

Direct Manipulation

- User interacts with visual representation of data objects (based on Shneiderman and Plaisant (2005)):
  - Continuous visual representation
    - Verbal or iconic
  - Physical actions or labeled button presses
    - most direct kind of action, analog to real world interaction
    - not everything can be easily mapped – convert a text to bold – so "command actions" are allowed
  - Rapid, incremental, reversible, immediately visible effects
    - within 100ms (why?)
    - drag a bit, see the change
    - physical or logical

Direct Manipulation II

- Examples
  - Files and folders on a desktop
  - Scrollbar
  - Dragging to resize a rectangle
  - Selecting text
- Visual representation and physical interaction are important
- It is powerful since it exploits perceptual and motor skills of the human user
- Some say it depends less on linguistic skills than command or menu/form interfaces
  - Only partly true and for a limited understanding of language

Comparison of Interaction Styles

- Knowledge in the head *vs.* world
  - CLI needs practice, training, references, manuals
  - M&F put much more information into the world
  - DM has information from affordances and constraints of metaphor
- Error messages regarding the interaction itself
  - DM rarely needs them – try to drag a scroll bar too far
- Efficiency
  - CLI good for experts
  - M&F demand good shortcuts
  - DM if appropriate for task, but mis-using can be labor intensive
- User experience
  - CLI best for experts
  - M&F, DM better for novices, infrequent users

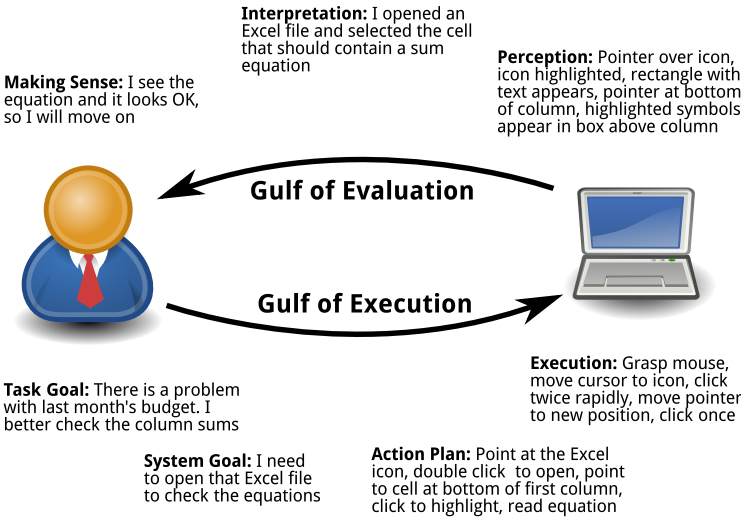
Comparison of Interaction Styles II

- Synchrony
  - CLI synchronous, M&F (user types, system does)
  - DM asynchronous, user can point anywhere, do anything
- Programming difficulty
  - CLI are easy, parsing rigid texts well understood
  - M&F, DM with substantial toolkit support
- Accessibility
  - CLI, M&F easier since both are text based
  - DM much harder

Stages of Interaction

- There are lots of places where interaction between human and machine can go wrong
  - Perception
  - Cognition
  - Action
- Stages of action proposed by Norman (1986)
- Two gaps
  - **Gulf of Evaluation:** the "cognitive distance" between what is displayed and the user's mental representation
  - **Gulf of Execution:** distance between the user's goals and the procedures and actions provided to pursue this goals

Gulfs Elaborated



Questions from the Action Cycle

1. What do I want to accomplish?
2. What are the alternative action sequences?
3. What action can I do now?
4. How do I do it?
5. What happened?
6. What does it mean?
7. Is this okay? Have I accomplished my goal?

(Norman, 2013)

Principles

1. Discoverability.
   - It is possible to determine what actions are possible and the current state?
2. Feedback.
   - Full and continuous information about the results of actions and the current state of the product or service.

3. Conceptual model.
    - Design projects all the information needed to create a good conceptual model.
4. Affordances.
    - The proper affordances exist to make the desired actions possible.
5. Signifiers.
    - Effective use of signifiers ensures discoverability and that the feedback is well communicated and intelligible.
6. Mappings.
    - The relationship between controls and their actions follows the principles of good mapping.
7. Constraints.
    - Constraints guide actions and eases interpretation.

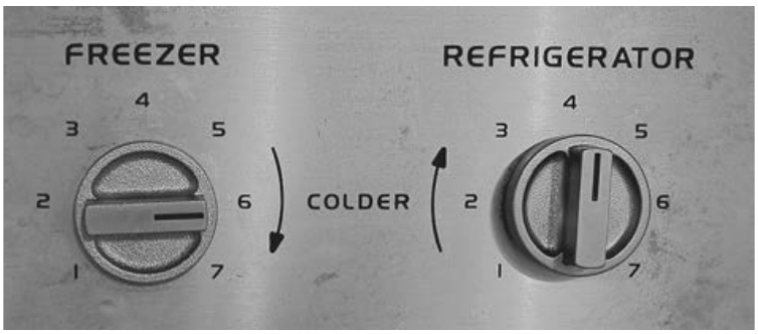
Discoverability by Visibility

- *Relevant parts* of system should be *visible*
- If the user cannot see an important control, they would have to
    - guess that it exists, and
    - guess where it is
- Not usually a problem in the real world
    - Look at a bike or a pair of scissors
    - Hiding often takes effort (hidden doors)
    - Design can come in the way
- But takes extra effort in computer interfaces
    - Mouse clicks can be interpreted in arbitrary ways

Feedback

- *Feedback:* what the system does when you perform an action
- When the user successfully makes a part work, it should appear to respond
- Actions should have immediate, visible effects
    - Push buttons depress and release
    - Scrollbars move
    - Drag & drop following the cursor
- Kinds of feedback
    - Visual – see above
    - Audio – clicks made by keyboard (or, artificially, touch screens)
    - Haptic – vibrating touch screens, force feedback 3D-mouse

Conceptual Model



Two compartments — fresh food and freezer — and two controls (in the fresh food unit). Your task: Suppose the freezer is too cold, the fresh food section just right. How would you adjust the controls so as to make the freezer warmer and keep the fresh food the same? (Norman, 2013).

Affordances

- *Perceived* and *actual* properties of a thing that determine how the thing could be used
  - Chair is for sitting
  - Knob is for turning
  - Button is for pushing
  - Listbox is for selection
  - Scrollbar is for continuous scrolling or panning
- Perceived *vs.* actual
  - A paper-mache chair still has a perceived affordance for sitting
  - A pole has no perceived affordance for sitting, but you can sit on it (albeit uncomfortably)
- The DM UI should agree on perceived and actual affordances

Signifiers

- Affordances exist even if they are not visible
- For designers, their visibility is critical: visible affordances provide strong clues to the operations of things
  - A flat plate mounted on a door affords pushing
  - Knobs afford turning, pushing, and pulling
- Perceived affordances help people figure out what actions are possible without the need for labels or instructions
- The signaling component of affordances is a signifier
  - Perceived affordances often act as signifiers, but they can be ambiguous
  - Signifiers signal things, in particular what actions are possible and how they should be done
- Signifiers must be perceivable, else they fail to function.

Mapping

- *Physical arrangement* of controls should match *arrangement of function*
- Best mapping is direct, but natural mappings do not have to be direct
  - Light switches
    - If the switches are arranged in the same fashion as the lights, it is much easier to learn which switch controls which light
  - Stove burners
    - Most stoves have four plates in a square and four controls in a row

  - Car turn signals
    * Up and down instead of left and right, but synchronous to turning wheel
  - DJ audio mixer
    * between turntable
- What is a direct mapping anyway?
  - Rudder of a boat *vs.* steering wheel of a car

Constraints I

- Graphical screen layout relies greatly on conventional interpretations of the symbols and placement
- Different types of constraints:
  - *Physical* constraints are closely related to real affordances
    * it is not possible to move the cursor outside the screen
    * Restricting the cursor to exist only in screen locations where its position is meaningful
  - *Logical* constraints use reasoning to determine the alternatives
    * If we ask the user to click on five locations and only four are immediately visible, the person knows, logically, that there is one location off the screen
    * It is how the user knows to scroll down and see the rest of the page
    * Logical constraints go hand-in-hand with a good conceptual model.

Constraints II

- Different types of constraints (contd):
  - *Cultural* constraints are conventions shared by a cultural group
    * That the graphic on the right-hand side of a display is a "scroll bar" and that one should move the cursor to it, hold down a mouse button, and "drag" it downward in order to see objects located below the current visible set is a cultural, learned convention
    * The choice of action is arbitrary: there is nothing inherent in the devices or design that requires the system to act in this way
    * "Arbitrary" does not mean that any random depiction would do equally well: the current choice is an intelligent fit to human cognition, but there are alternative methods that work equally well.

Description Error

- Intended action is replaced by another action with many features in common
- The user intends to do one action, but accidentally substitutes the other
  - Pouring orange juice into your cereal
  - Putting the wrong lid on a bowl
  - Throwing shirt into waste paper instead of hamper
- *Mitigation:* Avoid actions with very similar descriptions
  - Long rows of identical switches
  - Adjacent menu items that look similar

Capture Error

- A sequence of actions is replaced by another sequence that starts the same way
- The user starts executing one sequence of actions, but then veers off into another (often more familiar) sequence
  - Leave your house and find yourself walking to school instead of where you meant to go
  - Vi `:wq` command
- Picture for this: you have developed a mental groove from executing the same sequence of actions repeatedly, and this groove tends to capture other sequences that start the same way
- *Mitigation:* Avoid habitual action sequences with common prefixes

Mode Error

- Modes: states in which actions have different meanings
  - Vi's insert mode *vs.* command mode
  - Caps lock
  - Drawing palette
- Mode errors occur when the user tries to invoke an action that doesn't have the desired effect in the current mode
- *Mitigation:* Avoid modes ☺

Usability

> The official ISO 9241-210 definition of usability is the "extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use."

- Three measures
  - Effectiveness, efficiency and satisfaction
- Highly contextualised
  - Specified user, specified goals, specified context of use
- Focused on task achievement

User Experience

> In ISO 9241-210, we also read that user experience is a "person's perceptions and responses resulting from the use and/or anticipated use of a product, system or service."

- This includes all the users' emotions, beliefs, preferences, perceptions, physical and psychological responses, behaviours and accomplishments that occur before, during and after use.
- User experience is a consequence of brand image, presentation, functionality, system performance, interactive behaviour and assistive capabilities of the interactive system, the user's internal and physical state resulting from prior experiences, attitudes, skills and personality, and the context of use.

Working Definition

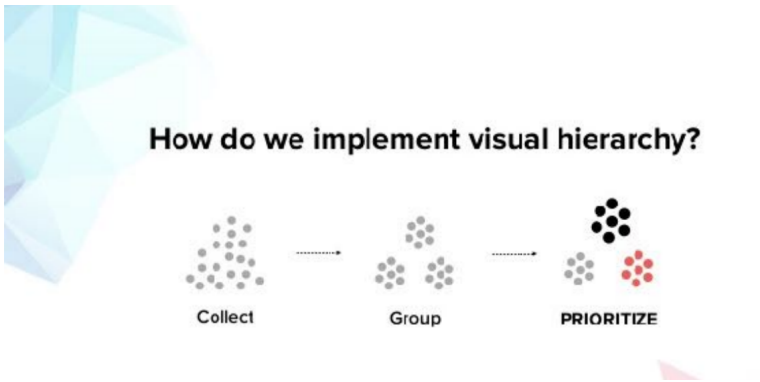
- We talk about Usability when we focus on pragmatic aspects
  - Accomplish a task, with minimal effort, without negatively affecting the user.
- We talk about User Experience when we focus on hedonic and affective aspects
  - Includes aesthetic and emotional factors, like appealing design or "joy of use".
- Not the only factors affecting the perceived quality of an artefact
  - Speed, reliability, safety
- We do not focus on such here, but will recognise them as "other" factors

11 Hierarchies

Establish Visual Hierarchy

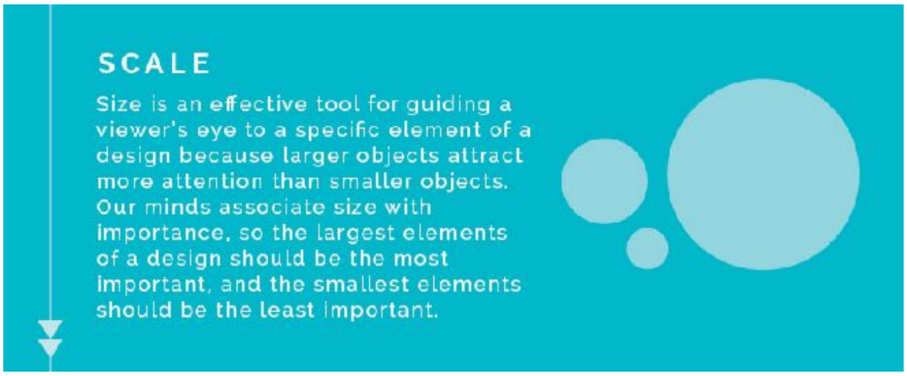
- Core technique based on Gestalt psychological theory:
  - Examines users' visual perception of elements in relation to each other
  - Shows how people tend to unify visual elements into groups
- Goals:
  - Present the content of apps and websites so users understand the level of importance for each element.
  - Organize UI components so the brain can distinguish objects based on their physical differences: size, color, contrast, style etc.

Achieving Visual Hierarchy



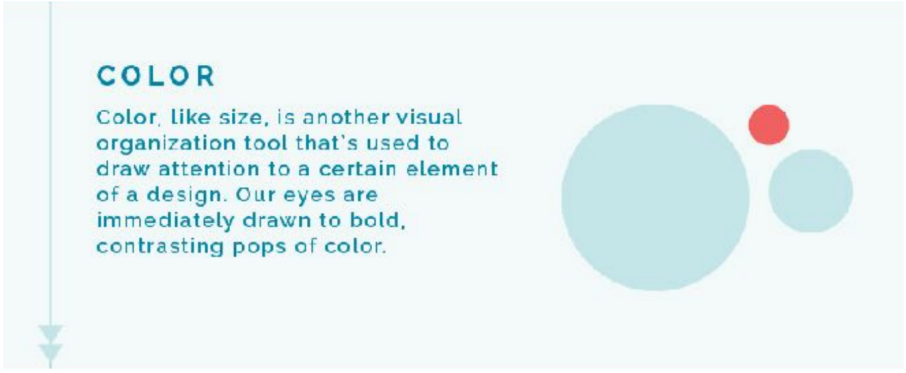
(Dell, 2018)

Achieving Visual Hierarchy



(Dell, 2018)

Achieving Visual Hierarchy

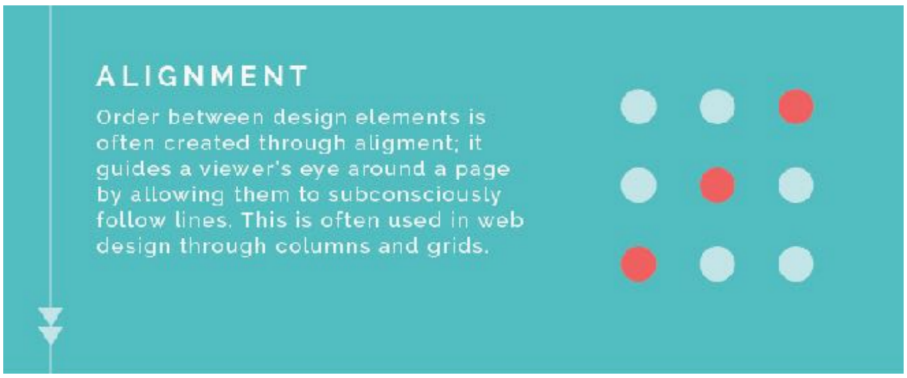


(Dell, 2018)

Achieving Visual Hierarchy

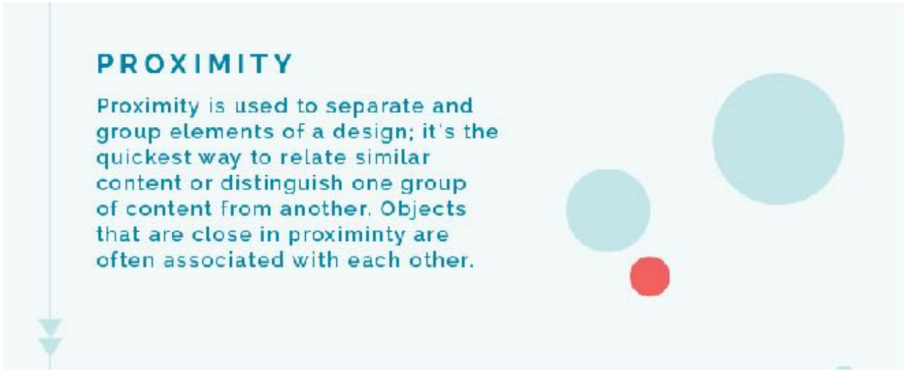
(Dell, 2018)

Achieving Visual Hierarchy



(Dell, 2018)

Achieving Visual Hierarchy



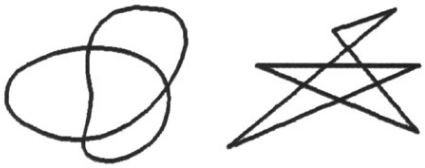
(Dell, 2018)

Typosemantik

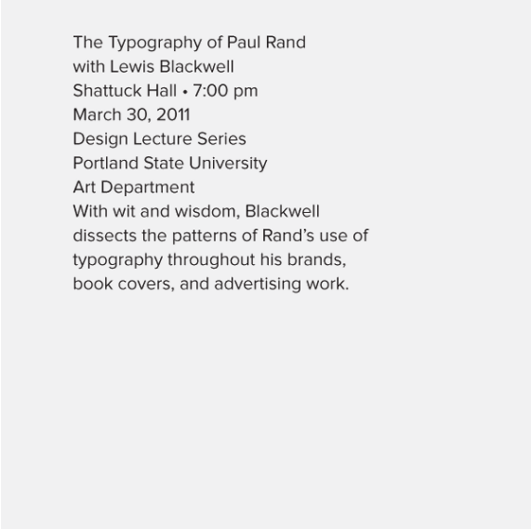
- Bedeutungsdimension von Zeichen
- Wesentlich: Auswahl einer zum Inhalt passenden Schrift
- Keine Frage des persönlichen Geschmacks, sondern eine der gestalterischen Sensibilität für emotionale Ansprache durch Typografie
- Ein Gestalter hat hauptsächlich die Aufgabe, eine bestimmte Botschaft für eine bestimmte Zielgruppe zu visualisieren
- Jede Schrift hat einen eigenen Schriftcharakter, der sie für bestimmte Anwendungsbereiche geeigneter macht als für andere

Assoziationen

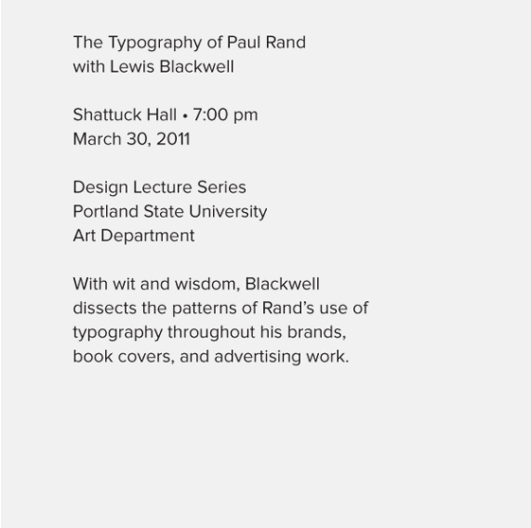
- Welche dieser Figuren heißt Maluma, welche Takete?



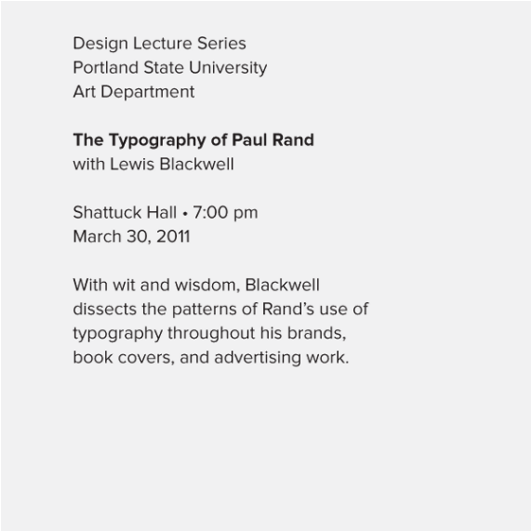
Wirth, 2002



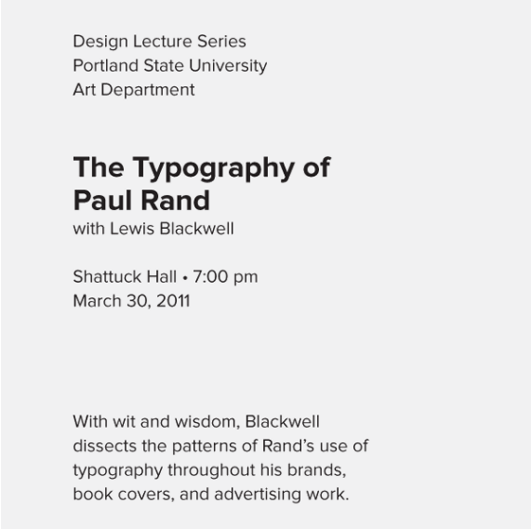
☞ Frank Chimero. Type study: Typographic hierarchy



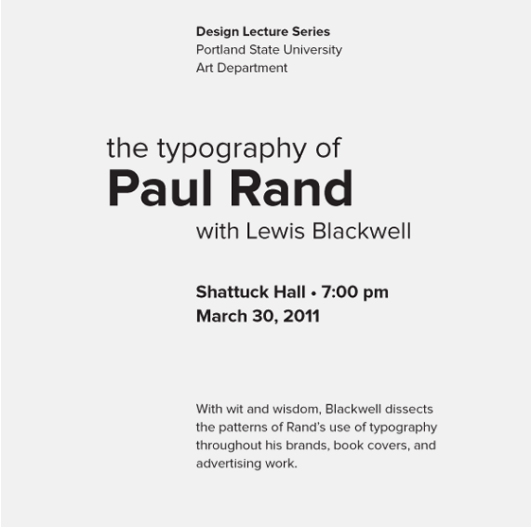
☞ Frank Chimero. Type study: Typographic hierarchy



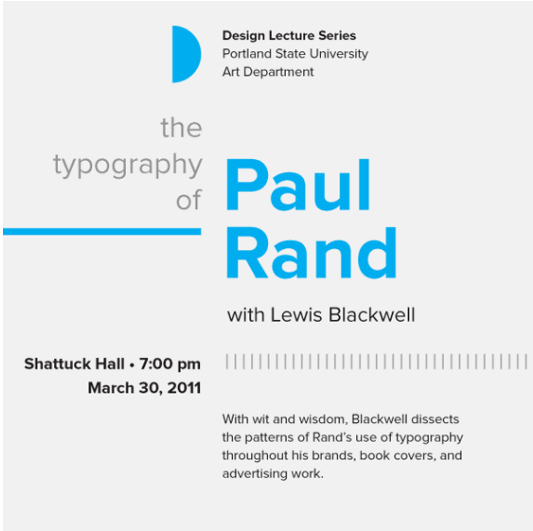
☞ Frank Chimero. Type study: Typographic hierarchy



☞ Frank Chimero. Type study: Typographic hierarchy



☞ Frank Chimero. Type study: Typographic hierarchy



☞ Frank Chimero. Type study: Typographic hierarchy

- Lesbarkeit ist nicht alles
  - Welche Emotionen möchte ich mit der gewählten Schriftart kommunizieren?
  - Welche Assoziationen möchte ich hervorrufen?
- Schriftarten schleppen einen großen Ballast mit sich herum
- Schrift als Bild geht in die Interpretation von Bildern über

Die Frage nach der Wirkung einer Schrift ist (auch) eine Frage der (multicodalen) Semiotik

12 Patterns

Posture

- Two primary types of desktop interfaces: sovereign and transient
- majority of actual work that gets done on desktop applications is done in sovereign applications
- Transients exist in supporting roles for brief, intermittent, or largely background tasks

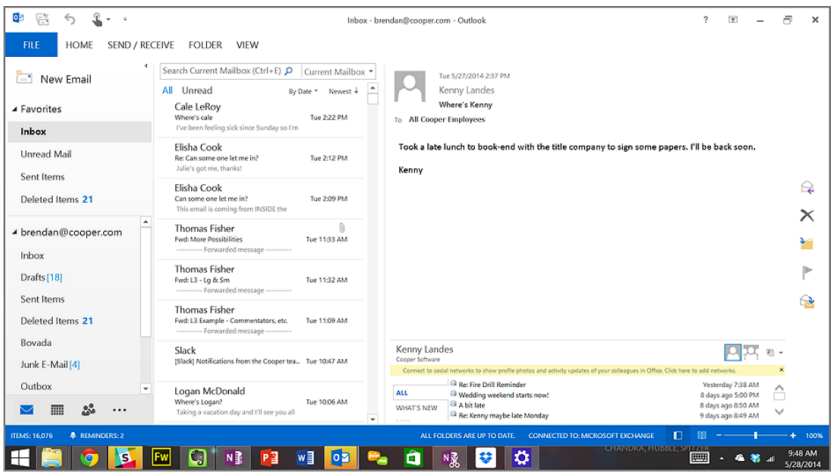
Primary and secondary windows

- The primary window contains your application's content, typically expressed in the form of documents that can be created, edited, and shared
  - Primary windows often are divided into panes that contain content, a means of navigating between different content objects, and sets of frequently used functions for manipulating or controlling the content
  - Primary windows typically are designed to assume sovereign posture, filling most of the screen and supporting full- screen modes
- Secondary windows support the primary window, providing access to less frequently used properties and functions, typically in the form of dialogs
  - If your application allows panes located in the primary window to be detached and manipulated separately, these floating panels or palettes also take on a role as secondary windows

Primary window structure

- **Menus and toolbars** are collections of related actions the user can instruct the application to perform, such as "close this document" or "invert the colors of the current selection."
- **Content panes** form the primary work area within most desktop applications, whether it is the editable view of a form or document or (as in the case of a software music synthesizer, for example) a complex control panel
- **Index panes** provide navigation and access to documents or objects that ultimately appear in the content view(s) for editing or configuration
- **Tool palettes** allow the user to rapidly switch between the application's modes of operation by selecting one tool from a set of tools
- **Sidebars** most often allow object or document properties to be manipulated without the need to resort to modal or modeless dialogs
- Question: What are ribbons?

Multipaned



(Cooper et al., 2014)

MDI vs SDI

- Multiple document interface, or MDI.
  - multiple windows reside under a single parent window.
- Tabbed document interface, or TDI
  - allows multiple documents or panels to be contained within a single window.
- Single document interface, or SDI.
  - all windows are independent of each other.

Menu

- Drop-down, pop-up
- Toolbars and direct-manipulation idioms can be too inscrutable for a first-time user to understand, but the textual nature of the menus explains the functions
- For an infrequent user who is somewhat familiar with an application, the menu's main task is as an index to known tools: a place to look when he knows there is a function but he can't remember where it is or what it's called.
- For a frequent user, menus provide a stable physical location at which to access one of hundreds of possible commands, or a quick reminder about keyboard shortcuts.

Accelerators and Mnemonics

- Accelerators or keyboard shortcuts provide an easy way to invoke functions from the keyboard.
  - These are commonly function keys (such as F9) or combinations involving modifier keys (Ctrl, Alt, Option, and Command).
- Access keys or mnemonics are another Windows standard (they are also seen in some UNIX GUIs) for adding keystroke commands in parallel to the direct manipulation of menus and dialogs
  - Mnemonics are accessed using the Alt key, arrow keys, and the underlined letter in a menu item or title.

Toolbars and menus

- Toolbars work together with menus to satisfy user needs as they mature
- Whereas menus are complete toolsets with the main purpose of teaching inexperienced users and organizing seldom-used advanced functions, toolbars are for frequently used commands and cater to perpetual intermediates
- They complement each other perfectly, addressing different user needs at different times
- Toolbars are modeless, but they don't introduce the conundrums that modeless dialogs do
- Toolbar button, or icon button
- ToolTips

Docked Palettes

(Cooper et al., 2014)

Sidebars, task panes, and drawers

(Cooper et al., 2014)

Mouse buttons and controls

- In general, the left mouse button is used for all the primary direct-manipulation functions, such as triggering controls, making selections, drawing, and so on
- The right mouse button enables direct access to properties and other context-specific actions on objects and functions via the ubiquitous context menu
- Rolling the wheel forward scrolls the window up, and rolling it backwards scrolls the window down, Pressing it acts like a third mouse button
- Using modifier keys in conjunction with the mouse can extend direct-manipulation idioms
- Metakeys include Ctrl, Alt, Command (on Apple computers), and Shift

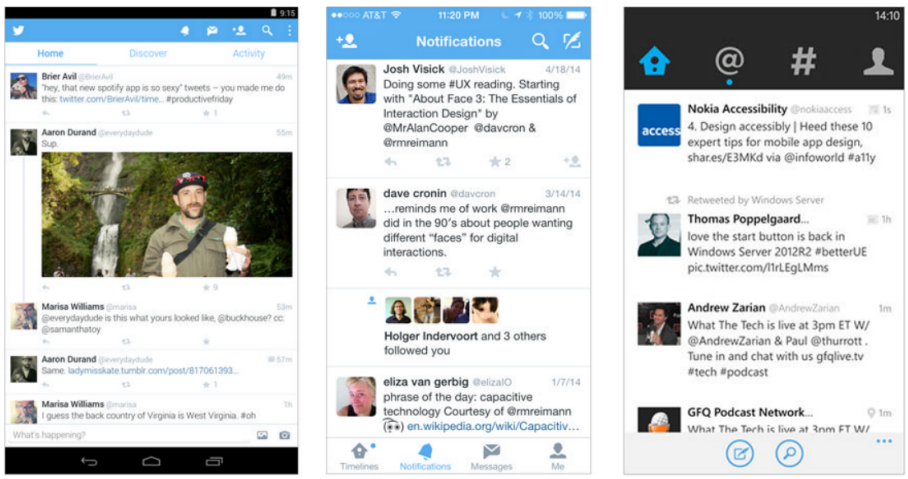
Point & Click

- Clicking and dragging
  - selecting, reshaping, repositioning, drawing, and dragging and dropping
- Double-clicking
  - Double-clicking means single-clicking plus action
- Chord-clicking
  - ???
- Double-clicking and dragging
  - ???

Drag and drop

- drag-and-drop operation: clicking and holding the button while moving an object across the screen and releasing it in a meaningful location
- Surprisingly, drag and drop isn't used as widely as we'd like to think, and it certainly hasn't lived up to its full potential.
- Drop candidates must visually indicate their receptivity.
- The drag cursor must visually identify the source object.

Stacks

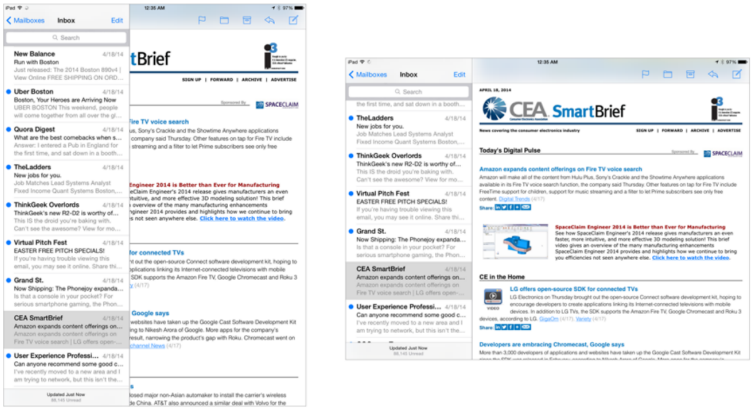


(Cooper et al., 2014)

Carousel

(Cooper et al., 2014)

Stacks and index panes

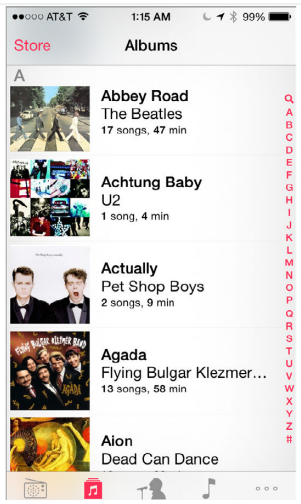


(Cooper et al., 2014)

Pop-up control panels

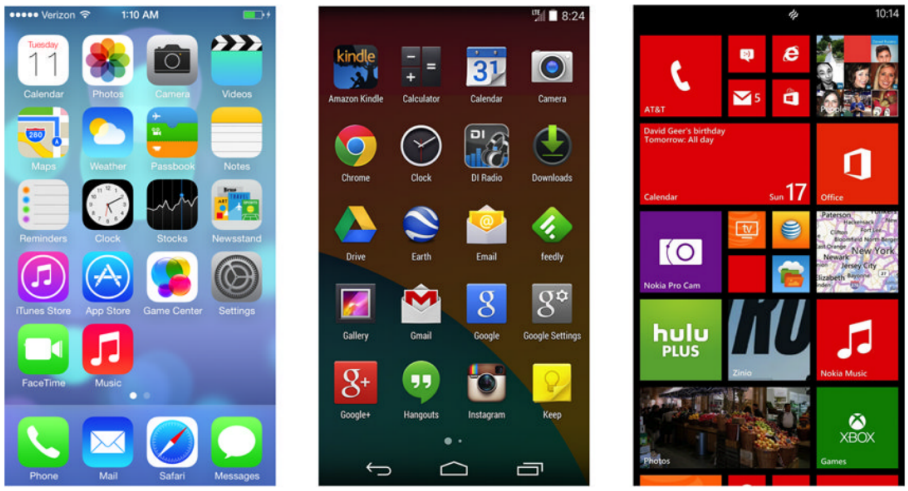
(Cooper et al., 2014)

Lists



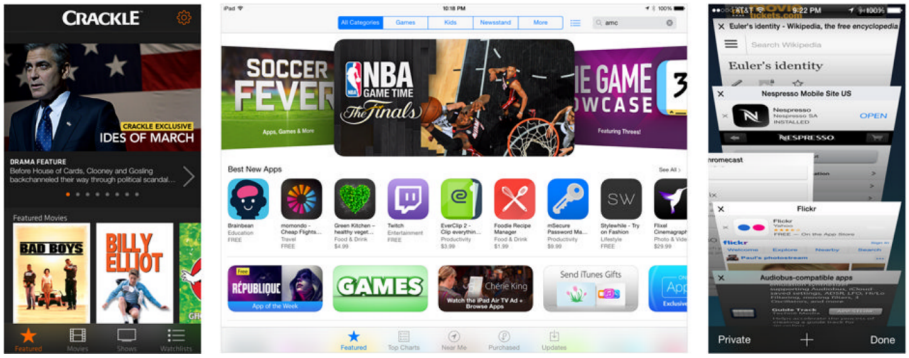
(Cooper et al., 2014)

Grids



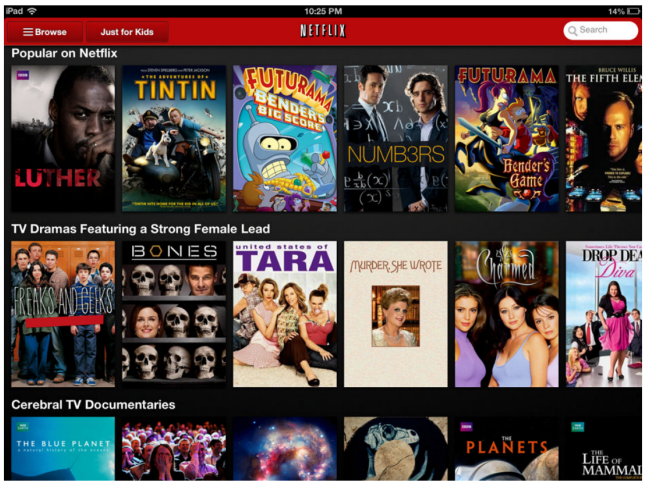
(Cooper et al., 2014)

Content carousels



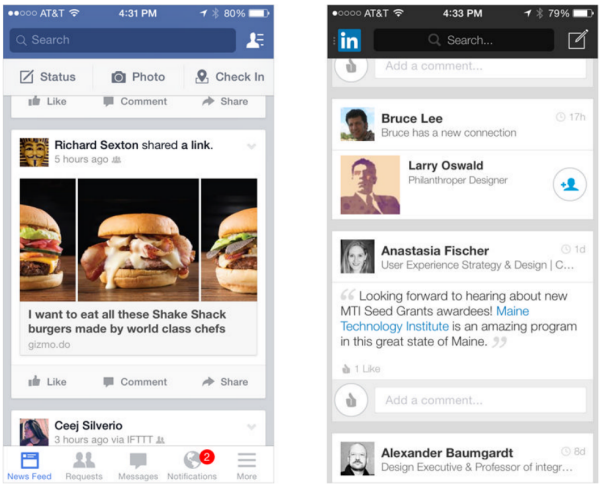
(Cooper et al., 2014)

Swimlanes



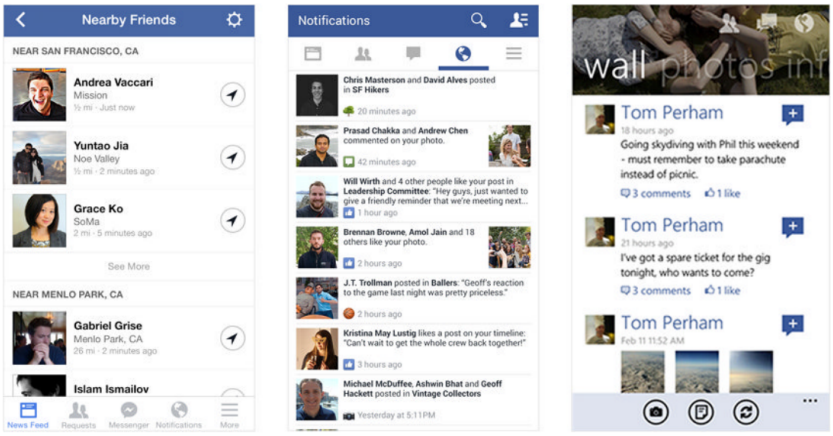
(Cooper et al., 2014)

Cards



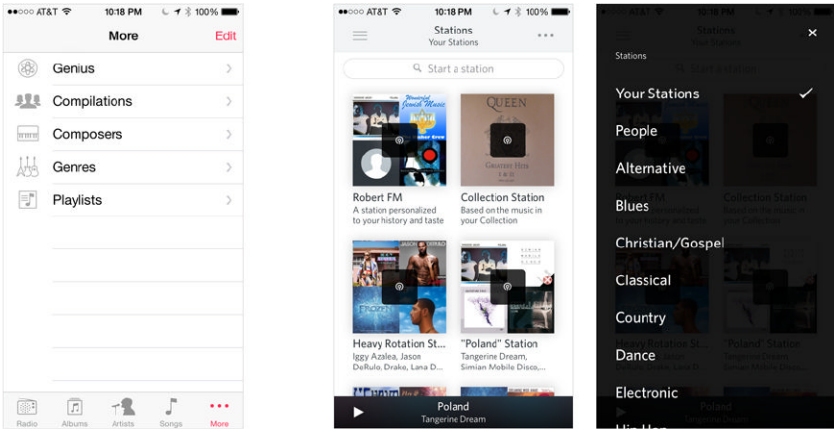
(Cooper et al., 2014)

Tab Bars



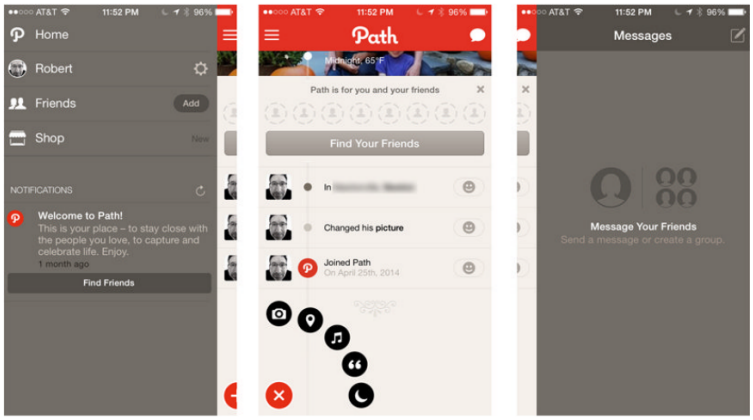
(Cooper et al., 2014)

More . . . Controls



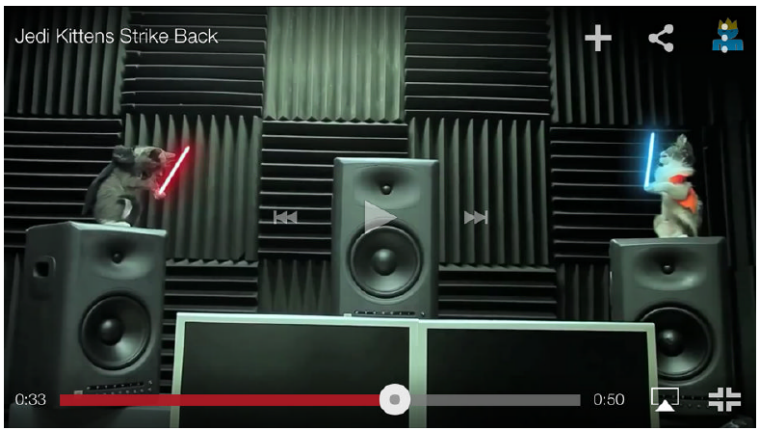
(Cooper et al., 2014)

Drawers



(Cooper et al., 2014)

Tap-to-reveal



(Cooper et al., 2014)

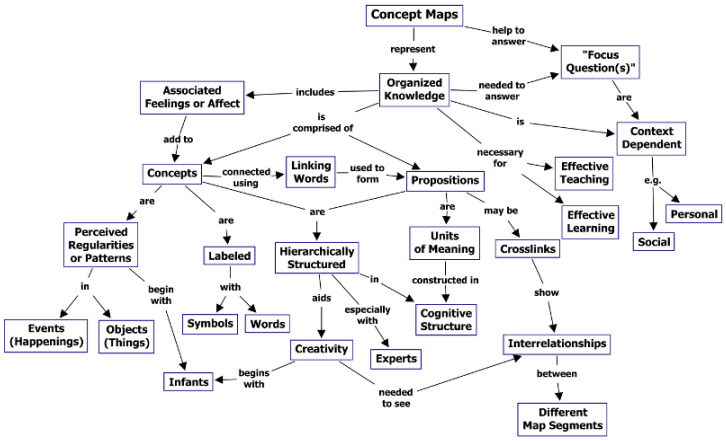
13 Closing

Assignment 15.1: Wissensmodellierung

- Bilden Sie Gruppen von 3-6 Personen
- Modellieren Sie Ihr Verständnis der Inhalte dieser Vorlesung
- Benutzen Sie dafür z.B.:
  - Mindmaps

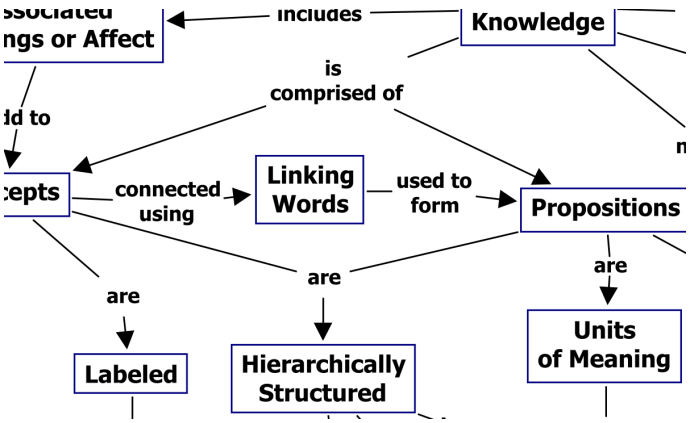
  - Semantic Nets
  - Concept Maps
- Stellen Sie Ihr Ergebnis vor
- Gelingt es uns, eine gemeinsame Modellierung zu finden?

Assignment 15.1: Wissensmodellierung



☞ Novak & Cañas: IHMC CmapTools

Assignment 15.1



☞ Novak & Cañas: IHMC CmapTools

Assignment 15.2: Prüfungsfragen

- Bilden Sie Gruppen von 3-6 Personen
- Überlegen Sie sich mögliche Fragen für eine Prüfung in Medieninformatik II
- Begründen Sie Ihre Auswahl an Fragen

Veranstaltungen Wintersemester 2019/2020

- Medieninformatik, Vorlesung und Übung
  - Dienstag, 10:00-12:00 Uhr
  - Mittwoch, 12:00-14:00 Uhr
  - Beginn: Dienstag, 22.10.
- Medieninformatik Seminar
  - Vorbesprechung: Mittwoch, 23.10., 14:00-16:00 Uhr
  - Bei Interesse gerne ☞ eine Mail

- Informatik und Gesellschaft
  - Vorbesprechung: Mittwoch, 23.10., 16:00-18:00 Uhr
  - Veranstaltung Lehramt Informatik, IMIT & AI
- Terminänderungen sind noch möglich! Bitte das LSF und das Learnweb beachten

References

Literatur

Bahill, A. T. and Gissing, B. (1998). Re-evaluating systems engineering concepts using systems thinking. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 28(4):516–527.

Benyon, D., Turner, P., and Turner, S. (2005). *Designing interactive systems: People, activities, contexts, technologies*. Pearson Education.

Beyer, H. and Holtzblatt, K. (1997). *Contextual design: defining customer-centered systems*. Elsevier.

Chapman, C. N. and Milham, R. P. (2006). The personas' new clothes: methodological and practical arguments against a popular method. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 50, pages 634–636. SAGE Publications Sage CA: Los Angeles, CA.

Cooper, A., Reimann, R., Cronin, D., and Noessel, C. (2014). *About Face (fourth edition): the essentials of interaction design*. John Wiley & Sons.

Dell, N. (2018). Hci and design. Course material, Cornell University.

Go, K. and Carroll, J. M. (2004). The blind men and the elephant: Views of scenario-based system design. *interactions*, 11(6):44–53.

Hix, D. and Hartson, H. R. (1993). *Developing user interfaces: ensuring usability through product & process*. John Wiley & Sons, Inc.

Holtzblatt, K. and Beyer, H. (2016). *Contextual design: Design for life*. Morgan Kaufmann.

McCracken, D. D., Wolfe, R. J., and Spool, J. M. (2004). *User-centered website development: A human-computer interaction approach*. Prentice Hall Upper Saddle River.

Nielsen, J. (1994). Enhancing the explanatory power of usability heuristics. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 152–158. ACM.

Norman, D. A. (1986). Cognitive engineering. *User centered system design*, 31:61.

Norman, D. A. (2004). *Emotional design: Why we love (or hate) everyday things*. Basic Books.

Norman, D. A. (2005). Human-centered design considered harmful. *interactions*, 12(4):14–19.

Norman, D. A. (2013). *The design of everyday things: Revised and expanded edition*. Basic Books.

Pruitt, J. and Grudin, J. (2003). Personas: practice and theory. In *Proceedings of the 2003 conference on Designing for user experiences*, pages 1–15. ACM.

Rosson, M. B. and Carroll, J. M. (2002). *Usability Engineering: Scenario-Based Development of Human-Computer Interaction*. Morgan Kaufmann/Academic Press.

Royce, W. W. (1970). Managing the development of large software systems: concepts and techniques. In *Proceedings IEEE WESTCON*, pages 1–9, Los Angeles.

Sauer, J. and Sonderegger, A. (2009). The influence of prototype fidelity and aesthetics of design in usability tests: Effects on user behaviour, subjective evaluation and emotion. *Applied ergonomics*, 40(4):670–677.

Scriven, M. (1966). The methodology of evaluation. Technical Report SEC-FUB-110, Social Science Education Consortium.

Shneiderman, S. B. and Plaisant, C. (2005). *Designing the user interface (fourth edition)*. Pearson Addison Wesley.