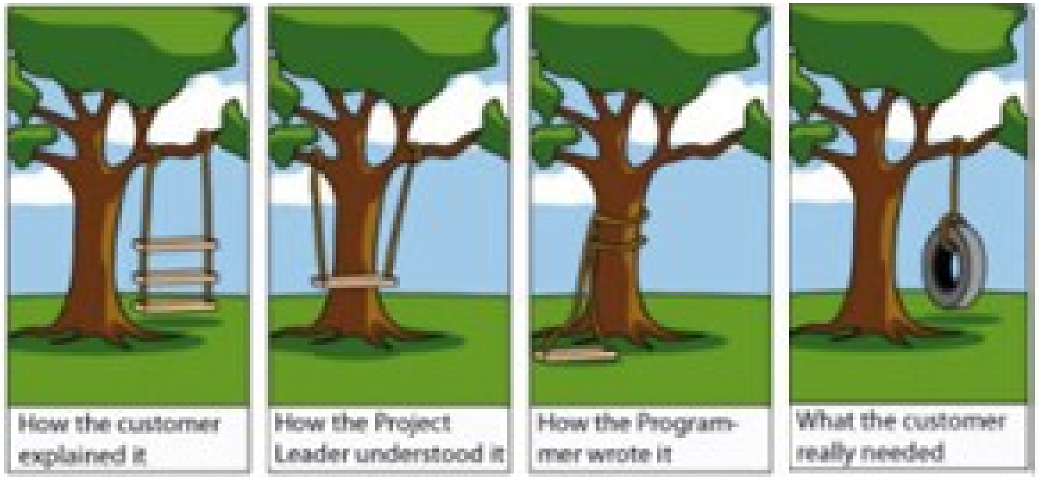
Universidade de Aveiro
Departamento de Electrónica,
Telecomunicações e Informática

Design of Interactive Systems

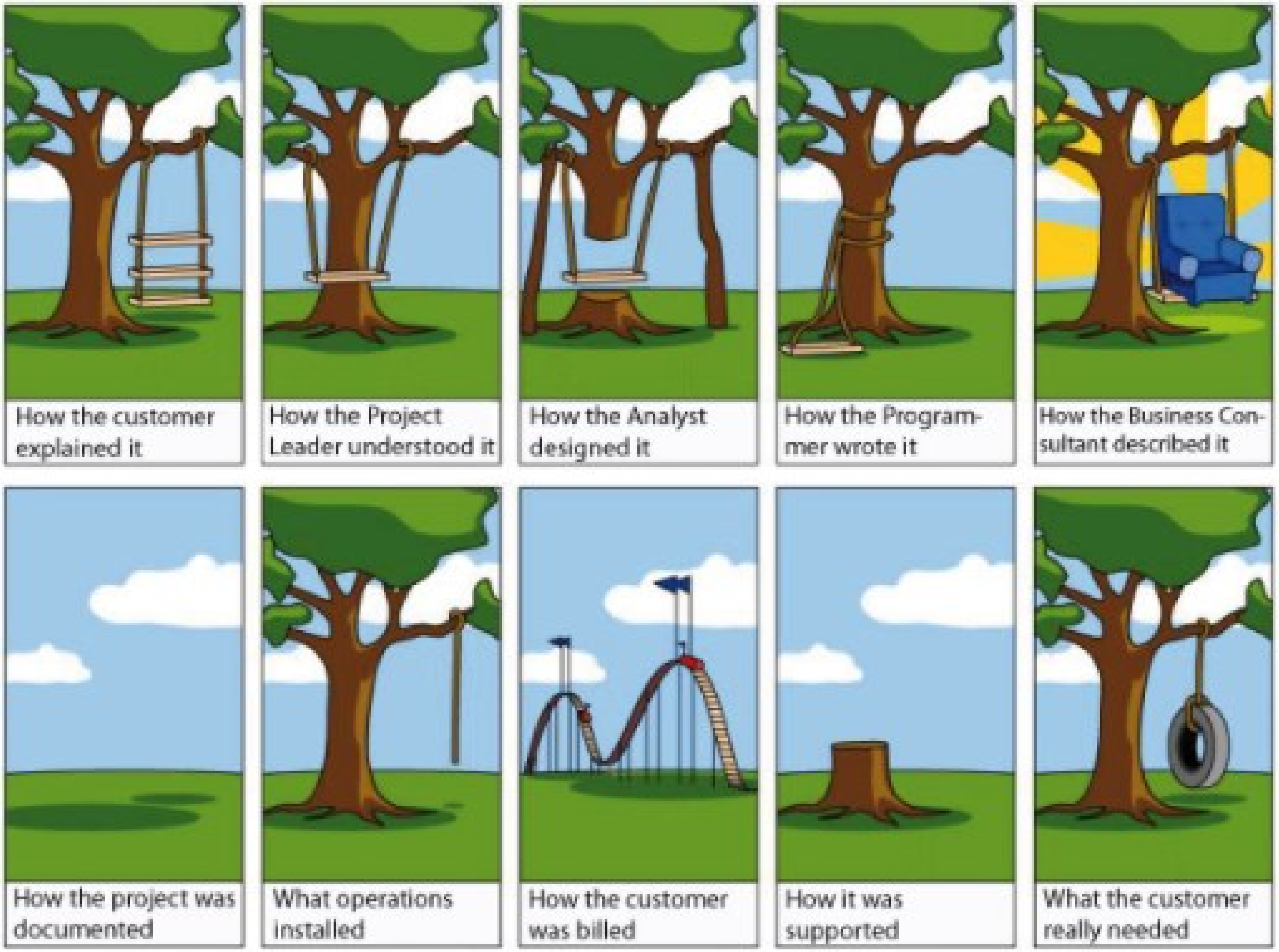
Human-Centered design of interactive systems and Usability Engineering Lifecycle



Human-Computer Interaction

Beatriz Sousa Santos, 2021

The problem of interactive systems design…



Human Centered Design of interactive systems

- Also known as User Centered Design (UCD)
- Complementary approaches to the design that should be used:

  Usability principles (technology independent)

  Usability paradigms (more technology dependent)

  Guidelines and standards (more specific)

- **S/W development technologies**

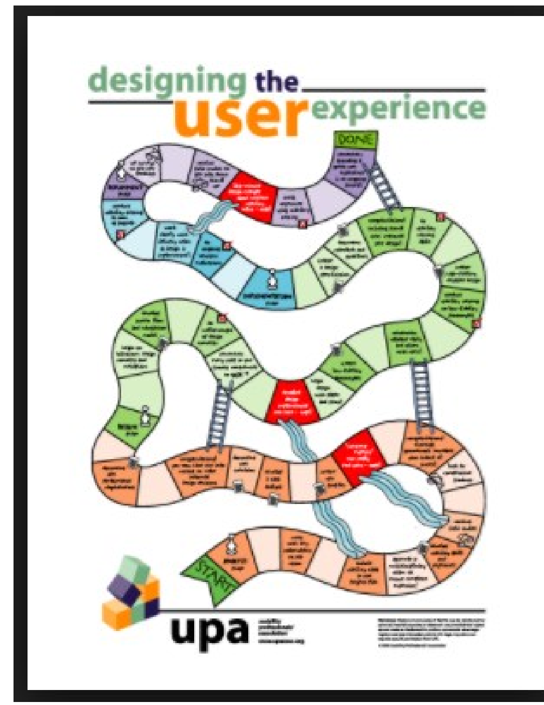
- **Evaluation**

- ISO standard 13407 (1999) - *Human centered design processes for interactive systems*:

"Human-centered design is an approach to interactive system development that focuses specifically on making systems usable. It is a multi-disciplinary activity."

- There are several proposals of UCD methodologies
- All are iterative
- And include usability evaluation in iterations

We must consider the situation at hand and ponder which are the best fitting and how to use them



https://uxpa.org/
https://www.w3.org/WAI/redesign/ucd
http://www.usability.gov/how-to-and-tools/methods/user-research/index.html

Benefits of Human-centered design

- Following the best practices, helps to identify challenges upfront so that a solution can be found early
- By putting a larger emphasis on principles and practices, iterative improvements can be made and avoid costly large scale rework
- The“10%” rules:
  - 10% of IT staff should be user experience (UX) professionals
  - 10% of budget dedicated to UX.

http://www.usability.gov/what-and-why/benefits-of-ucd.html

Top 12 reasons IEEE identified for why IT projects fail in 2005 (three can be fixed by using adequate UX methods)

- Unrealistic or unarticulated project goals
- Inaccurate estimates of needed resources
- Badly defined system requirements ← May be alleviated by good UX design
- Poor reporting of the project's status
- Unmanaged risks
- Poor communication among customers, developers, and users
- Use of immature technology
- Inability to handle the project's complexity
- Sloppy development practices
- Poor project management
- Stakeholder politics
- Commercial pressures

http://spectrum.ieee.org/computing/software/why-software-fails

A study about why IT projects fail
(and several causes are avoidable using adequate UCD methods)

Comparing with the situation in aviation an **IT accident** may be defined as:

"project with significant damage, e.g. large cost or schedule overrun, failing business goals, low usability, etc."

Five accident investigations of large government IT projects in Denmark

Identified 37 different causes, each causing damage to one or more projects, e.g.

- surprises with system integration
- wrong estimate of human performance

Only one cause related to programming

Proposes 22 cures that in combination could have prevented most of the damages

Half are familiar to developers, but were ignored in the project (e.g. usability test)

S. Lauesen, "IT Project Failures , Causes and Cures," *IEEE Access*, vol. 6, pp. 72059-67, 2020.

Some causes for IT projects failures
(and several causes are avoidable using adequate UCD methods)

Analysis
Doesn't identify user needs and win-win
Requirements don't cover customer needs
Describes solution in detail. No freedom to supplier
Makes heavy demands and believes it is for free
Oversells technology, e.g. SOA, web-based
Multi-vendor strategy - supplier independent ! ?
Wants everything at once, e.g. cover all types of debt
Doesn't plan the new work processes
No feasible solution, e.g. data missing, performance dubious.
Surprising rule complexity.

Study as-is and plan to-be
Problem oriented requirements (SL-07)

Plan to-be, SL-07 reqs (sections B1 and C)

Design
Doesn't ensure usability, even when they know how
Designs user screens too late
Accepts the solution description without understanding it
Cannot see how far the supplier is

Early prototypes & usability tests
Early prototypes & usability tests
Early prototypes & usability tests
Early prototypes, Monitor remaining work hours

Test
Deploys the system with insufficient testing

Pilot test, Ask expert developers

Deployment
Deploys system with insufficient support/training
The system is not used as intended
Wrong estimate of human performance

Usability tests, Pilot test, Deploy part-by-part
Observe at pilot test, Follow-up study
Check at POC, Check at pilot test

HCD/UCD involves

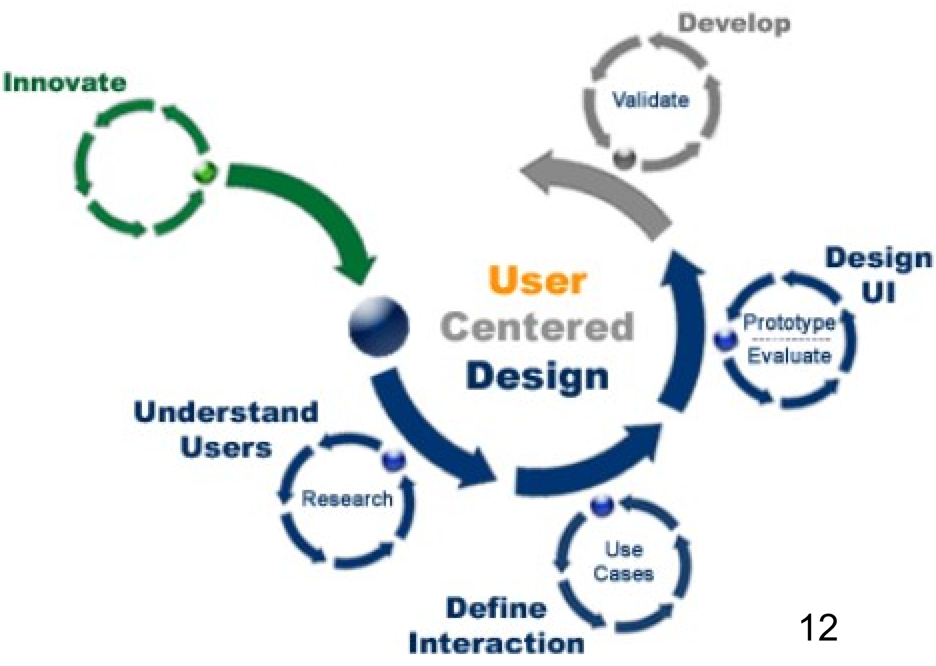
- Understanding users’ needs, motivations, contexts and activities
- Understanding business, technical, and domain opportunities, requirements, and constraints
- Using this knowledge as a grounds to create products whose form, content, and behavior is useful, usable, economically viable and technically feasible

- We must know the successful cases (usability paradigms)
- Understand why they work (usability principles)
- Adopt adequate methodologies
- And test, re-design, test, re-design

…

until usability goals are met

This is applicable not only to conventional interactive systems



Iteration

- Its role is to enable continual refinement through rapid prototyping and testing through repetition

- Observe and study,
- Decide what the problem might be
- Determine which parts of the design work and which don't

iterate again…

- Fosters getting the requirements right

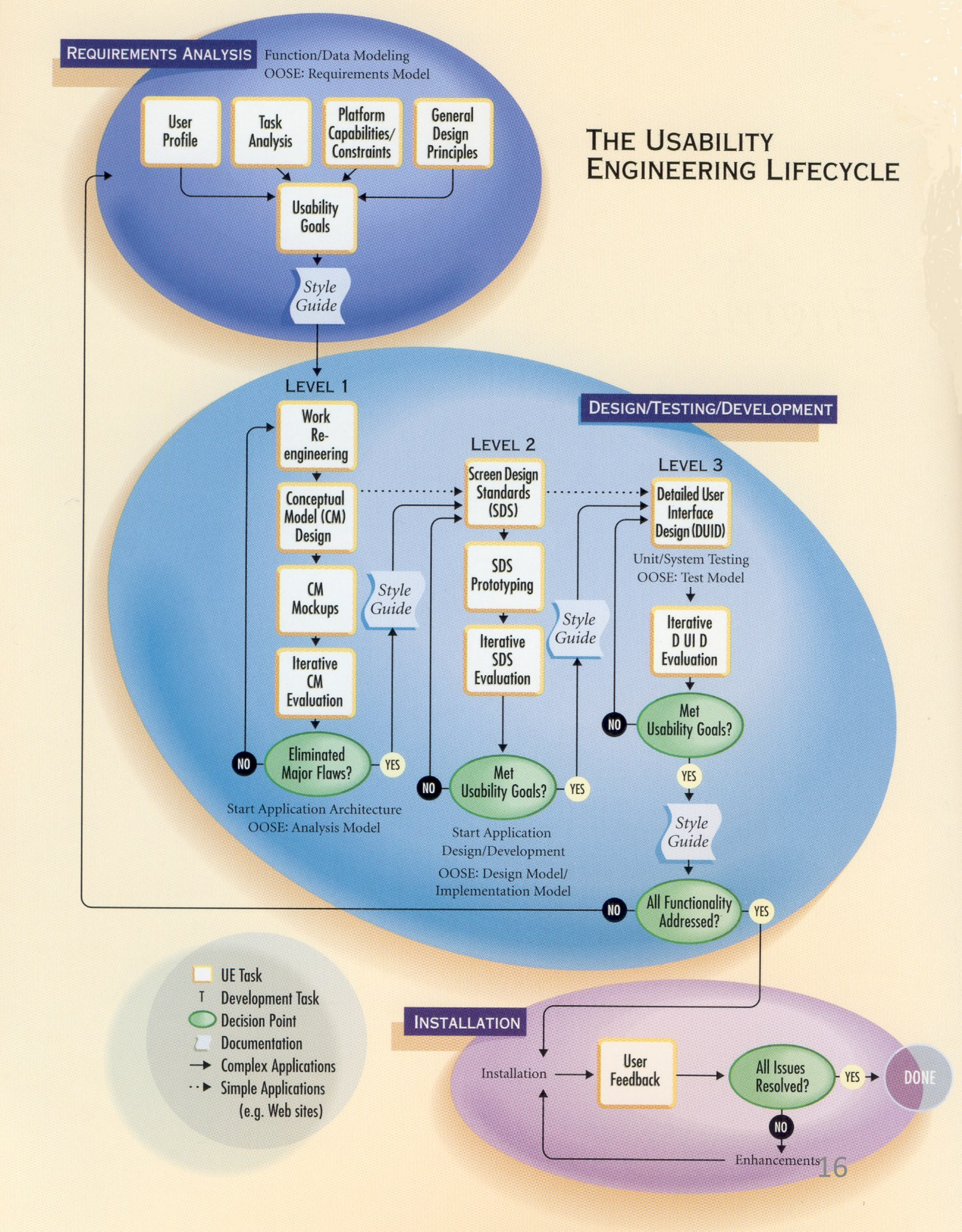
Examples of several approaches

- The Usability Engineering Lifecycle (Mayhew, 1999)
- Goal-directed design (Cooper, 2007)
- Activity-oriented design (Norman, 2013)
- Etc.

One of the first Human-Centered Design approaches:

The Usability Engineering Lifecycle

(Mayhew, 1999)



Requirement Analysis

User Profiles	Establish user characteristics important for UI design
Contextual Task Analysis	Obtain a user-centered model of work as it is currently done; extract the product usability requirements
Usability Goal Setting	Establish specific quantitative and qualitative usability goals to drive UI design
Platform Capabilities and Constraints	Establish capabilities and constraints of the technology platform which limit UI design alternatives
General Design Principles	Identify principles and guidelines that may be relevant for the product under development

Design, Testing, Development – Level 2

Screen Design Standards	Establish a set of design standards to set the stage for detailed UI Design
Screen Design Standards Prototyping	Support the evaluation, refinement and validation of the Screen Design Standards
Iterative Screen Design Standards Evaluation	Evaluate, refine, and validate the Screen Design Standards
Style Guide Development	Document the Conceptual Model Design, the Screen Design Standards and the output of Requirement Analysis

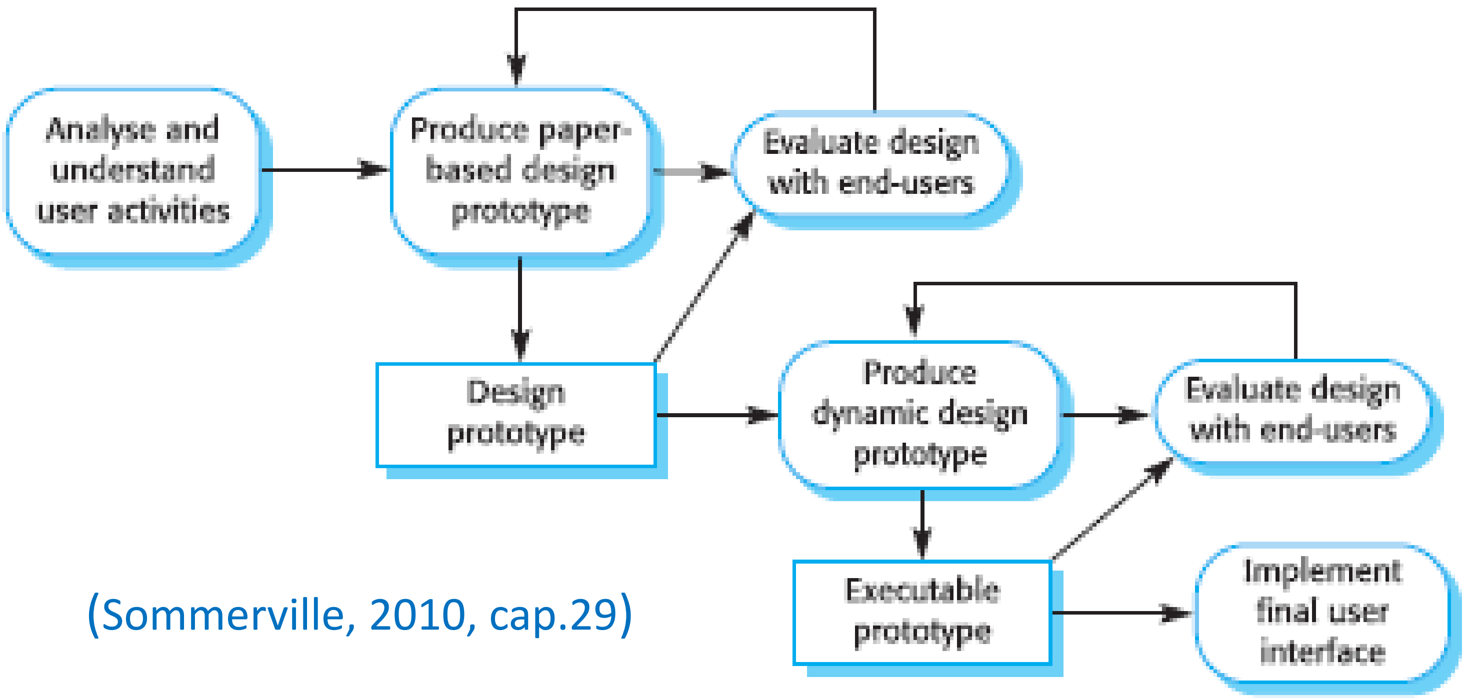
Design, Testing, Development – Level 3

Detailed User Interface Design	Design the complete, detailed product User Interface
Iterative User Interface Evaluation	Evaluate, refine, and validate key subsets of the detailed User Interface Design

Installation

User Feedback	Obtain usability data after a product has been installed and used Inform the UI Design for later releases or related products

User Interface design
from a S/W engineering perspective



(Sommerville, 2010, cap.29)

Goal-directed design (Cooper, 2007)

- Works on any platform, in any subject area, and with any users
- Is based on understanding the users and their desired end-state
- Not on any particular technology
- It involves personas and workflows

Activity-based design (Norman, 2013)

- Let the activity define the product
- Let the conceptual model be built around the activities
- Activity is a set of tasks performed together toward a goal

  It is different from task

  Is a high-level structure (e.g. “go shopping”)
- Tasks are lower level components (e.g. walk to the shop, ask for a product…)

User Research Methods

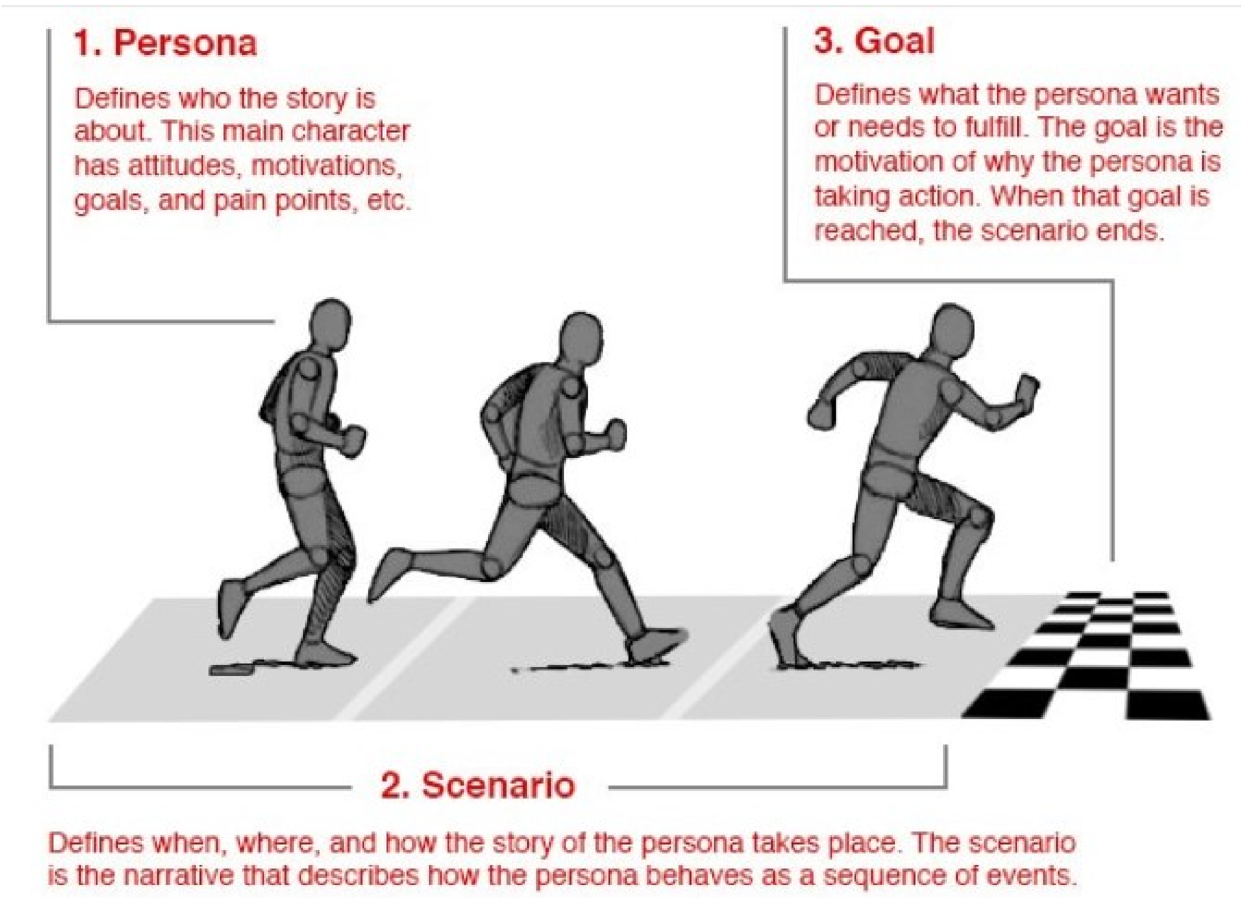
There are a lot of methods; we must consider the situation at hand to select and adapt the ones we use:

- Context interviews
- Focus groups
- Individual interviews
- On-line surveys
- **Personas**
- **Scenarios**
- **Task analysis**
- Activity analysis
- First click tests
- …

Methods to use in the mini-project

http://www.usability.gov/how-to-and-tools/methods/user-research/index.html

Personas and scenarios



https://www.interaction-design.org/literature/article/personas-why-and-how-you-should-use-them

Personas

- **Fictional characters based upon user research** to represent the different user types that might use a service/product in a similar way

- make the design task less complex,
  and guide the ideation process

- help:
  - understand users' needs, experiences, behaviors and goals
  - step out of oneself and recognize that different people have different needs and expectations
  - uncovering universal features and functionality
  - create a good user experience for your target users

History of Personas

Stem from IT system development during the late 1990s

How to best communicate an understanding of the users?

Various concepts emerged:

user archetypes, user models, lifestyle snapshots, …

Alan Cooper (1999) proposed personas to describe fictitious users

There is **no single definition** of what a persona should contain

Nor a unified understanding of **how to apply** the method

Benefits of Personas

- Offer a quick and inexpensive way to test and prioritize features throughout the development process

- Help to

  - focus decisions by adding a layer of real-world consideration

  - Stakeholders evaluate new feature ideas

  - Information architects develop informed wireframes, and interface behaviors

  - Designers create the overall look and feel

  - System engineers/developers decide which approaches to take based on user behaviors

https://www.usability.gov/how-to-and-tools/methods/personas.html

Types of Personas

- Several types (most based on previous user research):

  - **Goal-directed** Personas (Cooper, 2007)

  - **Role-based** Personas (goals + behavior)

  - **Engaging** Personas (goals + behavior + backgrounds)

  - **Fictional** Personas (based on assumptions, not user research)

- **Fictional personas can only be used as an initial sketch of user needs**

https://www.interaction-design.org/literature/article/personas-why-and-how-you-should-use-them

Best Practices for Developing Personas

- Create 2-4 personas of the product/service main audiences

- Conduct user research:
  - Who are the users
  - Why are they using the system?
  - What behaviors, assumptions, and expectations?

- Develop the appropriate descriptions of each persona's: background, motivations, and expectations

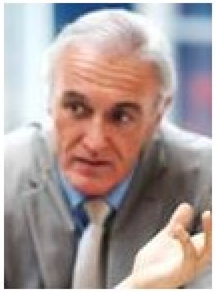
- Do not include too personal information
- Be relevant and serious

Elements of a persona

- Persona Group (i.e. web manager)
- Fictional name
- Job titles and major responsibilities
- Demographics such as age, and education
- The goals and tasks they are trying to complete using the product
- Their physical, social, and technological environment

Personas have no value in themselves, until they become part of a scenario they do not have real value!

Example of a Persona

Persona:	USDA Senior Manager Gatekeeper
Photo:	
Fictional name:	Matthew Johnson
Job title/ major responsibilities:	Program Staff Director, USDA
Demographics:	• 51 years old • Has a Ph.D. in Agricultural Economics.

Goals and tasks:	Spends his work time: • Requesting and reviewing research reports, • preparing memos and briefs for agency heads, and • supervising staff efforts in food safety and inspection.
Environment:	He is comfortable using a computer and refers to himself as an intermediate Internet user. He is connected via a T1 connection at work and dial-up at home. He uses email extensively and uses the web about 1.5 hours during his work day.

https://www.usability.gov/how-to-and-tools/methods/personas.html

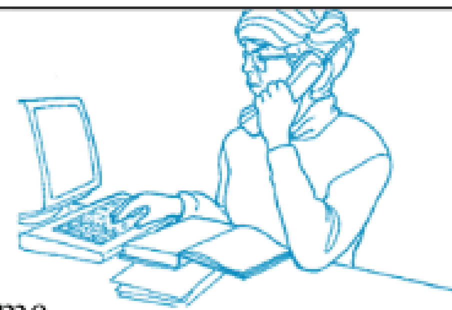
Example of using personas in VR

Model the people who will be using the VR application

Help to prevent the design from being driven by design/ engineering convenience

Personas should
- **Not be too detailed**

- be validated in later stages

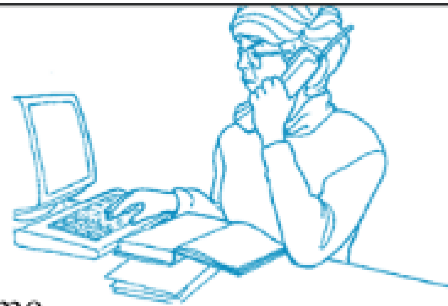
 Name	• Job • Experience • Activities • Attitude • Competencies • Age
• Problems • Pain points • Needs • Concerns • Fears • Desires	• Knowledge of VR • Dream VR system • Vision of VR • VR hardware access • Budget for VR • Activities that fit VR

(Jerald, 2016)

Describe 2–4 characters representing the range of targeted users

Sketch/photo and name

Basic description of the person

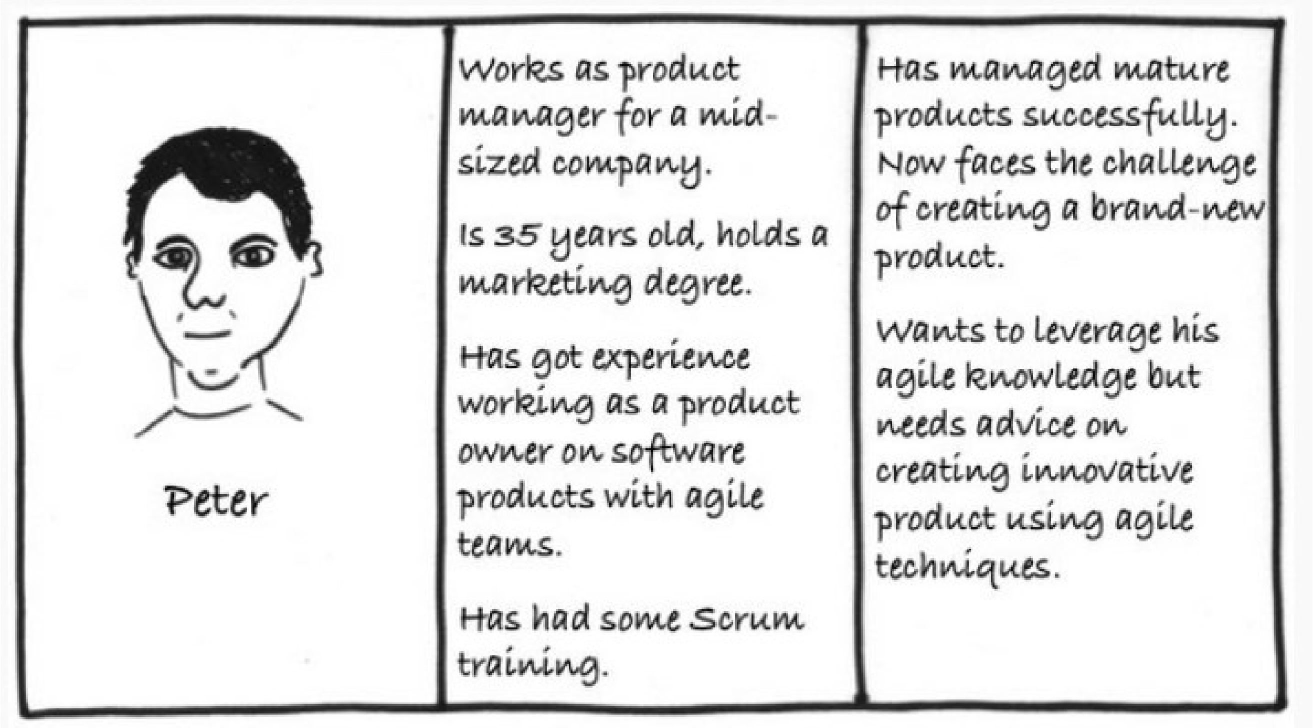
 Name	• Job • Experience • Activities • Attitude • Competencies • Age
• Problems • Pain points • Needs • Concerns • Fears • Desires	• Knowledge of VR • Dream VR system • Vision of VR • VR hardware access • Budget for VR • Activities that fit VR

Challenges the person has

Relation to type of system (VR in this case)

If personas are especially important (e.g., for therapy applications), then data should be carefully collected with interviews and/or questionnaires

Another example of a Persona



- A main difficulty of the persona method is getting the team members to use it
- The 10-step process of creating a persona can help

https://www.interaction-design.org/literature/article/personas-why-and-how-you-should-use-them

Personas: The Take Away

- Personas are fictional characters based on user research to help understand:

  - users' needs,
  - experiences,
  - behaviors
  - goals.

- Make the design task at hand less complex

- Guide the ideation processes, and help to achieve the goal of creating a good user experience for the target user group

- The 10-step process covers the entire process from the preliminary data collection, through active use, to continued development of personas.

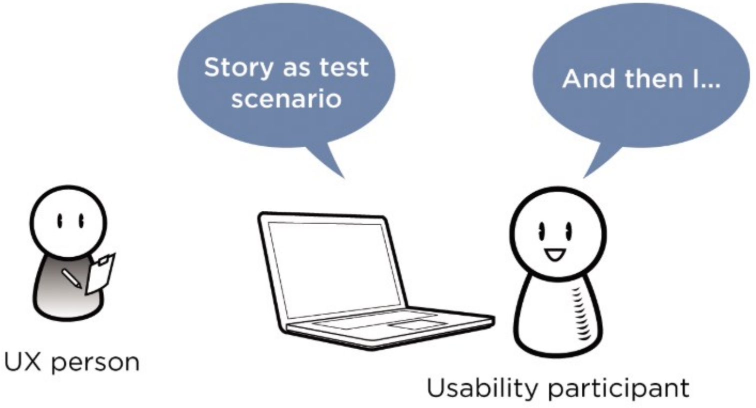
https://www.interaction-design.org/literature/article/personas-why-and-how-you-should-use-them

Scenarios

- Stories and contexts about how the user groups use a future product/service

- Note the goals and questions to be achieved and sometimes define the possibilities of how the user(s) can achieve them on the product/service

- Scenarios are critical for
  - designing
  - usability testing



https://www.usability.gov/how-to-and-tools/methods/scenarios.html

https://www.interaction-design.org/literature/topics/user-scenarios

- Scenarios should be used in the ideation phase of a project
- Scenarios need to be based on research with users
- Do ***not*** represent ***all*** possible users
- Typically account for the ***most common*** users or user motivations
- Are commonly based on personas
- Can be used to determine the most important areas to test during usability testing, and to provide guidance to the test

https://www.interaction-design.org/literature/topics/user-scenarios

What to Consider When Writing Scenarios

- Good scenarios are concise but answer the following questions:

  - **Who is the user?** Use the personas

  - **Why does the user uses the product?** Note what motivates the user and their expectations, if any

  - **What goals does s/he have**? Use task analysis

  - **How can the user achieve their goals with the product?**

https://www.usability.gov/how-to-and-tools/methods/scenarios.html

Types of Scenarios

- **Goal/Task-based Scenarios** state only what the user wants to do

Example: You are traveling to Paris for your job next week and you want to check on the amount you can be reimbursed for meals and other expenses

- **Elaborated Scenarios** give more user story details

- **Full Scale Task Scenarios** include the steps to accomplish the task

https://www.usability.gov/how-to-and-tools/methods/scenarios.html

Scenarios: The Take Away

- User scenarios are a great way of communicating the key tasks a user will perform with a system

- They can also help define the usability testing regime

- To create user scenarios is a simple process and should be used for developing and iterating interactive products

https://www.interaction-design.org/literature/topics/user-scenarios

Task Analysis

- The process of learning about ordinary users by observing them in action to understand in detail how they perform their tasks and achieve their intended goals.

- Helps identify the tasks that product/service must support

- Helps support other aspects of the user-centered design process

- It is important to perform a task analysis early in your process, in particular prior to design work

https://www.usability.gov/how-to-and-tools/methods/task-analysis.html

Types and benefits of Task Analysis

- Types
  - Hierarchical task analysis
  - Cognitive task analysis

- Help support several other aspects of the user-centered design process, including:
  - Requirements gathering
  - Developing structure
  - Prototyping
  - Usability testing

https://www.usability.gov/how-to-and-tools/methods/task-analysis.html

- **It is also useful for** understanding:

  - Users’ goals and what they are trying to achieve
  - The steps that users currently take to achieve their goals
  - The personal, social and cultural experiences that users bring to the tasks
  - The influence of the physical environment on the users while attempting to meet a goal

https://www.interaction-design.org/literature/article/task-analysis-a-ux-designer-s-best-friend

- **Task analysis may be performed**:

  - in a more formal way (e.g. Hierarchical Task Analysis)

    or

  - in a **more informal way**:

  - using several different methods

- First **use the 11 questions (at least 1, 2, 3 and 5)**

- Then **decompose the main tasks**

Standard/Informal Questions to be answered

1. **Who is going to use the system?**
2. **What tasks do they now perform?**
3. **What new tasks are desired?**
4. How are the tasks learned?
5. **Where are the tasks performed?**
6. What is the relationship between customer and data?
7. What other tools does the user have?
8. How do users communicate with each other?
9. How often are the tasks performed?
10. What are the time constraints on the task?
11. What happens when things go wrong?

Minimum set of questions to be answered

1. Who is going to use the system?

- Use **all the information obtained previously about the users** (e.g. to develop the personas), concerning:

age, needs, motivations, background, experience, technology literacy, physical characteristics…

2. What tasks do they now perform?

- Identify the **tasks that users perform currently**, without using the system under development, including:

relative importance, frequency of performing the tasks, if they are performed by one or more users, …

3. What new tasks are desired?

- Identify new tasks that might empower the users taking advantage of the new way of performing the tasks
- Be careful and prioritize the new tasks to support …

5. Where are the tasks performed?

- Observe the environment where users currently perform the tasks

- Identify other activities, the type of space (office, shop floor, hospital, class room, shopping mall, …), noise, light and dust conditions, stress level, …

How to conduct a Task Analysis

- Decompose a high-level task into the following steps:

  - Identify the task to be analyzed
  - Break this high-level task down into 4 to 8 subtasks
  - Draw a layered task diagram of each subtasks
  - Produce a written account as well
  - Present the analysis to someone else who knows the tasks

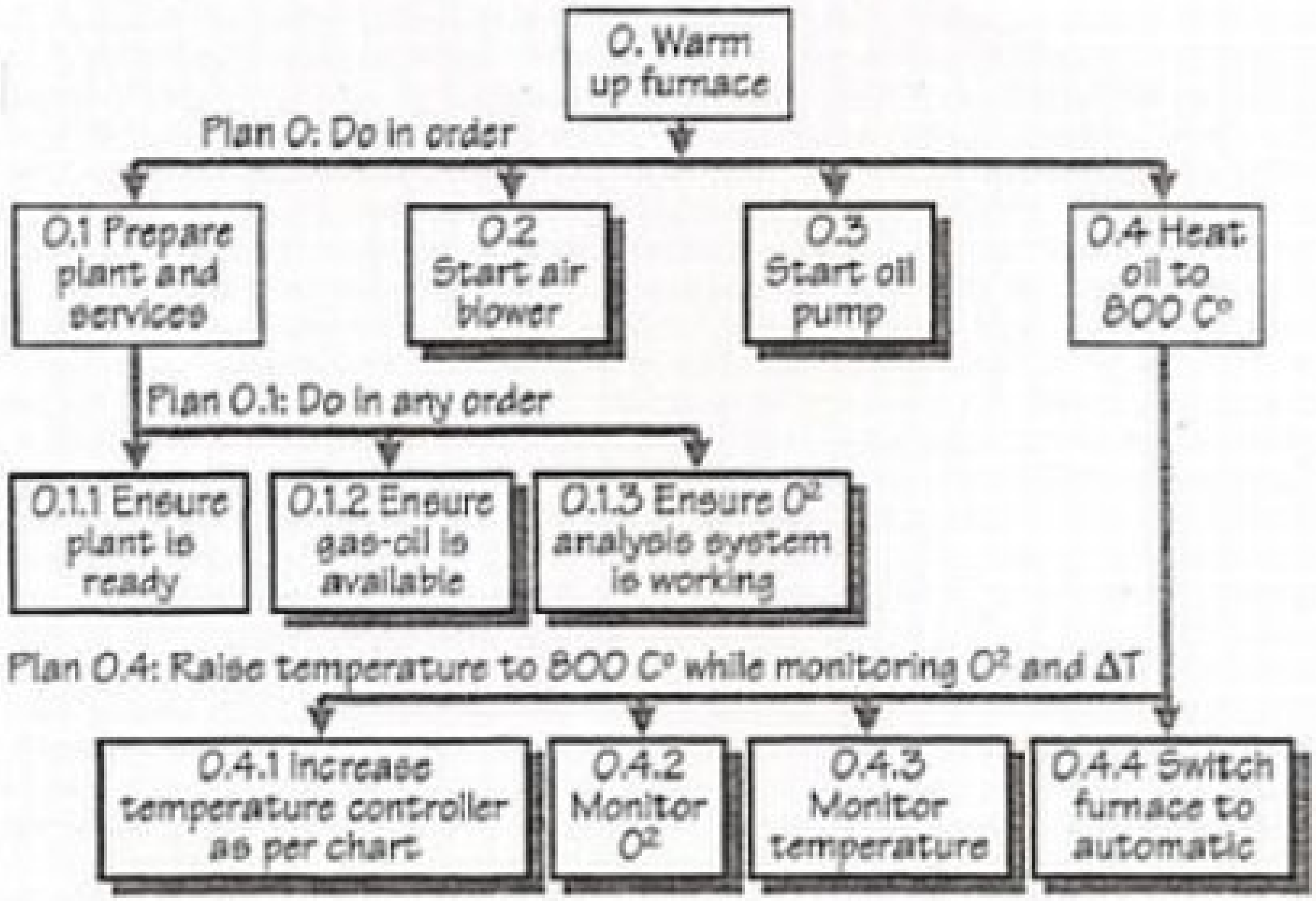
- The decomposition level of detail should be coherent across subtasks

http://www.usabilitybok.org/task-analysis

Example of a Hierarchical Task Analysis

(it will be addressed later in the semester)

Diagram for the goal: warm up a furnace



https://www.interaction-design.org/literature/article/task-analysis-a-ux-designer-s-best-friend

Task Analysis: The Take Away

- Is one of the most powerful tools in UX design
- It is not hard to get to know how to do it
- The difficult part is remembering to keep the user's perspective
- It is useless when it is not backed by rigorous user research
- is not a one-off process; can be repeated later in the process
- It requires time, resources, people and budget. Be sure to have a sufficient amount of all
- Like any other activity in UX design!

https://www.interaction-design.org/literature/article/task-analysis-a-ux-designer-s-best-friend

User Stories

Emerged from agile development methods, are **short concepts or descriptions of features customers would like to see**

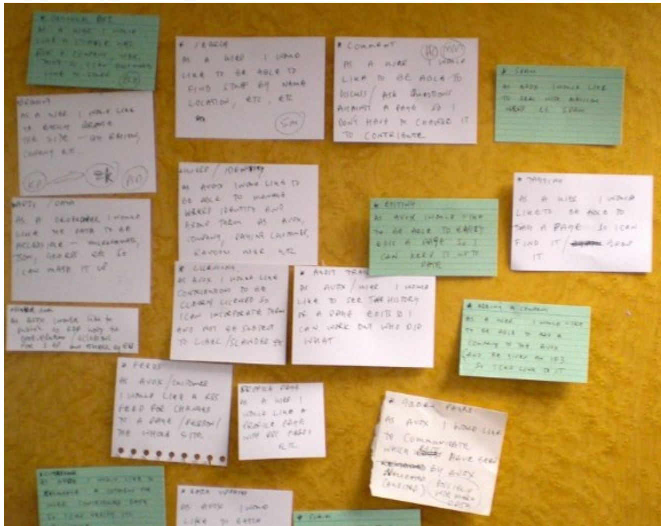
Should:

not go into too much detail

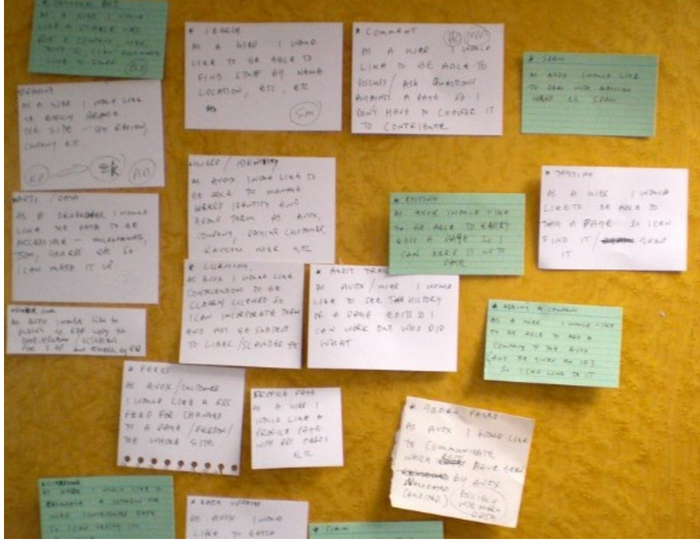
be written:

- from the **user's point of view**

- **with the client** and team members



User Stories



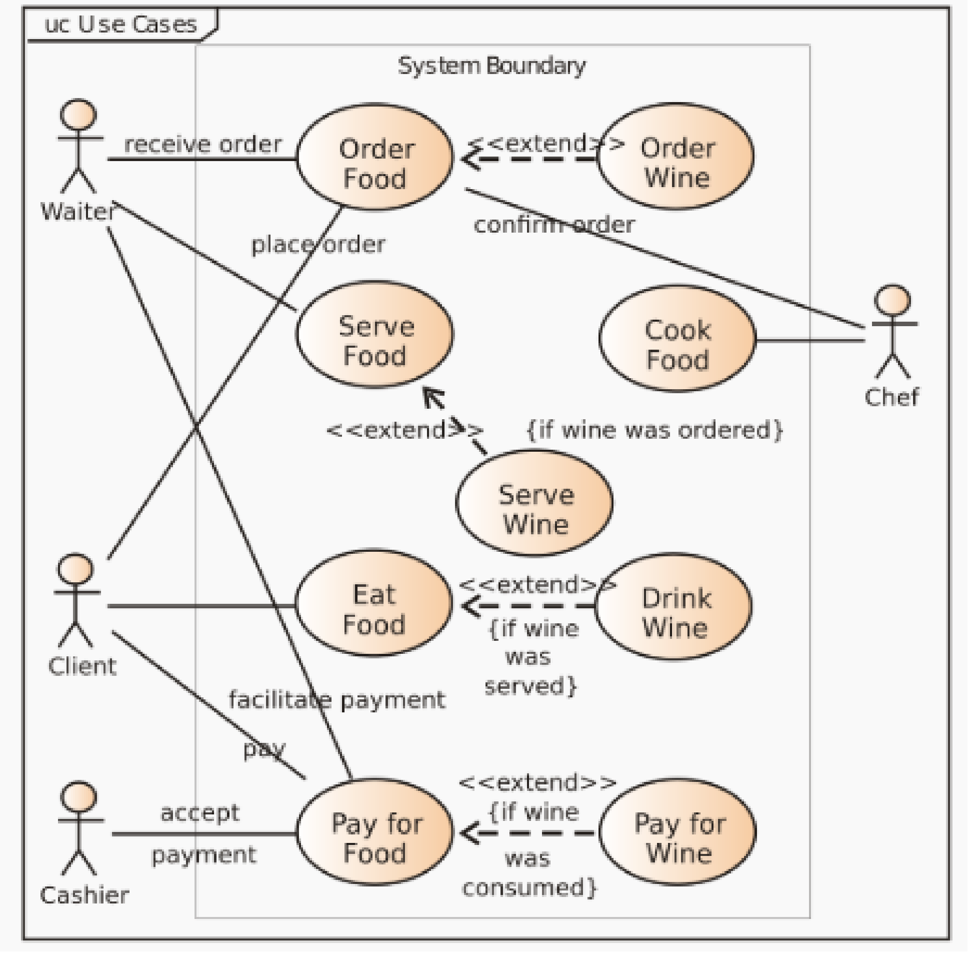
“As a <type of user> I want <some goal> so that <some reason>.”

Should be written with the **minimum amount of detail** necessary to fully **encapsulate the value that the feature** is meant to deliver

https://manifesto.co.uk/how-much-detail-should-a-user-story-have/
https://www.interaction-design.org/literature/topics/user-stories
https://www.interaction-design.org/literature/article/user-stories-capturing-the-user-s-perspective-quickly-and-simply

Use Cases

A common way for developers to explain user scenarios using UML



https://www.interaction-design.org/literature/topics/user-scenarios

Scenarios, User stories and Use cases

- Scenarios are created by user researchers to help communicate with the design team
- User stories are created by project/product managers to define the requirements prior to a sprint in agile development
- Use cases are created for developers to help with testing
- The difference in target audience means that the structure and information contained in the three approaches also varies.

Example

Scenario

“Jim, a second year internal medicine intern at Mount Pleasant Hospital, walks into the room of his patient, Andrew Ross. Since Andrew stayed the night in the hospital, Jim needs to review Andrew’s medical records to see if the nurses on the night shift had checked in and recorded any changes in Andrew’s condition.”

User Story

As a doctor, I need to get up to medical date records so that I know how to proceed with my patients’ treatment

(it does not reflect the context of use)

Use case: *Review Records*

Actor: *Doctor*

Steps:

Doctor walks into room

Doctor sees patient in bed

Doctor identifies patient in bed

Doctor sees medical charts on foot of bed

Doctor gets medical charts from foot of bed

Doctor opens medical charts

Doctor reads medical charts

Doctor changes pages to continue reading

Doctor closes medical chart

https://www.akendi.com/blog/scenarios-user-stories-and-use-casesoh-my/

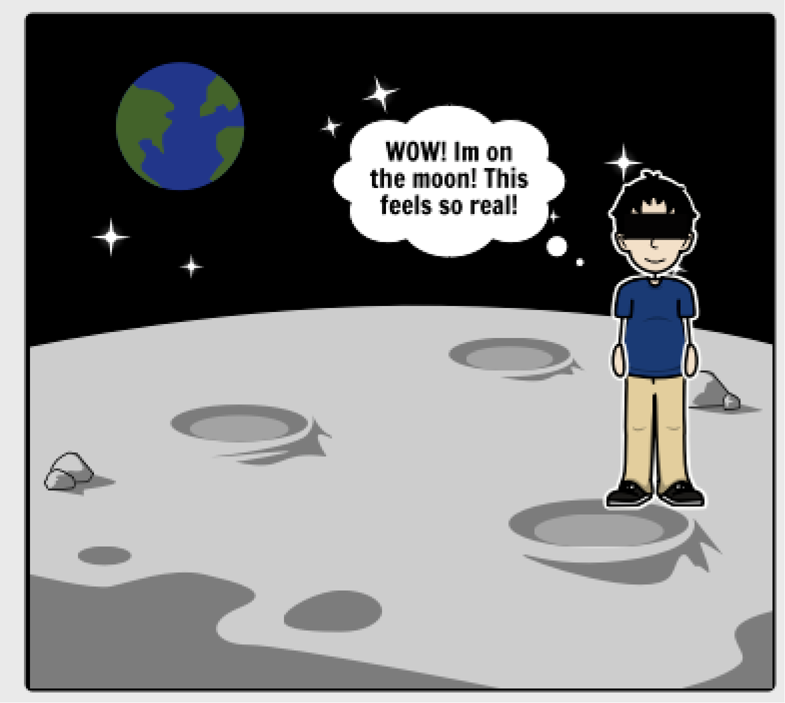
Storyboards

Are early visual forms of an experience

Derived from the film industry

Particularly useful for VR

The user can be shown directly interacting with objects



https://www.interaction-design.org/literature/topics/storyboards

https://www.storyboardthat.com

2nd practical assignment – next steps

Lab 5 - Presentation of the analytical evaluation assignment

Limit for selection of the 2nd assignment topic

2nd assignment - Development of an interactive application

2nd Assignment - topic selection

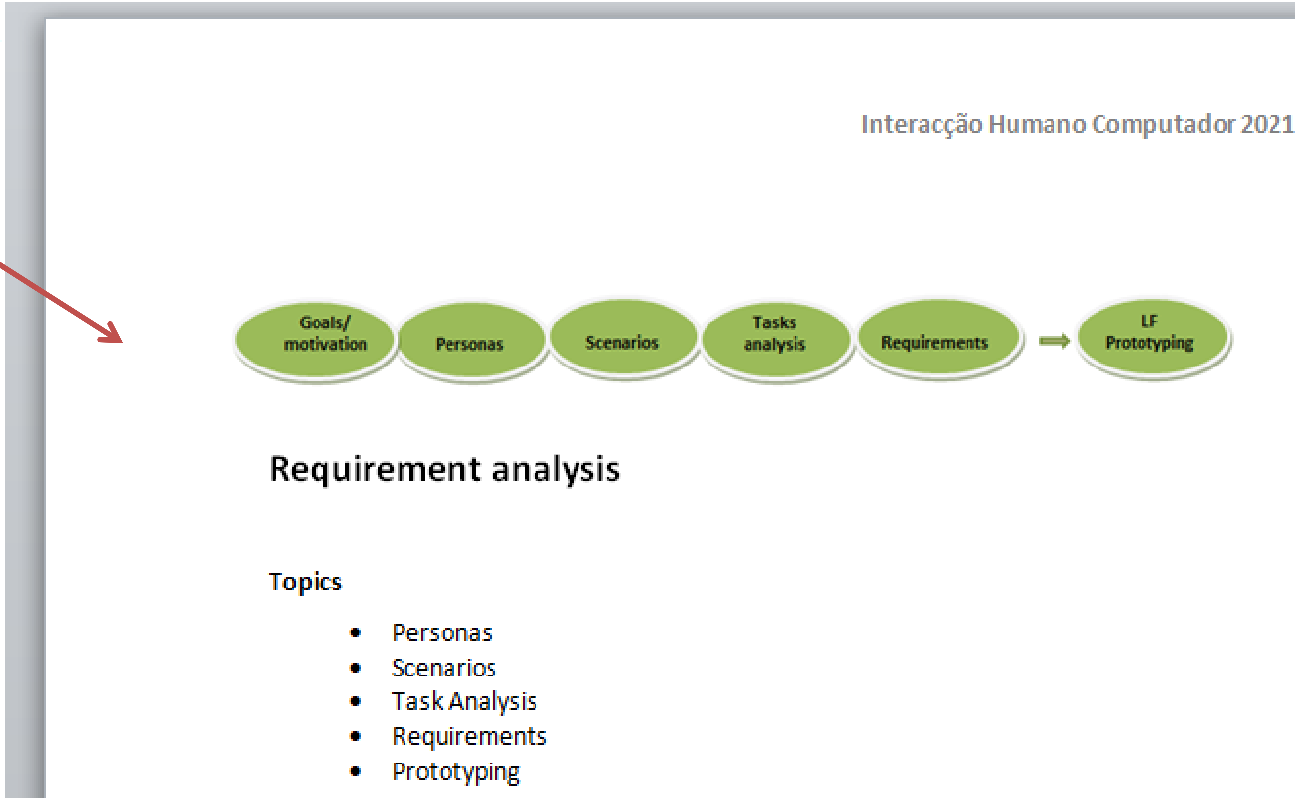
Examples of mini-projects 2020

Lab 6 - User Centered Design (UCD). 2nd Assignment

- Introduction to personas and scenarios and task analysis

- Introduction to the technologies to be used

How to do a requirement analysis



Interacção Humano Computador 2021

Requirement analysis

Topics

- Personas
- Scenarios
- Task Analysis
- Requirements
- Prototyping

Main bibliography

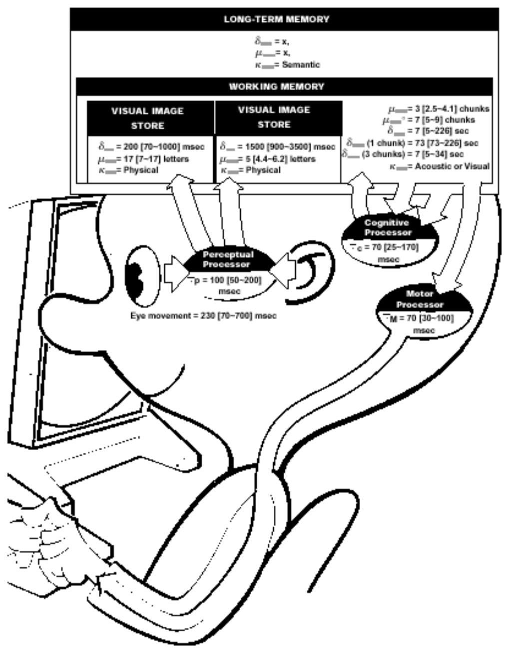
- Cooper, A., R. Reimann, and D. Cronin, *About Face 3 The Essentials of Interaction Design*. Wiley Publishing, Inc., 2007
- Mayhew, D., *The Usability Engineering Lifecycle*, Morgan Kaufmann, 1999
- Norman, D., *The Design of Everyday things, Revised and Extended Edition,* Basic Books, 2013
  https://books.google.pt/books/about/The_Design_of_Everyday_Things.html?id=I1o4DgAAQBAJ&printsec=frontcover&source=kp_read_button&redir_esc=y#v=onepage&q&f=false
- Sharp, H., Y. Rogers, *Interaction Design : Beyond Human-Computer Interaction*, 5th ed., Wiley, 2019
  https://books.google.pt/books/about/Interaction_Design.html?id=HreODwAAQBAJ&redir_esc=y
- Sommerville,, I., *Software Engineering*, 9th ed., Addison Wesley, 2010
  http://ifs.host.cs.st-andrews.ac.uk/Books/SE9/WebChapters/PDF/Ch_29%20Interaction_design.pdf

Interesting Links

- https://www.nngroup.com/
- http://www.usability.gov/;
- https://www.usability.gov/how-to-and-tools/index.html
- http://uxpa.org/
- https://www.w3.org/WAI/redesign/ucd

Universidade de Aveiro
Departamento de Electrónica,
Telecomunicações e Informática

The User



www.id-book.com

Outline

- Users Profile – relevant characteristics for interactive systems

- Human Information Processing System (HIPS)
  - Perceptual sub system
  - Senses
    - Sight, Hearing, Touch, Smell, Taste
      + Proprioception, Kinesthesia …

  - Cognitive sub-system
  - Memory

- Some implications on the design of interactive systems

Users profile

- Human Information Processing System (HIPS)

- Knowledge and experience

- Work and task

- Physical characteristics

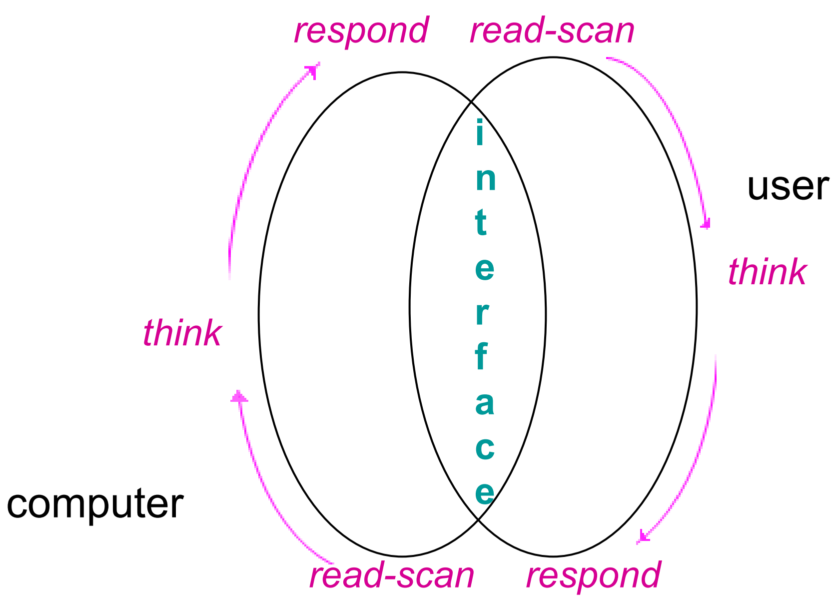
- Environment

- Tools

More variable among users

There are many user models to be used in the design of Interactive systems (e.g. personas, GOMS, KLM, …)

Dialog in an interactive system



Perceptual sub-system

Cognitive sub-system

Motor sub-system

Human Information Processing System (HIPS)

- Humans have different capabilities that might be considered when designing interactive systems
- Information is received through various I/O channels
- Information is stored in memory
- Emotions may influence capabilities
- Users share common characteristics but differences that cannot be ignored

Human Information Processing System (HIPS): main aspects relevant to interactive systems design

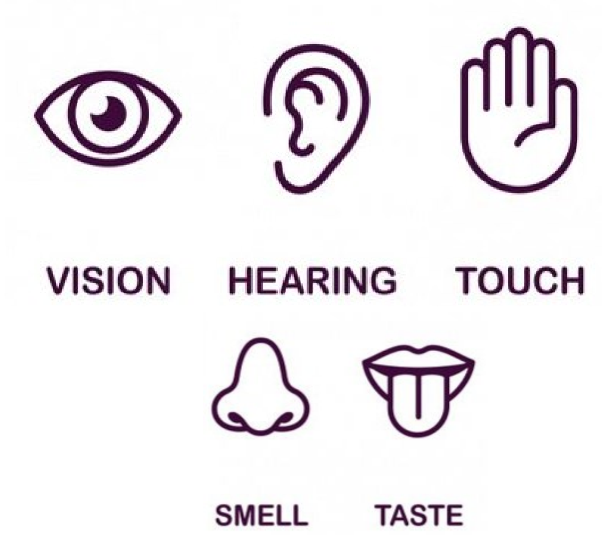
Perceptual sub-system
- memory – perceptual buffer (iconic, echoic, …)
- process – pattern recognition

Cognitive sub-system
- memory -
  - short term/working memory (STM)
  - long term memory (LTM)
- processes -
  - selective attention
  - problem resolution
  - learning
  - …

HIPS bottleneck

Motor sub-system

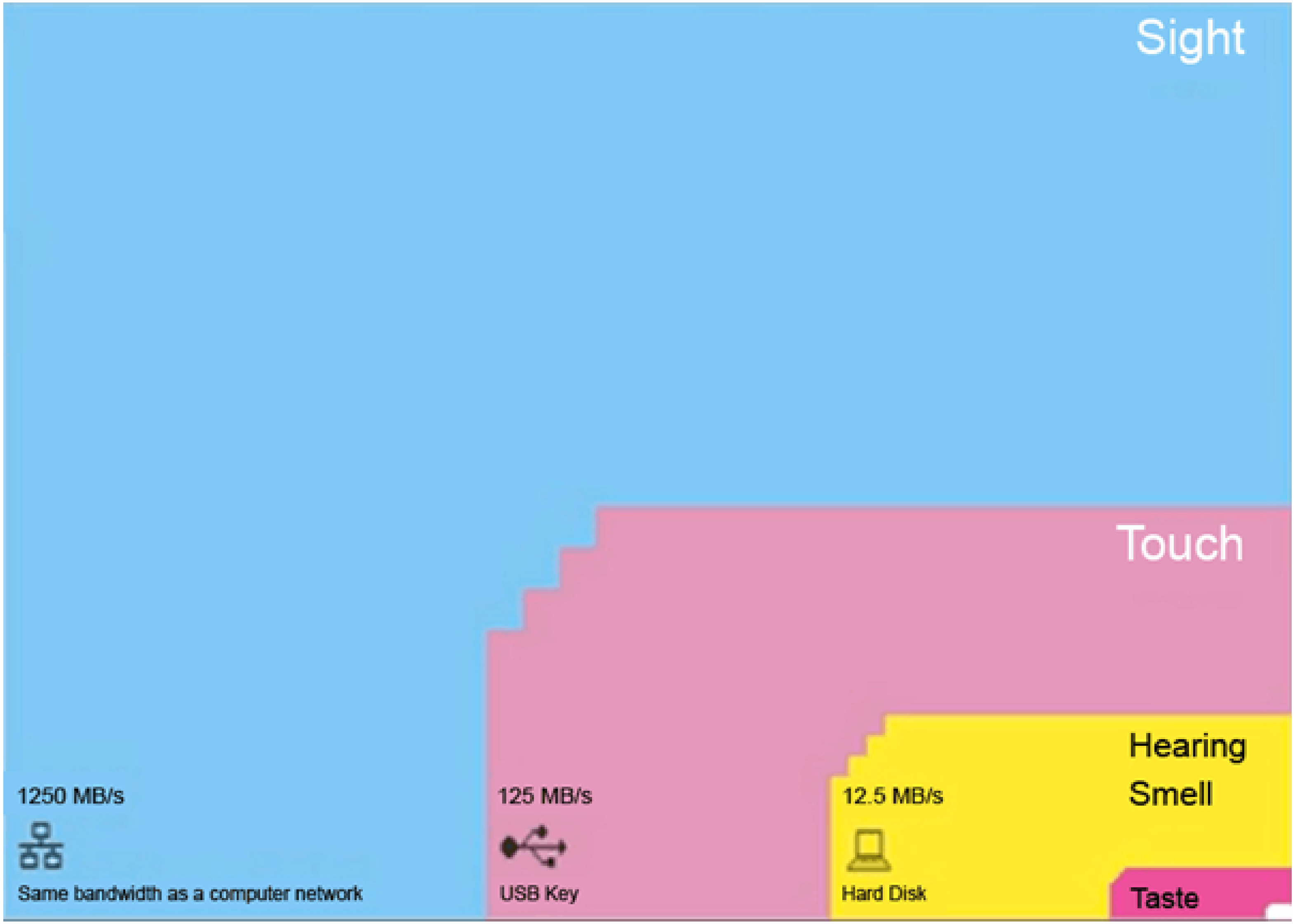
Perceptual sub-system – I/O



The five Aristotelian senses

- Input: 5+ senses
  Sense is a physiological capacity of organisms that provides data for perception
  - Some more relevant than other
  - For HCI, vision is preferred, but hearing and touch are more and more important …
- Output: communication system
  - vocal, gestures, eye gaze, ,…

Why vision is preferred

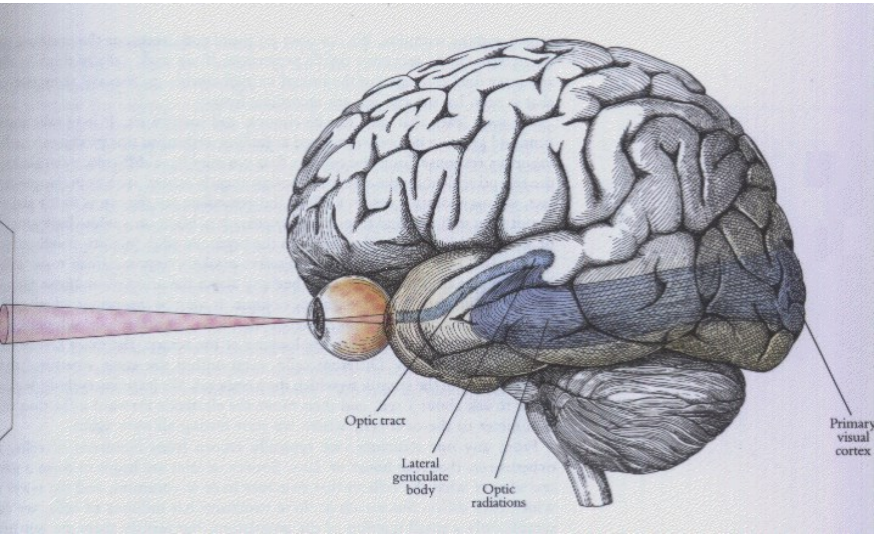


0.7% we are aware of

Bandwidth of our senses proposed by Danish Physicist - Tor Nørretranders

Vision

- Relevant for HCI:
  - Compensation of movements and illumination changes
  - Context used to solve ambiguities
  - May be tricked: Visual illusions come from excessive compensation.

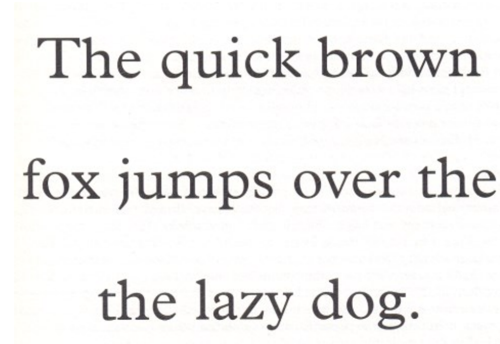


Eye – sensor

Brain - processor

(Hubel, 1988)

Pattern Recognition



Process that matches information from a stimulus with information retrieved from memory

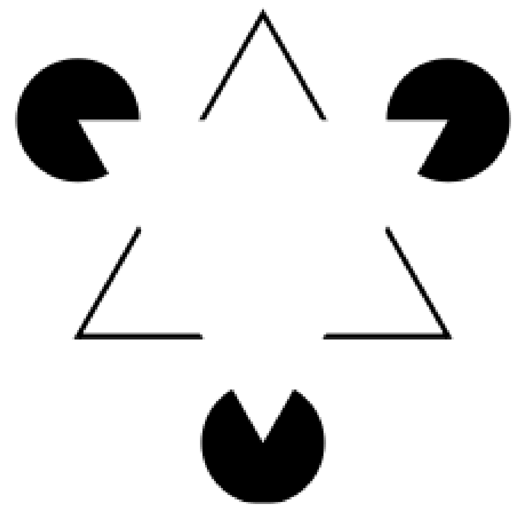
- It is a very powerful process

- It is subconscious

- It does not use only current data

- It solves ambiguities

- Occurs also in other senses

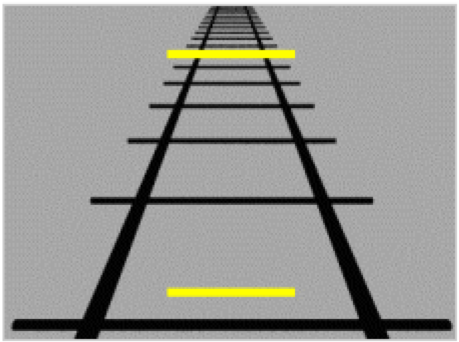


(Kaniza illusion, Wikepedia)

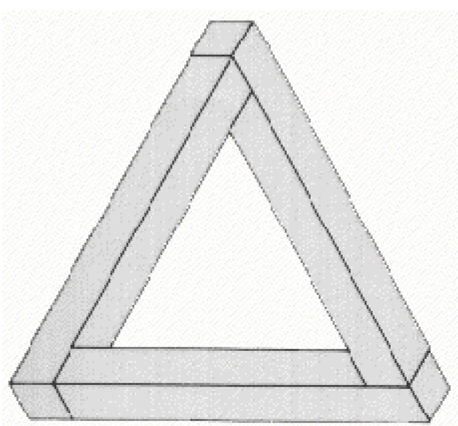
Visual Illusions illustrate that **what we see does not depend only of the stimulus**

- bring out good adaptations of our visual system to standard viewing situations

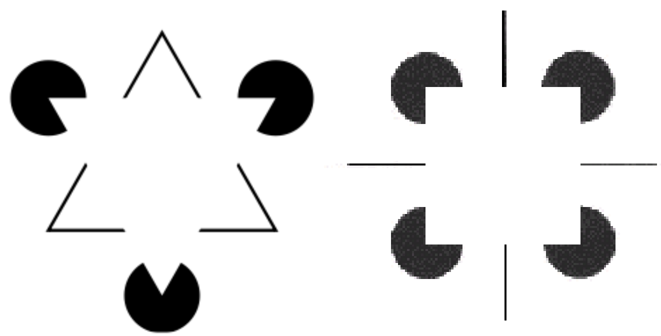
- under artificial manipulations can cause inappropriate interpretations of the scene



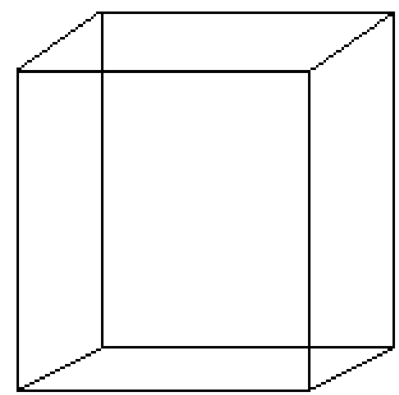
Ponzo illusion



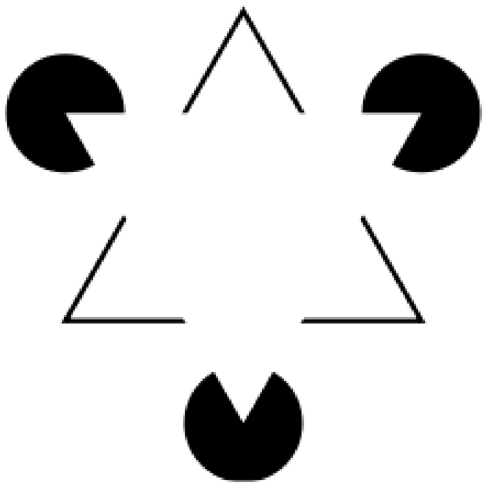
Pennrose triangle:
Impossible object?



Kanizsa illusion



Necker cube



Kanizsa illusion:

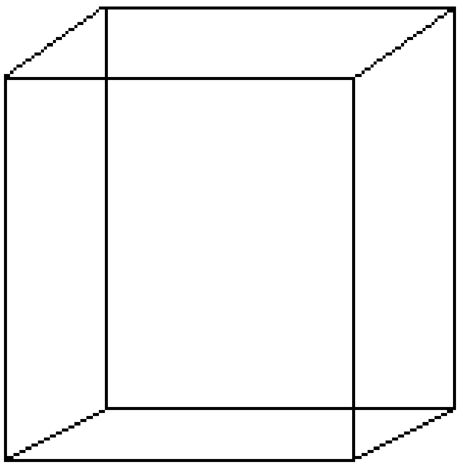
Although there are no actual triangles a sort of pattern recognition phenomenon is triggered and the image is interpreted as two overlapping triangles (simple explanation)

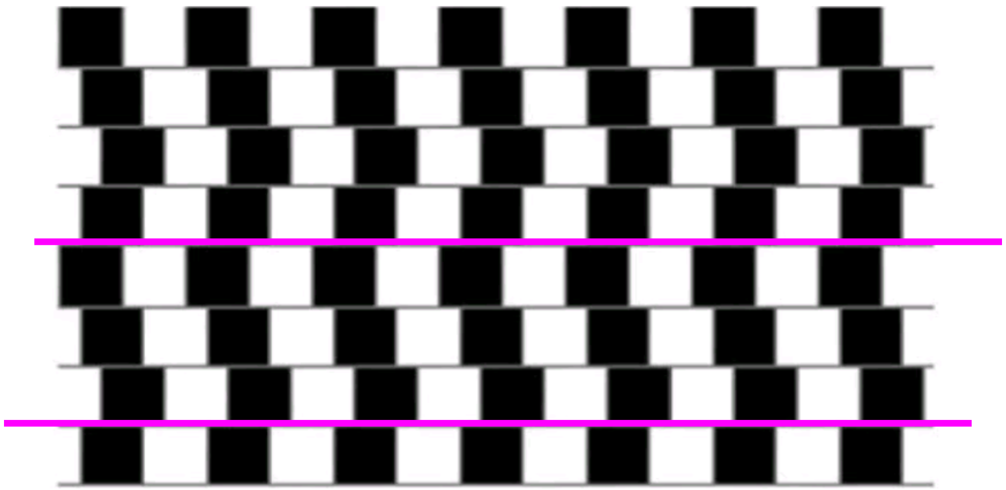
https://en.wikipedia.org/wiki/Illusory_contours

Necker cube:

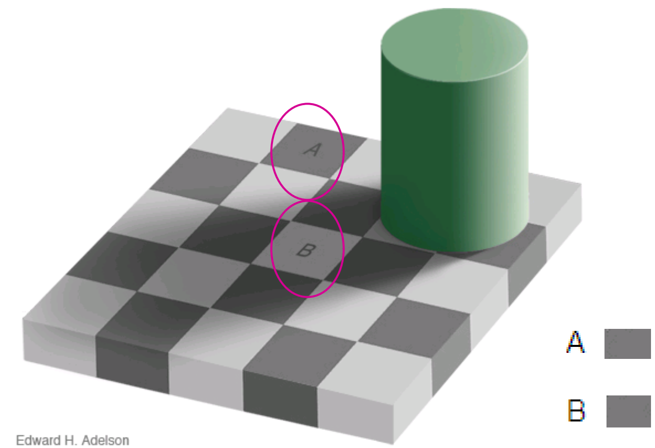
Cube with no visual cues as to its orientation; it can be interpreted to have either the lower-left or the upper-right square as its front side

https://en.wikipedia.org/wiki/Necker_cube





http://www.youtube.com/watch?v=2XqPOJLUM1s



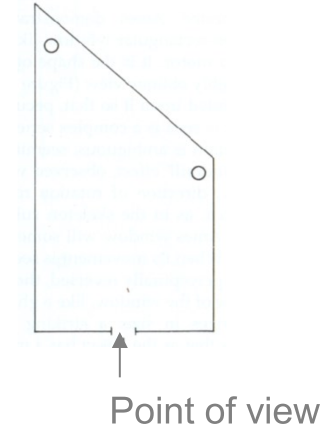
http://www.michaelbach.de/ot/

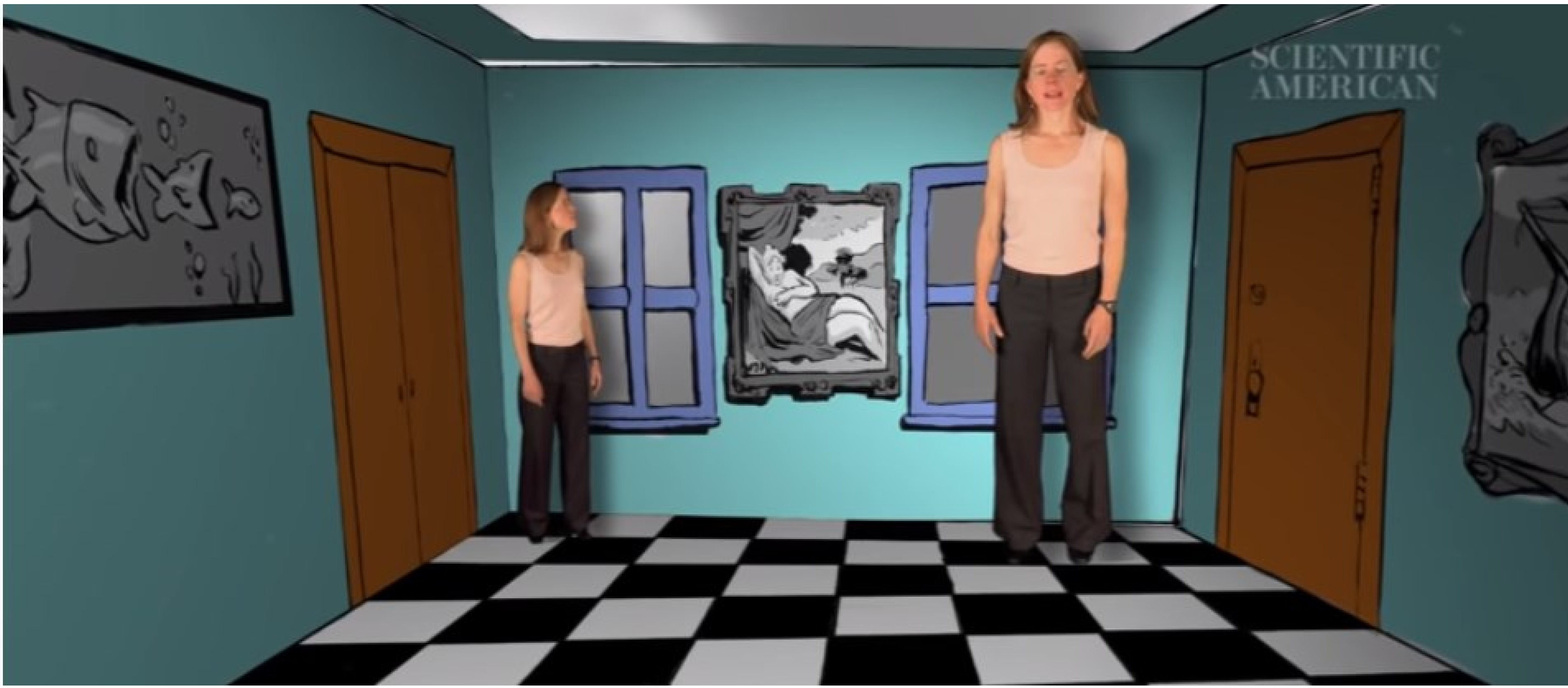
http://www.youtube.com/watch?v=URLRdcnU6Hk&NR=1

https://www.youtube.com/watch?v=-IWk5NkxQF8&ebc=ANyPxKoRFxfOSCgdPavBoMpgPrXjRRVqZmhiAvIBDgThnPfndq-gheNYZ-6cNRv2yYwN5SqX52DQGDWjvBnzUQQ-_N6iCVgMdQ

Ames Room
(what we see does not depend only of the stimulus)

A room that pushes the boundaries of human perception...





https://www.youtube.com/watch?v=gJhyu6nIGt8

https://www.youtube.com/watch?v=aS-vzPuZzuk

Other senses

https://en.wikipedia.org/wiki/Sense

- Hearing
  - Information on direction, objects and distance
  - Only sense that is really 3D
  - Cannot be “turned off”
  - Human hearing - 20Hz to 15KHz
  - Filtering is possible (Background noise – “cocktail party” example)

https://en.wikipedia.org/wiki/Sense

Other senses

- Touch
  - Important feedback
  - Key senses for people with sight problem
  - Several receptors in skin:
    - Termoreceptors: cold and hot
    - Nociceptor: pain
    - Mecanoreceptor: pressure
  - Some areas more sensitive (fingers)

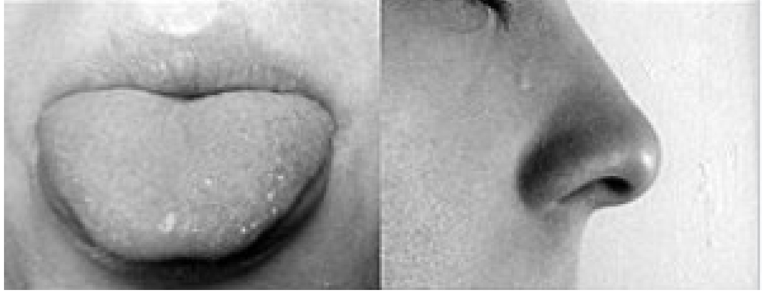
Simulators are complex interactive systems that stimulate several senses…

https://surgicalscience.com/systems/lapsim/

https://www.the737experience.co.uk/

https://en.wikipedia.org/wiki/Flight_simulator

Other senses



https://en.wikipedia.org/wiki/Sense

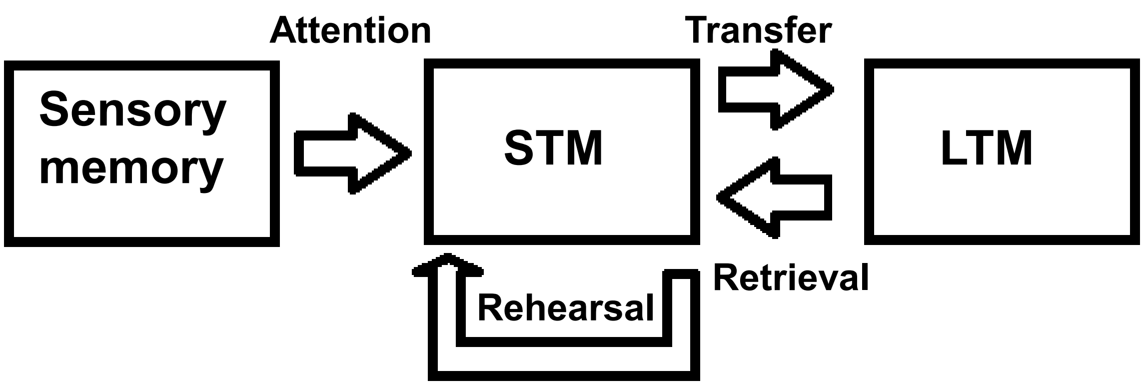
- Smell and Taste
  - Complex chemical senses
  - High latency
  - Difficult to use in HCI
  - Some experimental work exists
- And others (as proprioception - awareness of your body position)

https://www.khanacademy.org/test-prep/mcat/processing-the-environment/somatosensation/v/proprioception-kinesthesia

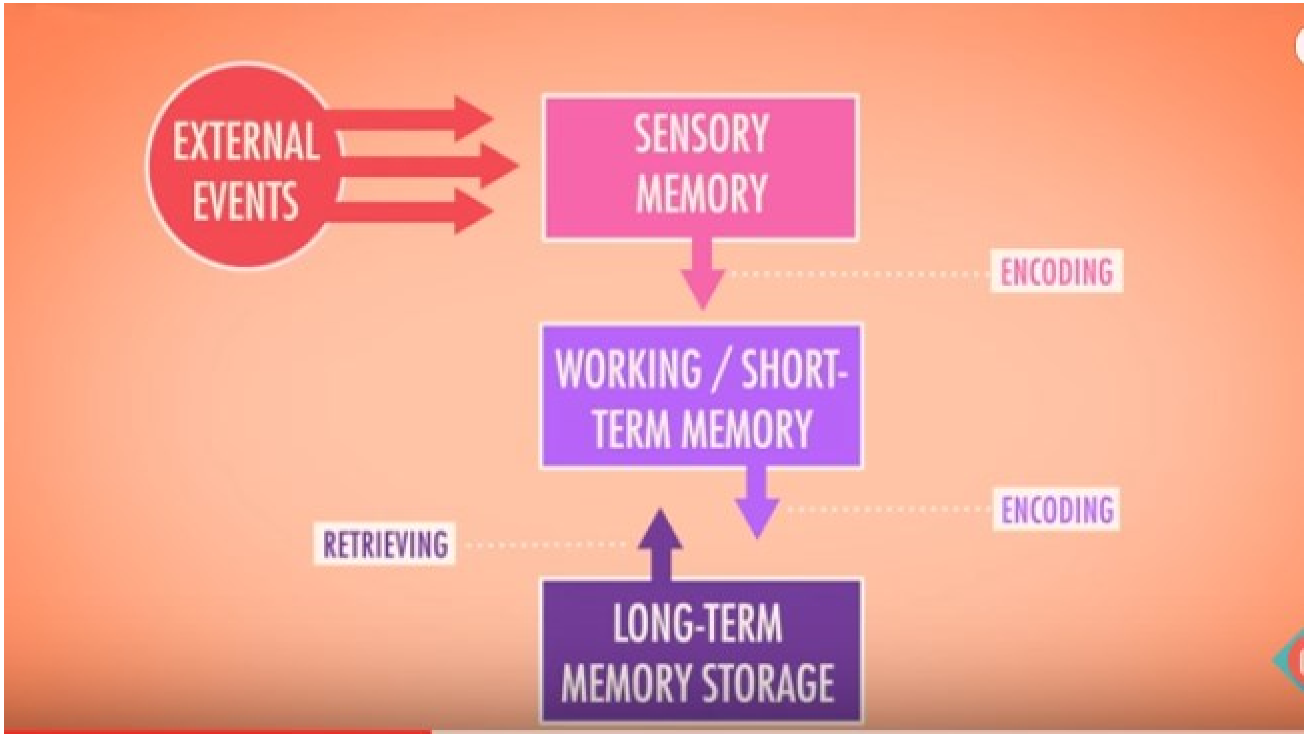
Memory

The **Atkinson–Shiffrin model** (1968) (a.k.a. multi-store model or modal model) asserts that human memory has three components:

- Sensory memory
- Short-term memory / working memory
- Long-term memory



https://en.wikipedia.org/wiki/Atkinson%E2%80%93Shiffrin_memory_model



How we make memories - Crash Course Psychology #13

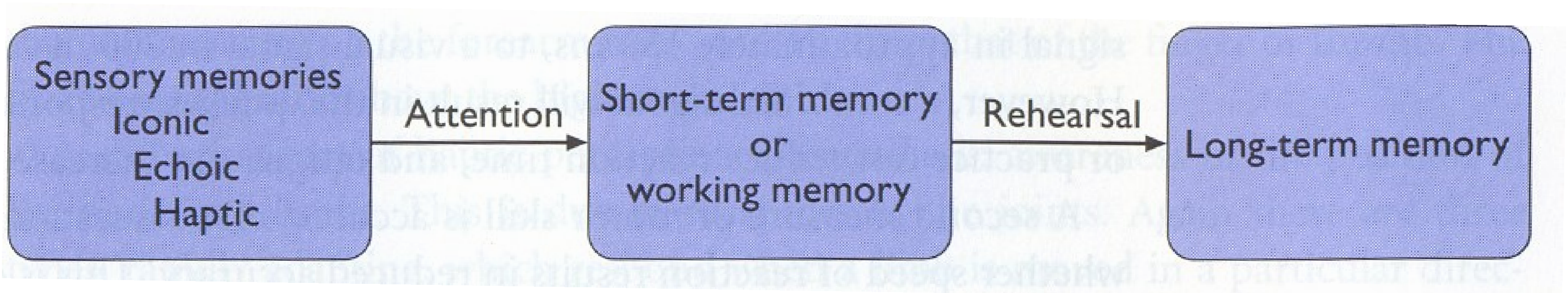
https://www.youtube.com/watch?v=bSycdIx-C48

How we remember and forget - Crash Course Psychology #14

https://www.youtube.com/watch?v=HVWbrNls-Kw

Memory

- The Atkinson–Shiffrin model (1968) memory has three components
  - Sensory memory / iconic memory, very short
    - A few seconds
  - Short-term memory / working memory
    - +/- 18 seconds, 7+/-2 items
  - Long-term memory
    - ~Infinite capacity

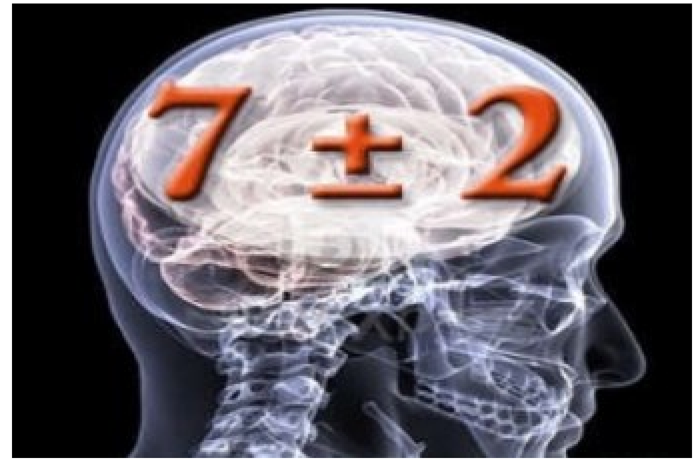


Short Term Memory (STM) / Working memory

- Working memory:
  - Mental calculus (6*35)
  - Reading (understand a sentence)
  - …
- Characteristics:
  - Quick access +/- 70ms
  - Quick forget
  - Limited capacity 7 +/- 2

  - Can be increased with chunking, to enter long-term memory

Short Term Memory (STM) / Working memory



a few characteristics:

- Short duration: a few seconds (<30s)
- Limited capacity: 7±2 elements

what is an element?

example: try to memorize the following numbers 649325401741
111122223333

which is easier?

and this one: 351234370517

Short Term/ working Memory (STM)

649325401741 $\longrightarrow$ 12 digits

111122223333 $\longrightarrow$ 3 digits and a rule

234370200 351 234 370 517

Portugal Aveiro University area IEETA building

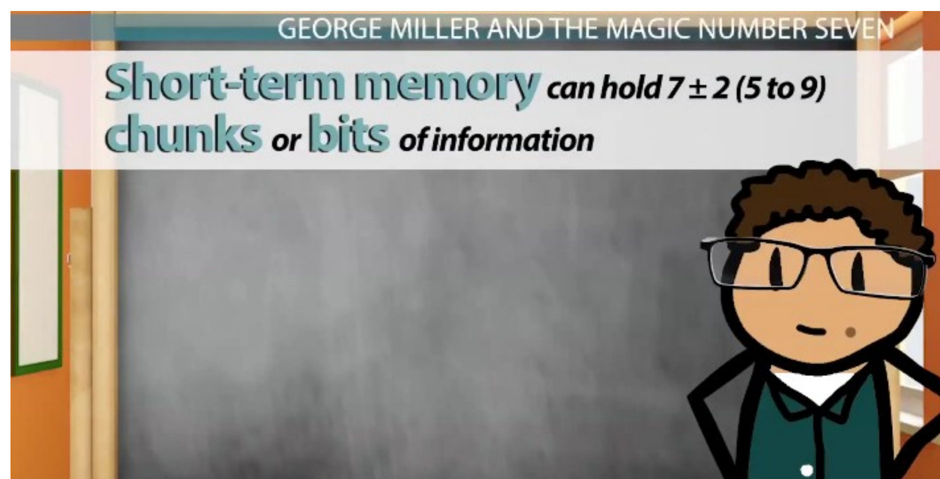
These numbers correspond to different “chunks”

Chunk: the largest meaningful unit that a person recognizes; depends on the person knowledge

- Chunking refers to an approach for making more efficient use of short-term memory by grouping information

- Resulting chunks are easier to commit to memory than a longer uninterrupted string of information.

- Can be used for making more efficient use of short-term memory by grouping information



http://study.com/academy/lesson/george-miller-psychologist-theories-on-short-term-memory-lesson-quiz.html

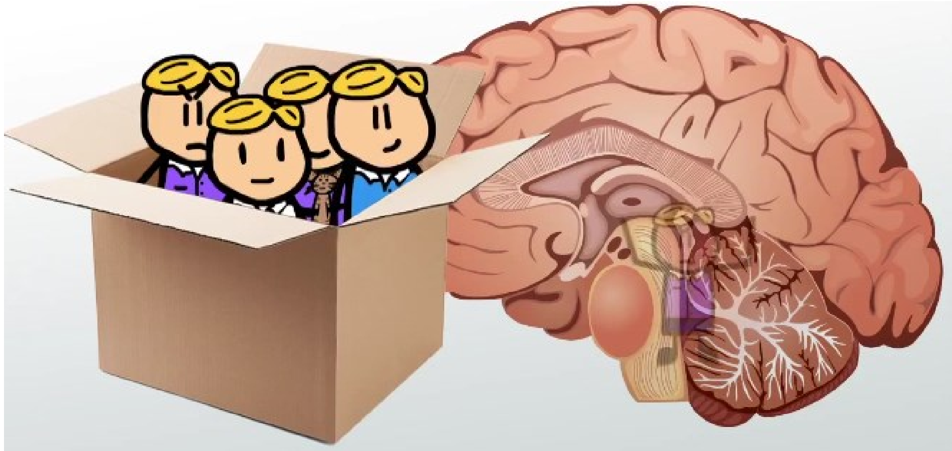
Long Term Memory

Stage of the dual memory model (Atkinson-Shiffrin memory model) informative knowledge can be stored for long periods of time

Main characteristics:

- "Infinite" capacity and duration

- non reliable access



http://study.com/academy/lesson/long-term-memory-definition-types-examples.html

Long-term Memory - Recognition vs. Recall

- Information retrieval - Recognition vs. Recall
  - Recognition: remembering with the help of a visible cue
    - aka “Knowledge in the world”
  - Recall: remembering with no help
    - aka “Knowledge in the head”

- Recognition is much easier
  - so menus are more learnable than command languages

Design Implications

Memory

- Reduce cognitive load by avoiding long and complicated procedures for carrying out tasks.

- Design interfaces that promote recognition rather than recall by using familiar interaction patterns, menus, icons, and consistently placed objects.

- Provide users with a variety of ways of labeling digital information (e.g. files, emails, and images) to help them easily identify it again through the use of folders, categories, color, tagging, time stamping, and icons.

Selective attention

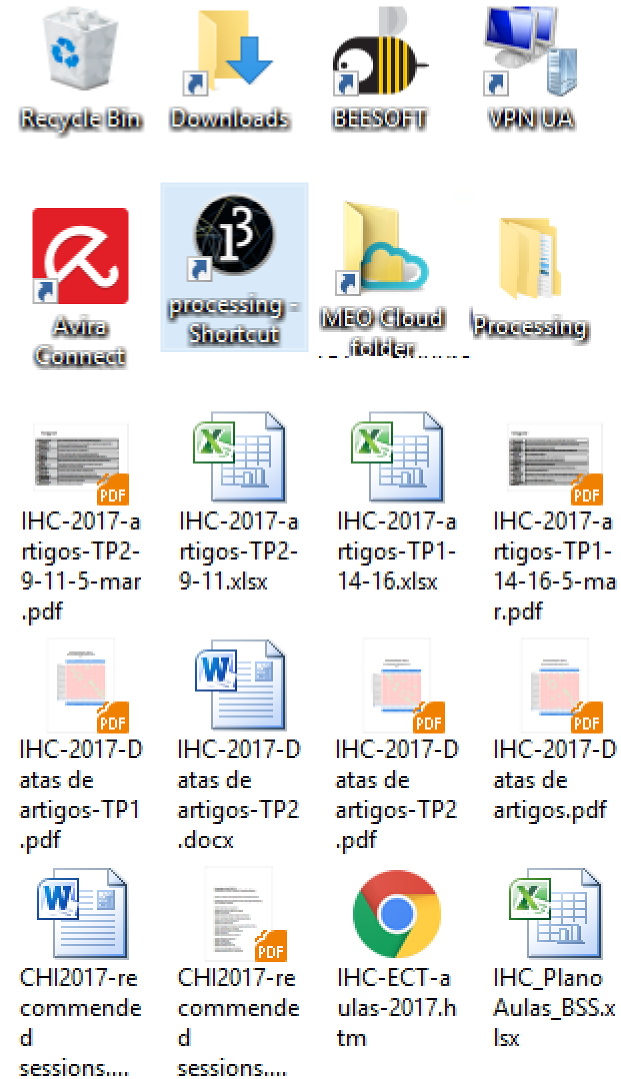
Occurs when we block out certain features of our environment and focus on one particular feature

It may be:

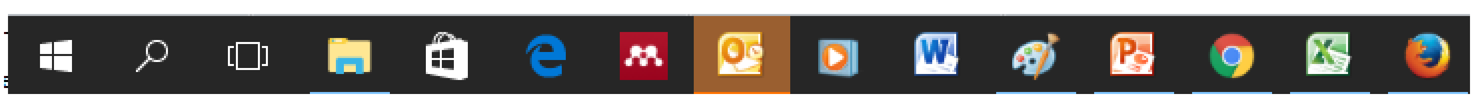
- Voluntary

- Involuntary

Both can be (and are) exploited in UIs



Calling your attention to an application



https://dictionary.apa.org/selective-attention
https://www.khanacademy.org/test-prep/mcat/processing-the-environment/attention-language/v/selective-attention

Design Implications

Attention

- Consider context. Make information salient when it requires attention at a given stage of a task.
- Avoid cluttering visual interfaces with too much information.
- Consider designing different ways of supporting effective switching and returning to a particular interface.

Design Implications

Learning

Design interfaces that encourage exploration

Design interfaces that constrain and guide users to select appropriate actions when initially learning.

Design Implications

Problem solving

- Provide information and help pages that are easy to access for people who want to understand more about how to carry out an activity more effectively (e.g., web searching).
- Use simple and memorable functions to support rapid decision-making and planning.
- Enable users to set or save their own criteria or preferences

HIPS Strengths and weaknesses (*versus* computer)

Strengths	Weakesses
LTM ~infinite capacity	STM limited capacity
LTM duration and complexity	STM limited duration
Capacity to learn	Error prone processing
Powerful selective attention	Non reliable access to LTM
Powerful pattern recognition process	Slow processing

Recommendation:
Assign tasks between user and computer according to the capacities of each

Example: minimize the users STM load
do not ask the user to perform computations

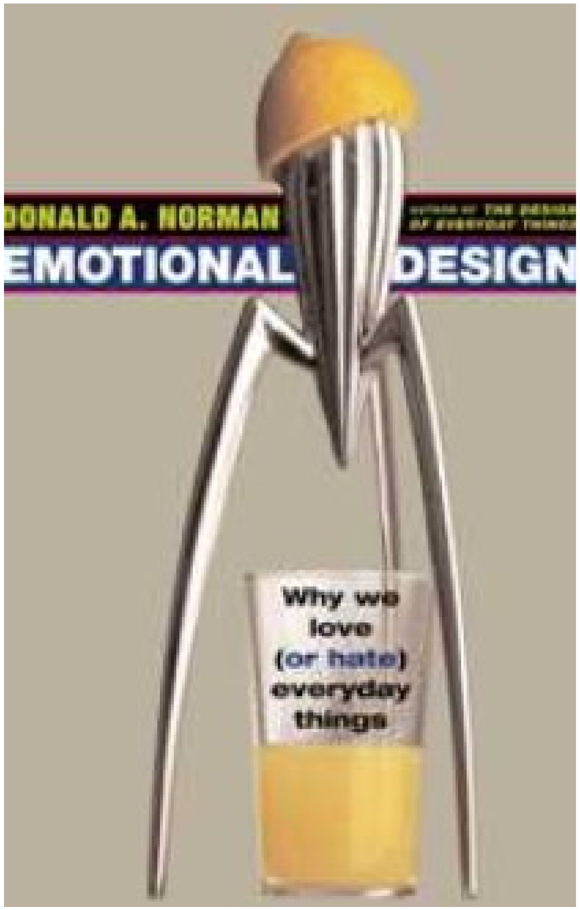
Emotion

- Various theories on how it works
- Involves cognitive and physical response to stimuli
- Biological response to a physical stimuli is called affect
- Affect Infuences how we react to situations

> “Negative affect can make it harder to do even easy tasks; positive affect can make it easier to do difficult tasks”
>
> (Donald Norman)

Not only the cognitive system is important

But also the **emotional system**



- Emotional design is a critical part of design

- Other Donald Norman videos:

  http://www.youtube.com/watch?v=WI2LkzIkacM

  http://www.youtube.com/watch?v=_PM3uqPNrWY

User Profile- other characteristics (besides HIPS)

- Experience and knowledge - education and reading level
  experience with the system and task
  mother language
  computer literacy …

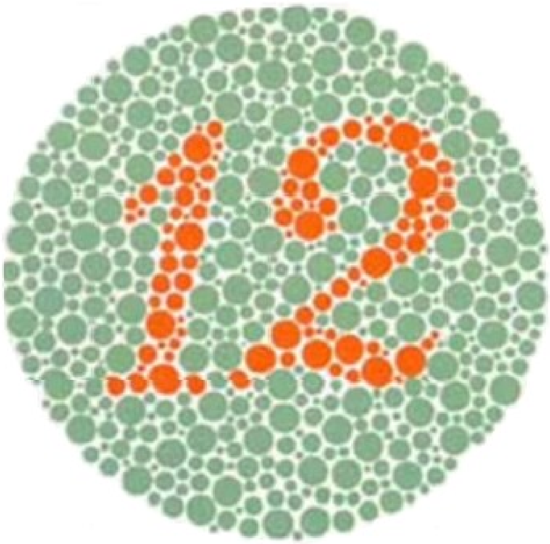
- Work and task - usage frequency
  training
  usage type (mandatory, optional)
  usage of other systems …

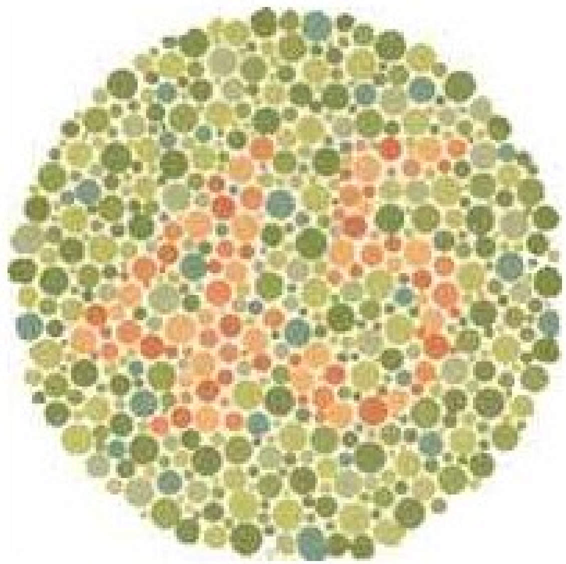
- Physical Characteristics – color vision deficiencies
  physical deficiencies
  handedness
  age …

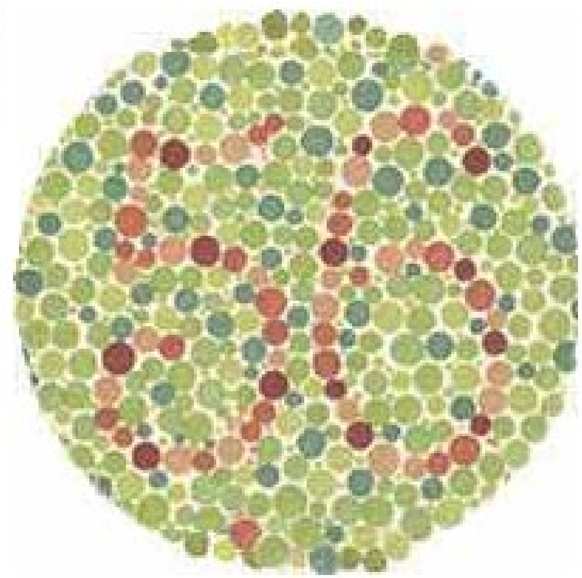
Cultural aspects!!…

Color blindness

Ishihara test for red-green blindness (daltonism or deuteranopia)







(Shinobu Ishihara,1917)

- The test includes 38 figures

How do these characteristics influence UI design?

- Much system experience, but low task experience -> more semantic help
- Much task experience but low system experience -> more syntactic help
- High usage frequency -> easy to use
- Low usage frequency -> easy to learn and remember
- Mandatory -> easy to use
- Optional -> easy to learn and remember
- Color (particularly red and green) should not be used as only cue to convey information
  - Etc., …

The take away:

- Users are much different from designers/developers

- Users vary a lot among thenselves

- Users change along time (evolve, forget...)

Final recommendation:

We should consider the users as an unknown species and study them scientifically

Main bibliography

- Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale, *Human-Computer Interaction*, 3rd edition, Prentice Hall, 2004

- Jenny Preece and Helen Sharp, *Interaction Design – Beyond Human-Computer Interaction*, 5th edition, John Wiley, 2019
 (http://www.id-book.com/chapter3_teaching.php)

- John Carroll, Human Computer Interaction - brief intro, *The Encyclopedia of Human-Computer Interaction,* 2nd edition https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/affective-computing

Universidade de Aveiro
Departamento de Electrónica,
Telecomunicações e Informática

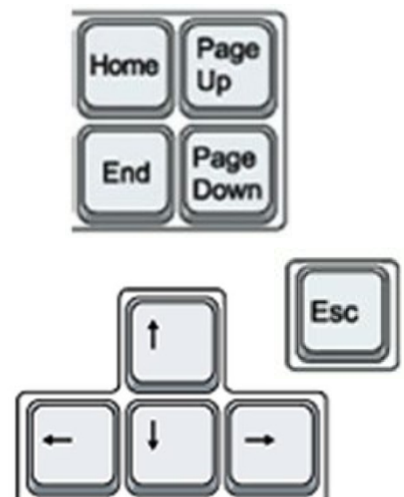
Input Devices

Main Input devices

- Keyboards
- Pointing devices
  - Mouse
  - Touch screen
  - Touch pad
  - Joy stick
  - Track ball, *…*
- Voice recognizers
- Eye trackers
- Motion and position trackers
- 3D input devices
- …

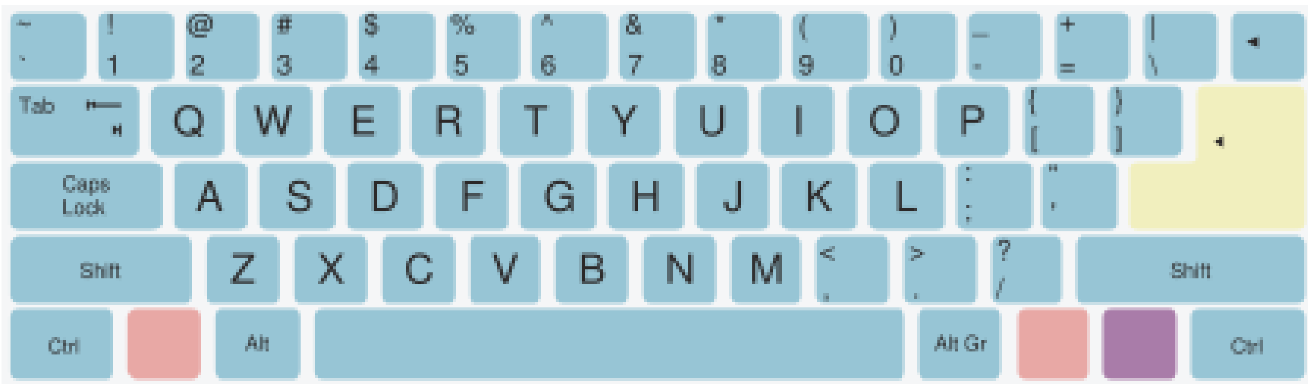
Keyboards

- Relevant issues in UI design:
  - Key layout
  - Operational characteristics:
    - Keyboard size
    - Keyboard angle
    - Hand resting area
    - Key spacing
    - Key activation force
    - Key surface and finishing
    - Key displacement
    - Activation feedback
    - Home row indicators



Keys layout

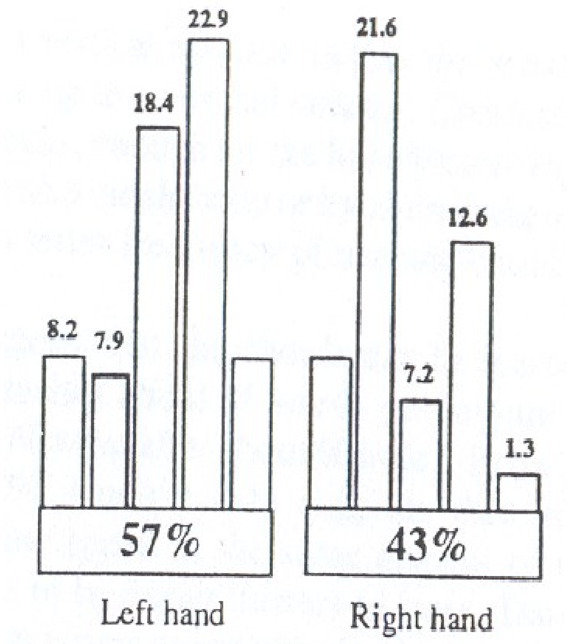
The Qwerty layout dates from the XIX century, and we still use it!

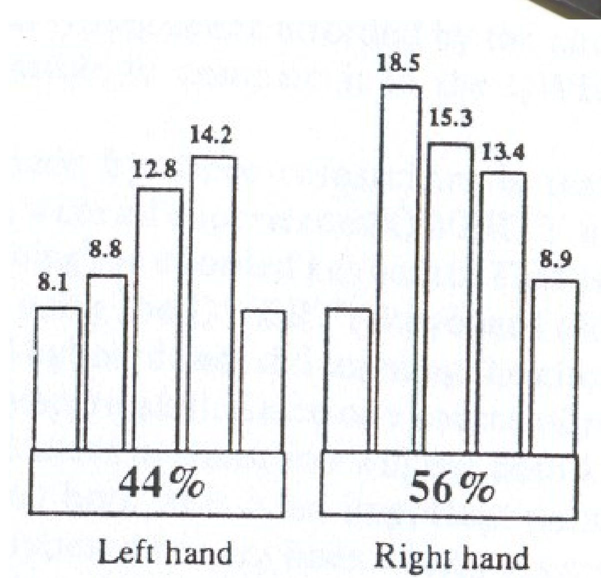


Dvorak

Combining both

Percentage of work performed by each hand (in English)





QWERTY **Dvorak**

QWERTY was devised to prevent jams in early typewriters.

http://www.dvorak-keyboard.com https://en.wikipedia.org/wiki/Dvorak_keyboard_layout

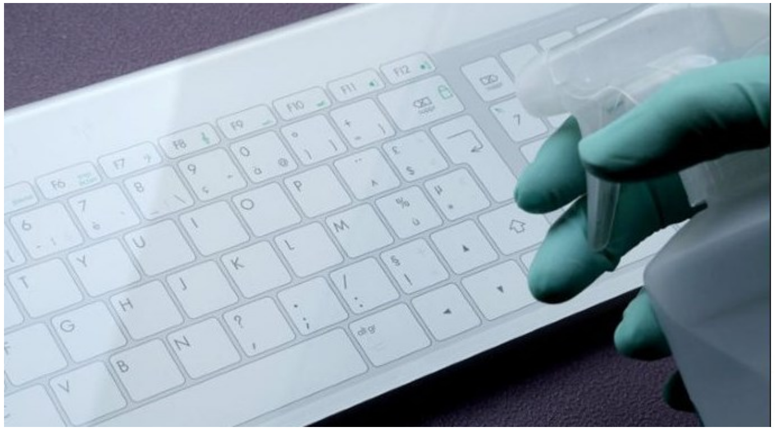
Ergonomic keyboards

Help avoid RSI (Repetitive Strain Injury) WRULD (Work Related Upper Limb Disorder) and KRP (Keyboard Related Pain)

1 Zoom
2 Customizable Hot Keys
3 Improved Number Pad
4 Ergonomic Design

https://en.wikipedia.org/wiki/Ergonomic_keyboard

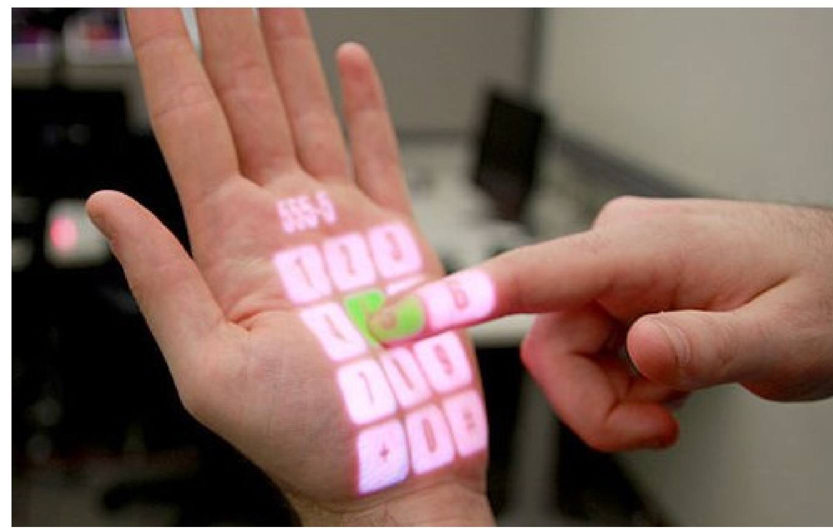
Keyboards for specific contexts of use



Chorded keyboard →
used in wearable computing

https://en.wikipedia.org/wiki/Chorded_keyboard

Virtual projection keyboards:
it is possible to project a keyboard on any surface:



http://en.wikipedia.org/wiki/Projection_keyboard
https://wiki.ezvid.com/best-virtual-keyboards

Pointing Devices

They are used to:

- Point a target
- Select a target
- Drawing
- Positioning objects
- Orient and rotate objects
- Define paths among objects
- Handle text
- etc.

- Their efficiency varies according to the tasks

- Shneiderman (98) divided them into:
  - Direct control —— touch screen
    light pen (deprecated)

  - Indirect control —— mouse
    track ball
    digitizing tablet
    joystick (track point)
    touch pad

Replica of the Engelbart's mouse

Mice

Currently are optical

- Relative coordinates
- Different shapes, n. of buttons,…

Advantages:

- Direct relation between hand and cursor movement
- Allow speed control
- Allow continuous movement in all directions

distance
speed
direction

Disadvantages:

- Require hand movement between mouse and keyboard
- Additional space (footprint)
- Hand-eye coordination

http://www.dougengelbart.org/firsts/mouse.html

http://www.computerhistory.org/revolution/input-output/14/350

Trackballs

- Relative coordinates
- Many different shapes

Advantages:

- Direct relation between hand and cursor movement (speed and direction)
- Allow speed control
- Allow continuous movement in all directions
- May not need additional space (footprint)

Disadvantages:

- Require hand-eye coordination
- May require hand movement between trackball and keyboard

Ergonomic Pointing Devices

Zero tension mouse

Quilll mouse

Whale mouse

Vertical mouse

Wireless Ergonomic Mouse

Gesture pad

For users with Repetitive Strain Injury,
Carpal Tunnel Syndrome or other problems
Or to avoid these problems

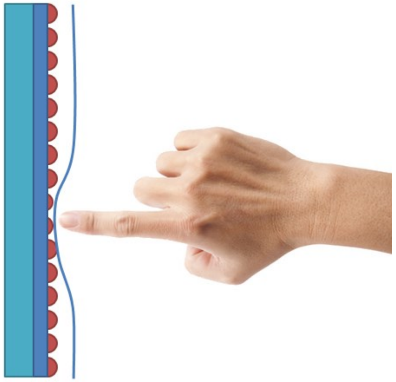
Touch screens

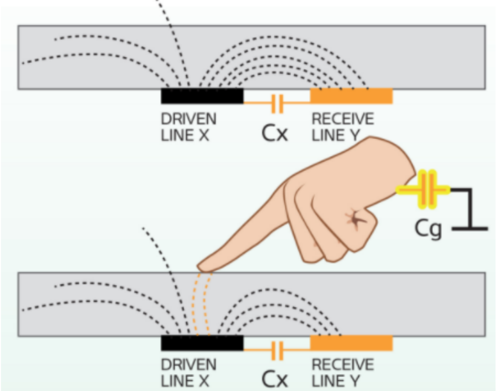
Resistive (older, less expensive)

optical

Sonic — stationary wave patterns

Capacitive





- Are combined with a display

https://en.wikipedia.org/wiki/Touchscreen

http://computer.howstuffworks.com/touch-screens.htm

Digitizing tablets adequate for digital art

Sophisticated models:
- Extended sizes
- Multitouch sensitive surface
- Pressure sensitive pen

http://www.wacom.com/products/pen-tablets

Some guidelines to select these interaction devices

- Choose a device after a careful task analysis and test
- Minimize hand and eyes movements
- Use cursor keys for tasks involving:
  - A lot of text manipulation
  - Traversing a structured array of discrete objects
- Use touch screens when
  - There is no training
  - Targets are large, discrete and scattered
  - Space is important
  - No (or little) text entry
  - Are not used for a long time

Voice recognition systems

- The first system was developed in 1972 at Bell Lab
- It is becoming more used
- Has two types of challenges:
  - Technological (have improved a lot …)
  - Human factors

Voice recognition as input

Independently of the technology state of the art,

- Has advantages when the user:
  - Has physical deficiency
  - Must move around
  - Has eyes busy
  - Is in a low visibility or cluttered environment

- Has inherent disadvantages:
  - Voice is transient
  - Does not have natural feedback
  - May disturb other people
  - May result in lack of privacy
  - May be slower and more tiresome (overloading STM)

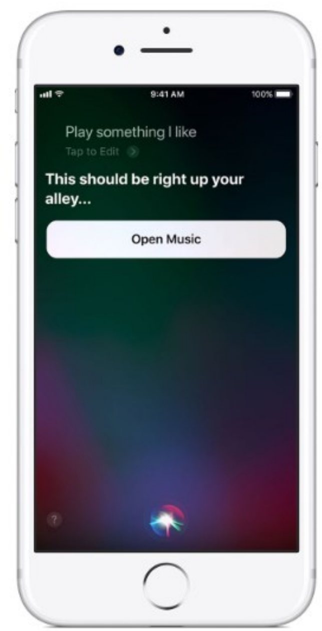
- Consider voice input when:
  - The user has to move
  - Has eyes or hands busy

amazon alexa

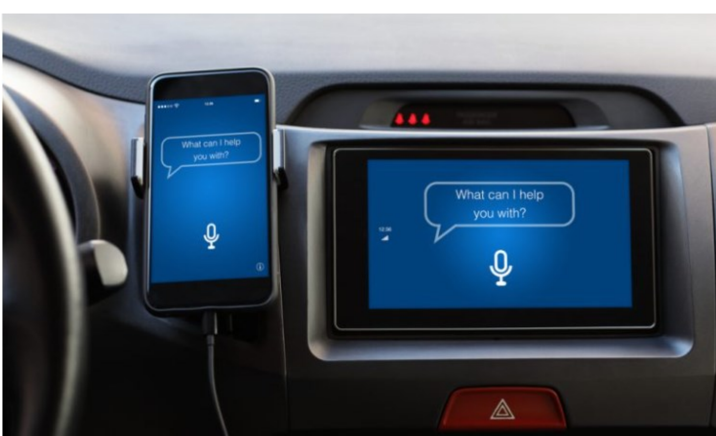
- Avoid voice input when:
  - Privacy is important
  - Error taxes, even low, are not acceptable
  - Usage frequency is high
  - Speed is important

Google Assistant

Siri



- Voice input/output has became more used



Some guidelines for voice interfaces

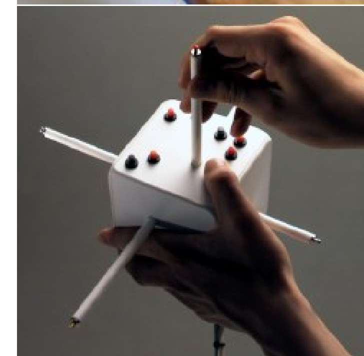
- Provide output dialog with structure to guide input
- Use a distinct and familiar vocabulary to avoid errors
- Consider voice input if technology constraints are acceptable considering:
  - Ambient noise
  - Privacy
  - Vocabulary extent
  - Error cost

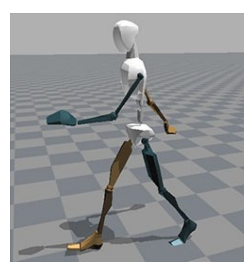
Input devices for 3D interfaces

- Trackers:
  - Magnetic
  - Optical
  - Ultrasonic
  - Inertial, ...

- Navigation and manipulation interfaces:
  - Tracker-based
  - Trackballs
  - 3D mice, ...

- Gesture interfaces:
  - Gloves
  - Spatial gestures sensors
  - …

Other input devices for 3D

Cyber Glove
http://www.cyberglovesystems.com/cyberglove-iii/

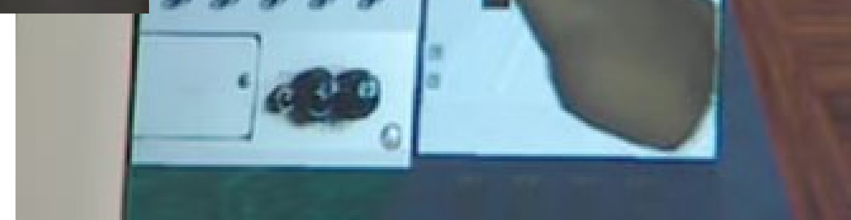
Tangible interfaces and pedals
http://www.youtube.com/watch?v=zJmrcEM-uvA

Leap motion (allows for hand gestures interfaces)
https://www.leapmotion.com/

Depth cameras (allows body gesture interfaces)
https://en.wikipedia.org/wiki/Kinect

https://www.vive.com/eu/
HTC Vive trackers

CyberTouch Glove: input + output

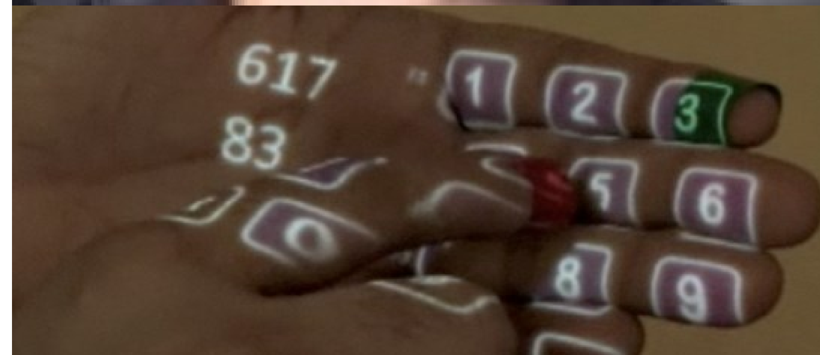


http://www.cyberglovesystems.com/cybertouch/
https://www.youtube.com/watch?v=32f2UxKjydI

What future?

It seems likely that we will use more often:

- gestures
- two hand input
- voice
- 3D pointers
- wearable devices
- whole-body environments
- tactile/force feedback, …



Conclusion

When choosing an input device, consider:

- Ergonomics / human factors
- Typical scenarios of use
- Cost
- Generality
- DOFs (Degrees Of Freedom)
- Output devices
- Interaction techniques

Technology shall not be used only because it is new and interesting!

- It is necessary to understand the usability of devices for the users and the tasks they have to perform in a specific context

Don’t forget that:

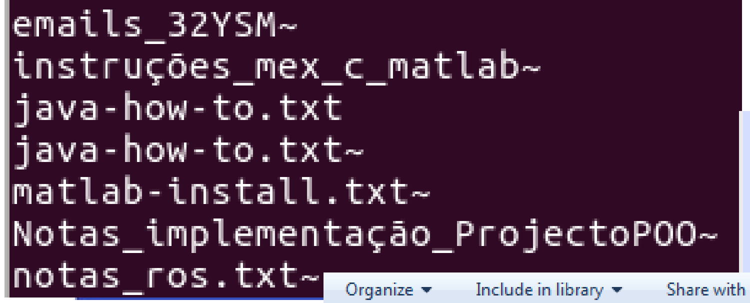
“The interface between humans and computers is harder than ever to define, we can interact with computers just by walking through a public space.”

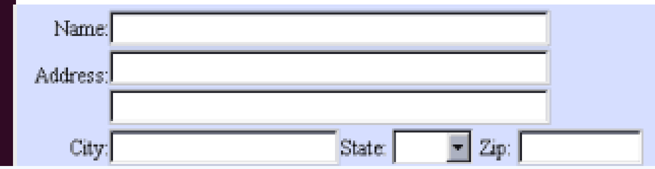
Sellen, A., Rogers, Y., Harper, R., & Rodden, T., “Human Values in the Digital Age”, *Communications of the ACM*, *52*(3), March 2009, pp. 58–66

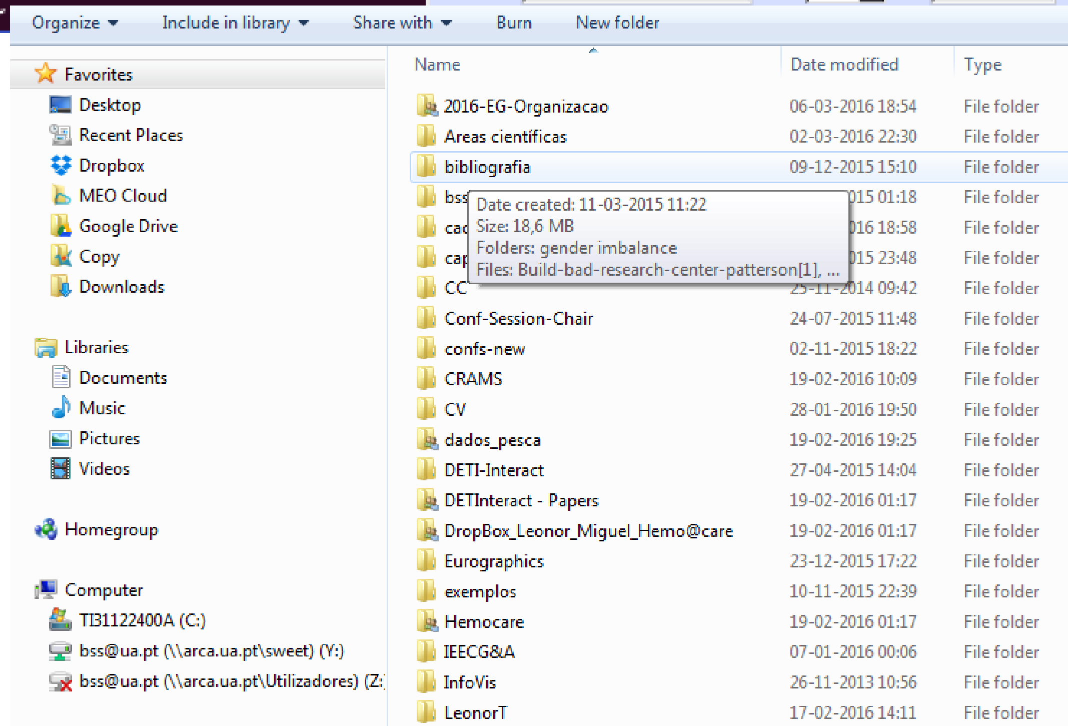
Universidade de Aveiro
Departamento de Electrónica,
Telecomunicações e Informática

Direct Manipulation

Interaction/ Dialog styles



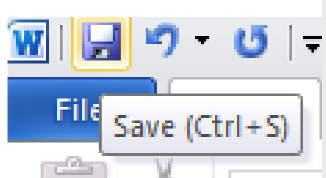


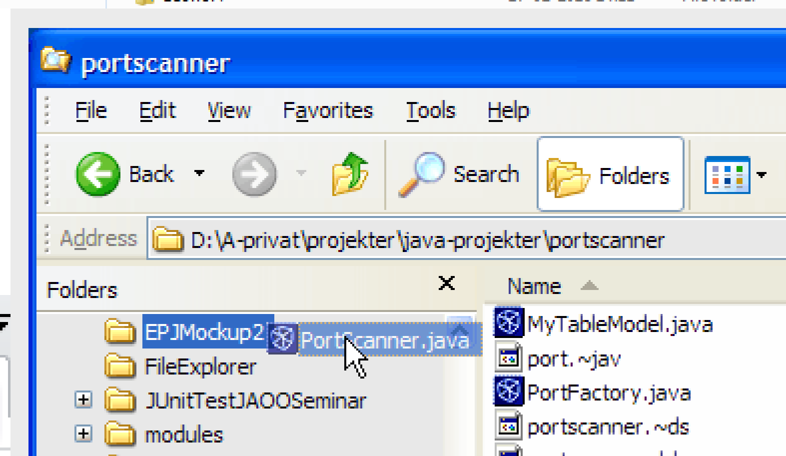


A possible classification:

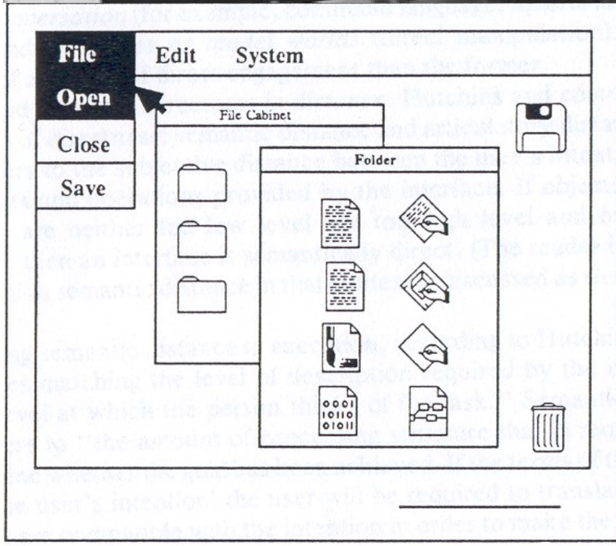
- Menus
- Fill-in-forms
- Direct manipulation
- Function keys
- Question and answer
- Command languages
- Natural languages

Often two or more styles are used simultaneously





Direct manipulation
(name coined by Shneiderman, 1982)



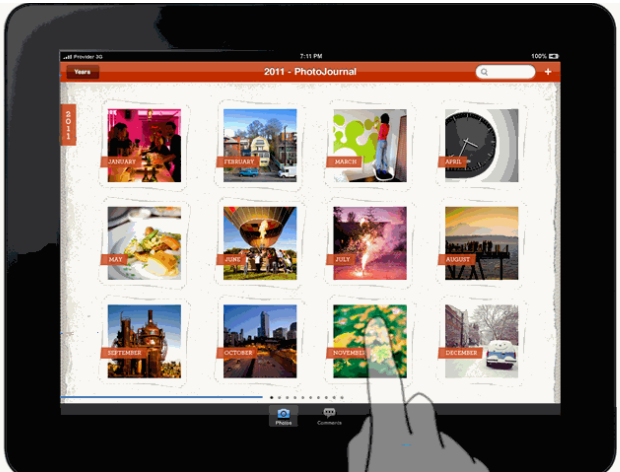
- Can be traced down to Sketchpad
https://www.youtube.com/watch?v=495nCzxM9PI

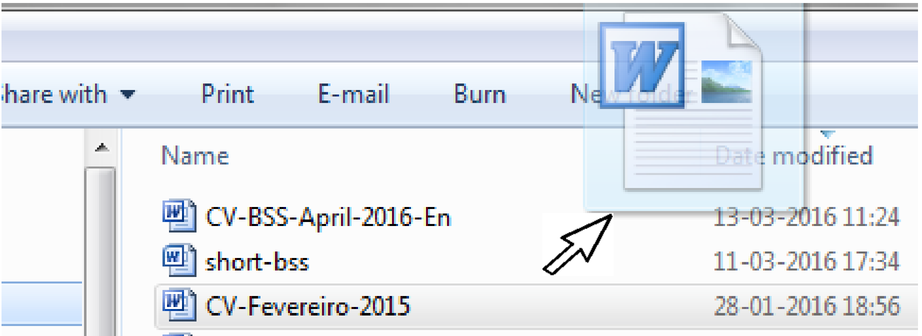
- Actions are performed directly on visual representations of the objects
- It is characterized by:

1- Continuous representation of objects

2- Physical actions instead of command languages

3- Fast, incremental, reversible actions with visible results





Direct manipulation does not necessarily imply icons; however, in most situations they are involved

2 RELATED WORK

Despite the growing interest in usability related research in the VE community, not as many papers concerning usability evaluation exist, as compared to papers proposing new methods, techniques or systems. For instance, in a research recently conducted, we were able to find only a few studies directly comparing user performance while using VEs in desktops and systems including a HMD [Sousa Santos,2008] [Sousa Santos,2009]. Analyzing these studies, it can be observed that controlled experiments involving users have been the most used evaluation method, complemented in some cases with a questionnaire. We can also observe that most studies were performed in a general context (as opposed to applied to a specific situation), and that search and navigation were the chosen tasks in a significant part of them.

User studies have been considered an important method in other contexts, as Scientific Visualizations and Augmented Reality [Kosara,2003][Gabbard,2008]. We believe that they can, likewise, contribute to optimize VEs informing their design within a usability engineering approach; however, they can also be used to compare alternatives, validate solutions, and more fundamentally help seeking insight into why a particular solution is effective, thus allowing establish design guidelines.

Example: When a section of a text is selected and dragged elsewhere icons are not used, yet an action is performed on a visual representation of an object (text section)

2 RELATED WORK

User studies have been considered an important method in other contexts, as Scientific Visualizations and Augmented Reality [Kosara,2003][Gabbard,2008]. We believe that they can, likewise, contribute to optimize VEs informing their design within a usability engineering approach; however, they can also be used to compare alternatives, validate solutions, and more fundamentally help seeking insight into why a particular solution is effective, thus allowing establish design guidelines.

Despite the growing interest in usability related research in the VE community, not as many papers concerning usability evaluation exist, as compared to papers proposing new methods, techniques or systems. For instance, in a research recently conducted, we were able to find only a few studies directly comparing user performance while using VEs in desktops and systems including a HMD [Sousa Santos,2008] [Sousa Santos,2009]. Analyzing these studies, it can be observed that controlled experiments involving users have been the most used evaluation method, complemented in some cases with a questionnaire. We can also observe that most studies were performed in a general context (as opposed to applied to a specific situation), and that search and navigation were the chosen tasks in a significant part of them.

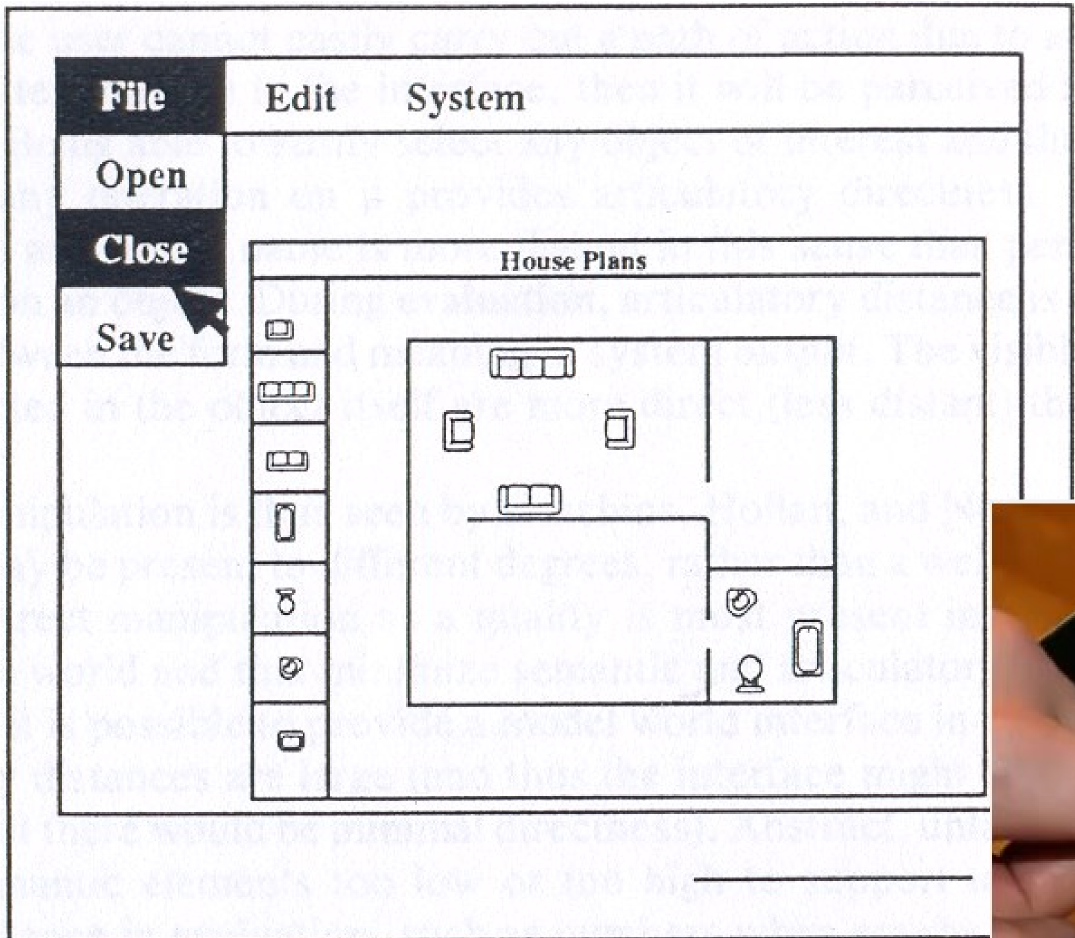
To study and compare usability issues concerning our low cost platforms we had to choose a context of use since usability cannot be defined in abstract. In fact, it is associated to users performing certain tasks [Nielsen,1993] (page 27) [Dix,2004] (page 192). Given that we had not a

Direct manipulation does not necessarily imply icons; however, in most situations they are involved

Another example:
On a mobile phone you can pinch out/in to zoom into an image or to zoom out

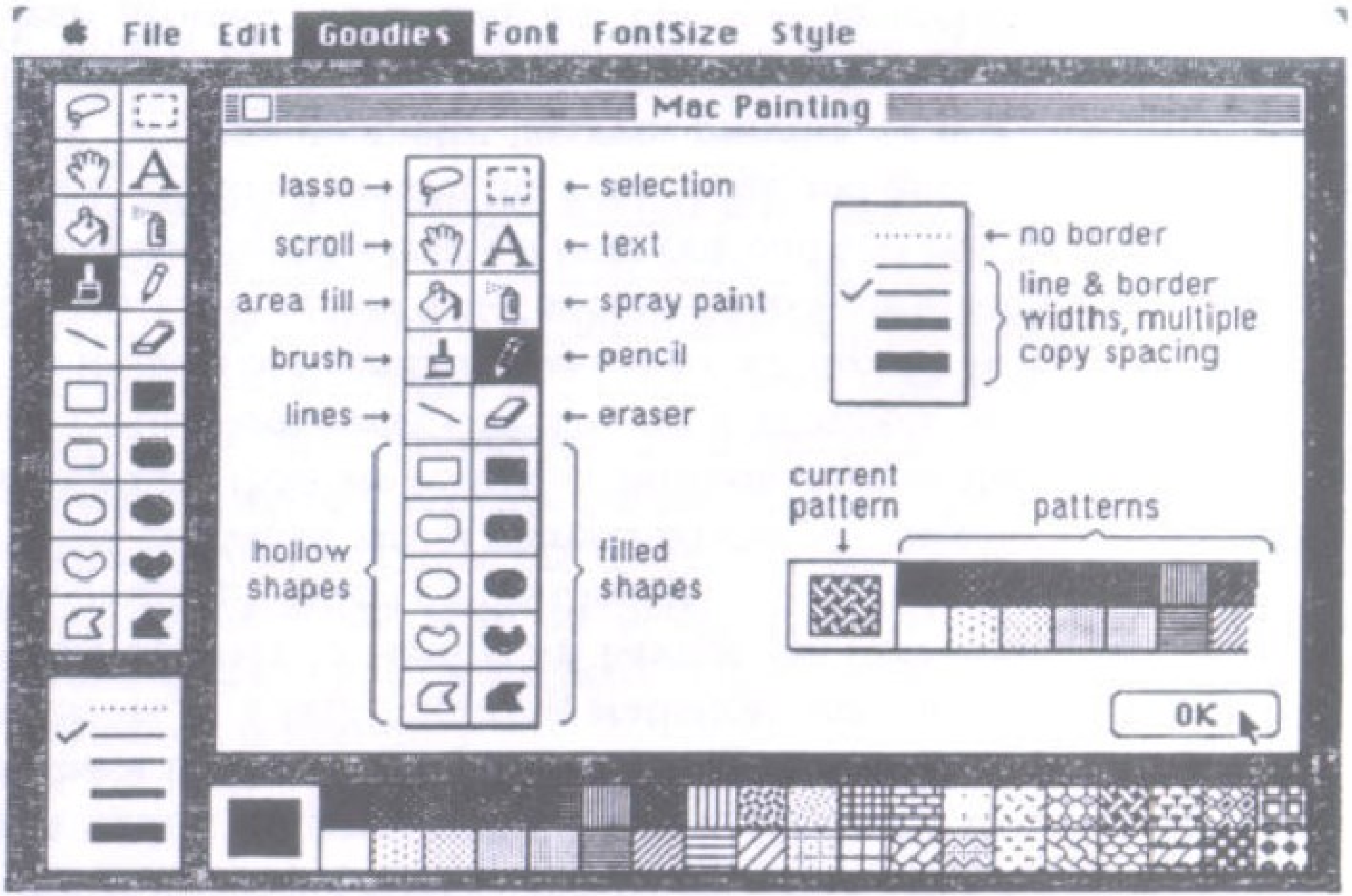
https://www.nngroup.com/articles/direct-manipulation/

Some applications are adequate to use direct manipulation:



https://www.youtube.com/watch?v=yqlyzTezT_Q

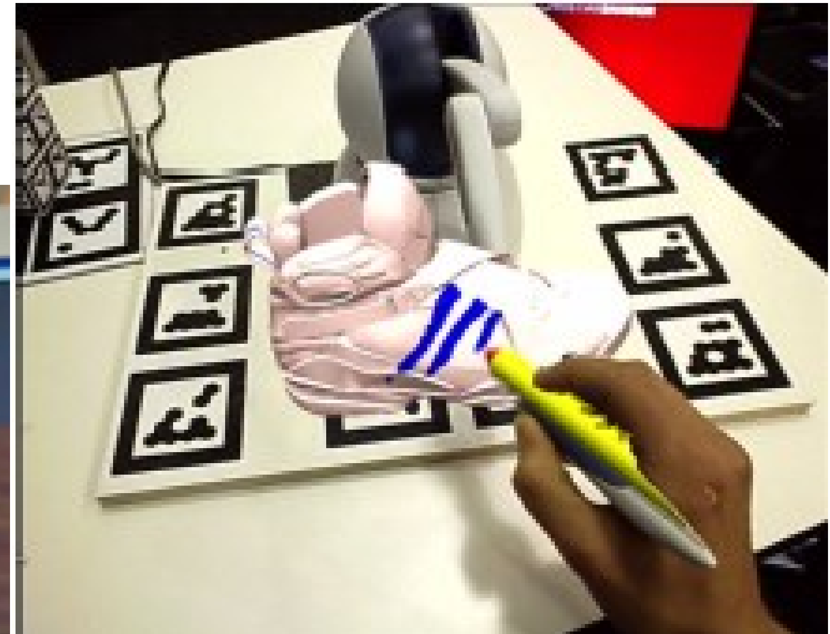
One of the earliest commercially available UI using Direct Manipulation (MacPaint)



http://www.magicvisionlab.com/pub/eck_ieeevr13/paper.pdf

Virtual and augmented reality

Take direct manipulation to another level



http://www.cyberglovesystems.com/cad-evaluator

- It does not exist a “pure” direct manipulation User Interface (UI)

- Direct manipulation is a quality which may be present in different degrees

- According to Hutchins, Hollan e Norman (1986) a UI has the following aspects:

Distance
- semantic
- articulatory

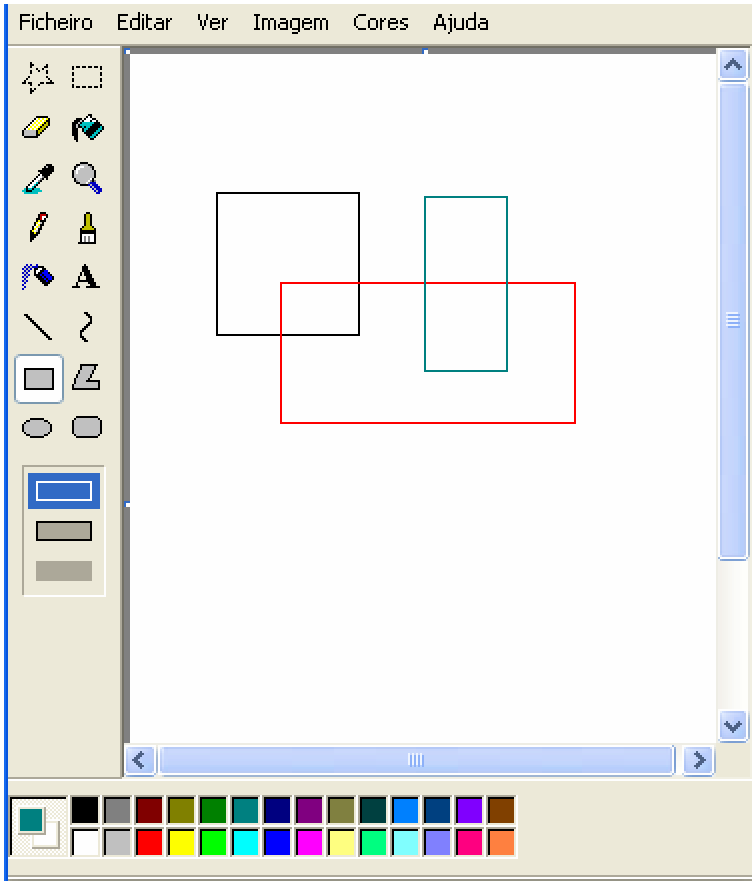
Semantic and articulatory distance

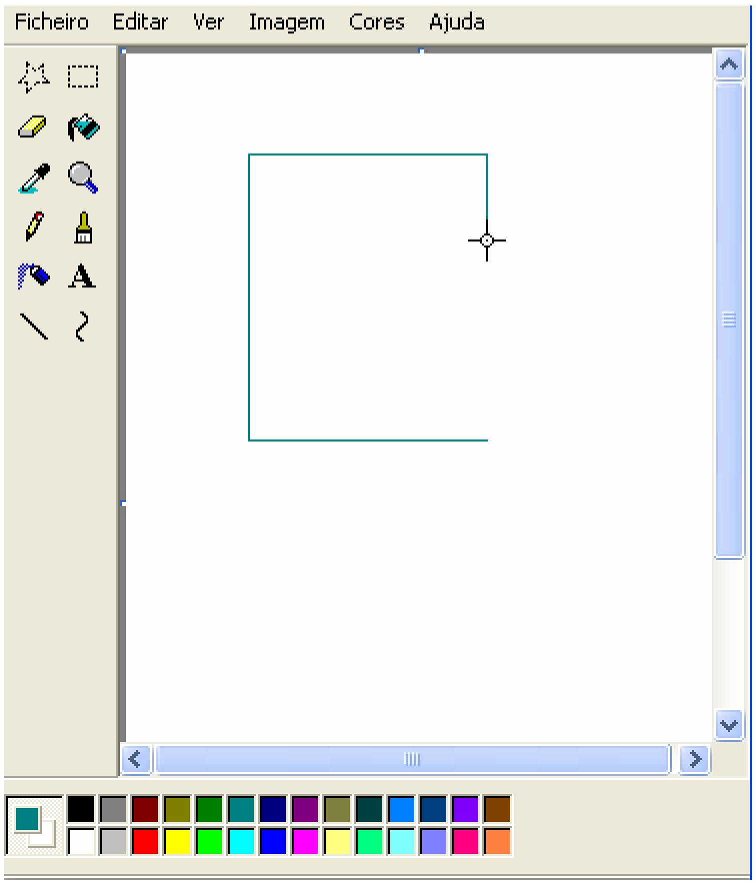
- Semantic Distance – subjective distance between the user’s goal and interface semantics

- Articulatory distance – distance between the meaning of the actions and their physical form

Semantic Distance

If the objects and actions do not support the users' goals, semantic distance is high



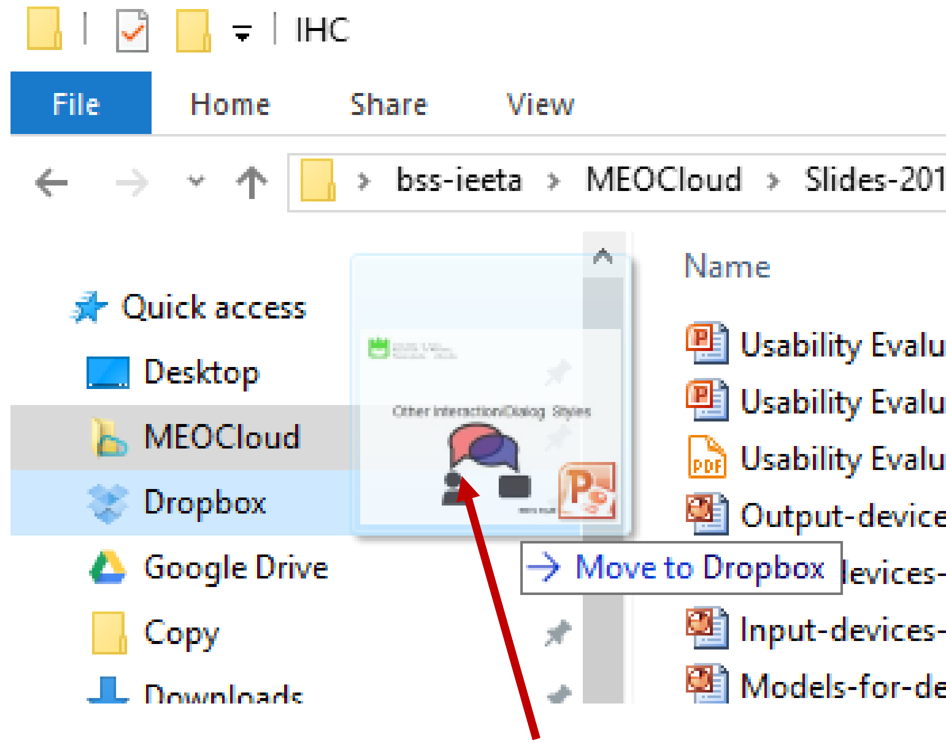


If the user wants to draw rectangles this application has a smaller semantic distance

,

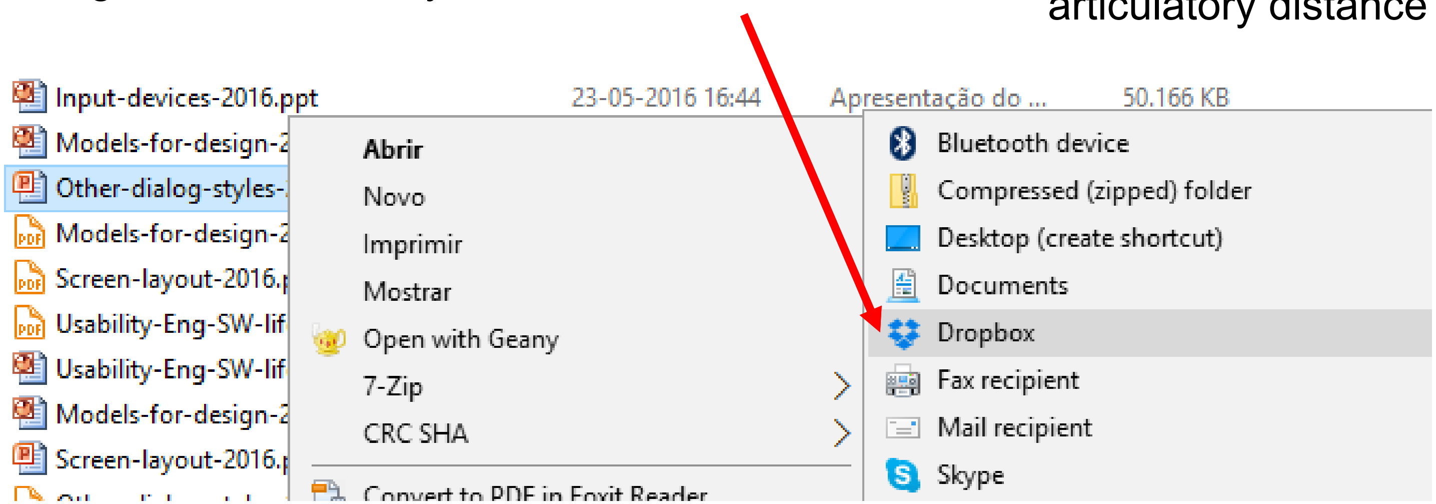
Articulatory Distance

When the physical way actions are performed is more similar to their meaning, articulatory distance is smaller



Selecting an option corresponds to a greater articulatory distance

Dragging an icon corresponds to smaller articulatory distance



- According to Wolf e Rhyne (1987) there are two relevant aspects in any user interaction:

Object specification	Action specification
name generation	name generation (write a name)
visual correlation	visual correlation (select)
	gesture generation (draw a symbol)
	analogous action
	coded selection (write a command)

- Specifying objects by visual correlation implies the presence of direct manipulation

- How actions are specified defines the degree of direct manipulation

- direct UI

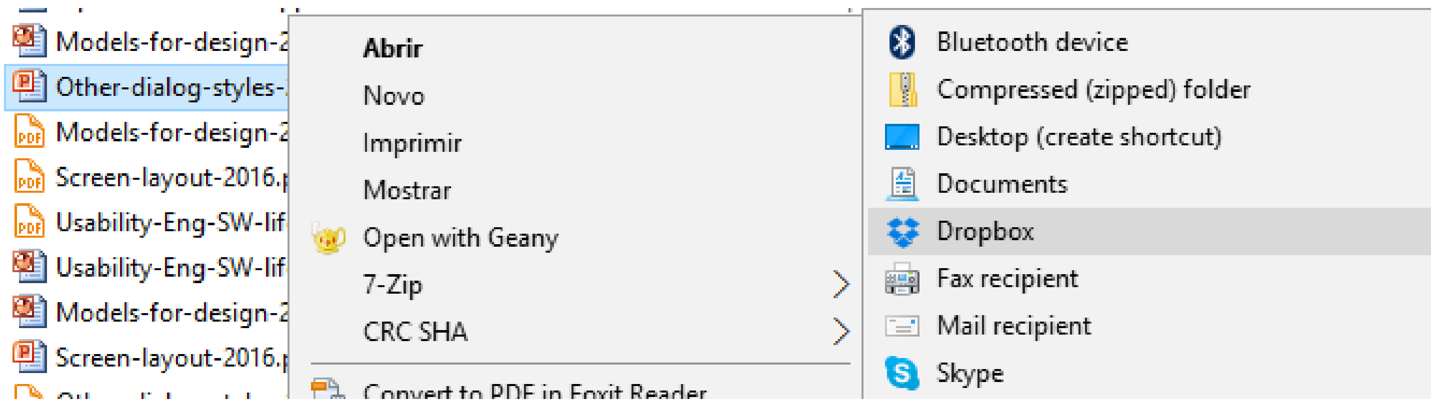
Objects		Actions
visual correlation	+	name generation
		visual correlation
		analogous action

+ direct UI

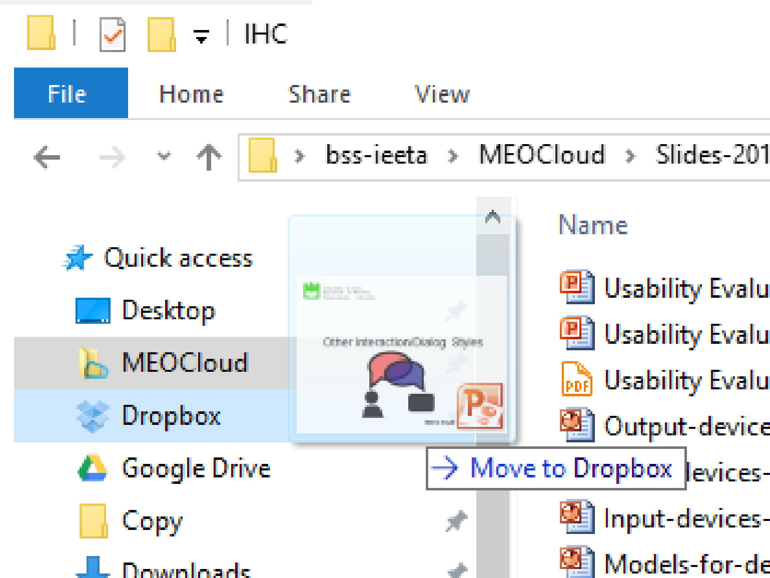
Examples

```
bi@ub:~/Desktop$
bi@ub:~/Desktop$ mv java-how-to.txt smartbike_paper/
bi@ub:~/Desktop$
bi@ub:~/Desktop$
```

Not direct manipulation UI: name generation + name generation



Direct manipulation UI:
visual correlation + visual correlation



+ Direct manipulation UI:
visual correlation + analogous action

Main advantages and disadvantages of direct manipulation UIs

Advantages (potential)

- Easy to learn and remember (are great for novices with good design
- Direct, WYSIWYG (What you see is what you get)
- Flexible, easily reversible actions
- Immediate visual and context feedback
- May be less prone to errors

Disadvantages

- Not auto-explanatory
- May be inefficient
- Repetitive tasks are not well supported
- Some gestures can be more error-prone than typing
- Difficult to draw recognizable icons (particularly for actions)
- Icons occupy more screen real estate then text

https://www.nngroup.com/articles/direct-manipulation/

User profile to whom direct manipulation is adequate:

Knowledge and experience:

- Moderate system experience
- Moderate to high task experience
- Frequent usage of other systems
- Low computational literacy

Work and task

- Low frequency of use
- Moderate training
- Optional usage
- Low structured tasks

Direct Manipulation design: some guidelines

Minimize articulatory and semantic distance

Use general guidelines to design a usable UI:

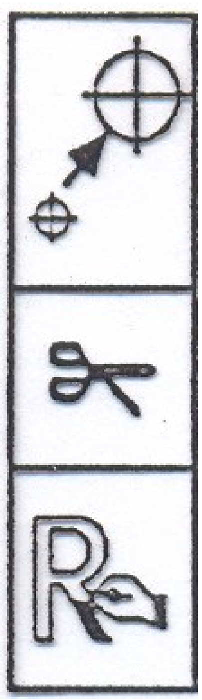
- Coherence
- Good conceptual model
- Feedback
- Adequate organization of functionality
- Adequate screen layout
- Adequate colour usage
- Adequate error handling
- Etc.

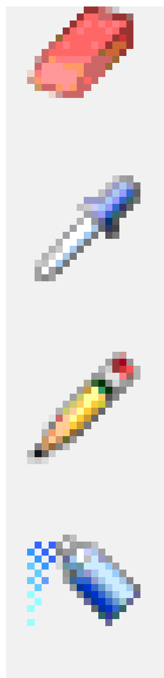
https://www.interaction-design.org/literature/article/a-brief-history-of-the-origin-of-the-computer-icon

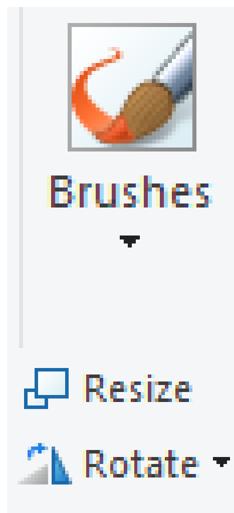
Use a coherent Icon scheme

Different schemes:

better: same scheme

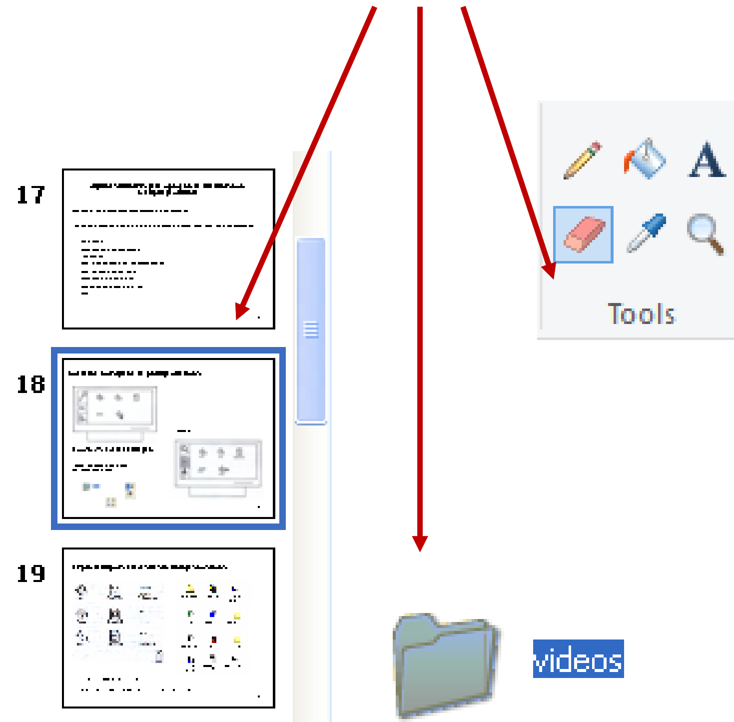




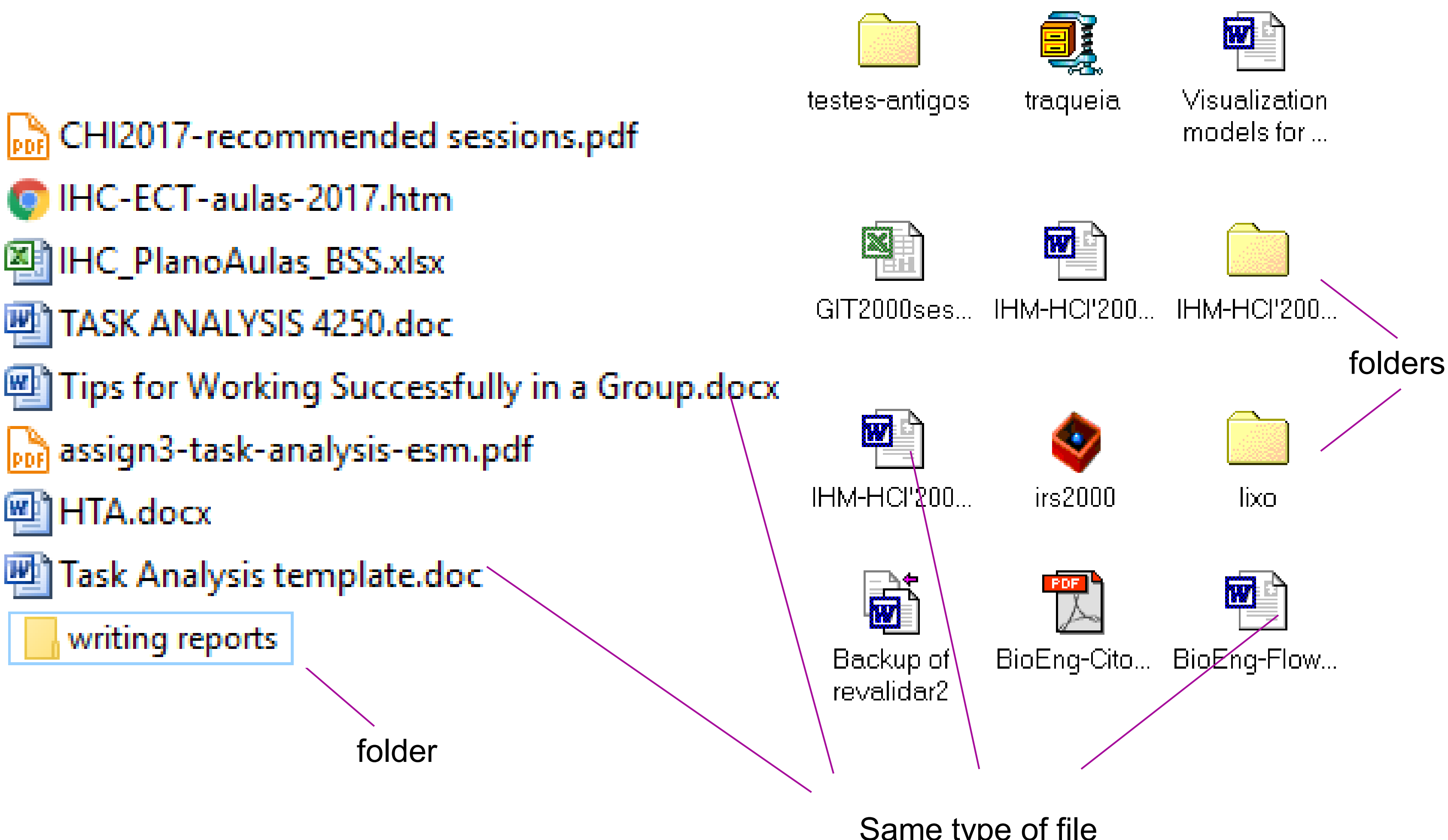


Adding names
(+ recognizable)

Visual selection feedback

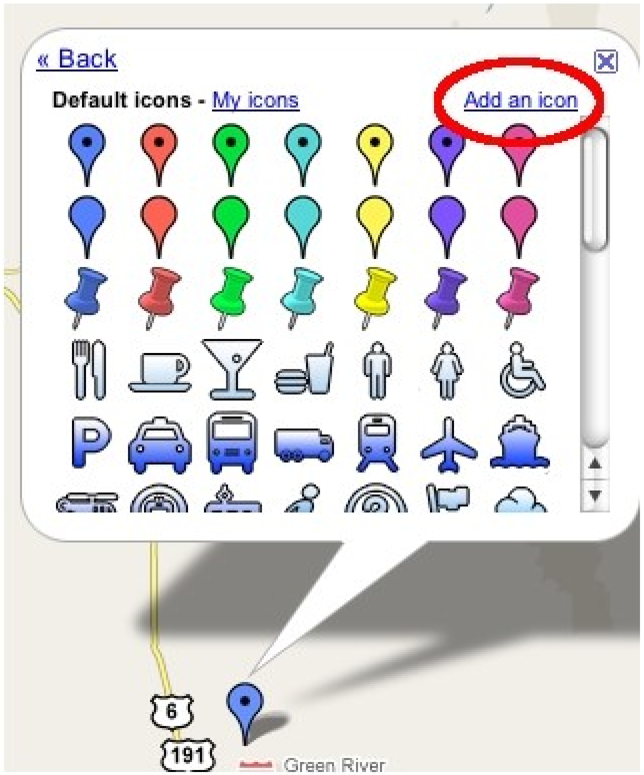


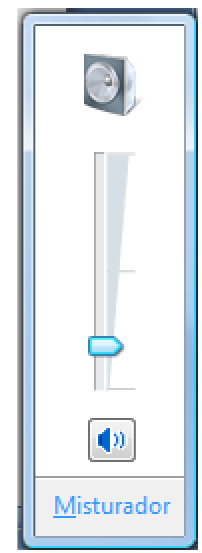
Express relation through icon similarity



Coherence in the icons production scheme

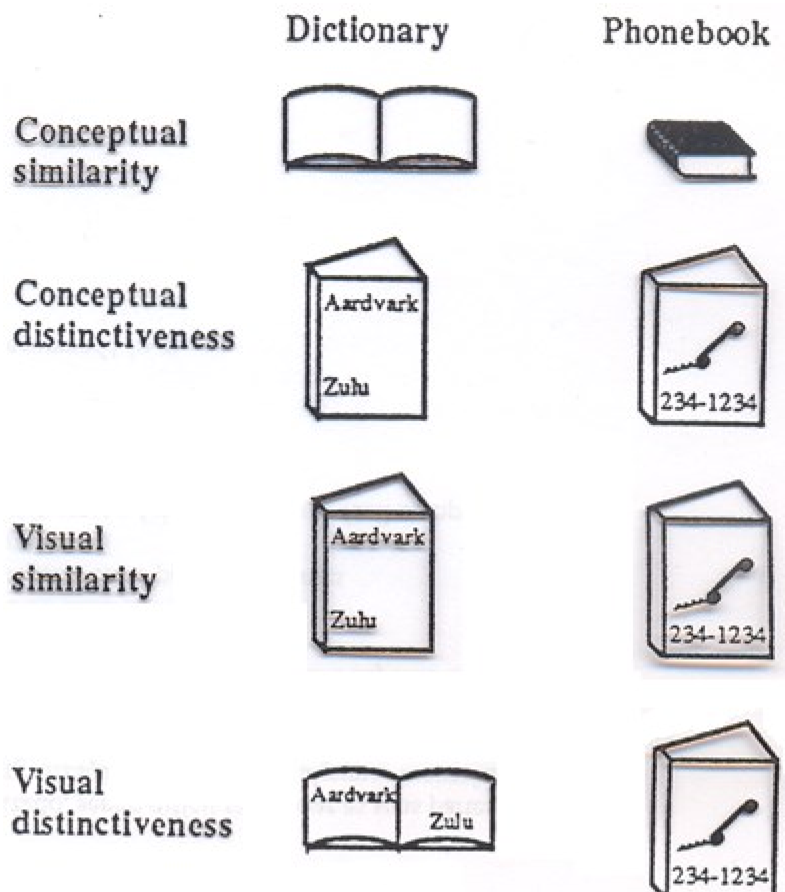
Add names to icons to make them more recognizable (recognition rather than recall)

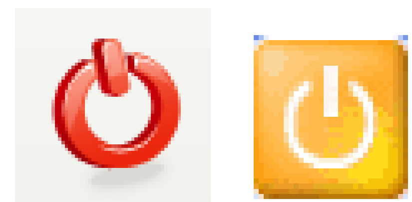




Allow name definition

Icons must be conceptually and visually distinctive
(recognition rather than recall)

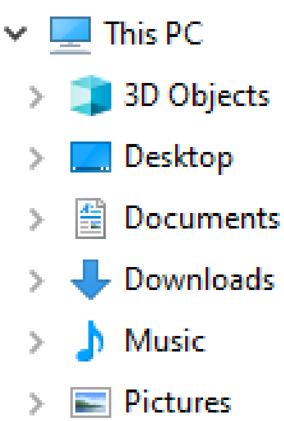




Conceptually similar icons

Best solution: conceptually and visually distinct

Icons should be specific/familiar not abstract/non-familiar (familiarity)



Familiar Icons?

Express objects' attributes through icons
(visibility of the system status)

synced

empty

In progress

Concluding remarks

“It’s hard to imagine modern interfaces without direct manipulation …

Augmented-reality and virtual-reality systems will push DM to even newer limits …

Despite the many downsides, we still recommend a heavy dose of direct manipulation for most UIs”

https://www.nngroup.com/articles/direct-manipulation/

Main Bibliography

- B. Shneiderman, C. Plaisant, M. Cohen, S. Jacobs, *Designing the User Interface- Strategies for Effective Human–Computer Interaction*, 5th ed., Addison Wesley, 2010

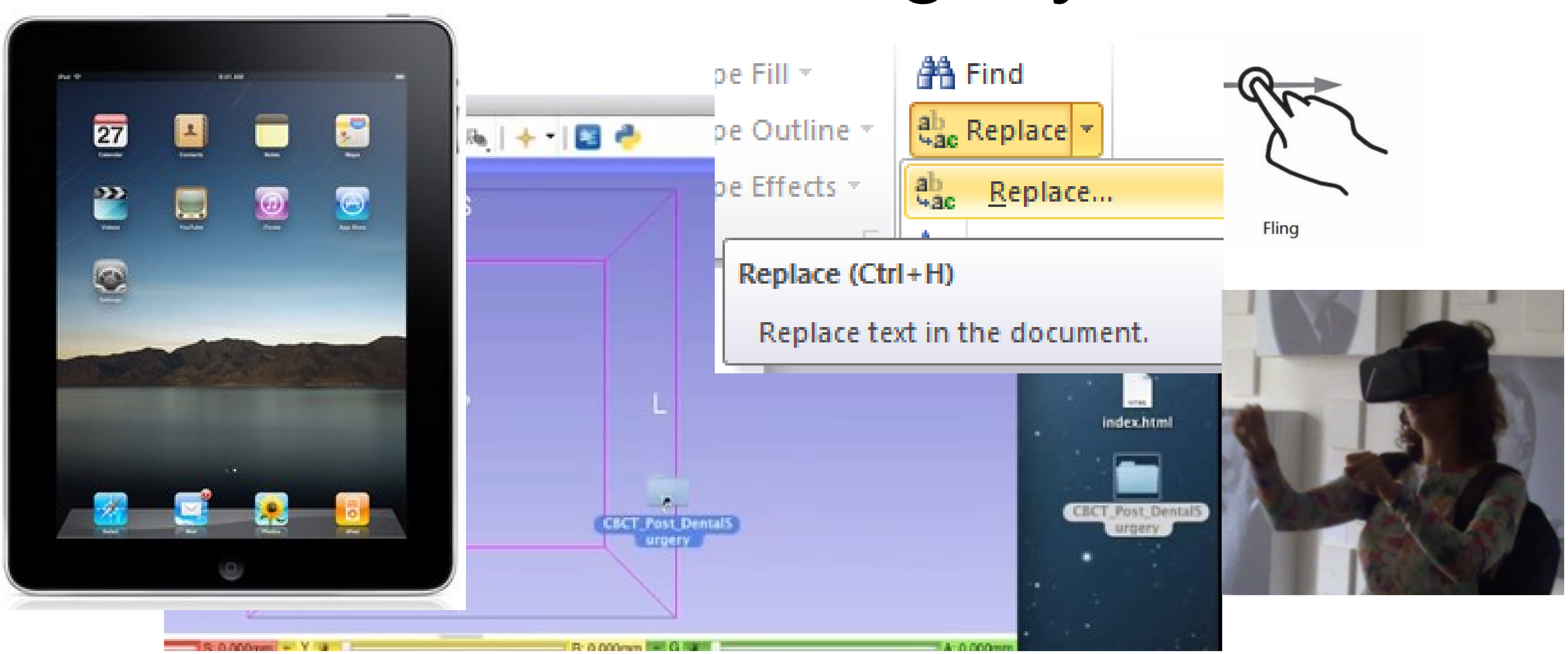
- H. Sharp, J. Preece, and Y. Rogers, *Interaction Design: Beyond Human-Computer Interaction*, 5th Edition Wiley, 2019

- M. Soegaard, Interaction Styles, *Interaction Design Foundation Encyclopedia*, 2nd edition,

  http://www.interactiondesign.org/encyclopedia/interaction_styles.html

Universidade de Aveiro
Departamento de Electrónica,
Telecomunicações e Informática

Interaction/Dialog styles



Beatriz Sousa Santos, 2021

Interaction Styles

“The concept of Interaction Styles refers to all the ways the user can communicate or otherwise interact with the computer system.”

Soegaard, Mads. Interaction Styles, 2010 (Retrieved March 2020)
http://www.interactiondesign.org/encyclopedia/interaction_styles.html

There are a lot of studies and design guidelines

Shneiderman's Eight Golden Rules of Dialogue Design

1. Strive for consistency
2. Enable frequent users to use shortcuts
3. Offer informative feedback
4. Design dialogues to yield closure
5. Offer simple error handling
6. Permit easy reversal of actions
7. Support internal locus of control
8. Reduce short-term memory load

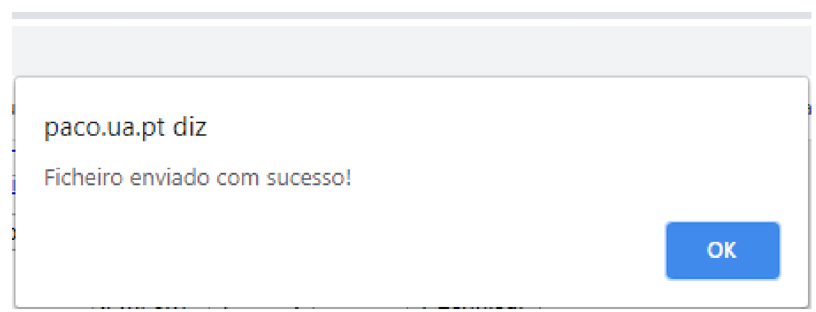
These golden rules are paramount in the design process

https://www.interaction-design.org/literature/article/shneiderman-s-eight-golden-rules-will-help-you-design-better-interfaces

Support internal locus of control

- Allow users to be the initiators of actions
- Give users the sense that they are in control of events

Design dialogue to yield closure

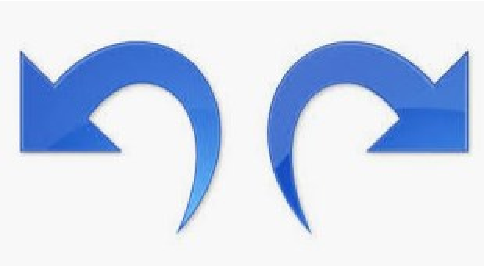


- Don't keep users guessing.
- Tell them what their action has led them to

Allow easy reversal of actions

encouraging exploration of unfamiliar options

undo – CTRL z



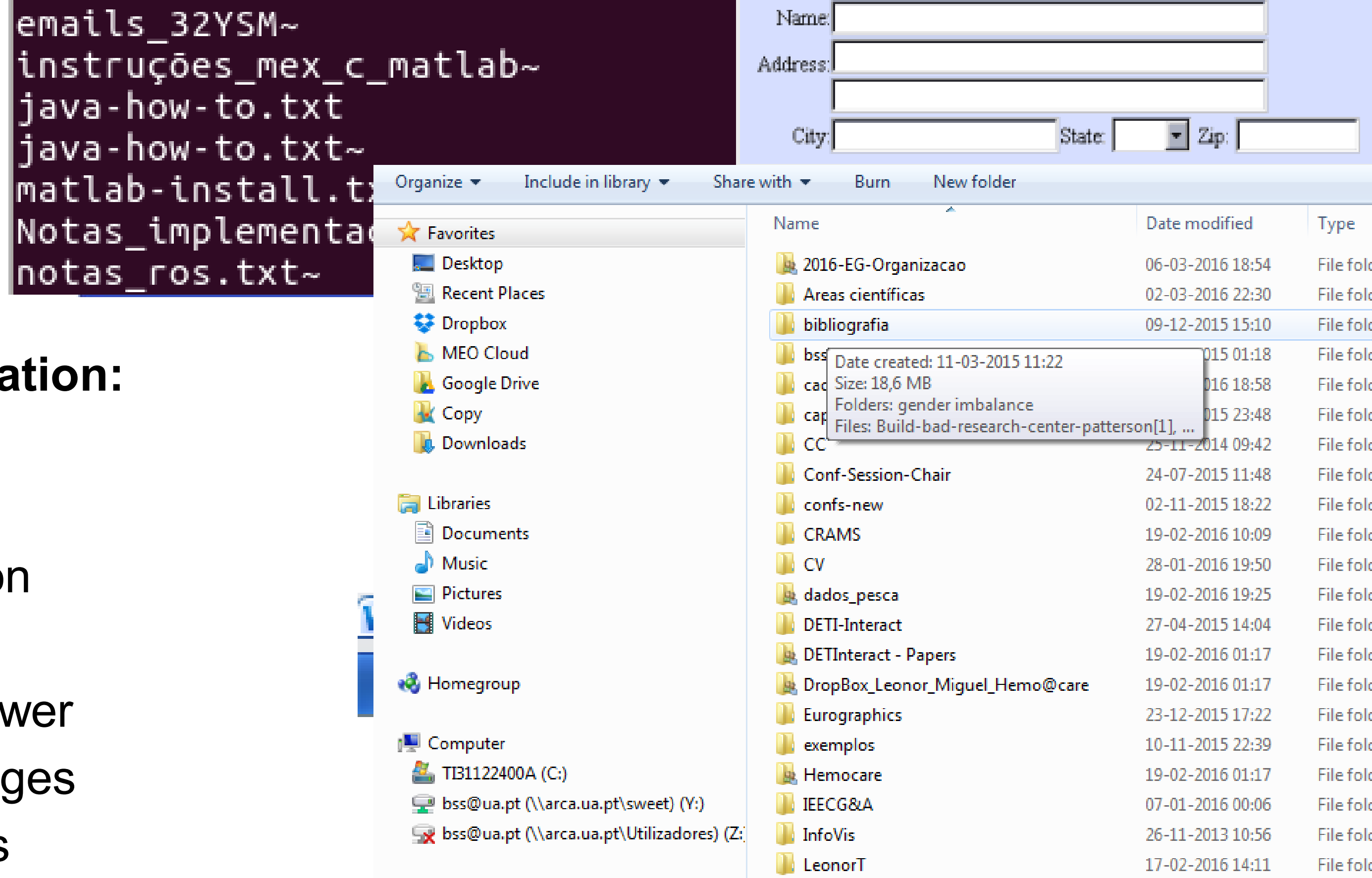
Allow frequent users to use shortcuts

Common examples:

save – CTRL s

copy – CTRL c

Interaction/ Dialog styles



A possible classification:

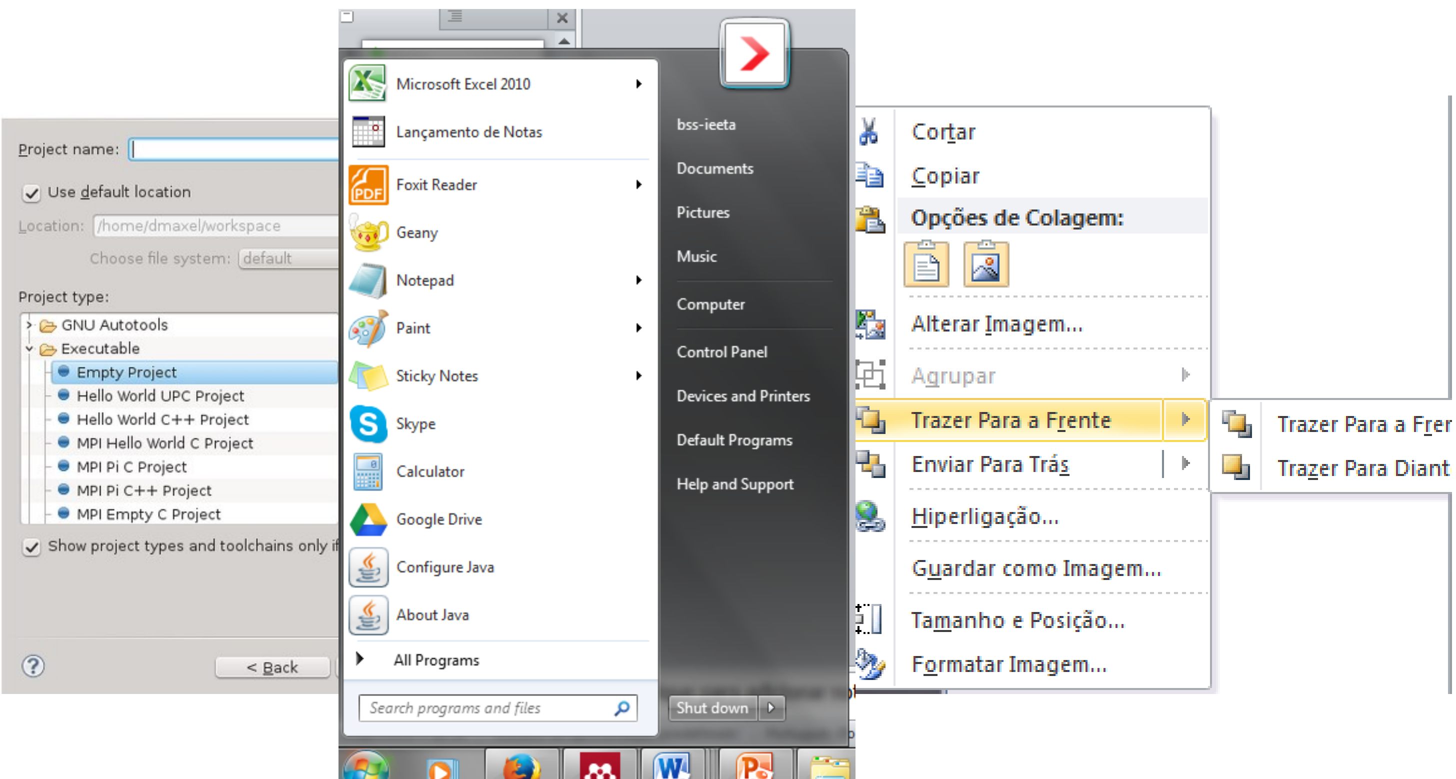
- Menus
- Fill-in-forms
- Direct manipulation
- Function keys
- Question and answer
- Command languages
- Natural languages

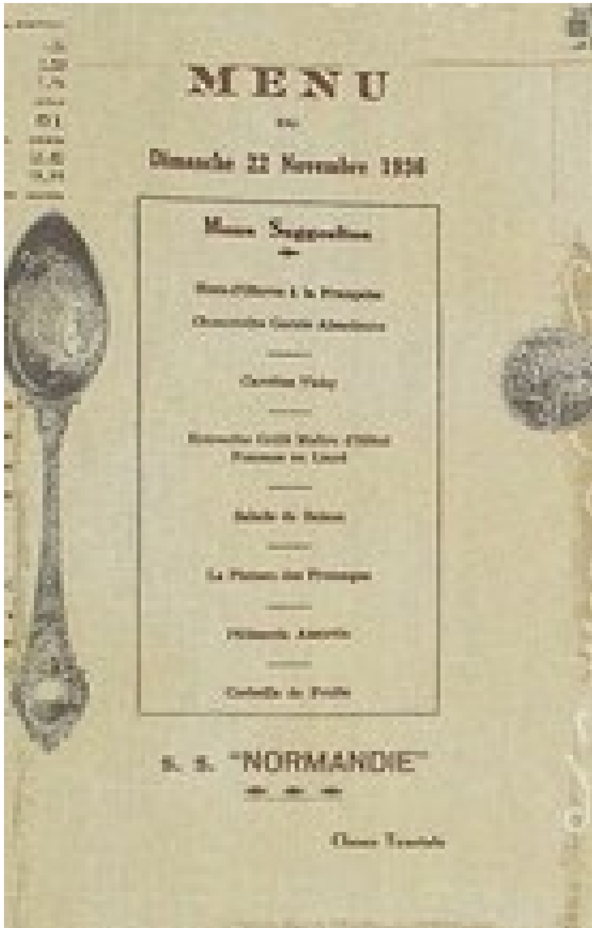
Often two or more styles are used simultaneously

Less traditional interfaces:

- 3D interfaces
- Tangible interfaces
- etc.…

Menus

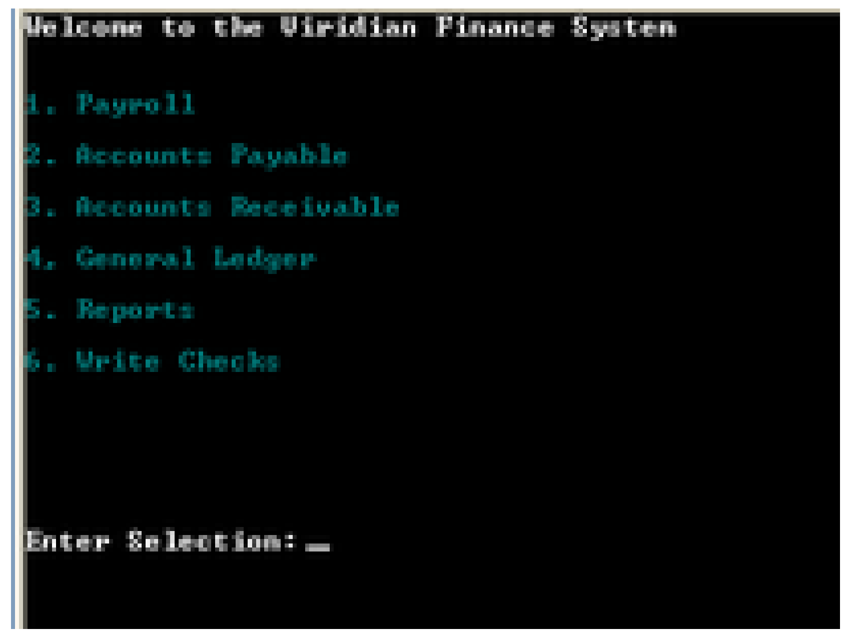

Project name:
Use default location
Location: /home/dmaxel/workspace
Choose file system: default
Project type:
GNU Autotools
Executable
Empty Project
Hello World UPC Project
Hello World C++ Project
MPI Hello World C Project
MPI Pi C Project
MPI Pi C++ Project
MPI Empty C Project
Show project types and toolchains only if
< Back
Microsoft Excel 2010
Lançamento de Notas
Foxit Reader
Geany
Notepad
Paint
Sticky Notes
Skype
Calculator
Google Drive
Configure Java
About Java
All Programs
Search programs and files
bss-ieeta
Documents
Pictures
Music
Computer
Control Panel
Devices and Printers
Default Programs
Help and Support
Shut down
Cortar
Copiar
Opções de Colagem:
Alterar Imagem...
Agrupar
Trazer Para a Frente
Enviar Para Trás
Hiperligação...
Guardar como Imagem...
Tamanho e Posição...
Formatar Imagem...
Trazer Para a Frer
Trazer Para Diant



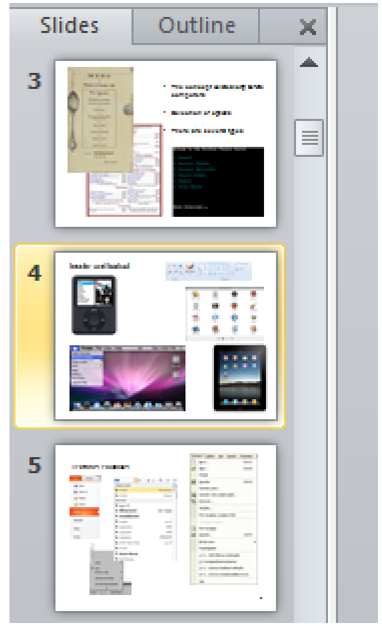
- The concept existed long before computers

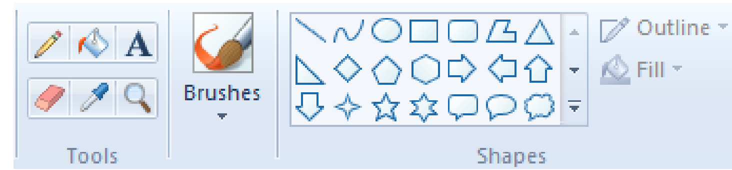
- Selection of options

- There are several types

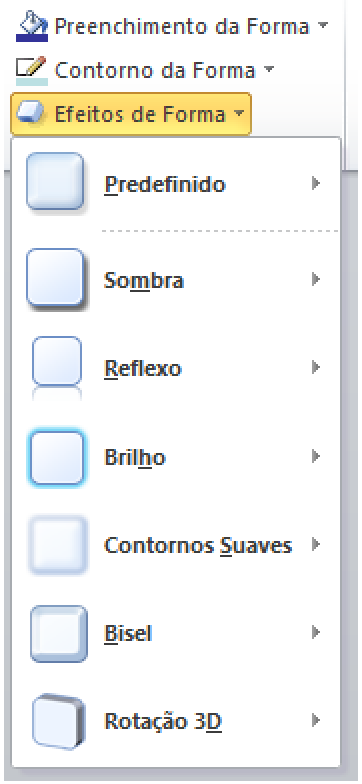


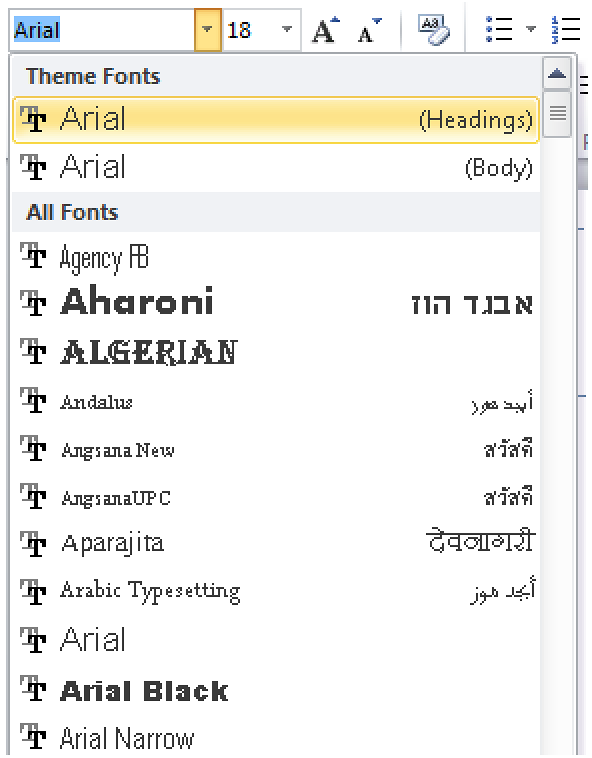
Iconic and textual

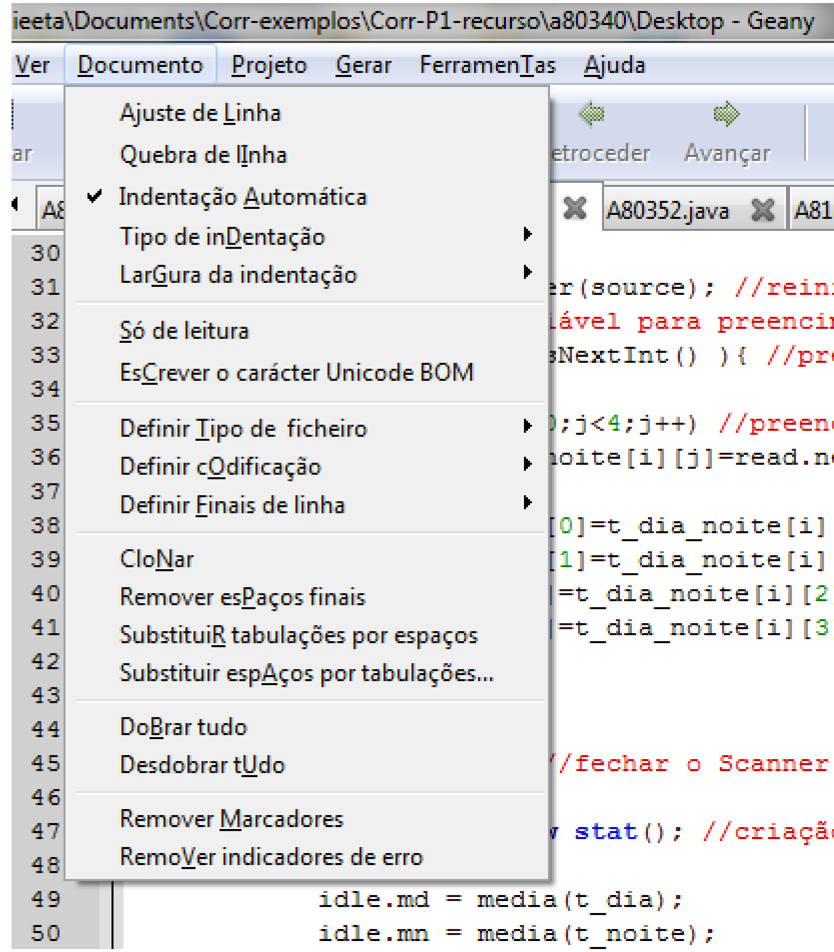


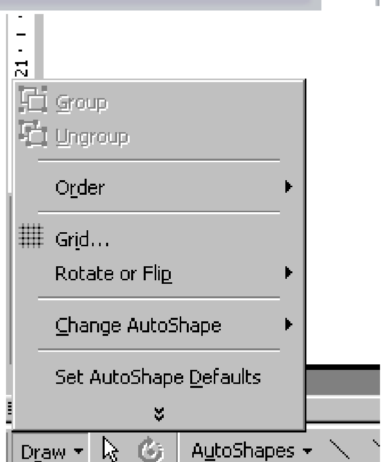


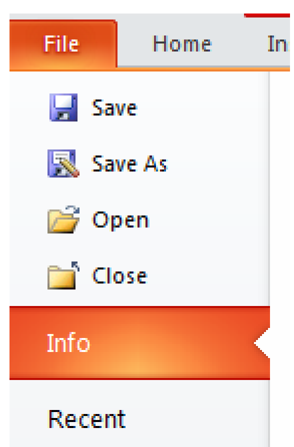
Textual and Iconic





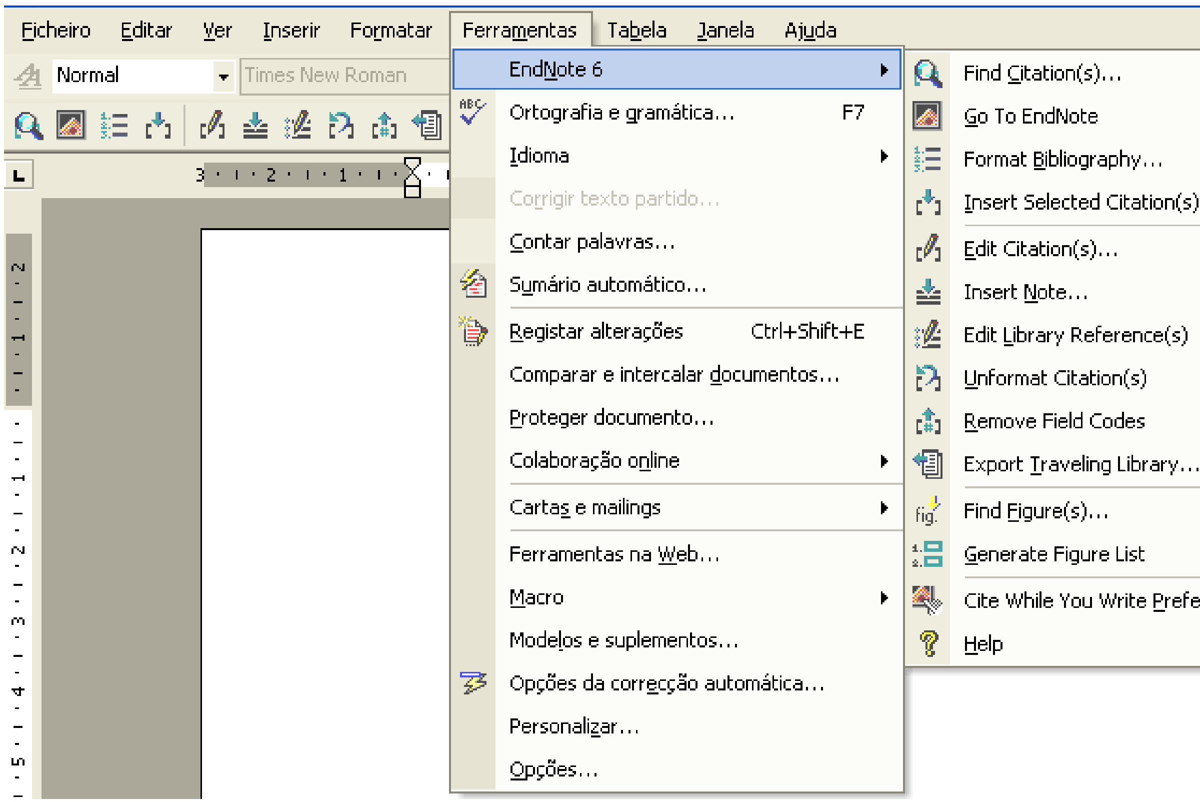


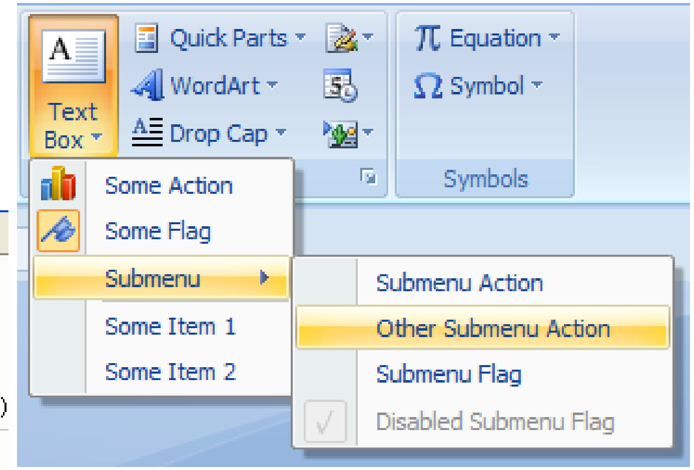


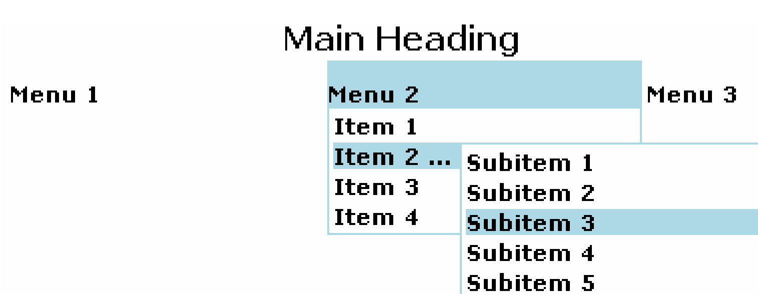


Pull-down/ pull-up menus

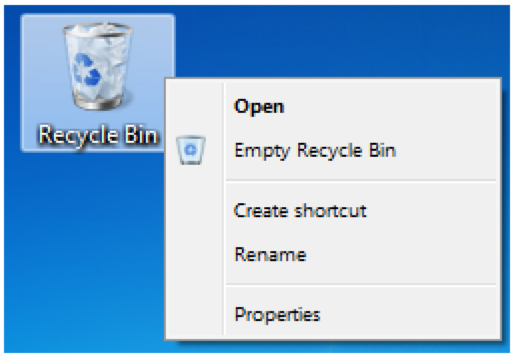
Cascading menus

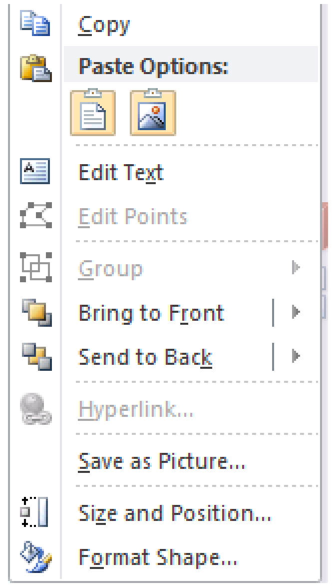


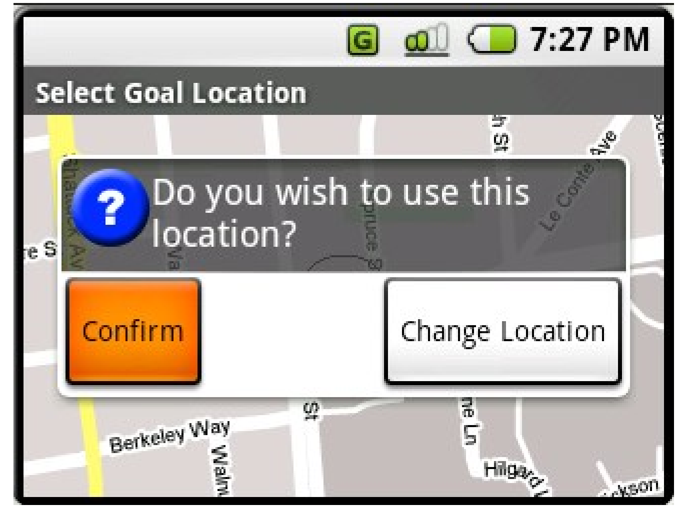


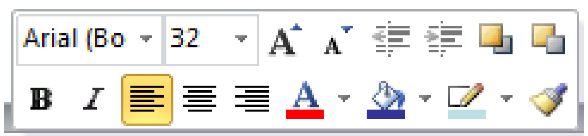


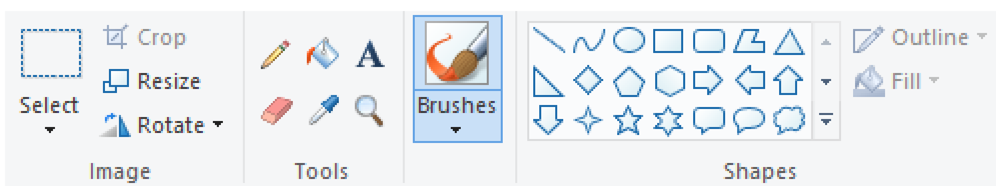
Always visible / *Pop-ups*

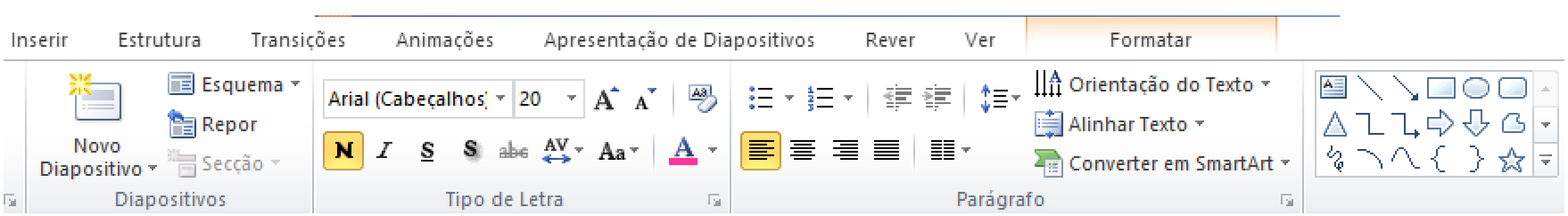












Menus: main advantages and disadvantages

Advantages (potential, i.e. if properly designed)

- Auto-explanatory
- Do not load memory (recognition rather than recall)
- Prevent syntactic errors
- Visible improvements

Disadvantages

- Not efficient
- Not flexible
- Not practical for many options

User profile to whom menus are adequate:

Knowledge and experience:

- Low system and task experience
- Frequent usage of other systems
- Low computational literacy

Work and task:

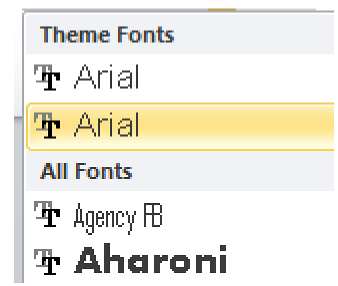
- Low frequency of use
- No training
- Optional usage
- Highly structured tasks

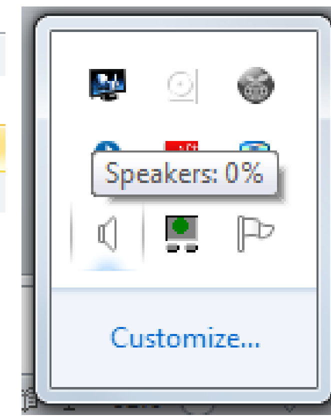
Menu design: relevant aspects

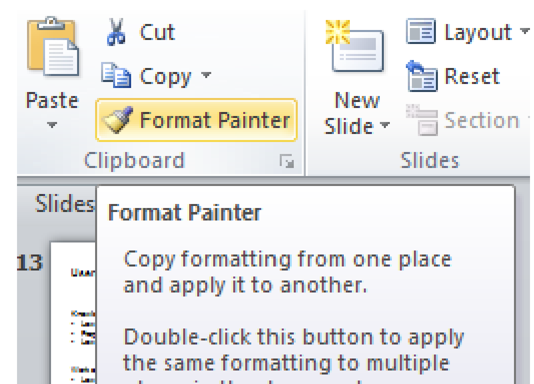
- Menu structure
- Option ordering
- Option selection
- Menu invocation
- Navigation

Menu design: guidelines

- Adequate the menu structure to the task structure
- Minimize depth increasing breadth (within reasonable limits)
- Use an adequate ordering method
- Be coherent (design, option names, etc.)
- Give selection feedback to the user
- Include tooltips if names or icons are not auto-explanatory
- Indicate currently unavailable options
- …







Menu design: guidelines

- Find the adequate structure using card sorting:

  a low-cost method that helps understanding how users expect to find content or functionality

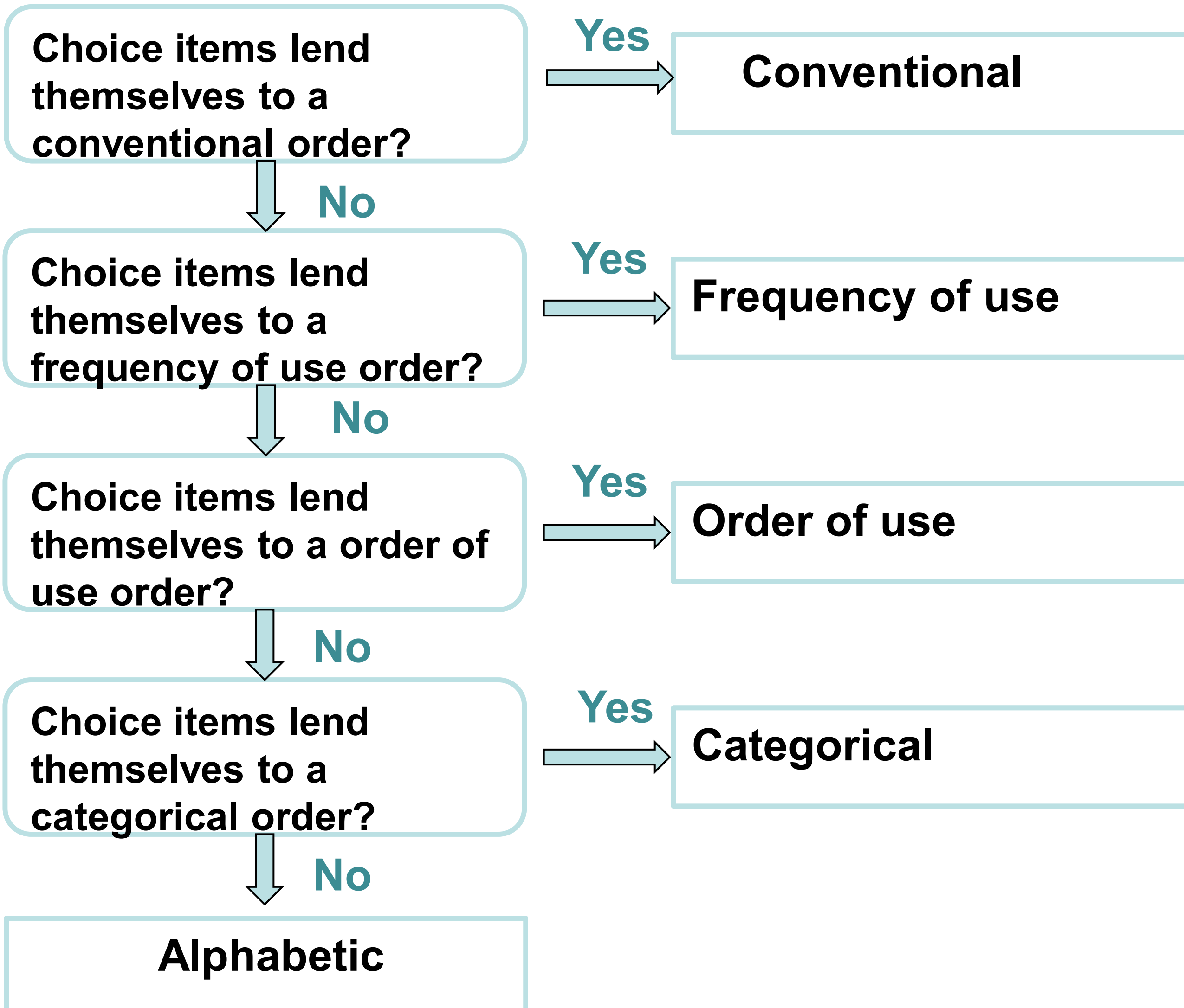
http://www.interaction-design.org/encyclopedia/card_sorting.html

- Card sorting (usually performed by potential users of an interactive solution) provides information on:
  - Terminology (what people call things)
  - Relationships (proximity, similarity)
  - Categories (groups and their names)
- that can be used to decide upon:
  - which items should be grouped together in displays
  - how menu contents should be organized and labeled
  - what words should be employed to describe the objects of our users' attention

Card sorting example: think about how to sort the fruits and vegetables sold in a supermarket (may be it is not as easy as it seems…)

http://www.interaction-design.org/encyclopedia/card_sorting.html

Select adequate option ordering

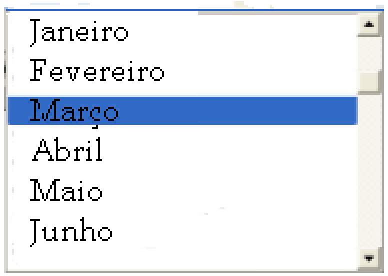


Which ordering scheme would you select?

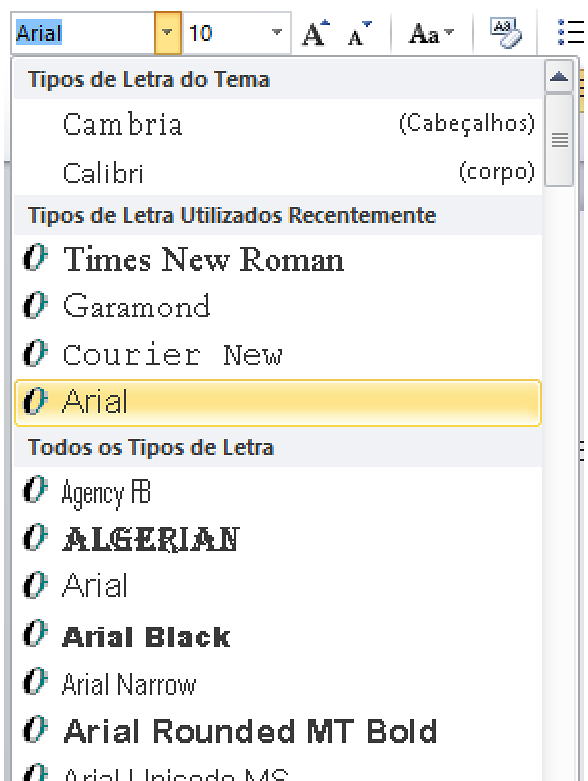
Why?

Option ordering

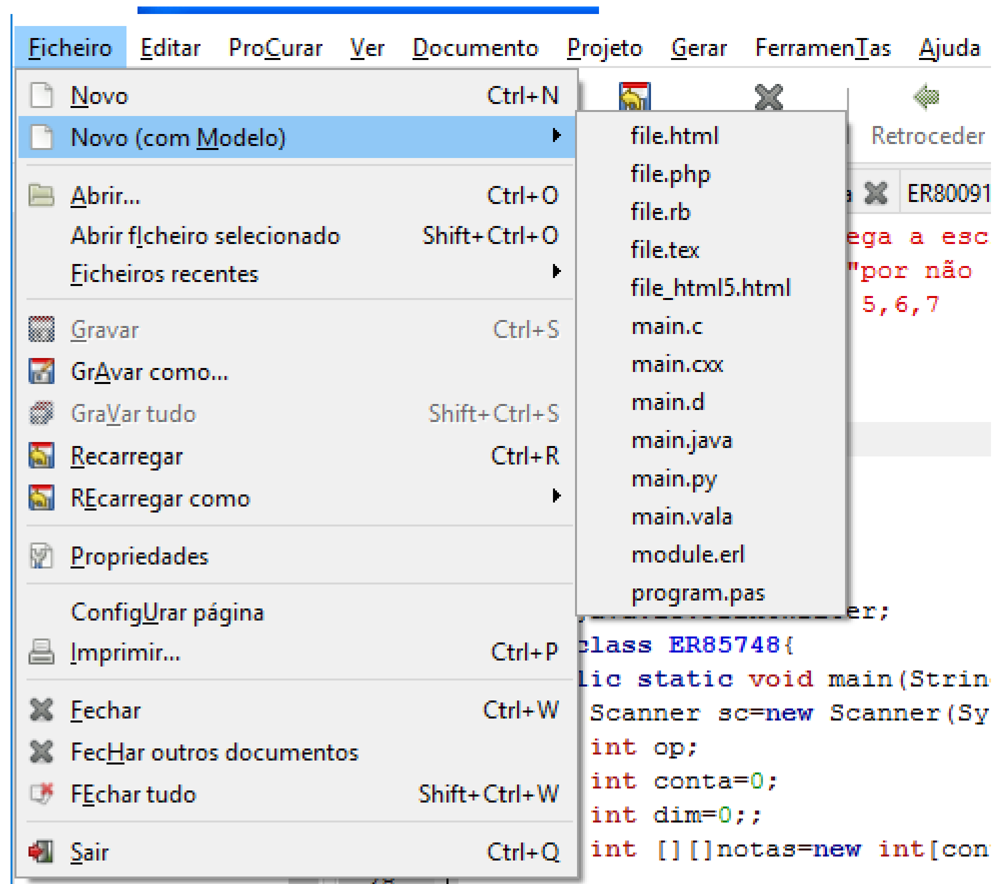
Conventional

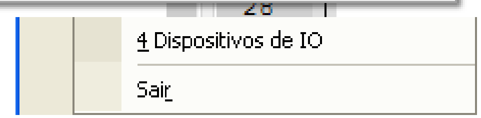


Alphabetic + frequency



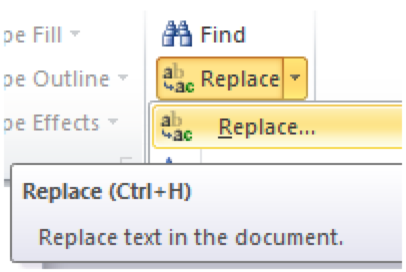
Order of use

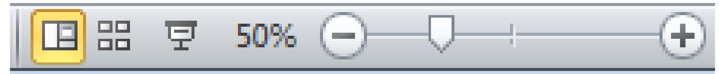


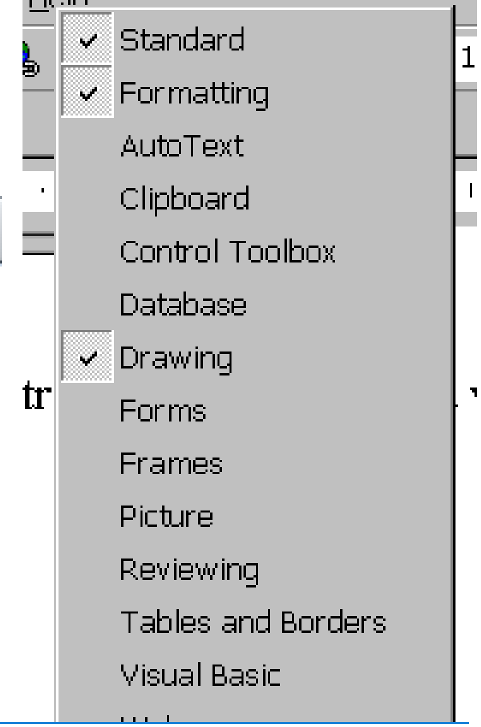


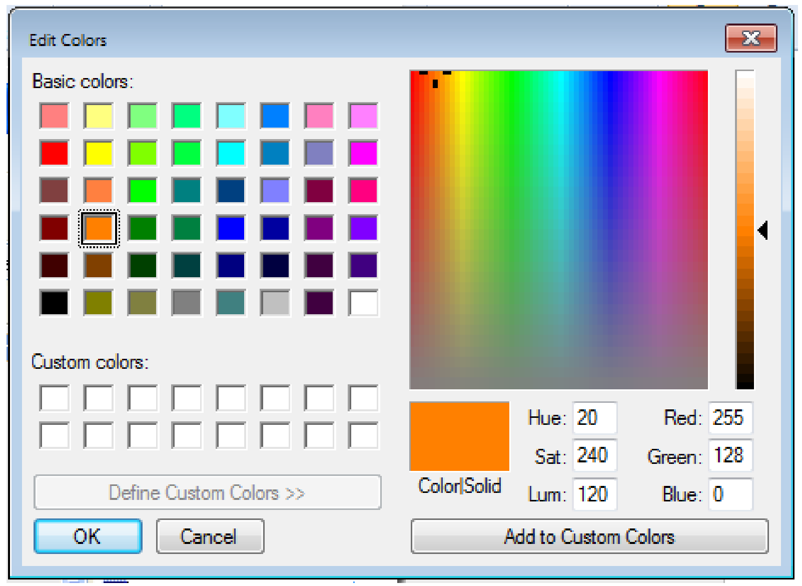
Category

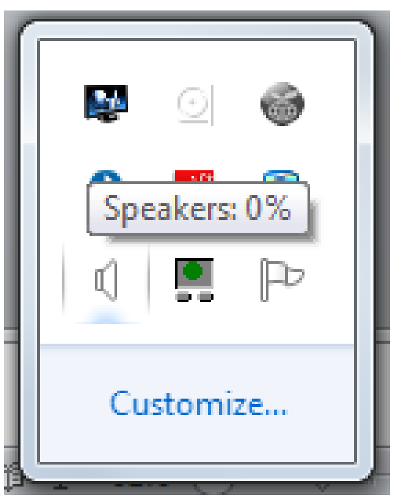
Give selection feedback

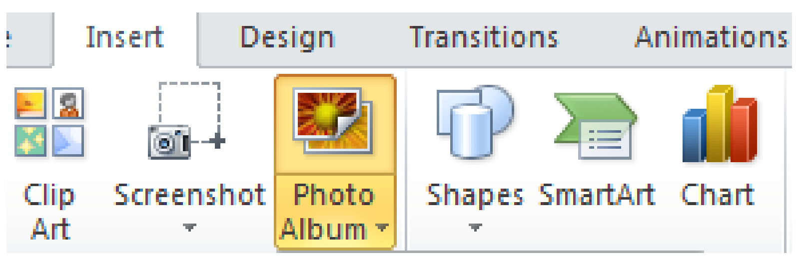


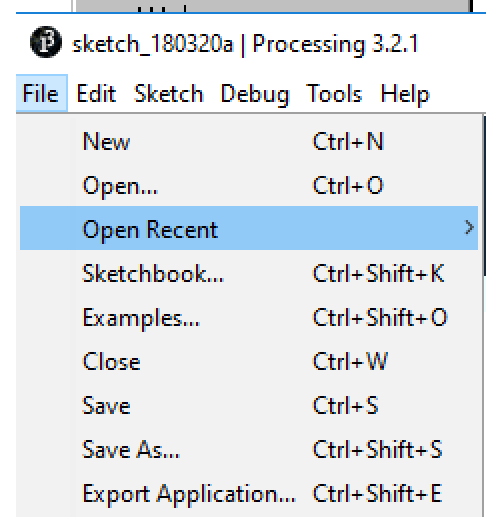








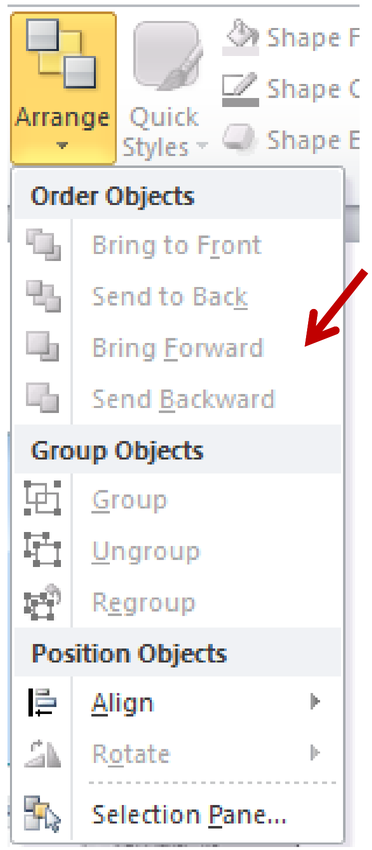


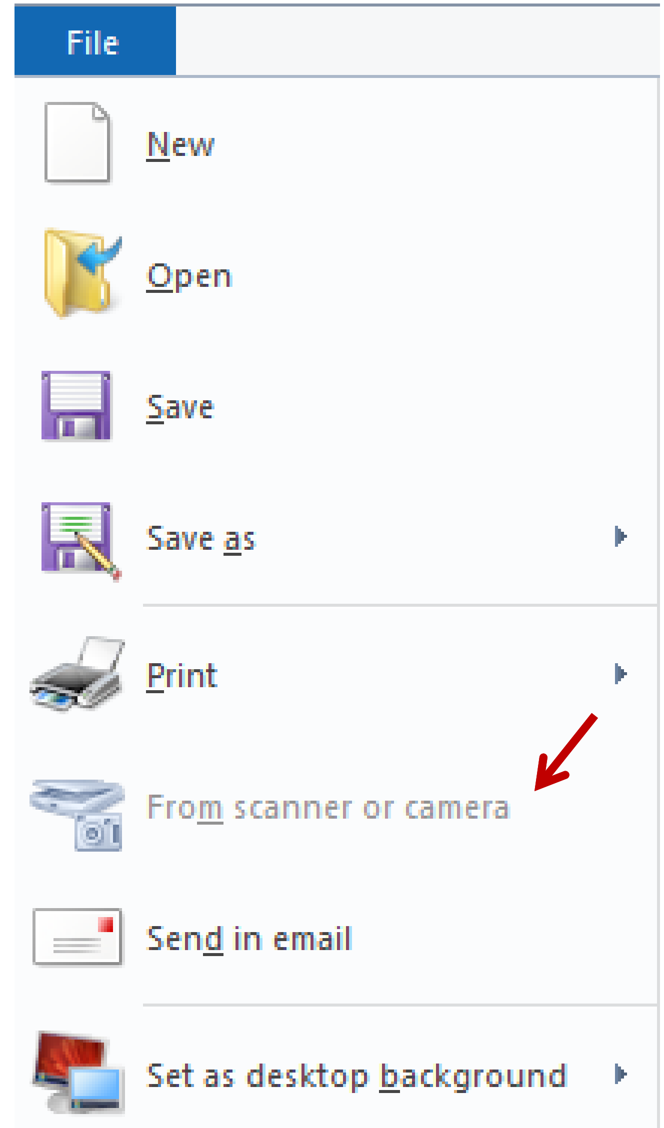


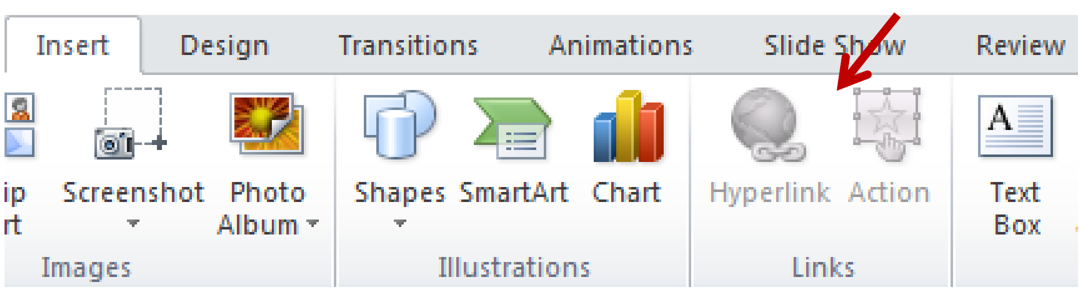
Indicate currently unavailable options

In grey to let users know they exist, but are unavailable

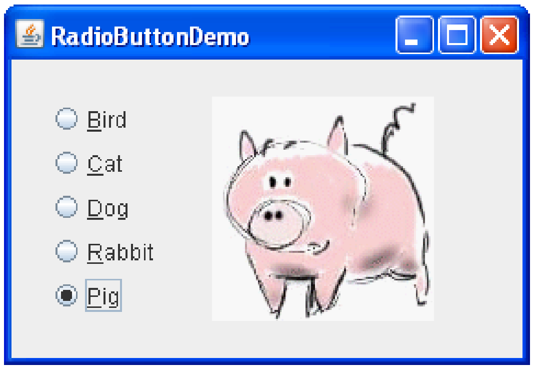
- Preventing errors
- And showing existing options (functionality)





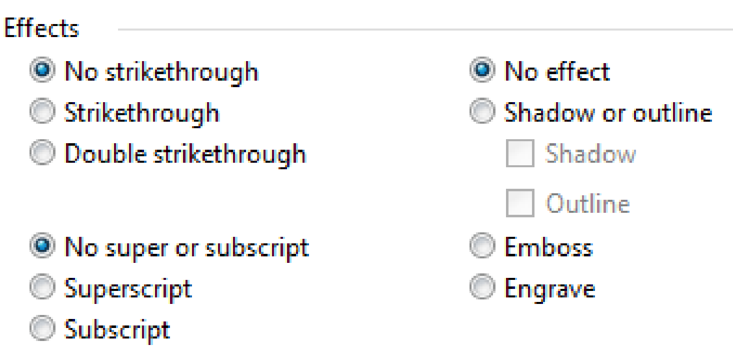


Make clear the difference between choices of only one or several

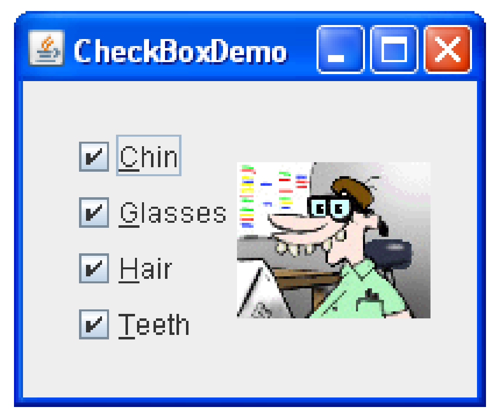


"Radio button":
Only a single option can be selected from several mutually exclusive options

https://en.wikipedia.org/wiki/Radio_button



"Check box":
Permits to make a binary choice.
A series of checkboxes may be presented
The user may select several of the choices



Select only one alternative:

	Service	From	To
	AP No. 180	06:21	08:22
	AP No. 130	07:21	09:22
	IC No. 520	07:31	09:52

Possible to select more than one alternative:

Additional seat options

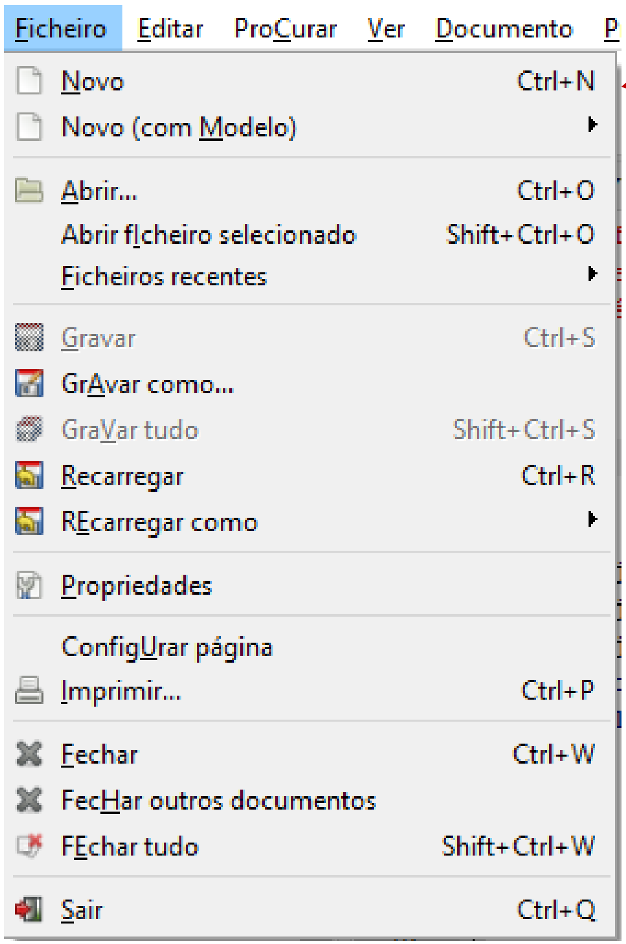
Special needs seat ? Bikes ?

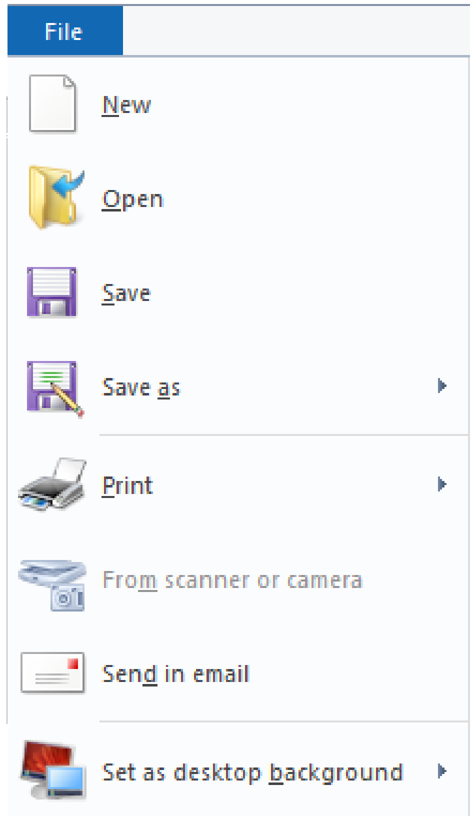
May we send you updates using e-mail?

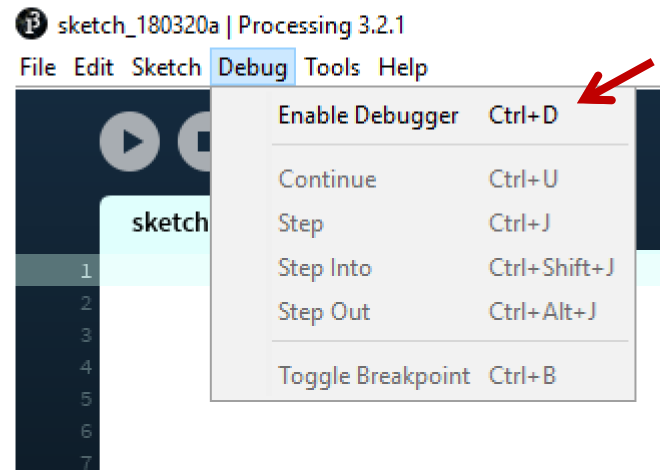
Yes, please use e-mail to send me information about other offerings.

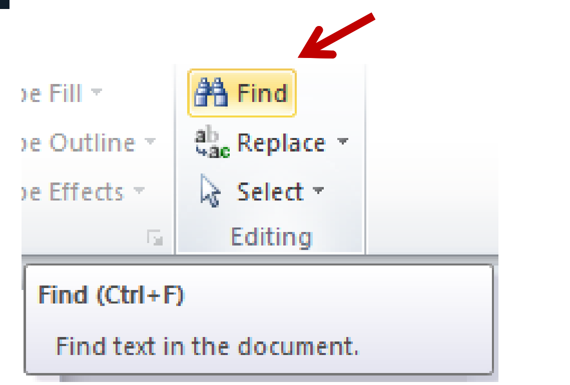
https://www.nngroup.com/articles/checkboxes-vs-radio-buttons/

Show alternative ways of accessing the same functionality





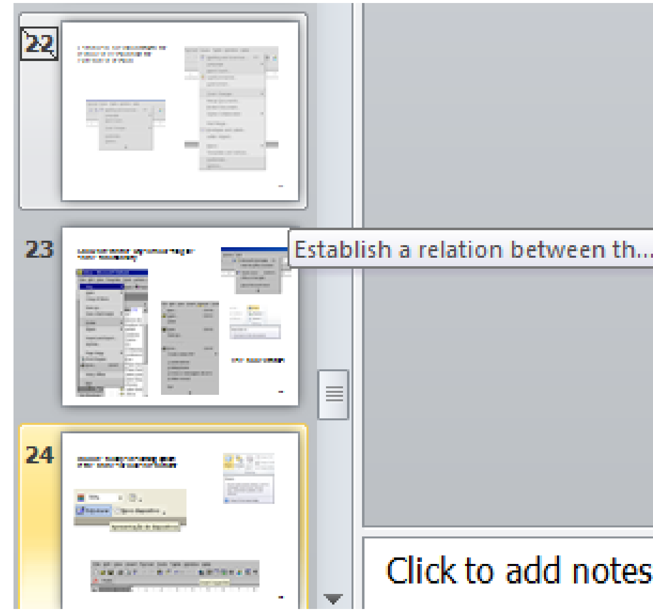


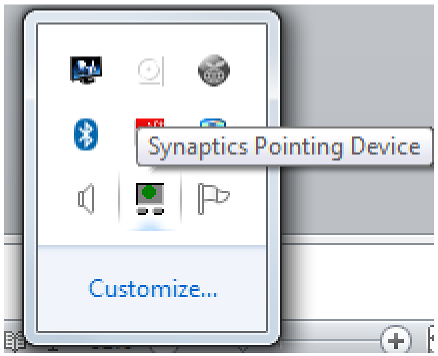


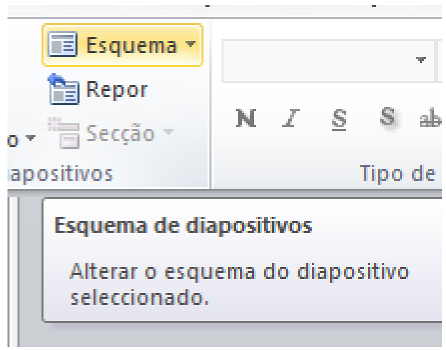
Use accelerators

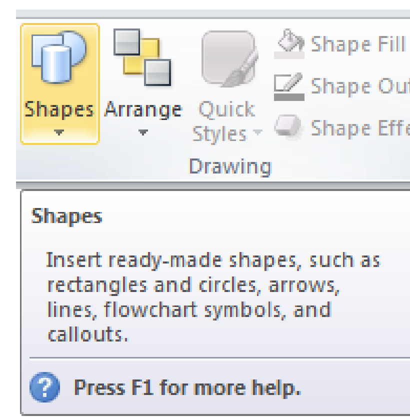
(Flexibility and efficiency of use)

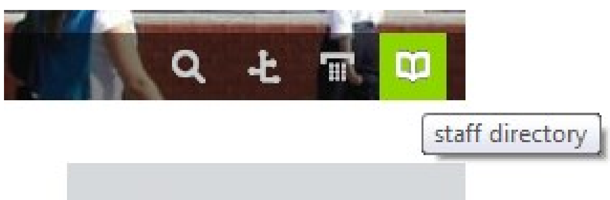
Include tooltips describing options if the names or icons are not clear

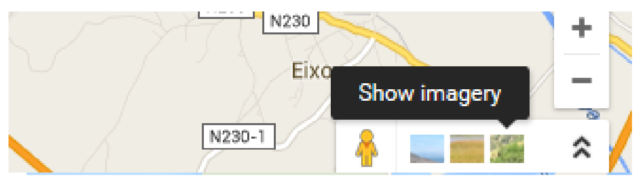


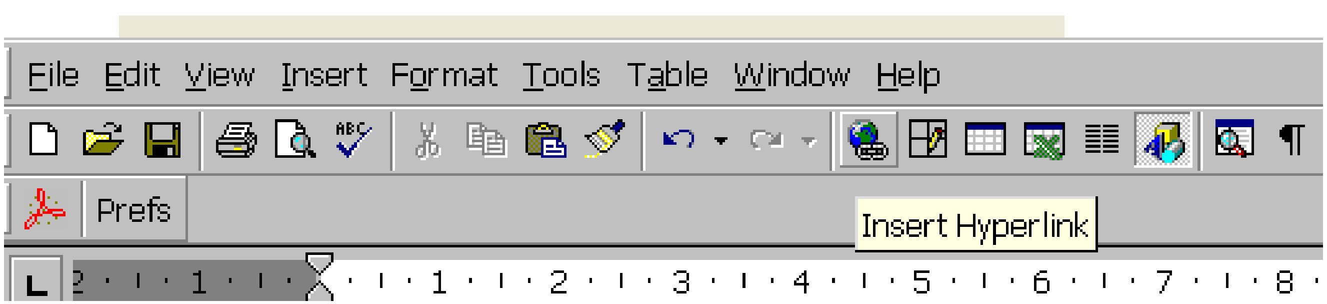


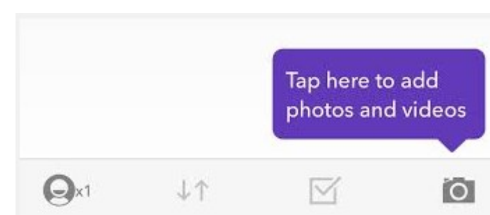




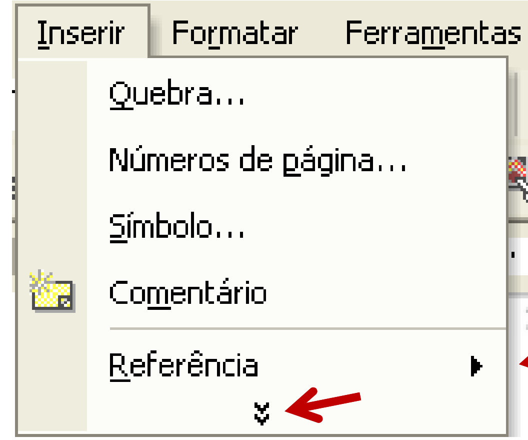


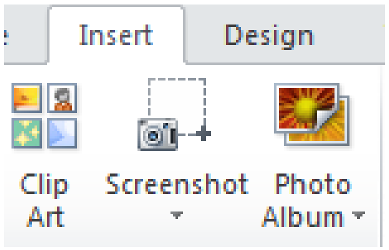


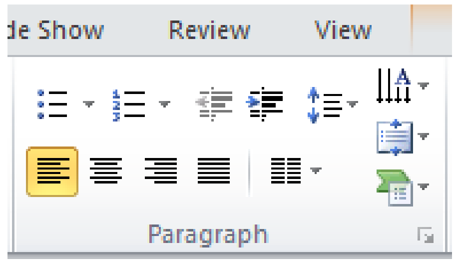


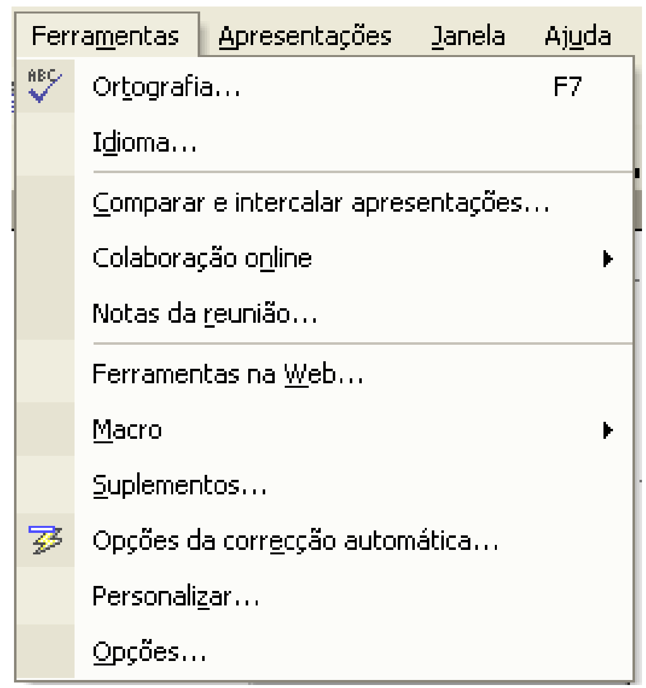


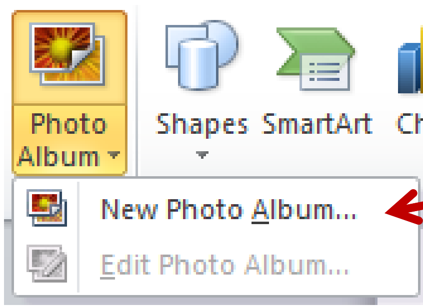
Make clear that there are more options

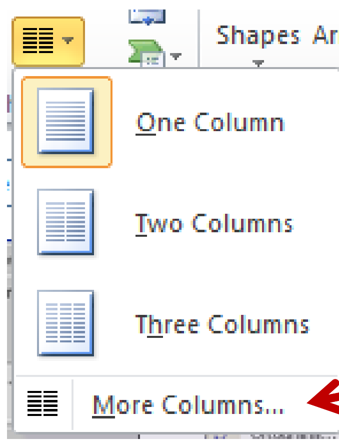




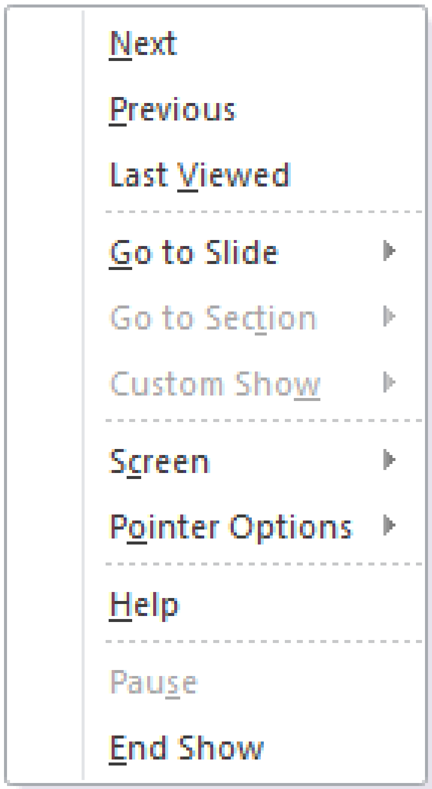


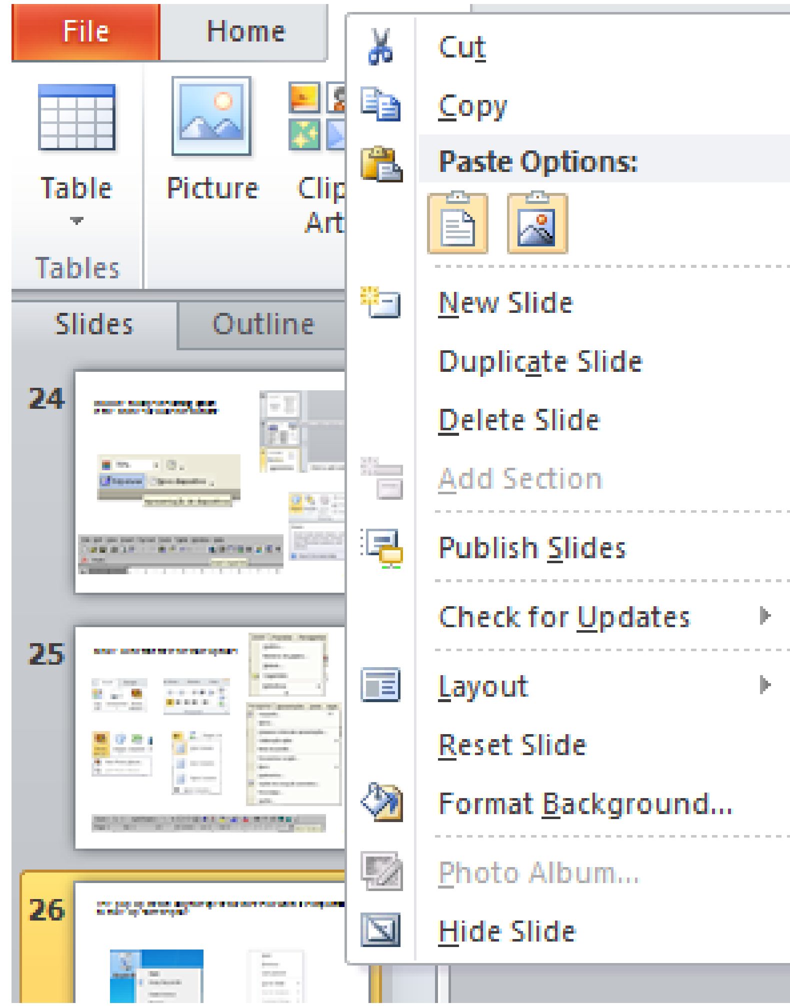






Use pop-up menus (context menus) only for experienced users or when it is very important not to take up screen space





(e.g.:
during a Power Point presentation)

Main Bibliography

- Shneiderman, B., C. Plaisant, M. Cohen, S. Jacobs, *Designing the User Interface- Strategies for Effective Human–Computer Interaction*, 5th ed., Addison Wesley, 2010

- Soegaard, Mads. Interaction Styles (Retrieved March 2020)

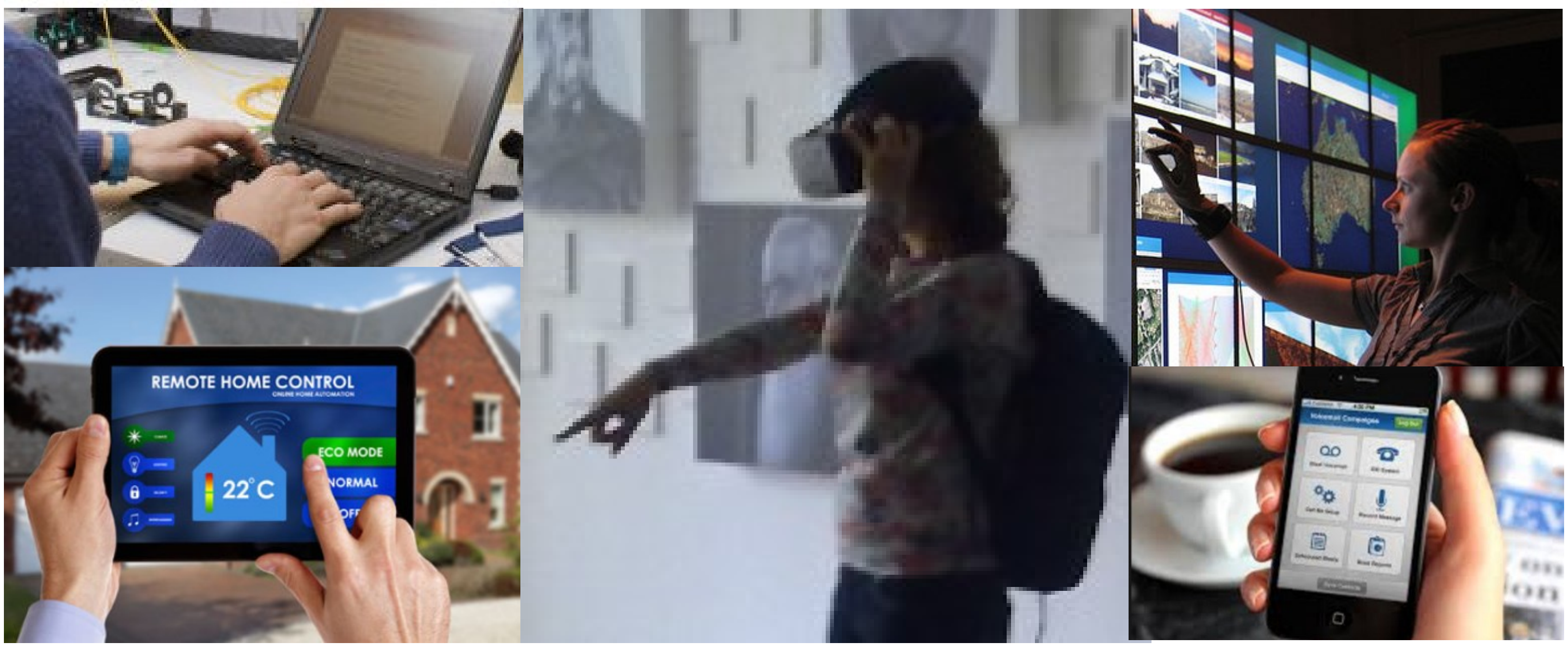
  http://www.interactiondesign.org/encyclopedia/interaction_styles.html

- Hudson, W.,Card Sorting. In: Soegaard, Mads and Dam, Rikke Friis (eds.). "The Encyclopedia of Human-Computer Interaction, 2nd Ed.". Aarhus, Denmark: The Interaction Design Foundation (Retrieved March 2020)

  http://www.interaction-design.org/encyclopedia/card_sorting.html

Universidade de Aveiro
Departamento de Electrónica,
Telecomunicações e Informática

Human-Computer Interaction
2020/21



Beatriz Sousa Santos

Outline

- Introduction
- Course Information
- Lectures and lab classes organization
- Lectures and lab classes schedule
- Assessment
- Bibliography

“ the HCI discipline investigates and tackles all issues related to the design and implementation of the interface between humans and computers. “

“It expanded from early graphical user interfaces to include myriad interaction techniques and devices, multi-modal interactions, …, and a host of emerging ubiquitous, handheld and context-aware interactions”

Carroll, John M., “Human Computer Interaction - brief intro”. In: Soegaard, Mads and Dam, Rikke Friis (eds.). "The Encyclopedia of Human-Computer Interaction, 2nd Ed.". Aarhus, Denmark: The Interaction Design Foundation.
https://www.interaction-design.org/encyclopedia/human_computer_interaction_hci.html

Interaction and Interface

“Roughly speaking, **interaction** refers to an **abstract model** by which humans interact with the computing device for a given task, and an **interface** is a choice of **technical realization (hardware or software)** of such a given interaction model.”

(Kim, 2015)

"The interface between humans and computers is harder than ever to define, we can interact with computers just by walking through a public space."

Sellen, A., Rogers, Y., Harper, R., & Rodden, T., "Human Values in the Digital Age", *Communications of the ACM*, *52*(3), March 2009, pp. 58–66

- “What will Human - Computer Interaction (HCI) be like in 20 years?

- “That question is important because HCI … has a pivotal part to play in the 21st, when computers will become so pervasive that how humans interact with them will be a crucial issue for society”

http://research.microsoft.com/en-us/um/cambridge/projects/hci2020/reading.html

About this course:

Main objectives you should attain:

- understanding of what is the Human-Computer Interaction field

- understanding the importance of the User Interface (UI) of an interactive system;

- knowledge of the fundamental concepts, methods and techniques for the:
    - design
    - implementation
    - evaluation of Interactive Computer Systems

Course information

- Web
  - http://sweet.ua.pt/bss
  - More materials in moodle.ua.pt

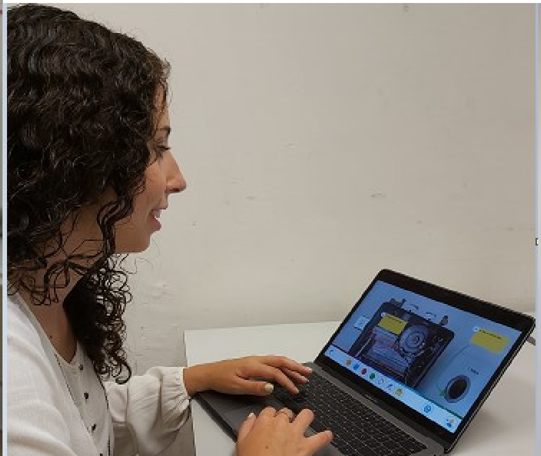
- Team:
  - Beatriz Sousa Santos
    - bss@ua.pt
  - Paulo Dias
    - Paulo.dias@ua.pt
  - Pedro Almeida
    - pma@ua.pt
  - Fábio Barros
    - fabiodaniel@ua.pt

Lectures and Lab classes

Lectures - slides, discussion and paper presentation

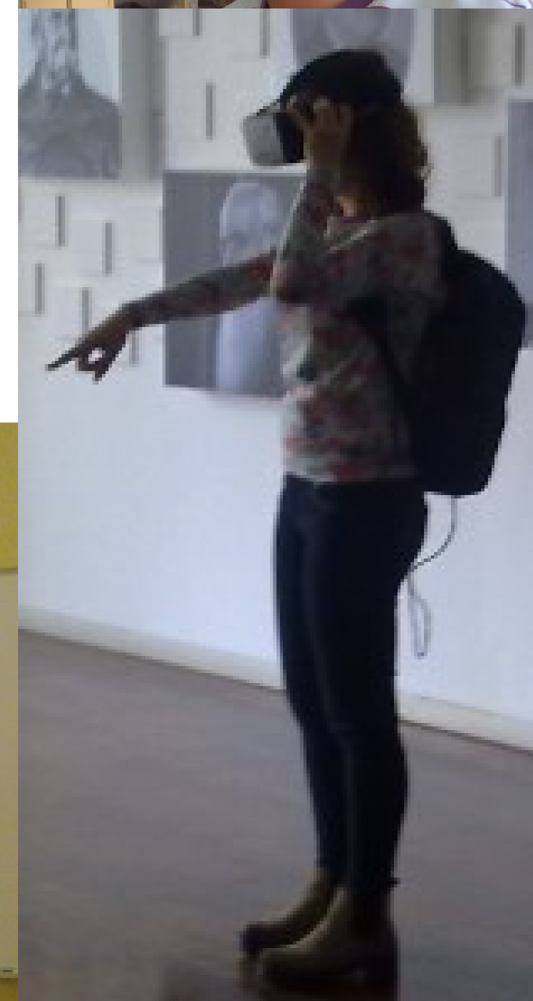
Lab classes – design, implementation and evaluation of User Interfaces (UIs) and interactive systems

- participation in user studies (if possible …)



You will have the opportunity to:

- Learn the fundamentals of this pivotal field
- Attend the presentation of cutting edge research
- Test and use new interaction and display equipment
- Develop for various platforms
    - Perhaps participate in user studies



Attending lectures and lab classes

- Attending lectures will help you in several ways.
- Attending face-to-face lab classes is mandatory, will be registered formally and you cannot pass if you do not have the minimum required (80%).
- Working students must contact faculty members during the two first weeks of the semester

Lectures (subject to minor changes)
Wednesdays

- **1-** Introduction to the course
- **2-** Definition of HCI, User Interface (UI), Usability and UX principles and paradigms
- **3-** The user: the Human Information Processing System (HIPS)
- **4-** The user (cont). Mental models and conceptual models
- **5-** Interaction Styles: Menus and direct manipulation
- **6-** Other Interaction styles
- **7-** Introduction to User-Centered Design and S/W patterns for UIs
- **8-** Models for UI design (user models, task analysis)
- **9-** Models for UI design (cont)
- **10-** Screen layout. Color models and color usage
- **11-** Evaluation methods (more detailed study)
- **12-** Input devices
- **13-** Output devices
- **14-** Introduction to 3DUI: Mixed, Virtual and Augmented Reality
- **15-** Paper presentation

Lab classes (subjected to minor changes)

- Introduction to the Lab classes.
- **Assignment n.1 (evaluate an interactive system) (groups of 3 students)**
- Evaluation of UIs/Interactive systems using analytical methods
- **Presentation and discussion of assignment n. 1.**

- **Assignment n.2 (develop a prototype of an interactive system) (3 students)**
- Human-Centered approach to design and develop interactive systems: requirements analysis; prototyping and evaluation; prototyping and evaluation
- Introduction to Android Studio
- Introduction to Web programming
- **Presentation and discussion of assignment n. 2**

Lectures – online all semester
Lab Classes – online + face to face

Terça	Quarta	Quinta	
	IHC E@D-4 TP2 (TP)	IHC ANF. V P4 (P)	IHC ANF. V P7 (P)
		IHC ANF. V P2 (P)	IHC ANF. V P7 (P)
IHC 04.2.03 04.2.07 P3 (P)	IHC E@D-4 TP1 (TP)	IHC ANF. V P1 (P)	IHC ANF. V P6 (P)
IHC E@D-4 OT1 (OT)			

Pandemic situation

Zoom sessions:

- Configure your zoom name
- If possible associate a photo to your profile

- If the zoom session end after 40 min, use the same link …

Volunteers for a “IHC students Committee” ?

Devices that can be used in the lab classes

Assessment

Final Mark -> Exam (50%) + group assignments (50%)

Minimum mark in each component – 7.5/20

- paper presentation (10%) + assignment n. 1 (10%) + assignment n.2 (30%)

- paper from a conference -> 15 min presentation (grous of 2 students)

- assignment n. 1: **evaluation with analytic methods** -> presentation and discussion (groups of 3 students)

- assignment n. 2: design, implementation and test of a interactive prototype following **User Centered Design** -> presentation, demo, discussion (groups of 3 students)

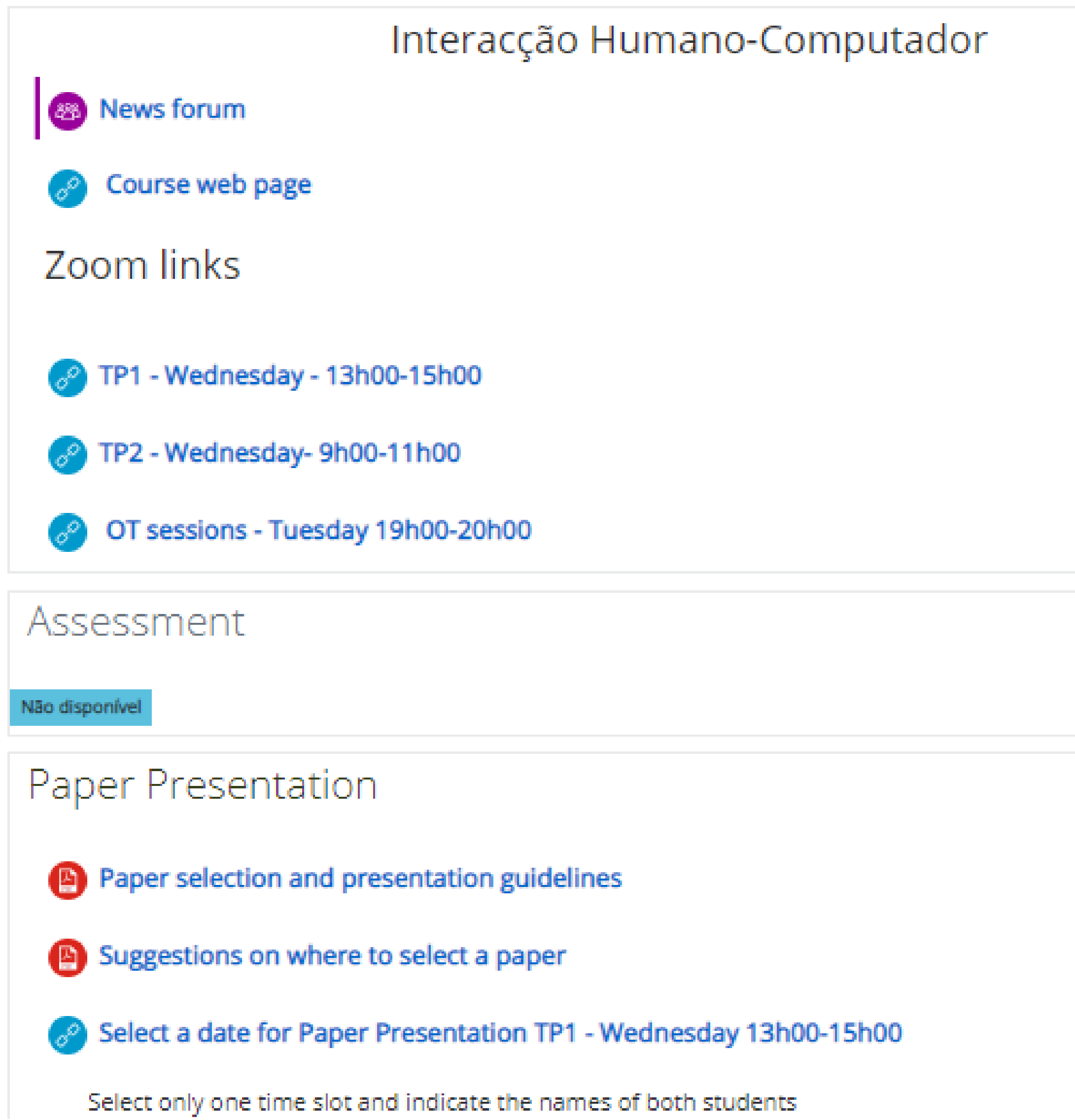
Bibliography

- Sharp, H., Preece, J., and Rogers, Y., Interaction Design- beyond Human-Computer Interaction, Wiley, 2019
- Dix, A., J. Finley, G. Abowd, B. Russell, *Human Computer Interaction*, 3rd. ed., Prentice Hall, 2004
- Kim, G. J., Human–Computer Interaction-Fundamentals and Practice, CRC Press, 2015
- Cooper, A. et al.., About Face 4: The Essentials of Interaction Design, 4th ed., Wiley, 2014
- Shneidermen, B., *Designing the User Interface, Strategies for Effective Human-Computer Interaction*, 6th ed., Addison Wesley, 2016
- Soegaard, M. and, Rikke Friis, D.(eds.). "The Encyclopedia of Human-Computer Interaction, 2nd Ed.". Aarhus, Denmark: The Interaction Design Foundation. https://www.interaction-design.org/encyclopedia/interaction_design.html
- Mitchell, P., *A Step-by-step Guide to Usability Testing*, iUniverse, 2007
- Nielsen, J., *Usability Engineering*, Academic Press, 1993

Portuguese bibliography

- Manuel J. Fonseca, Pedro Campos, Daniel Gonçalves, *Introdução ao Design de Interfaces*, FCA, 2012

Moodle Walkthrough



Paper presentation assignment (groups of two students)

- Wednesday 9h-11h – 35 paper presentations
- Wednesday 13h -15h - 35 paper presentations

This year you may read and present papers from one of these conferences

HRI 2020
CAMBRIDGE / UK

ACM/IEEE International Conference on Human-Robot Interaction

http://humanrobotinteraction.org/2020/

https://mobilehci.acm.org/2020/

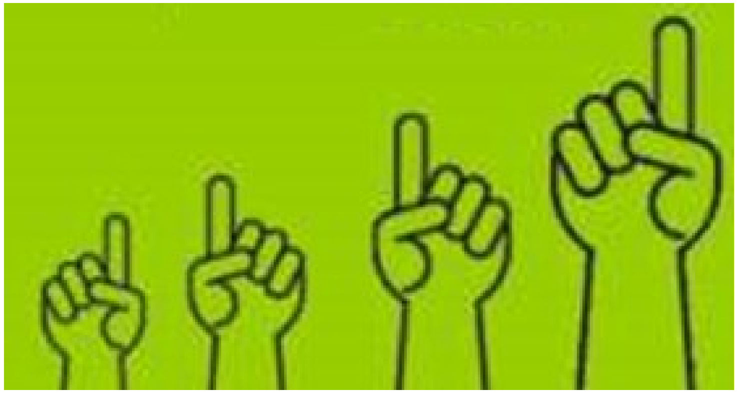
2020 IEEE VR ATLANTA
ATL

March 22nd - 26th

IEEE COMPUTER SOCIETY · IEEE

http://ieeevr.org/2020/

- Volunteers to present a paper next week?



Note that:

- Volunteers have absolute priority in selecting the paper

- And will have this assignment done (10% of final mark) soon in the semester !

Until March 18

Each group of two students should:

- select paper (with >=8 pages) from the conference proceedings
  (HRI2020, MobileCHI2020 IEEEVR2020, ISMAR2020)

- indicate the preferred paper via a form and select the date via doodle

- wait for approval of the paper and date (posted on Moodle)

- read the paper presentation guidelines (available at the course web page)

- prepare a 15 min presentation (~15 slides)

- submit the slides to bss@ua.pt before the lecture at the defined date

“ the HCI discipline investigates and tackles all issues related to the design and implementation of the interface between humans and computers. “

Some Present and Future trends:

Gesture interfaces

Virtual and augmented reality, Mixed reality

Large public displays

Brain-computer interfaces

Human-Robot interfaces

Natural Conversational Speech Interfaces

Affective States and Human-Computer Interactions

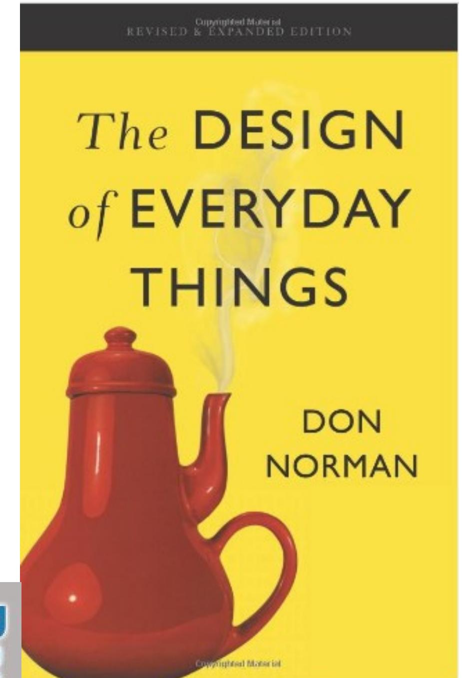
...

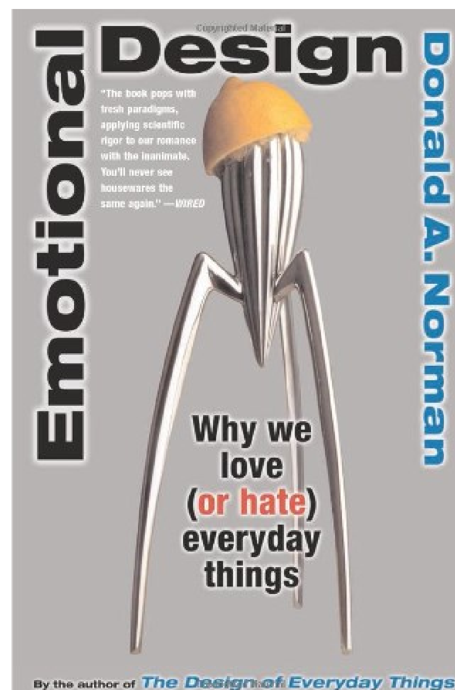
P Montuschi, P., Sanna, A., Lamberti, L, and Paravati, G., "Human-Computer Interaction: Present and Future Trends," *Computing Now*, vol. 7, no. 9, September 2014
http://www.computer.org/web/computingnow/archive/september2014

Interesting books

- Donald Norman, The design of everyday things, Basic Books, Revised Edition, 2013

- Donald Norman, Emotional Design: Why We Love (or Hate) Everyday Things, Basic Books, 2010





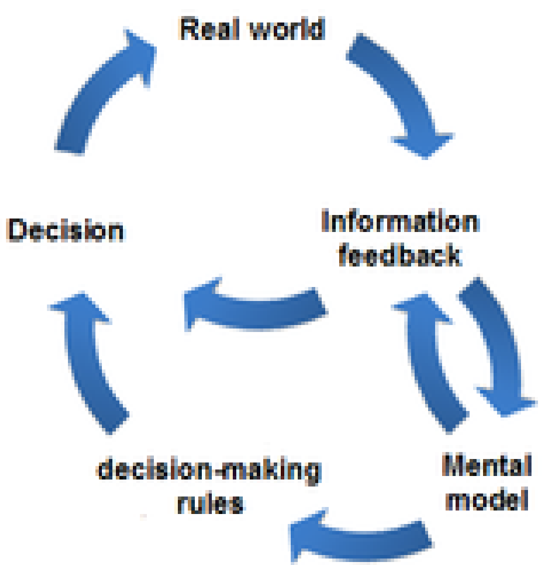
- For next week:

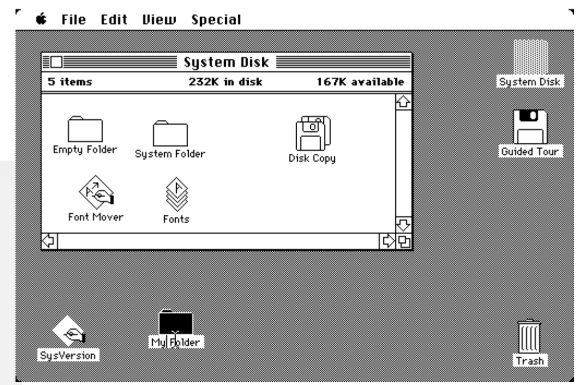
  - Select the presentation dates you prefer via doodle
  - And the papers you prefer via google form
  - Think about two interactive systems/applications to evaluate

Good luck with your work
in this course and have fun!

Universidade de Aveiro
Departamento de Electrónica,
Telecomunicações e Informática

Mental and conceptual models





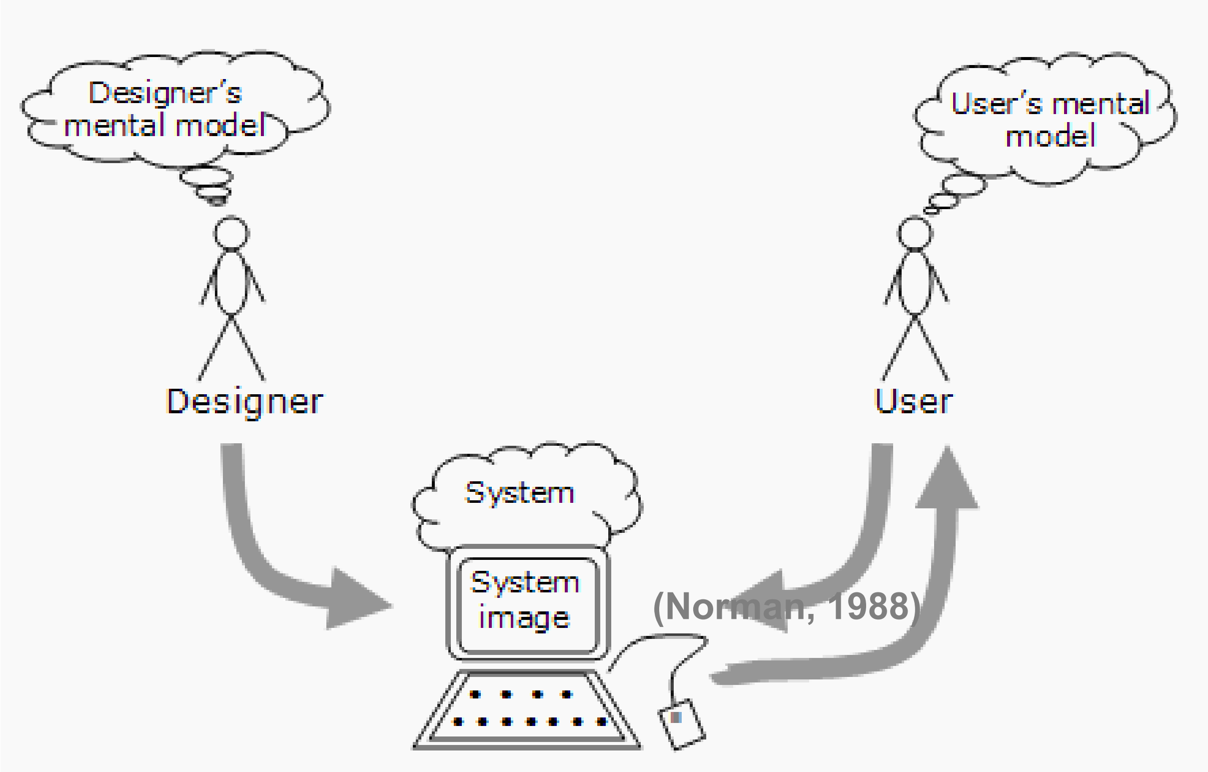
Relevant issues:

- What are mental models?
- How do we construct them?
- What is known about them?
- What are they used for?
- What are conceptual models?
- Guidelines to obtain good conceptual models

- The conceptual model is the UI highest level
- The conceptual model is the conceptual framework in which the functionality is provided to the user
- To understand how to design a good conceptual model it is necessary to understand mental models
- A mental model (in a simple way) is the user’s internal representation of the current conceptualization and understanding of the system
- A conceptual model is the designer’s atempt to foster good mental models through UI aspects

- The user develops a **mental model** of how he/she thinks the system works

- And uses it to:
  - reason about the system
  - anticipate system behavior
  - explain why the system reacts as it does



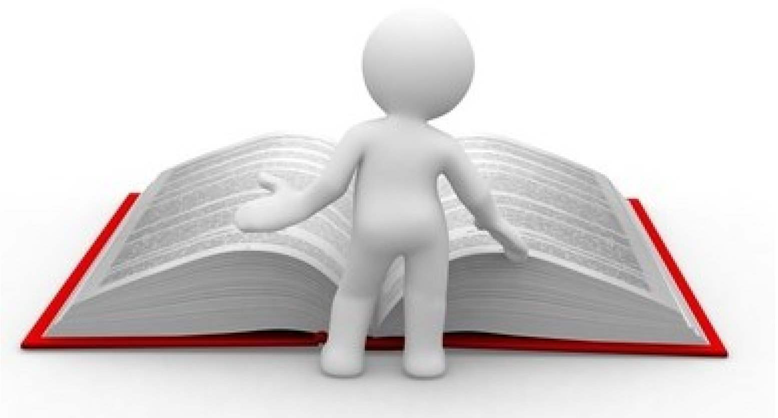
(Norman, 1988)

https://www.interaction-design.org/literature/book/the-glossary-of-human-computer-interaction/mental-models

How do we create a mental model?

- Using the system
- Observing others using the system
- Reading documentation
- …

and thus all these are important ways to train the user to use a system



Mental models allow:

- Make predictions
- Determine causes of observed events
- Determine adequate actions to produce the wanted changes
- Understand analogous devices
- …

“What users believe they know about a UI strongly impacts how they use it. Mismatched mental models are common, especially with designs that try something new.”

https://www.nngroup.com/articles/mental-models/

Which button shall I press?

Example: remote controller of some projectors @ DETI

- This device has a different UI from the others I am used to
  (not complying with the consistency and standards heuristic)

- I will try to infer how to use it based on the mental models I have

- But it is ambiguous and it does not give prompt feedback
  (not complying with the visibility of the system status heuristic)

- Determining adequate actions to produce the wanted changes fails!

I press one button, nothing happens, I press the other ...

… low efficiency, low efficacy, low satisfaction

poor usability and UX

Mental models:

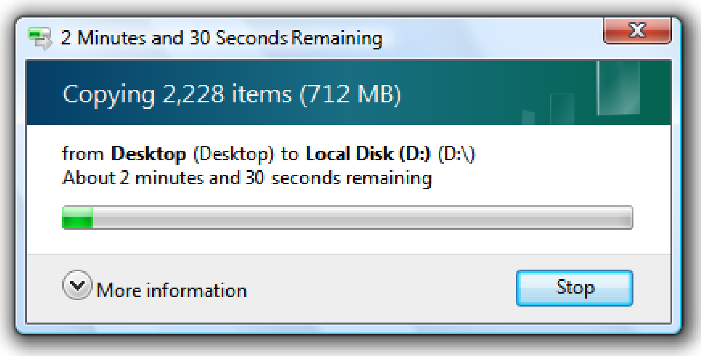
- Are incomplete
- Are unstable
- Are not scientific
- don´t have specific limits
- …

“We must give up finding elegant mental models, and instead learn how to understand the incomplete and confuse structures people have”

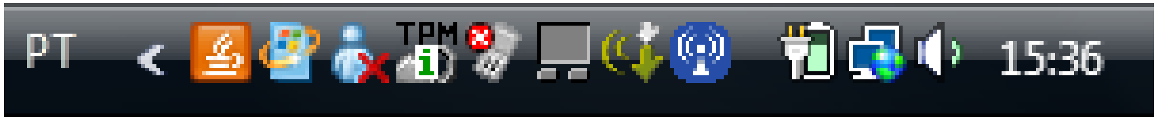
Donald Norman

Main guidelines to obtain a good conceptual model
(that fosters a good mental model)

- Make visible invisible parts and processes



- Give *feedback*

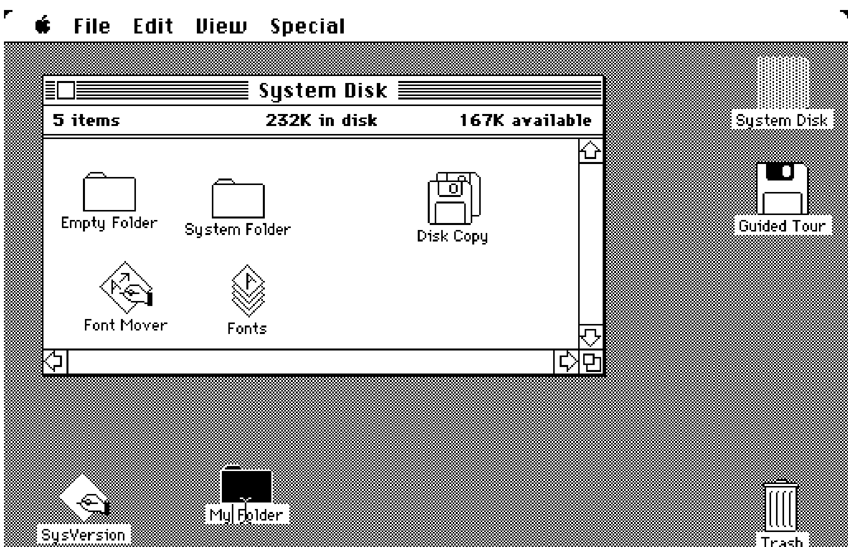


- Use coherence (colors names, command sintax, dialog styles, information location on the screen, etc., etc.)

- Use a metaphor (optional)

All this may help the user to understand better how the system works

Desktop metaphor



Metaphors

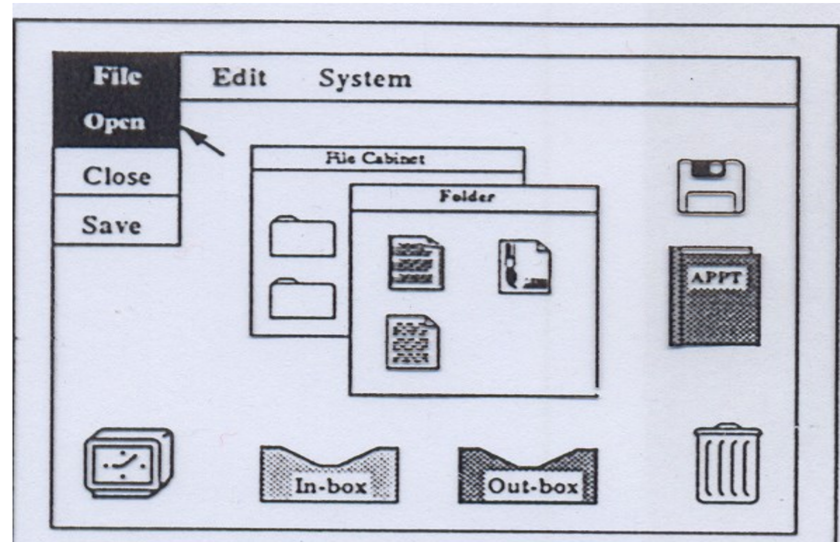
- Exploit existing mental models of the real world

Metaphors can be misleading since the “the essence of metaphor is understanding and experiencing one kind of thing in terms of another” (Lakoff and Johnson 1983)

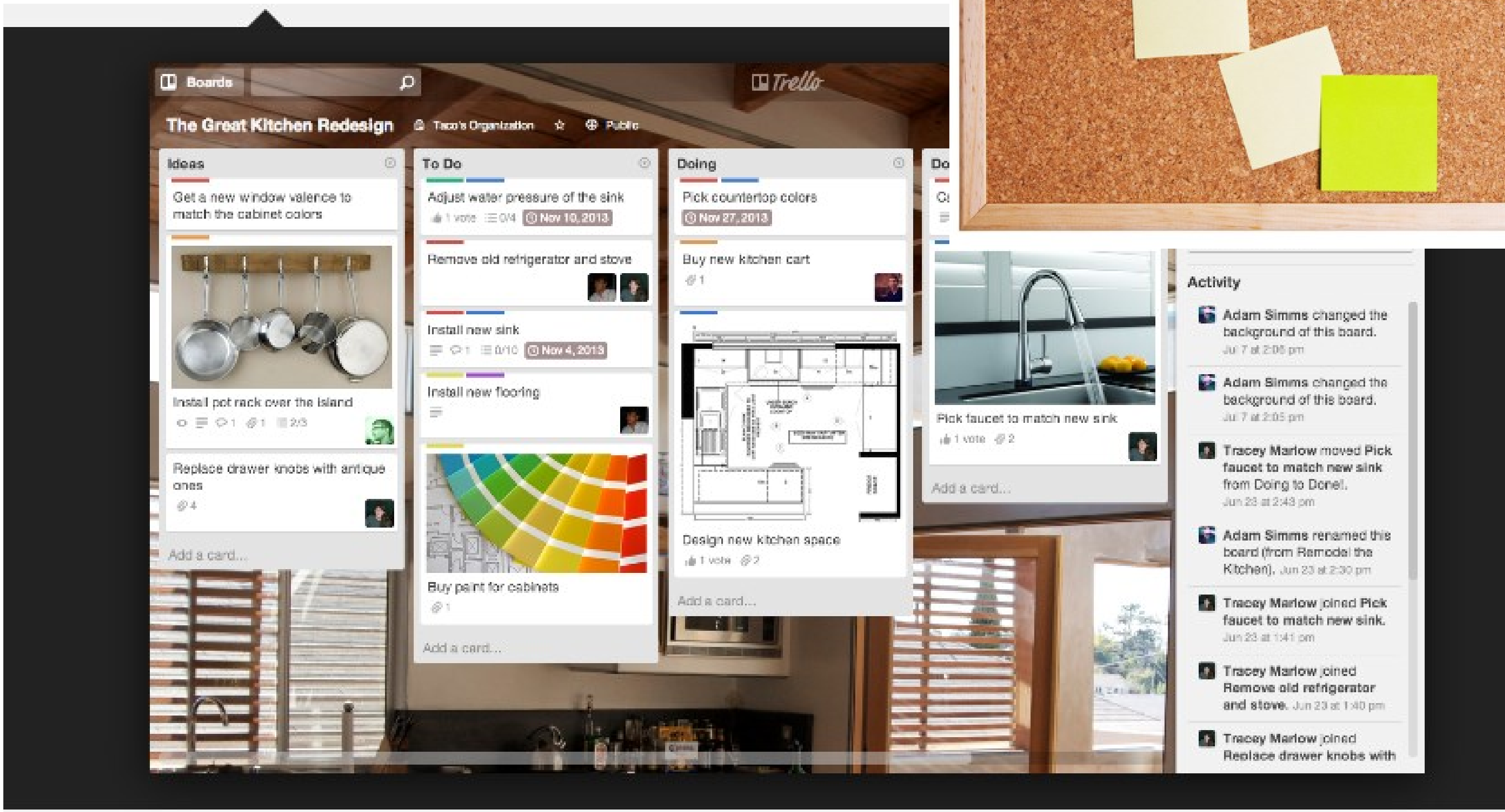
- Which, by definition, makes a metaphor different from what it represents or points to

https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/human-computer-interaction-brief-intro

The Desktop metaphor:

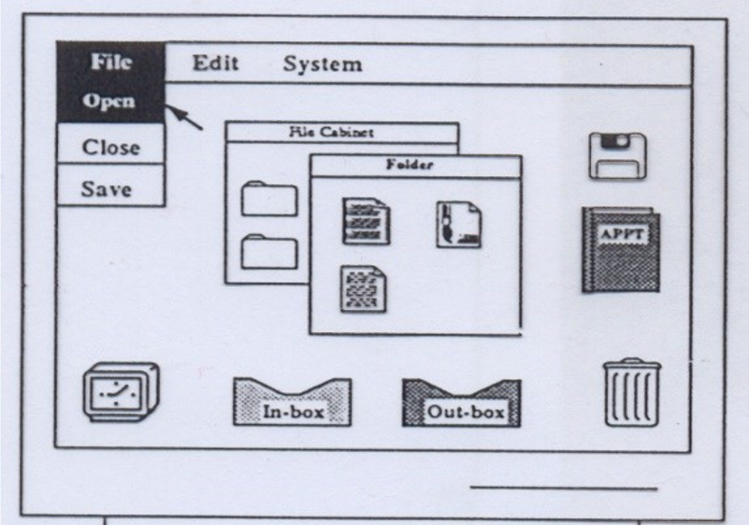


Another example:
the bulletin board (Trello)



https://trello.com/tour

Potential problems in using metaphors



• Incomplete metaphors may confound the user

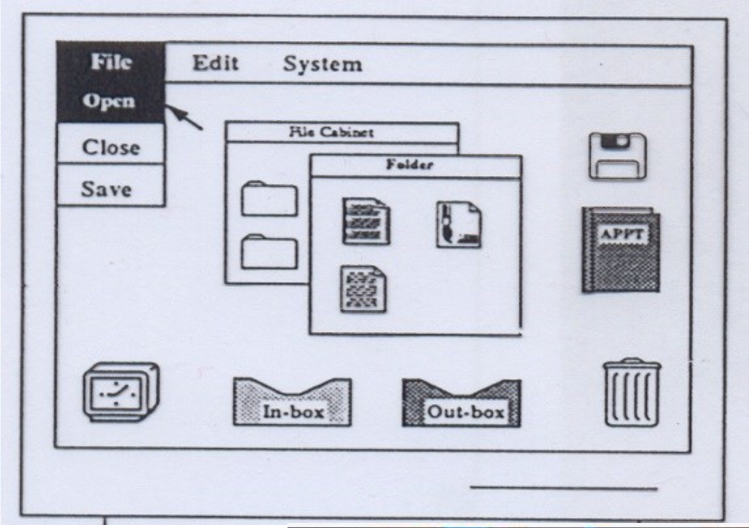
• Risk of under-utilization of the system's capacities

• Less experienced users (e.g. children) seem to expect more "literal" metaphors

• Sophisticated users seem to expect more "magical" metaphors

Thus, the use of a metaphor should be carefully pondered …

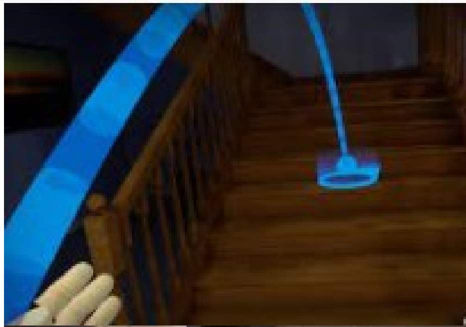
Examples of using metaphors



• In Apple's original desktop metaphor:

remove the diskette from the system ->

-> drag it to the recycling bin !! (unlike the real world…)



• Navigation in VR systems:

Magical metaphor -> "Teleportation"

Less magical (more literal) -> "physically" walking

Conceptual models - summary

- A conceptual model is a **high-level description** of a product in terms of:
  - what users can do with it
  - the concepts they need to understand how to interact with it
- Developing a conceptual model involves:
  - Understanding the problem space
  - Specifying how the proposed design will support users
- Conceptual models **must foster good mental models**
- Paradigms, visions, theories, models, and frameworks
  - Provide ways of framing design and research

- Prepare the first assignment presentation and submit the slides according to the guidelines and recommendations in Moodle:

Lab 3 - Analytical methods: Cognitive walkthrough

Assignment Presentation guidelines and examples

- P1, P2, P4, P5, P6, P7 – April, 8   P3 - April, 13

- 2 or 3 volunteer groups to present tomorrow in each Lab class

For next week:

- Think about what you would like to develop in your mini-project, in doubt talk with us!

Lab 5 - Presentation of the analytical evaluation assignment

Limit for selection of the 2nd assignment topic

2nd assignment - Development of an interactive application

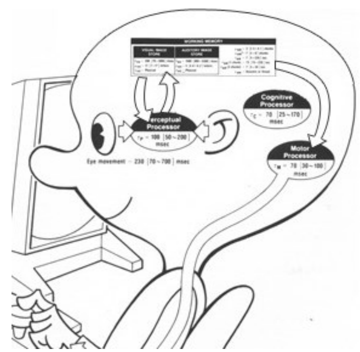
2nd Assignment - topic selection

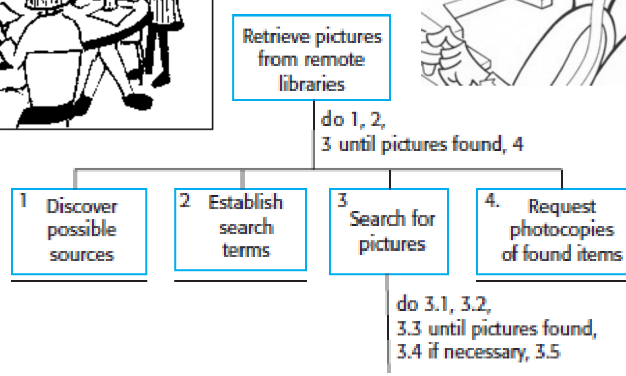
Main bibliography

- Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale, *Human-Computer Interaction*, 3rd edition, Prentice Hall, 2004
- Jenny Preece and Helen Sharp, *Interaction Design – Beyond Human-Computer Interaction*, 5th edition, John Wiley, 2019 (http://www.id-book.com/chapter3_teaching.php)
- John Carroll, Human Computer Interaction - brief intro, *The Encyclopedia of Human-Computer Interaction,* 2nd edition https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/affective-computing
- Mental Models, *The Encyclopedia of Human-Computer Interaction,* 2nd edition https://www.interaction-design.org/literature/book/the-glossary-of-human-computer-interaction/mental-models
- Jakob Nielsen, Mental Models, https://www.nngroup.com/articles/mental-models/

Universidade de Aveiro
Departamento de Electrónica,
Telecomunicações e Informática

Models for design





- All engineering fields use models:
  - To evaluate – does the design have the needed characteristics?
  - To prescribe – directly contribute to the design
- Models are needed also in UI design
- Several types of models may be used throughout the design of user interfaces:
  - User models
  - Task analysis
  - Dialog notation
  - System models

I- User Models

- Models to obtain user requirements in their social and organization context

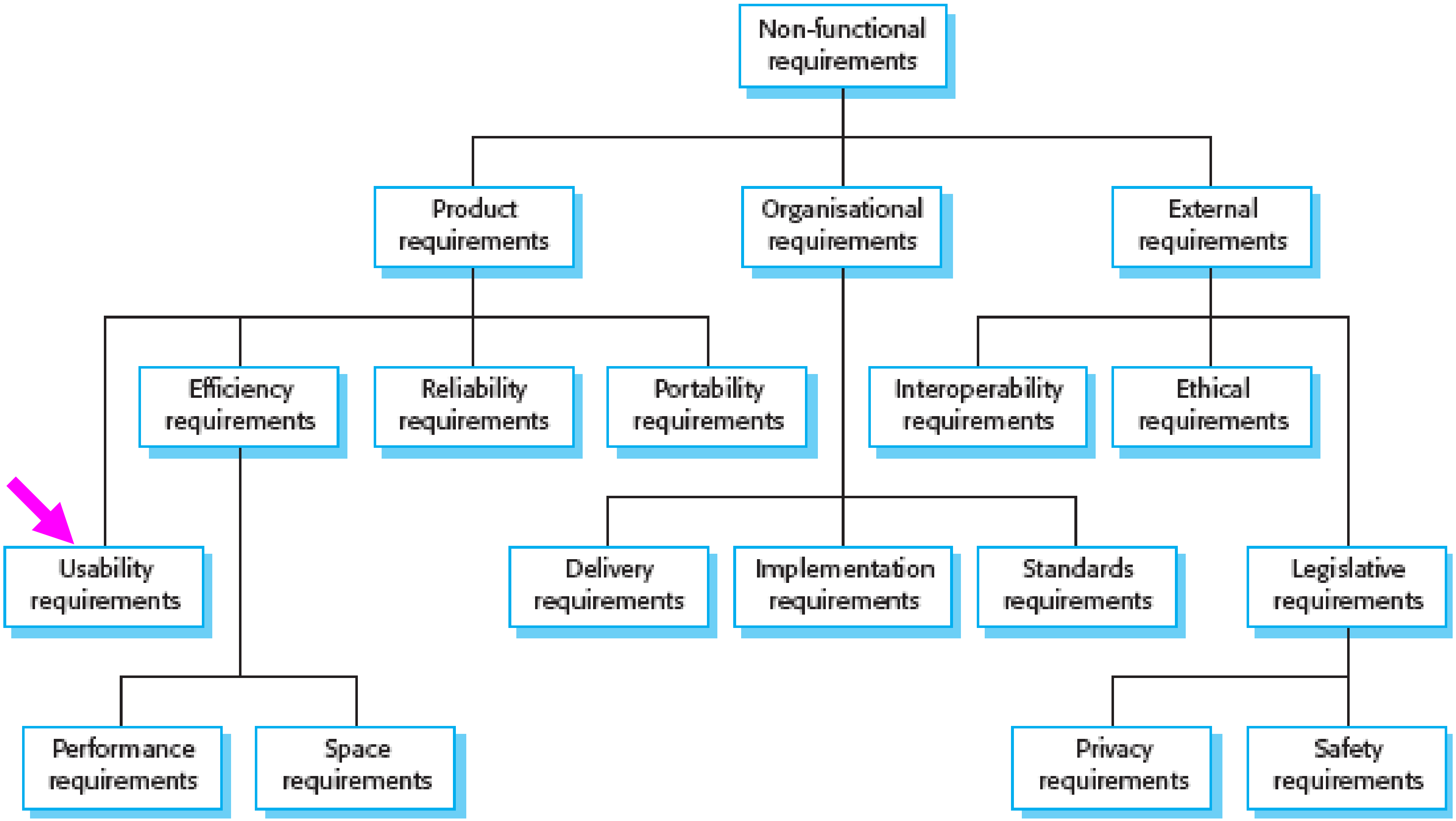
- User models- of the users’ mental, perceptual and motor processes

- Personas- fictional characters based upon research in order to represent the different types of users

Models to obtain user requirements

- Obtaining user requirements is very important in S/W engineering
- But often focuses on functional requirements: what the system must do
- Overlooking non-functional requirements, as if it is:
    - acceptable
    - usable
- There are several models to capture user requirements

Non-functional requirements



(Sommerville, 2016)

Participatory design

- Users are involved as domain experts (e.g. business representatives and users) along the complete process and work together with developers to design a solution
  - It is work oriented and not system oriented
  - It is collaborative- users contribute to all phases
  - It is iterative – design is evaluated and reviewed in every phase

Participatory design

- Uses a set of techniques (that can be used in other contexts) to help transfer information from users to designers:
  - Brainstorming
  - Scenarios
  - Story boarding
  - Workshops
  - Paper and pencil exercises ...

http://infodesign.com.au/usabilityresources/participatorydesign
https://www.usabilityfirst.com/usability-methods/participatory-design/index.html

Personas

- Use a set of techniques (that can be used in other contexts) to help transfer information from users to designers:
  - Based on research
  - Represented as individual people
  - But represent groups of users
  - Explore ranges of behavior
  - Must have motivations

https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/personas

Cognitive models

- Represent the user in an interaction with the system; i.e. model aspects of user knowledge, intentions or processing

- The representation level varies from model to model, from:

  High level models...→ motor activity

- There are several types of cognitive models:
  - Object and tasks hierarchies

    (GOMS- Goals, Operators, Methods and Selection) ✓
  - Linguistic models
  - Physical and device

    (KLM- Keystroke Level Model) ✓

GOMS- Goals, Operators, Methods and Selections

- Proposed by Card, Moran and Newell, 1983
- A GOMS decomposition has the following elements:
- ***Goals***: what the user wants to attain
- ***Operators***: basic operations that the user has to perform to use the system; may affect the system or not (press a key or read a message)
- ***Methods***: possible decompositions of the goal into sub-goals (e.g. Select an option “Save” or press “ctrl S”)
- ***Selections***: rules to select the possible methods (taking into account the type of user and the system status)

GOMS- Goals, Operators, Methods and Selections

- A typical GOMS analysis consists in decomposing a high level goal in a sequence sub-goals

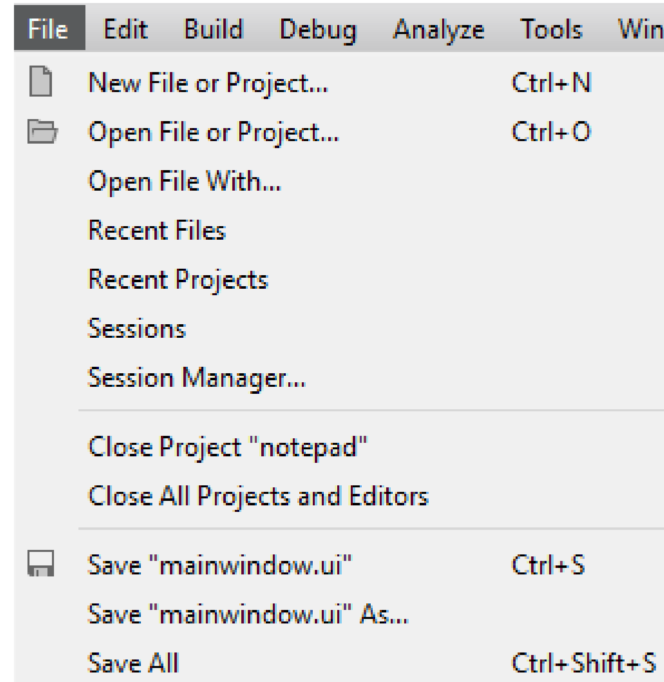
- Selection rules must be adjusted to the user profile

- Analyzing the structure of the GOMS decomposition may give an **approximate measure** of :

  – Short Term Memory load (depth of the goal structure)

  – Time needed (a time for each operator)

Example: ‘save’ a file: using two common ways

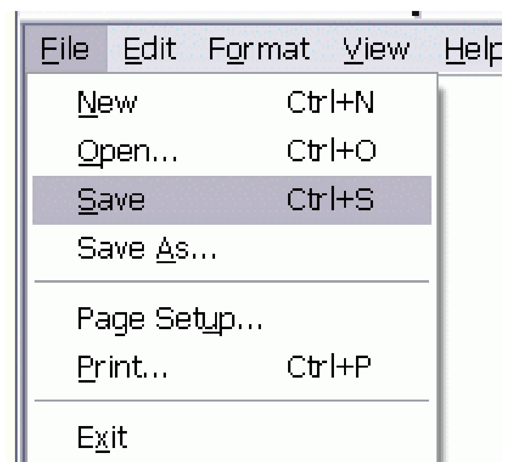


- GOAL: **SAVE-A-DOCUMENT**
  - . [select GOAL: USE-SAVE-OPTION-METHOD
  - . . MOVE-MOUSE-TO-MENU-BAR
  - . . CLICK-OVER-FILE-OPTION
  - . . MOVE-MOUSE-TO-SAVE-OPTION
  - . . CLICK-SAVE-OPTION
  - . GOAL: USE-CTRLS-METHOD
  - . . PRESS-’CTRL’+’S’-KEYS]

User BSS:

Rule 1: **USE-CTRLS-METHOD** unless other rule applies

Rule 2: If has hand on mouse **USE-SAVE-OPTION-METHOD**



Another example: Copy an article from a journal

- Goal: Photocopy-paper
- .   Goal: Locate-article
-            Goal: Photocopy-page repeat
-                        Goal: Orient-page
-                                    open cover
-                                    select-page
-                                    position-paper
-                                    close-cover
-                        Goal: Press-copy-button
-                        Goal: Verify-copy
-                                    locate-out-tray
-                                    examine-copy
-            Goal: Collect-copy
-                        locate-out-tray
-                        remove-copy (outer goal satisfied)
-            Goal: Retrieve-journal
-                        open-cover
-                        remove-journal
-                        close-cover

Closure problem
(the user attains the goal before the task is complete)

The "closure problem"

The copies usually are available to the user before they remove the original from the photocopier and walk away!

To prevent this, the overall goal should be satisfied only after removing the original

In earlier ATMs the money was given before returning the card

... many users left the card:
their goal was getting money!

This was changed.

The “closure problem” in MultiBanco

In stores usually these are the following steps:

- Insert the card
- Insert the pin code
- Transaction approval -> audio signal
- Remove the card
- Receipt is handed to the client

At the ATMs the money is given (goal satisfaction) only after the card is removed by the client

These procedures help not to forget the card!

GOMS- Goals, Operators, Methods and Selections

- Capacities:
  - It has been used in cognitive model research
  - It may describe adequately how **experienced users** perform **routine tasks**
  - Associated to a device model allows time estimates

  Limitations:
  - It does not give information concerning user knowledge to estimate training or transfer times

KLM- Keystroke-Level Model

- Proposed by Card, Moran e Newell, 1980
- Predicts user performance based on motor system characteristics
- Models unitary interaction tasks (simple command sequences <20s) (e.g. change the font of a word, use search and replace)
- These tasks have two phases:
  - Acquisition (building the mental representation of the task)
  - Execution (using the system)
- KLM only models the execution phase (the user has already decided how to use the task)

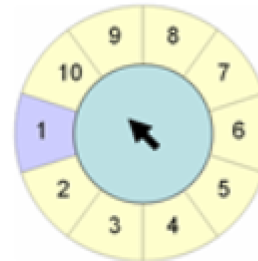
KLM- Keystroke-Level Model

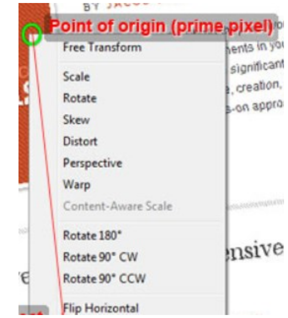
- The execution phase may be decomposed in 7 operators:
  - K- *Keystroke* (varies with typing skill)
  - B- *Button press of the mouse*
  - P- *Pointing at a target* (Fitts' law)
  - H- *Homming between mouse and keyboard*
  - D- *Drawing using mouse*

  motor

  - M- *Mentally preparing for physical action* → mental
  - R- *System Response* (often may be ignored) → system

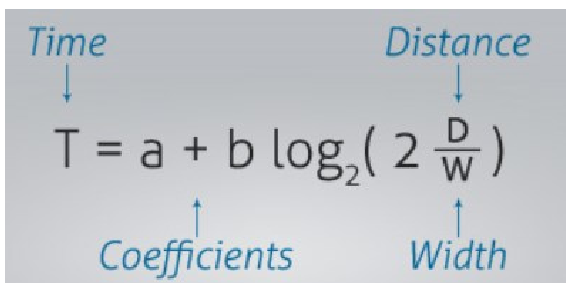
Fitts’s law (1954)

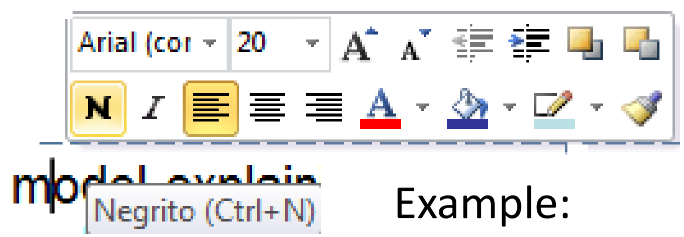




- Empirical model explaining <u>speed-accuracy tradeoff </u>characteristics of human muscle movement with some analogy to Shannon’s channel capacity theorem

- Estimates the average time a user takes to select a target considering the distance (D) from the cursor and the Width (W) of the target:





Example:
Minimizing D

- The larger the target the easier to select (no fine control needed)

- The farther the target from the cursor the longer it will take

http://www.interaction-design.org/encyclopedia/fitts_law.html

Example

Using a mouse based editor – correct a character

Point at the wrong character, delete it, write a new character, return to the original place

1- Move hand to mouse → H [mouse]

2- Place the cursor after the error → PB [left]

3- Return hand to keyboard → H [keyboard]

4- Delete wrong character → MK [delete]

5- Write correct character → K [char]

6- Replace cursor → H [mouse] MPB [left]

Adding all times:

$$T\ total = 2T_k + 2T_P + 2T_B + 3T_H + 2T_M$$

This is an **estimate of the total time** the user will take to perform the task

Times for KLM operators:

(empirically established and may be vary for different types of users, e.g. naïve or experienced)

Operator	Remarks	Time (s)
K	Press key	
	good typist (90 wpm)	0.12
	poor typist (40 wpm)	0.28
	non-typist	1.20
B	Mouse button press	
	down or up	0.10
	click	0.20
P	Point with mouse	
	Fitts' law	$0.1 \log_2(D/S + 0.5)$
	average movement	1.10
H	Home hands to and from keyboard	0.40
D	Drawing – domain dependent	–
M	Mentally prepare	1.35
R	Response from system – measure	–

wpm = words per minute

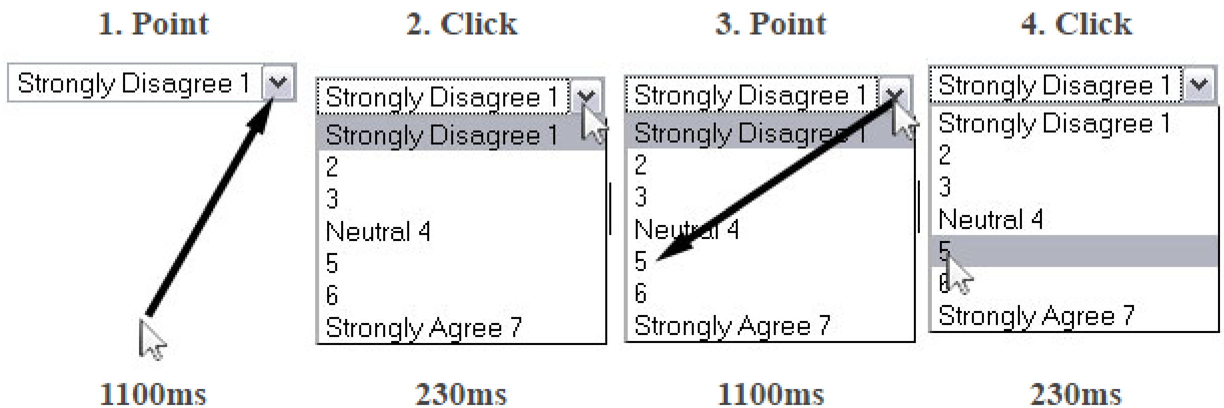
(Dix et al., 2004)

Another example: which survey alternative takes less time?

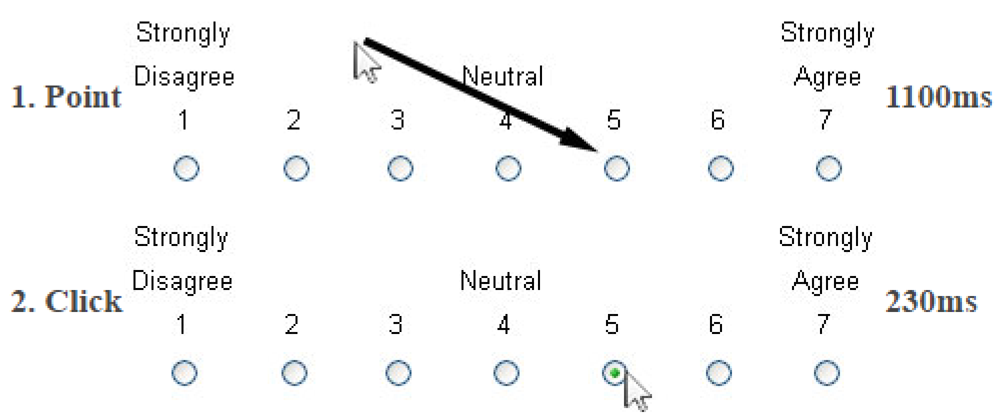
- Drop down list

  or

- Radio buttons?



- The radio button option will probably take ~**half the time**



- **If a task is done repeatedly small changes to an interface can save a lot of time!**

https://measuringu.com/predicted-times/

More examples:

Design A: drag the file into the trash can[29]	Design B: use the short cut "control + T"[30]
method encoding (operator sequence)[31]	method encoding (operator sequence)[32]
1. initiate the deletion (M) 2. find the file icon (M) 3. point to file icon (P) 4. press and hold mouse button (B) 5. drag file icon to trash can icon (P) 6. release mouse button (B) 7. point to original window (P)	1. initiate the deletion (M) 2. find the icon for the to-be-deleted file (M) 3. point to file icon (P) 4. press mouse button (B) 5. release mouse button (B) 6. move hand to keyboard (H) 7. press control key (K) 8. press T key (K) 9. move hand back to mouse (H)
Total time	Total time
3P + 2B + 2M = 3*1.1 sec + 2*.1 sec+ 2*1.35 sec = 6.2 sec	P + 2B + 2H + 2K + 2M = 1.1 sec + 2*.1 sec + 2*.4 sec + 2*.2 sec + 2*1.35 sec = 5.2 sec

This shows that Design B is 1 second faster than Design A, although it contains more operations.

https://en.wikipedia.org/wiki/Keystroke-level_model

- KLM has been extended to mobile

Paul Holleis, Friederike Otto, Heinrich Hussmann, and Albrecht Schmidt. 2007. Keystroke-level model for advanced mobile phone interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '07).

KLM- Keystroke-Level Model

- Placing times T_M is related to chunking and the type of user
- This model has an applicability limited to micro-dialog
- **It allows only approximate results**; thus reasonable estimates concerning the user are enough
- Can predict a skilled user's task time (error-free) to within10-20% of the actual time.
- Its main application is **alternative comparison**

II- Task analysis

- What it is and how it can be performed (more or less formally)

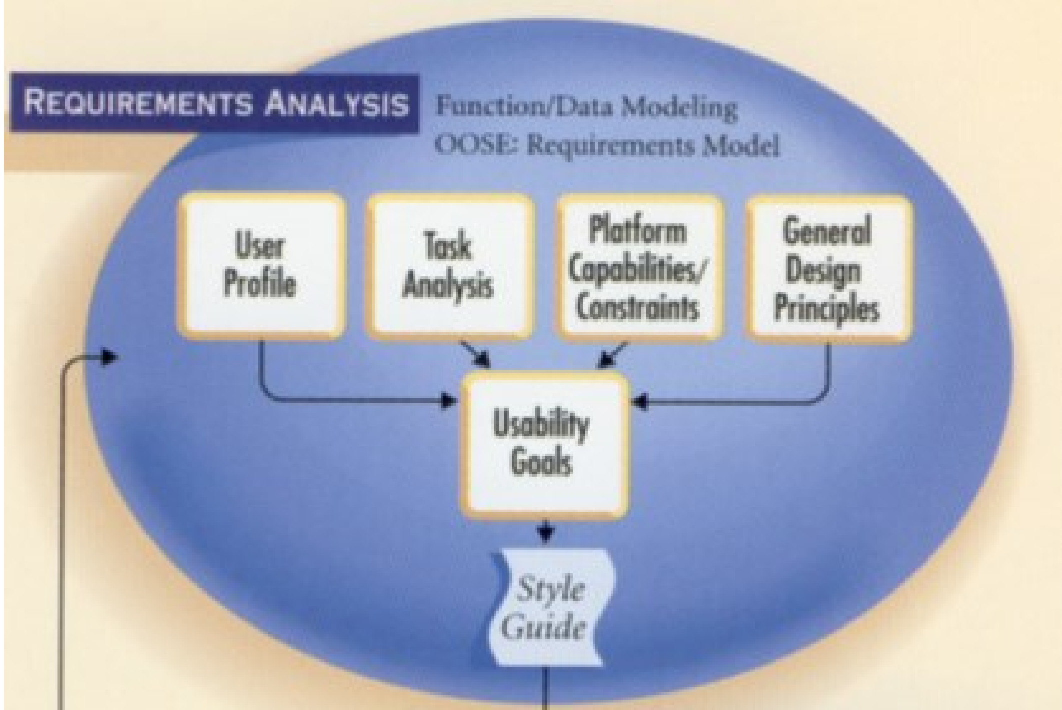
(Note that in Lab classes we started by doing it in an informal way…)

- Techniques

(Mayhew, 1999)

- Sources of information

- How to use it



Task analysis

- It is the analysis of how people perform their work
  - what they do
  - what they use
  - what they need to know

- Example: vacuum cleaning a house

  Get the vacuum cleaner

  Choose the adequate attachment

  Clean the rooms

  Empty the bag when it is full

  Put the vacuum cleaner and attachments away

- Users have to know about vacuum cleaners, rooms, …

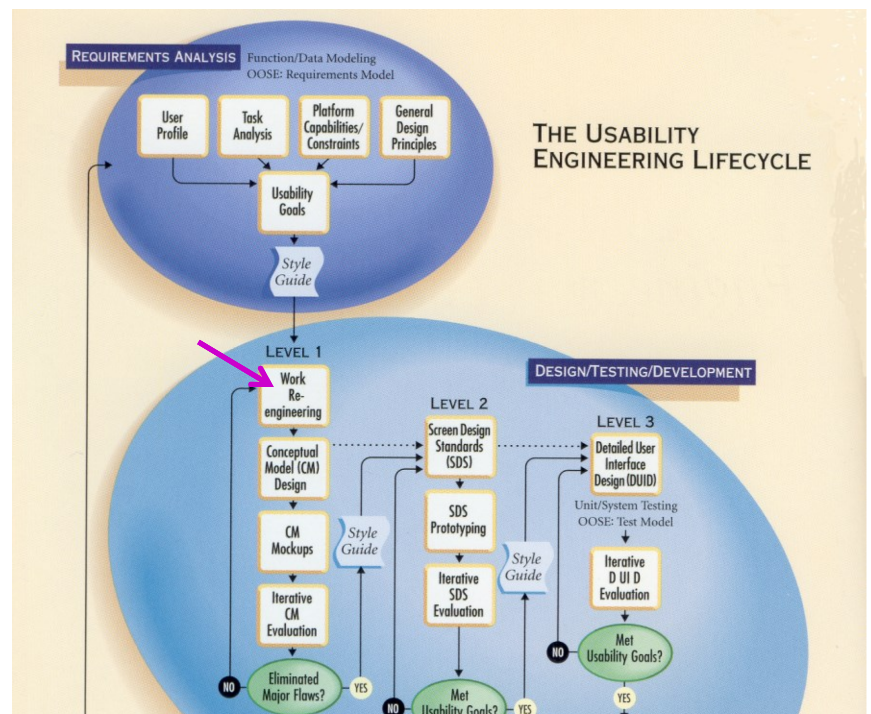
Task Analysis approaches

- There are a lot of approaches and methods; we have used an informal way in the practical classes, but there are more formal methods
  - Task decomposition ✓
  - Knowledge based
  - Relation and entities based

- Different approaches:
  - <u>Task decomposition</u> - divides tasks into subtasks that must be performed in a specific sequence
  - <u>Knowledge based</u> – considers what users need to know about the objects and actions involved in performing the task and how knowledge is organized
  - <u>Relation based</u> - is focused on actors and objects, relations among them and the actions they perform

Task Analysis

- Observation (of various types) is a fundamental tool
- It can be used to:
  - produce documentation and training materials
    (the observation of how existing
    systems are used is enough)
  - design new systems
    (work-re-engineering is
    usually necessary)



Task Decomposition

- Hierarchical Task Analysis (HTA) is one of the most used task analysis techniques and produces:

    - a task and sub-task hierarchy

    - plans with a sequence and execution conditions

Simple example: vacuum cleaning the house:

0. in order to clean the house
    1. get the vacuum cleaner
    2. fix the appropriate attachment
    3. clean the rooms
        3.1. clean the hall
        3.2. clean the bedrooms
    4. empty the dust bag
    5. put the vacuum cleaner and attachments away

Plan 0: do 1 – 2 – 3 – 5 in that order
when the dust bag gets full do 4

Plan 3: do any of 3.1, 3.2, or 3.3 in any order depending on which rooms need cleaning

Plan 3 could be more specific; what if it were varnishing the house?

- Where should the decomposition stop?
- The decomposition detail depends on the goals of each task analysis
- Example: in a factory what should be done in an emergency

  0. in a emergency

     1. Read the alarms
     2. Determine the corrective actions
     3. Execute the correction actions

- If the goal is
  - installing a monitoring system → expand 1 e 3
  - produce operation manuals → expand 2

- Where should the decomposition stop?

- A stop decomposition rule:

Stop if P x C < a specific value

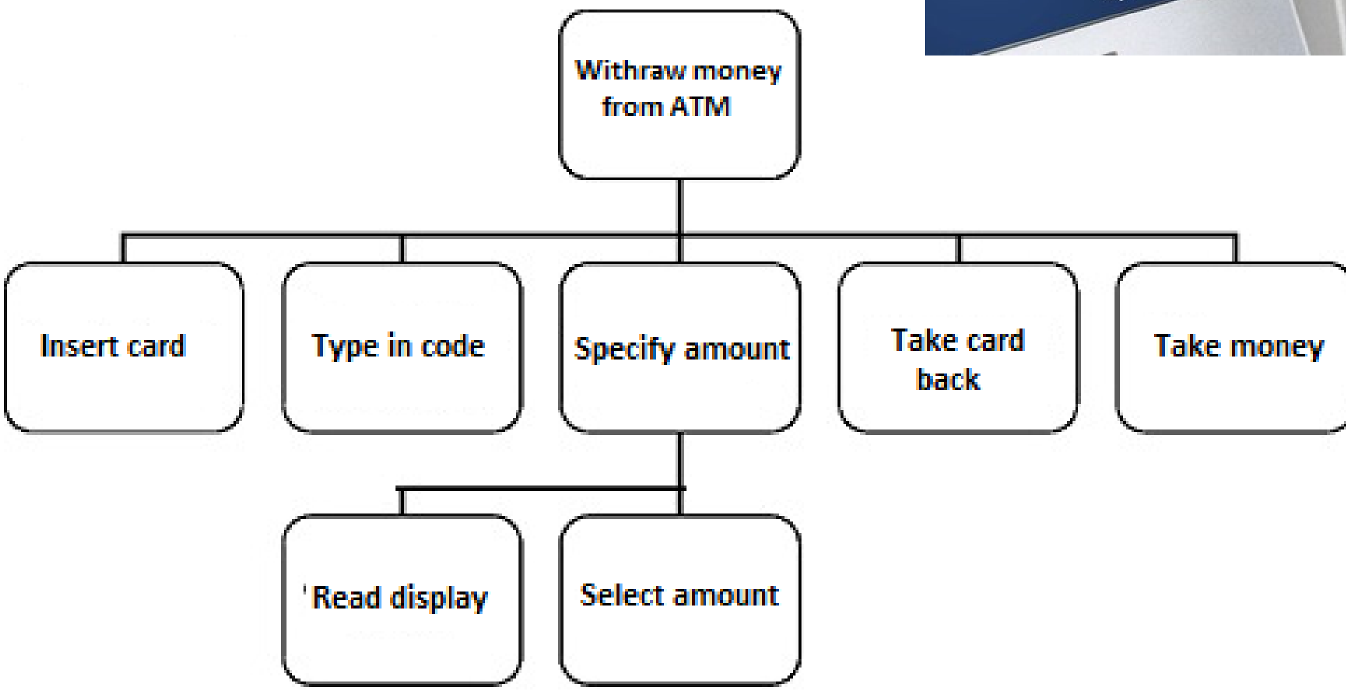
error cost

Probability of making an error

i.e.: simple tasks do not need decomposition unless they are critical!

Incomplete decomposition of the task:
Withdraw money from an ATM

Can you fix and complete it?

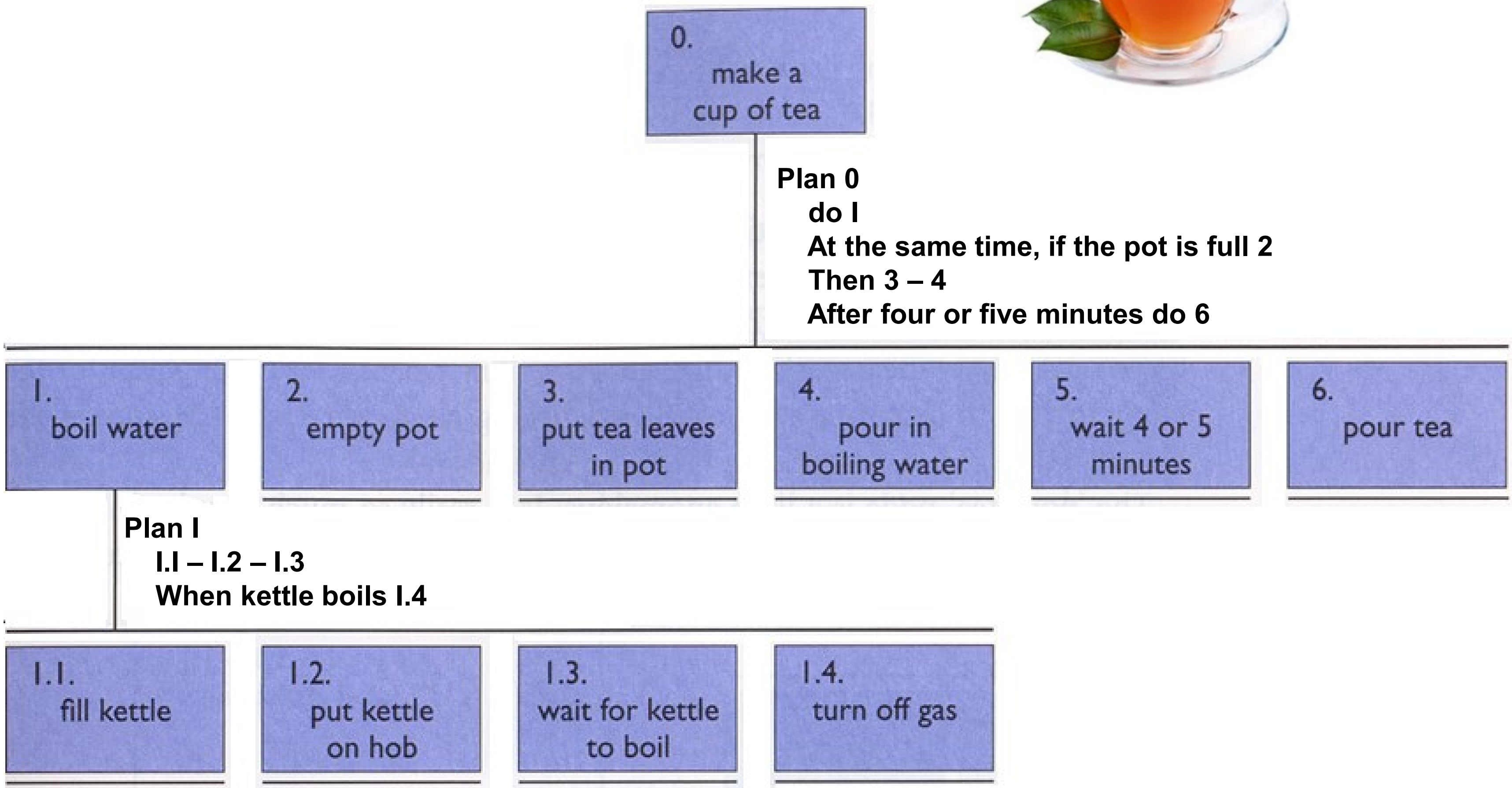


Another example:

Preparing a cup of tea

Can you do a HTA describing this task?

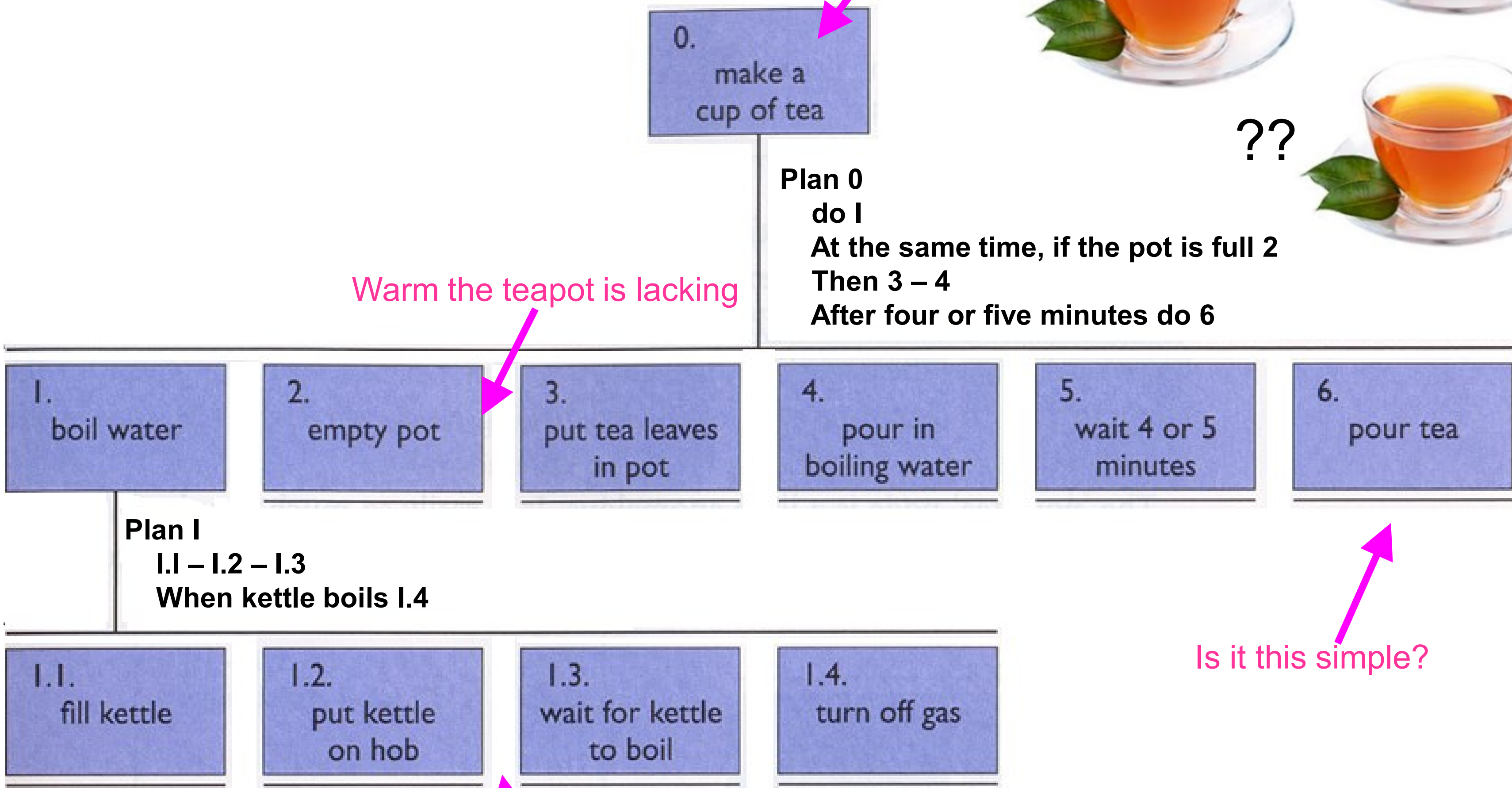
HTA- Preparing a cup of tea
(graphical representation
of first approach)



HTA- Preparing a cup of tea
(analysis of the of first approach)

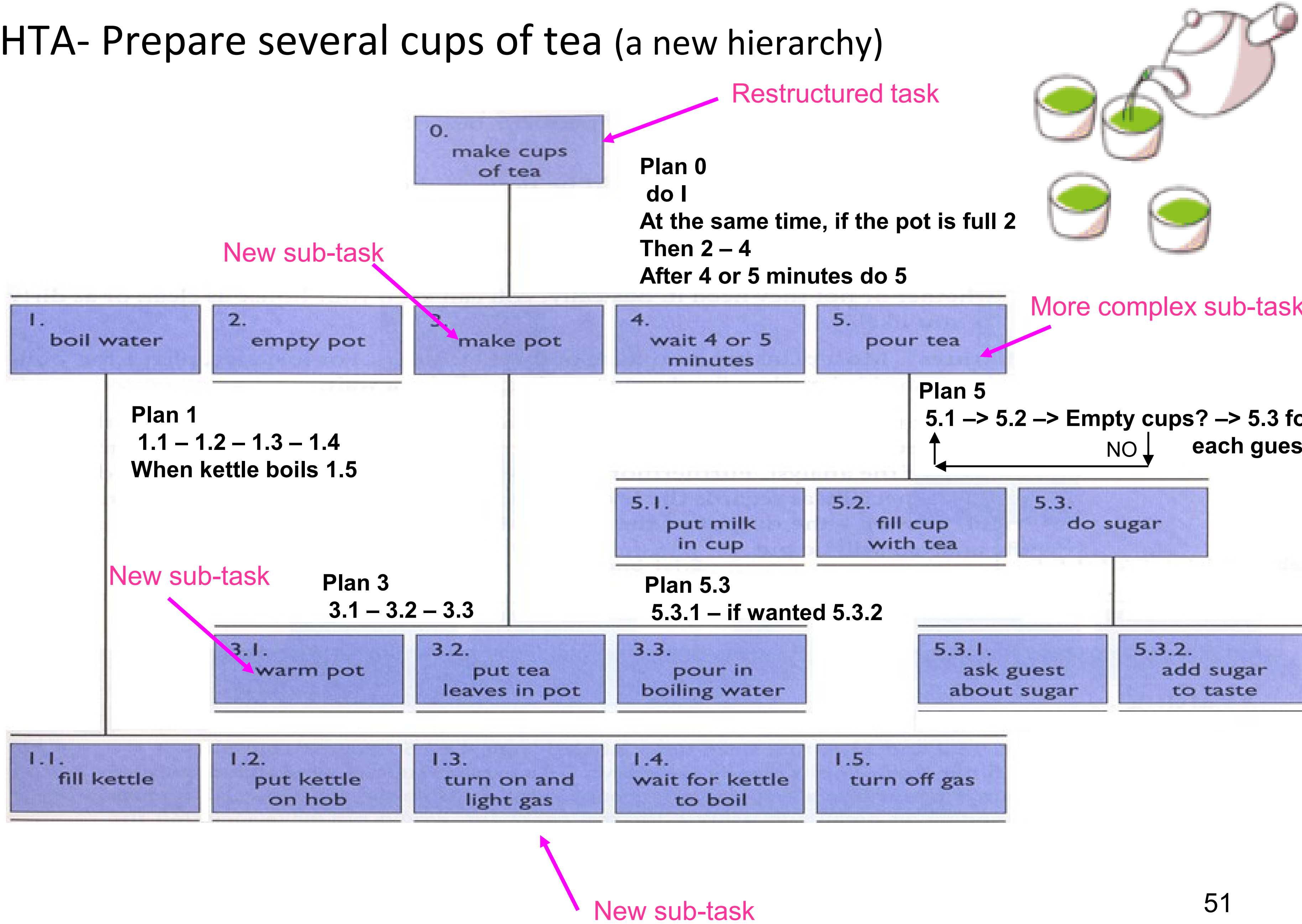
Do we want only one cup of tea?

??



Turn on the gas?

HTA- Prepare several cups of tea (a new hierarchy)



Plan types in HTA

- Fixed sequence (plan 3 – prepare the teapot)
- Optional tasks (5.3 sugar?)
- Waiting for events (4- wait 4 or 5 minutes)
- Cycles (plan 5 – serve tea)
- Time sharing (1 and 2 prepare teapot, boil water)
- Random (vacuum cleaning rooms)
- Mix of several types

- The result of the analysis depends a lot on the experience of the analyst
- **Different analysts usually produce different results** (mainly at the detail level) varying with the goal of the analyst

Task analysis information sources

- The quality of task analysis results cannot be better than the original data

  "garbage in garbage out"

- The process of analysis in general triggers new questions, thus <u>several phases</u> of data collection and analysis are needed

- There are several types of information sources:

  - Documentation
  - Observation (expensive)
  - Interviews

Documentation:

- Manuals, instruction books, training documentation ... are very good information sources

- But they describe what people are supposed to do, not what they actually do

- System's manuals usually describe functionality, not how they are used

- Observation and user interviews should be performed based on this information

- Be careful with user interviews!!

Interviews:

- Interviewing domain experts is a good way of getting information about the task; should include:
  - General questions (e.g. a typical day)
  - Specific questions (e.g. why did you do that?)
  - Task decomposition (~ HTA)

Observation:

- It is always necessary to perform (formal or informal) observation to understand the tasks

- Reading documentation and observing users is a good starting point

- More observation should follow:

  - In the lab
  - In the field

  - Passive (only observation)
  - Active (questions, post-task walkthrough)

Using Task Analysis:

- May be used in:

  - Manuals and teaching materials

  - High-level system design

  - Detailed design of the system user interface

- In the first case users are observed while performing tasks using the system

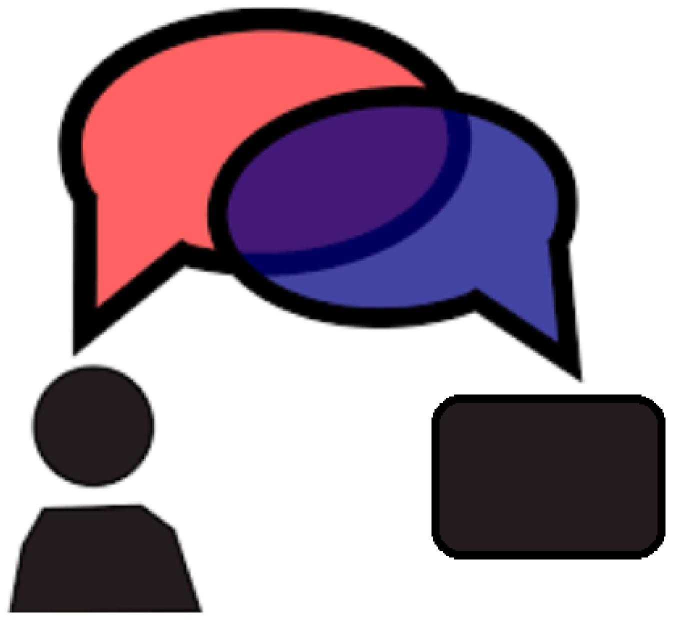
- In the other cases task analysis contributes to the design of the new system

Main bibliography

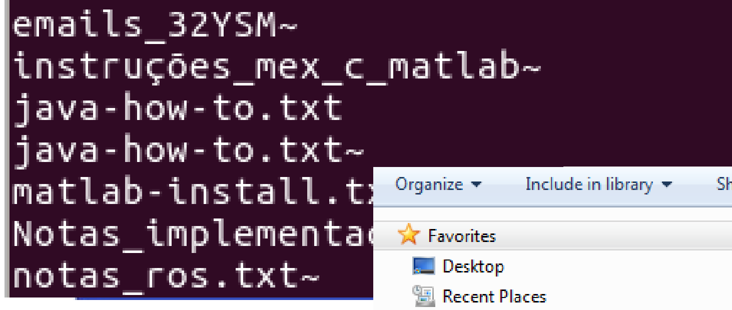
- Alan Dix, J. Finley, G. Abowd, R. Beale, *Human-Computer Interaction*, 3rd ed., Prentice Hall, 2004
- Dan Diaper, Neville Stanton, *The Handbook of Task Analysis for Human-Computer Interaction*, CRC Press, 2003
  http://www.amazon.com/Handbook-Task-Analysis-Human-Computer-Interaction/dp/0805844333/ref=sr_sp-atf_title_1_1?s=books&ie=UTF8&qid=1396779216&sr=1-1&keywords=Handbook+task+analysis#reader_0805844333
- Ian Sommerville, *Software Engineering*, 10th ed., Pearson, 2016
  https://www.amazon.com/Software-Engineering-10th-Ian-Sommerville/dp/0133943038
- Lene Nielsen, "Personas" In: Soegaard, Mads and Dam, Rikke Friis (eds.). *The Encyclopedia of Human-Computer Interaction*, 2nd Ed. Aarhus, Denmark: The Interaction Design Foundation, 2014
  https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/personas

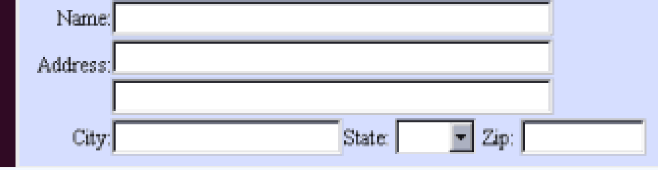
Universidade de Aveiro
Departamento de Electrónica,
Telecomunicações e Informática

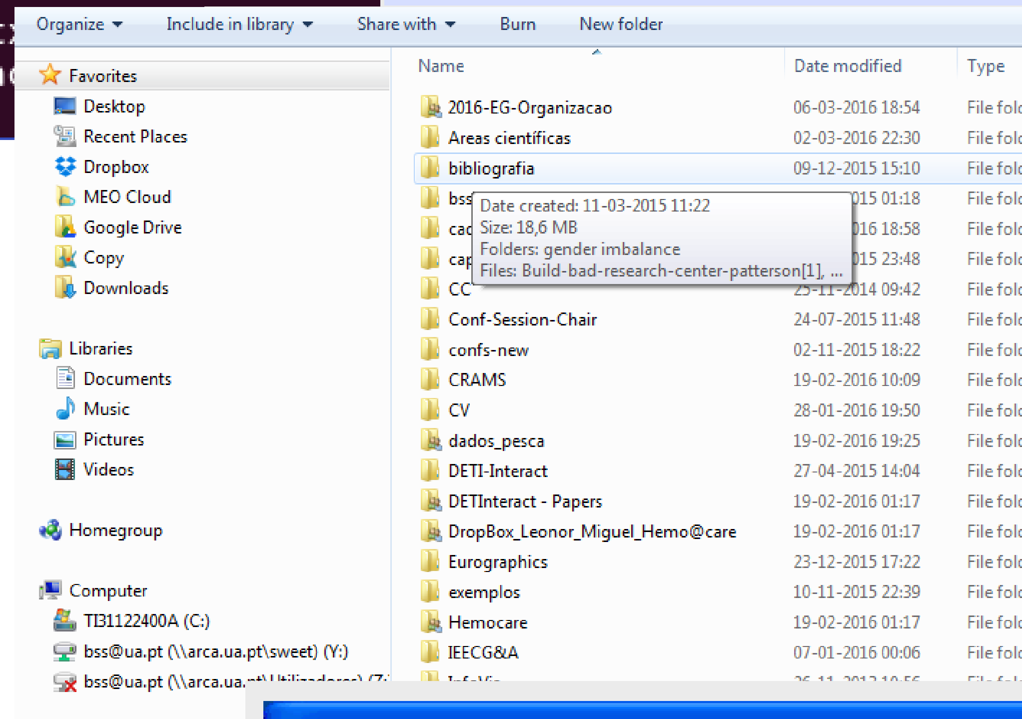
Other Interaction Styles



Interaction/ Dialog styles

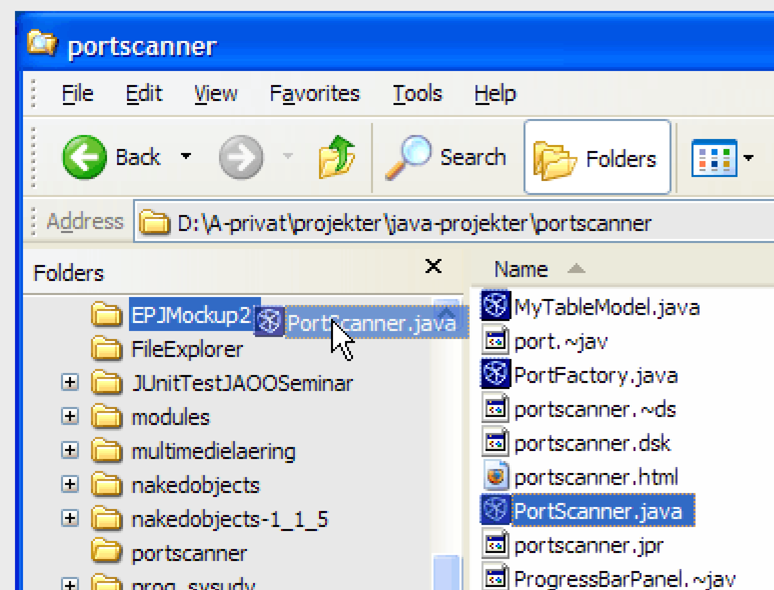






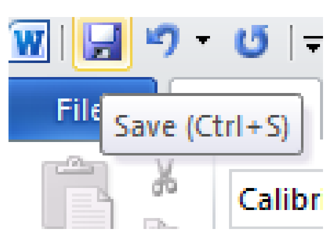
A possible classification:

- Menus
- Direct manipulation
- Fill-in-forms
- Function keys
- Question and answer
- Command languages
- Natural languages

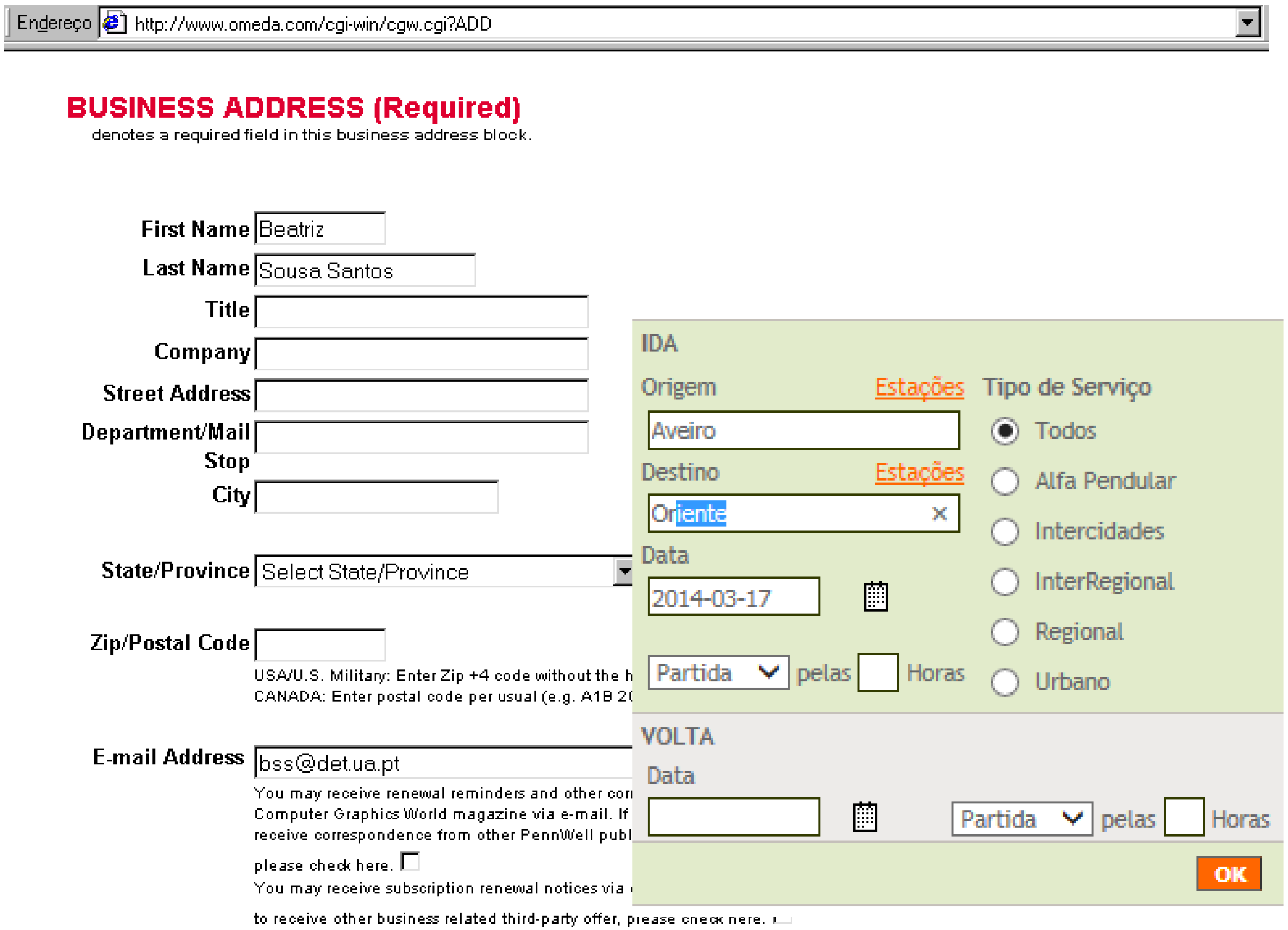


Often two or more styles are used simultaneously;

Why?



Fill in forms



- Fill in forms are particularly useful for routine, clerical work or for tasks that require much data entry

- The concept already existed long ago

- Currently they are often used with other styles

Main advantages and disadvantages

Advantages (potential)

- Self-explanatory
- Recognition instead of recall
- Allow many different inputs (unlike menus)
- Give context and guide the user
- New functionality is visible (unlike command languages)

Disadvantages

- Imply knowledge of valid inputs
- Error prone
- Not very flexible

Fill in form design: relevant aspects in design

- Organization and layout
- Titles and fields
- Input formats
- Instructions and help
- Navigation
- Error handling

Fill in form design: guidelines

Avoid unfamiliar layouts

Example:

Zip code:
Name:
Country:
Address:
City:

Better:

Name:
Address:
Zip code:
City:
Country:

Alignment of field titles

Not a good solution

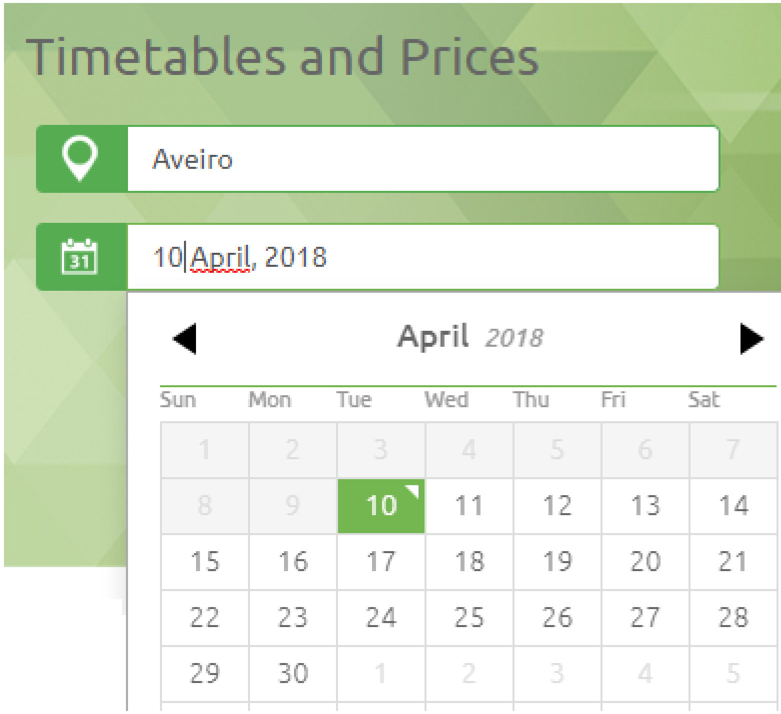
Name: --
Title: --
Rank: --
Telephone number: --

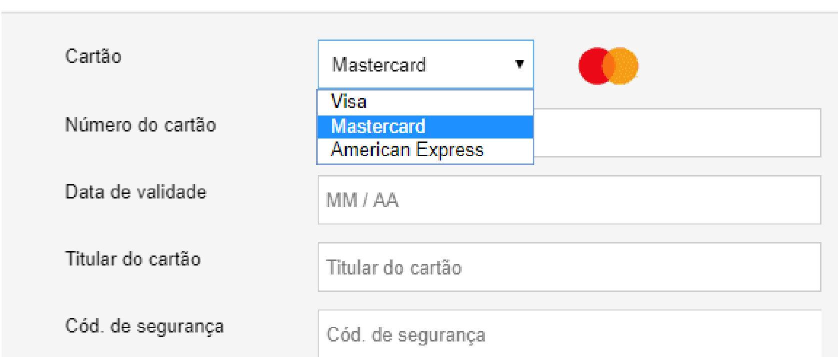
Name: .... --
Title: .... --
Rank: .... --
Telephone number: --

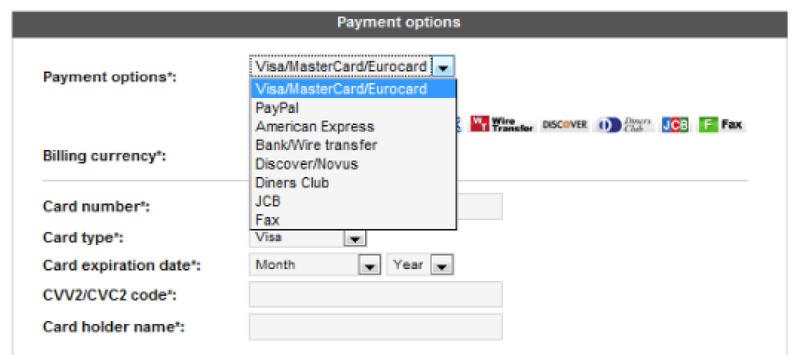
Better solutions

Name: --
Title: --
Rank: --
Telephone number: --

Provide a menu when possible inputs are known (combining two interaction styles…)

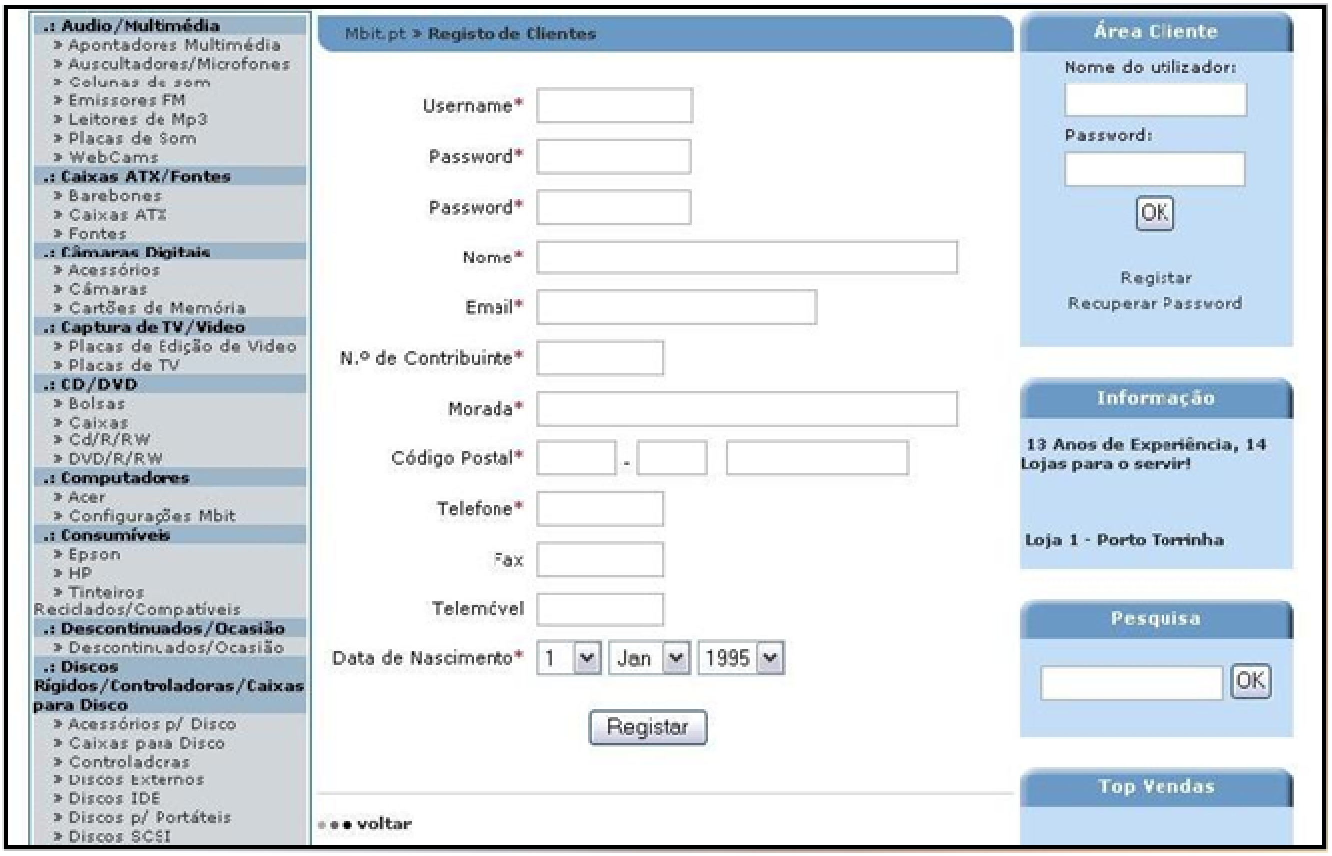






Provide a format for fields that may be ambiguous

Show which fields are mandatoty

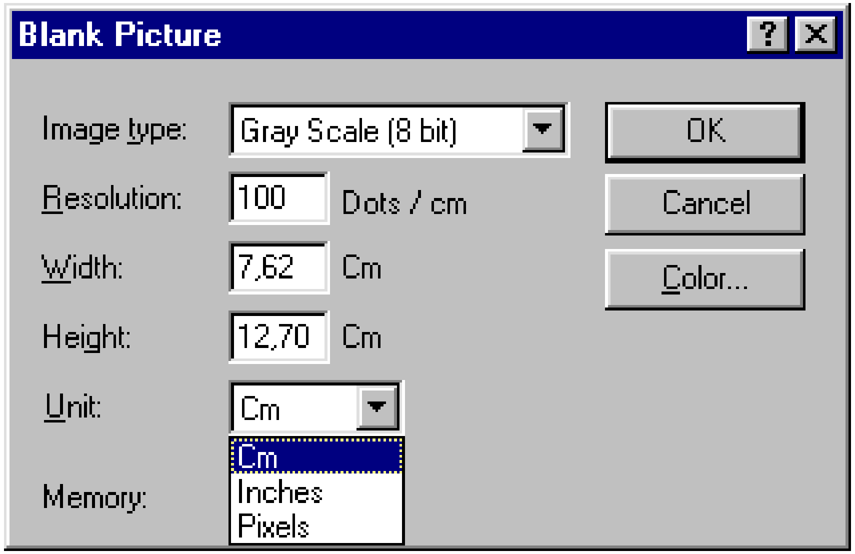


Usually indicated by *

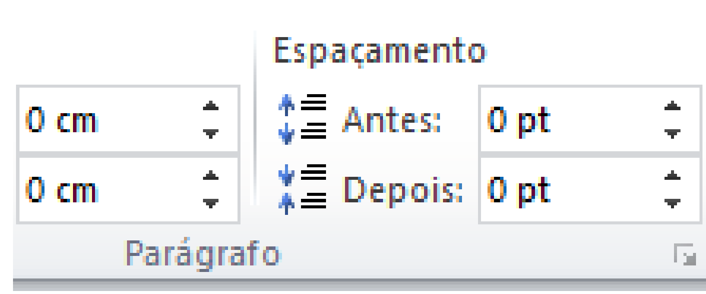
Input format must be familiar and clear

	Better:
Date:______________________ (eg. 1/12/2000)	Date:___/__/__________ (e.g. 1/ 12 /2000)
Date:______________________ (e.g. 01122000)	
Time:_____________________ (eg. 8-15)	Time:___-________________ (e.g. 08-15)
Time:_____________________ (e.g. 0815)	
Card#:____________________ (e.g. 123456789012)	Card#:___-_____-________ (1234-5678-9012)

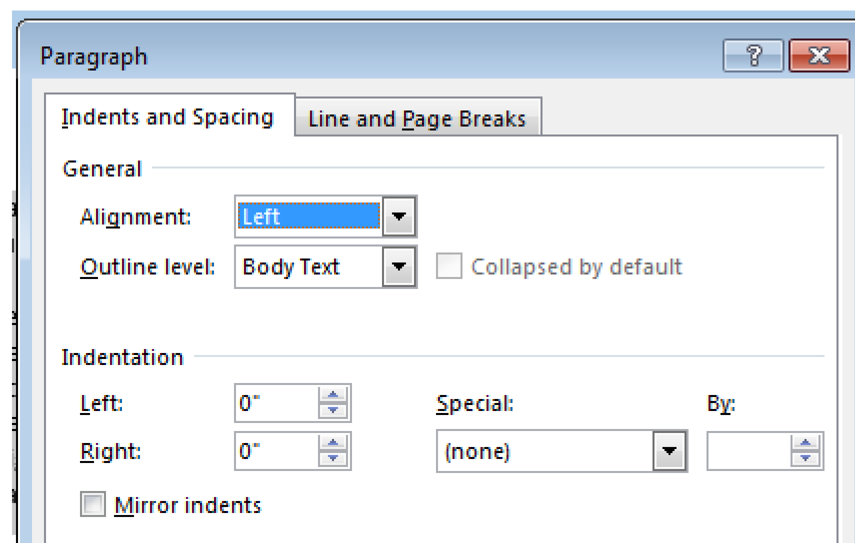
It should be possible for the user to choose the type of input (it prevents errors) or adapt to the context



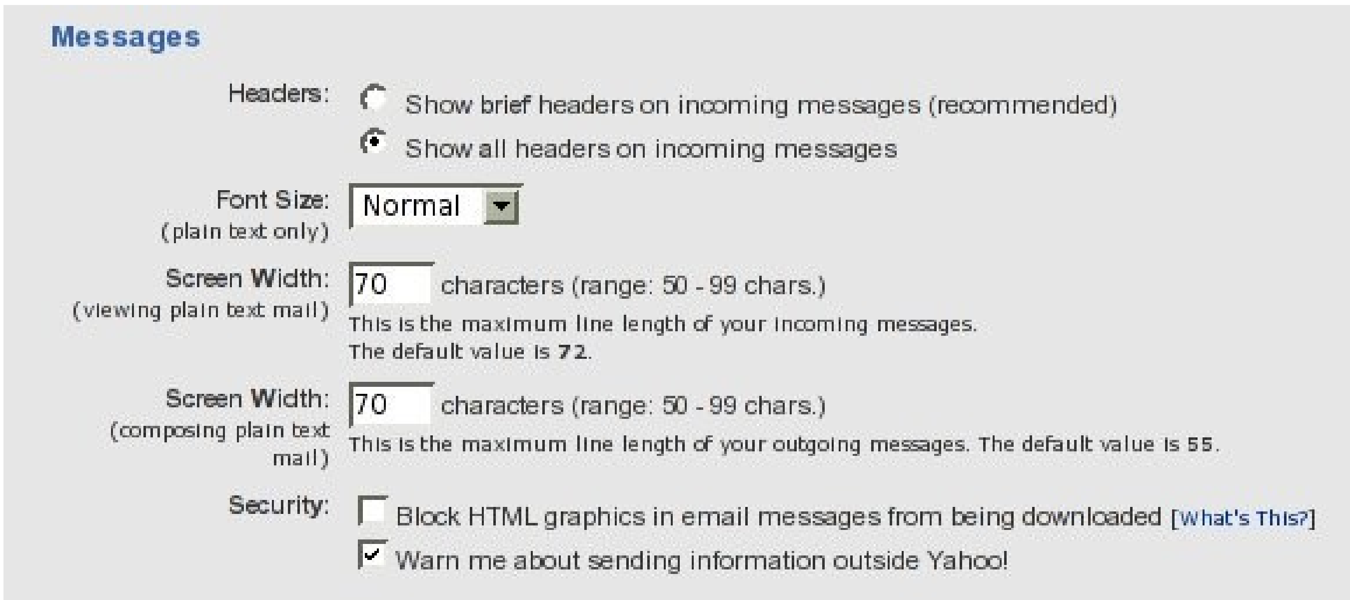
Portuguese version (cm):



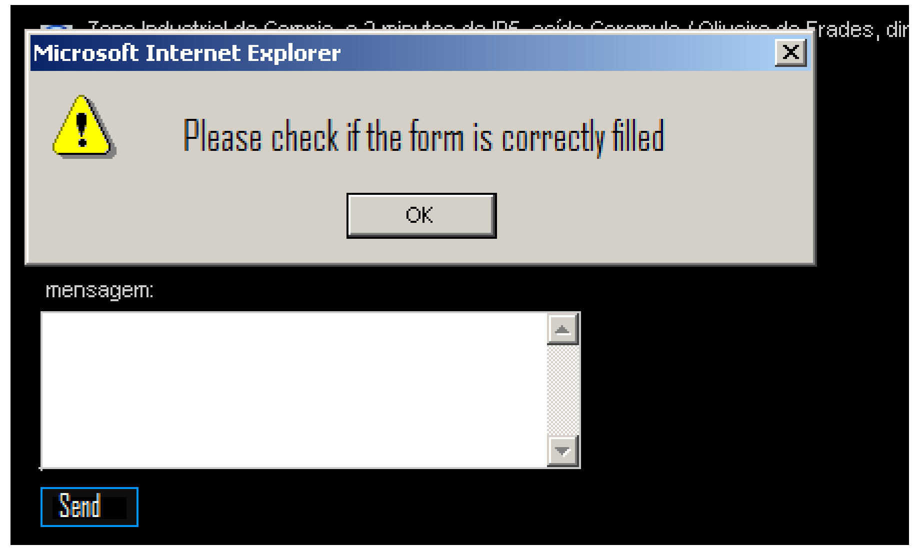
English version (inches):

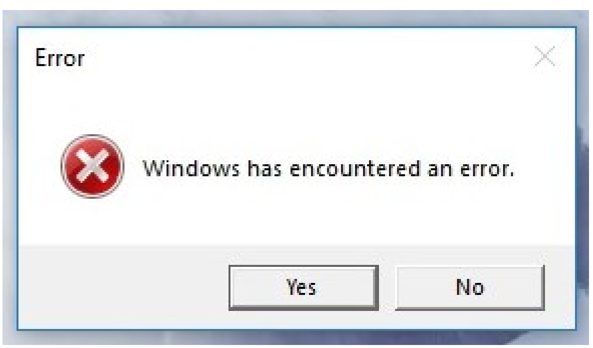


Instructions to fill the fields should be clear

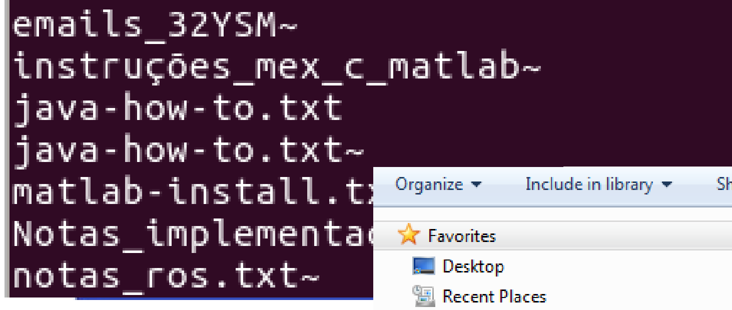


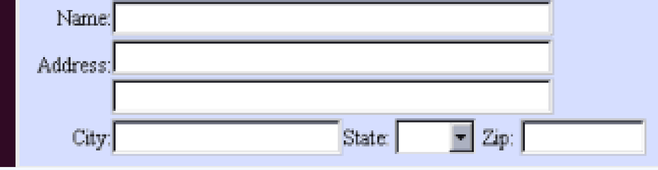
Messages not clear, nor helpful





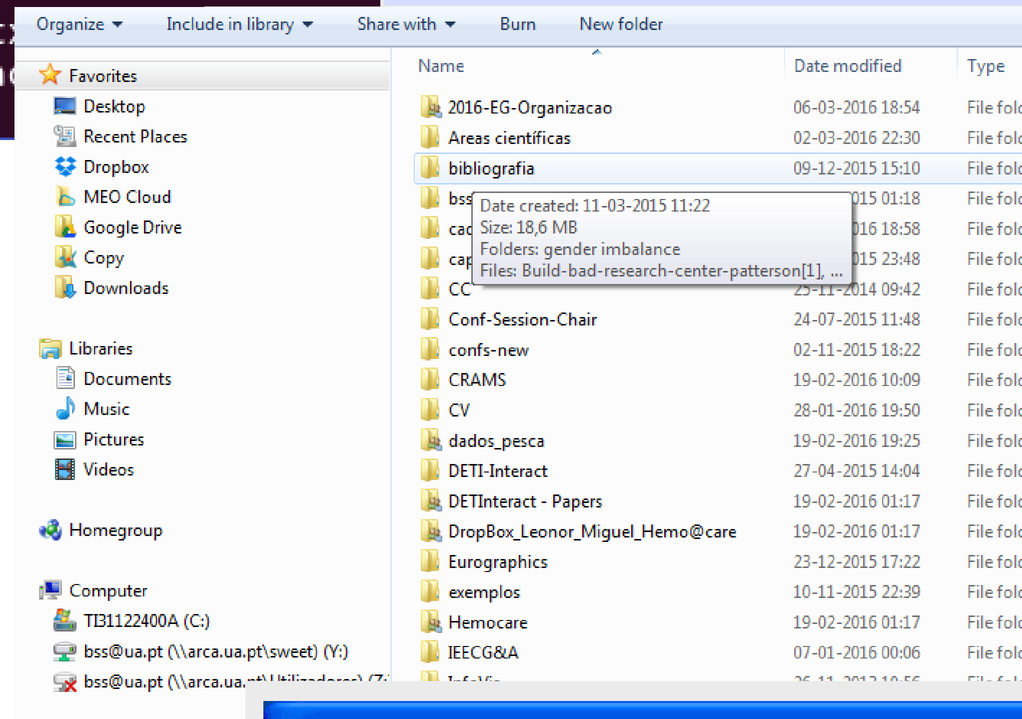
Interaction/ Dialog styles

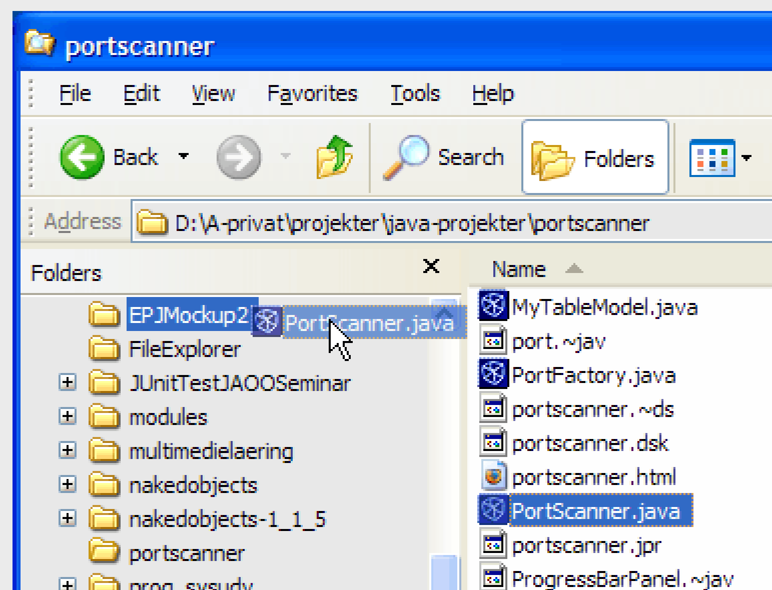




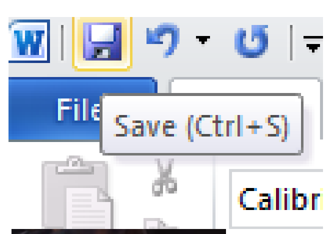
A possible classification:

- Menus
- Direct manipulation
- Fill-in-forms
- Function keys
- Question and answer
- Command languages
- Natural languages





Often two or more styles are used simultaneously



Function keys

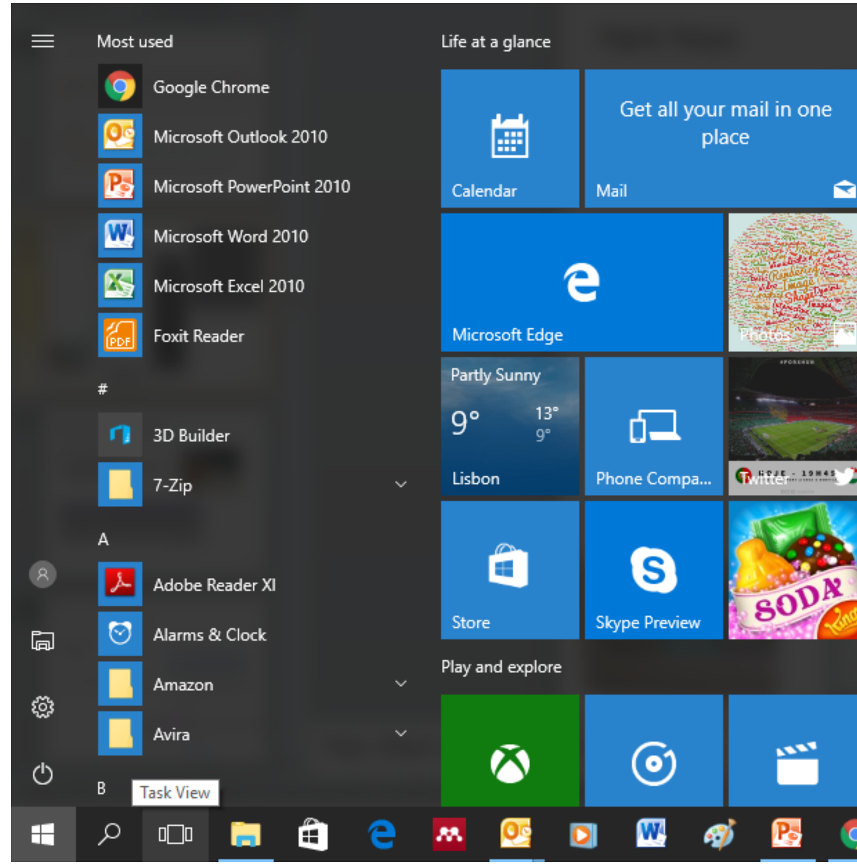
- Two types:
  - *Hard Keys* – Always invoke the same functionality (as the keys of a calculator and some specific keys of PCs)
  - *Soft Keys* – invoke different functionality according the context of use (as the keys (F1...Fn) and the generic keys of an Automated Telling Machine, e.g. Multibanco)

- PCs have 12 generic Keys (F1 a F12) and a few other specific keys

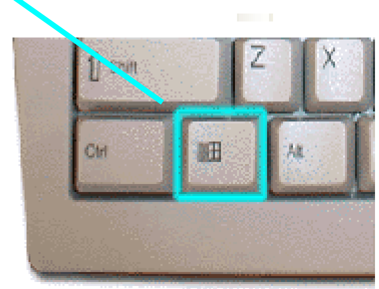
Keys that invoke specific functionality in PCs and MACs

Hard Keys

Hard function keys have abbreviations of default actions printed on/besides them



Specific keyboard



Start menu key

Soft Keys

Soft function keys don't have abbreviations of default actions printed on/besides them, they may have "F-number" designations.

Function keys (generic)

https://en.wikipedia.org/wiki/Function_key

Touch bar: is it a new type of function keys?

Discuss the advantages and disadvantages for several types of users and contexts of use

Main advantages and disadvantages

Advantages (potential)

- Self-explanatory
- Recognition instead of recall
- Easy to use
- Flexible
- Require little or no screen real estate

Disadvantages

- Limited number of keys
- Hardware expansions are expensive

Function keys design: guidelines

Provide enough keys to call the functionality

But no too many as not to make it difficult to learn

Use:

- free space
- different size, color and shape to different groups
- category groups
- clear and distinctive names

Multi-media remote control keyboard

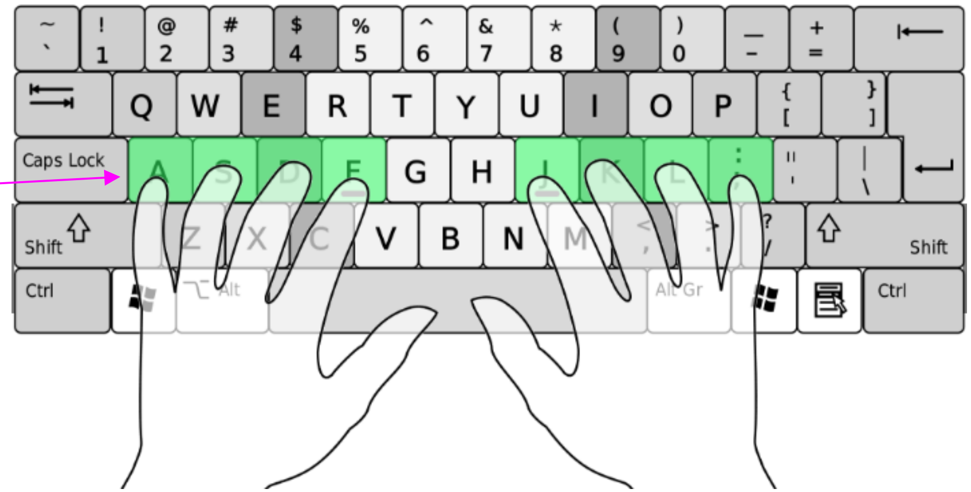
Industrial keyboard

ATM keyboard

Shop system keyboard

TV remote control

Often used keys should be near the “home row”



Keys with serious consequences should not be easy to activate

(e.g. ctrl Alt Del)

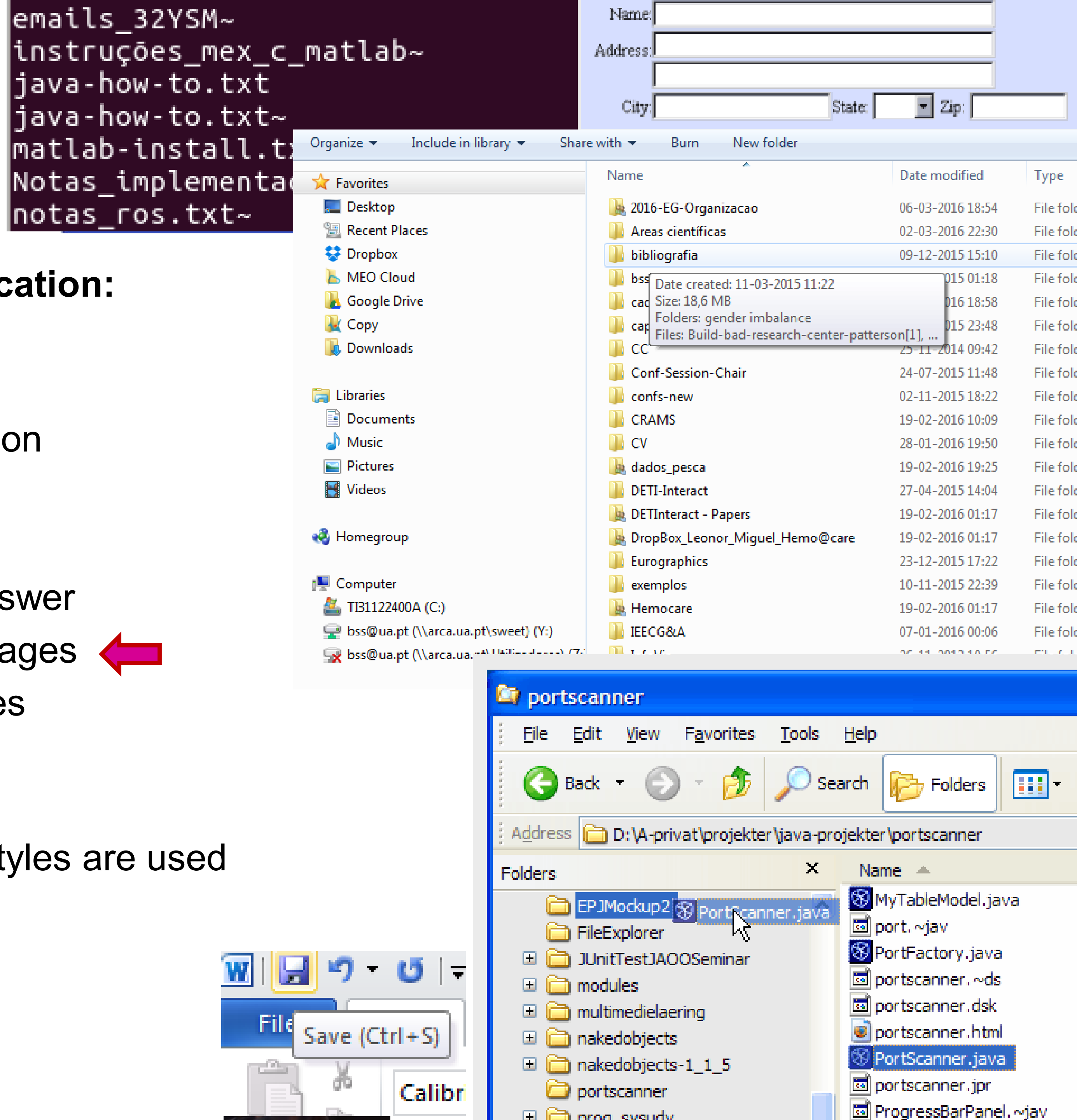
Interaction/ Dialog styles

A possible classification:

- Menus
- Direct manipulation
- Fill-in-forms
- Function keys
- Question and answer
- Command languages
- Natural languages

…

Often two or more styles are used simultaneously



Command languages

```
cd /tmp
echo "line 1
line 2
line 4" > tmp1$$
echo "line 2
line 3" > tmp2$$
diff tmp1$$ tmp2$$
rm tmp1$$ tmp2$$
```

```
guru99@VirtualBox:~$ history
    1  cat > sample
    2  cat sample
    3  cat sample ^a
    4  cat sample a
    5  cat sample | grep a
    6  cat sample | grep ^a
    7  useradd home
    8  useradd mycomputer
    9  sudo useradd mycomputer
   10  sudo adduser MyLinux
   11  sudo adduser mylinux
   12  vi scriptsample.sh
```

Command languages shall also be designed as to be as usable as possible

Basic Goals of Language Design

- Precision
- Compactness
- Ease in writing and reading
- Speed in learning
- Simplicity to reduce errors
- Ease of retention over time

Usability Questions concerning a command language

- Does the language support necessary functions?
- Is it fast to enter a command?
- Is it easy to recognize what the command might do?
- Is it easy to recall a command?
- Are there few errors when using the language?

Main advantages and disadvantages

Advantages (potential)

- Powerful
- Flexible
- Efficient
- Do not take much screen real estate

Disadvantages

- Difficult to learn
- Not self-explainable
- Error prone
- Improvements are not visible

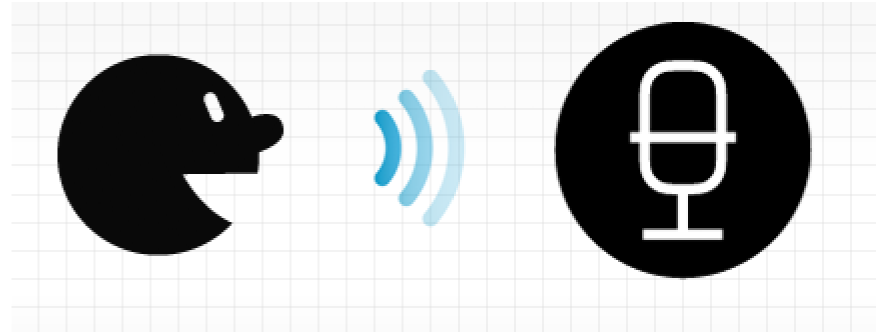
User profile to whom Command languages are adequate

Knowledge and experience:
- High task experience
- High application experience
- High computational literacy
- High typing skill

Task characteristics:
- High usage frequency
- Formal training

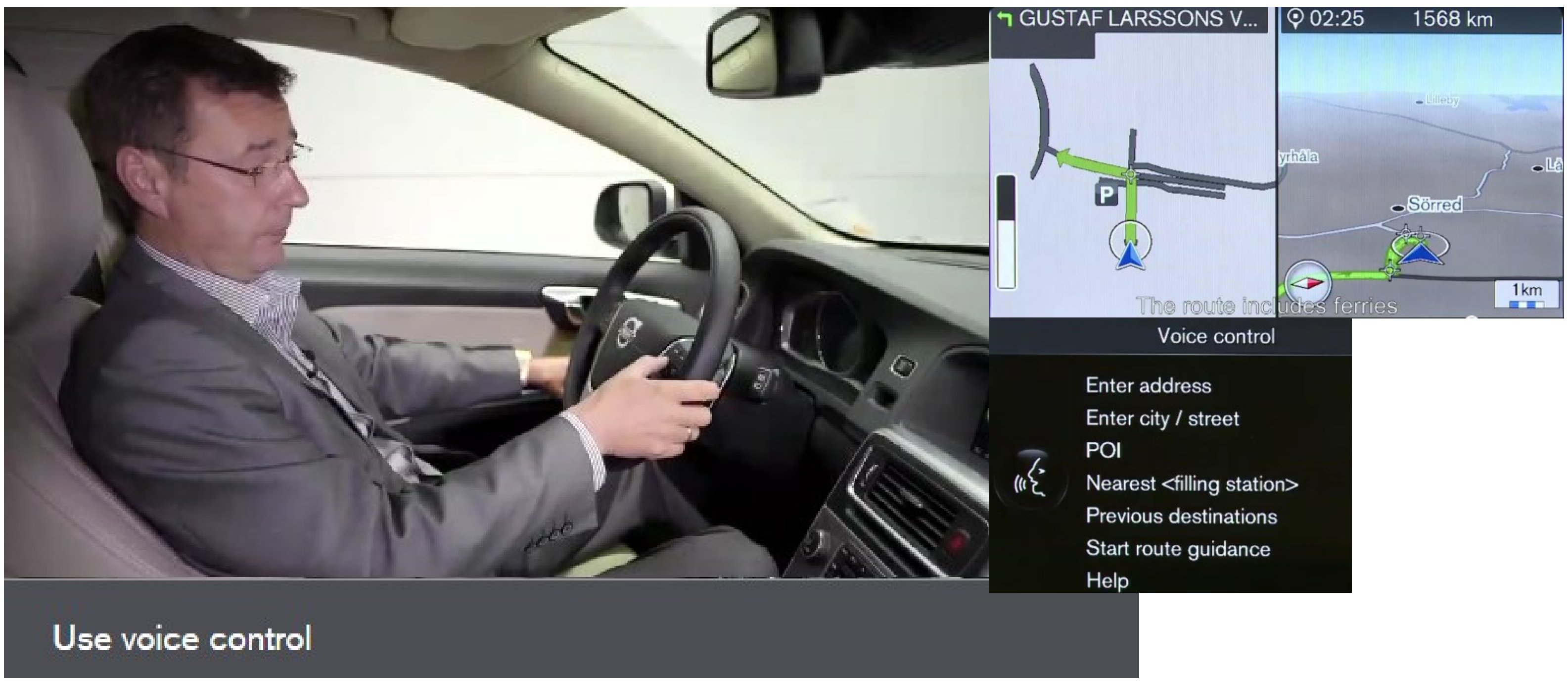
Note that:

Command languages may be used not only through text but also via voice



e.g.

While driving a car to control the media, the phone or navigate



http://support.volvocars.com/uk/Pages/article.aspx?article=a8275b1eb0ed6a0fc0a8015159f7fdd6

Relevant issues in Command Language design

- Semantics
- Syntax
- Lexicon
- Interaction

Command Languages Design guidelines

Balance richness and minimalism
(similar to semantic distance in direct manipulation)

Examples :

Rich	**Minimal**
Delete	Delete
Insert	Insert
Replace	

Copy	Copy
Move	Delete
Rename	
Delete	

(the functionality is the same)

Use a coherent syntaxe

Use a natural and easy to remember action-object grammar

```
VolB!FileA!D$$
FileA!VolB!ER$L!:KO:!*$$
```

Uncoherent syntax and unfamiliar commands

```
search filea volb.
open filea volb.
list all lines with "KO".
```

or

```
s filea volb.
o filea volb.
lal "KO".
```

Command abbreviations should be simple and coherent

Easy to remember (not easy to recognize as for function keys)

Name	Abbreviations Poor:	Abbreviations Improved:
Move forward	MovF	MovF
Move backward	Mvb	MovB
Insert	I	Ins
Delete	Dl	Del
Replace	Repl	Rep
Search	Srch	Sea
Delete	X	Del
Send	Sn	Sen
Print	Prt	Pri
Search	Srch	Sea
Send	Sn	Sen
Find	Fi	Fin
Choose	Ch	Cho

Allow the following interaction features:

- Defaults
- Command edition
- Intelligent interpretation
- Type-ahead
- Feedback
- Help and documentation
- Make the language “user tailorable”

Example of intelligent interpretation:
“delate”: did you mean “delete”? Y or N

Example of a (complex) command with defaults

ls - Linux man page

You don't need to use all arguments; there are default values

Name

ls - list directory contents

Synopsis

ls [*OPTION*]... [*FILE*]...

Description

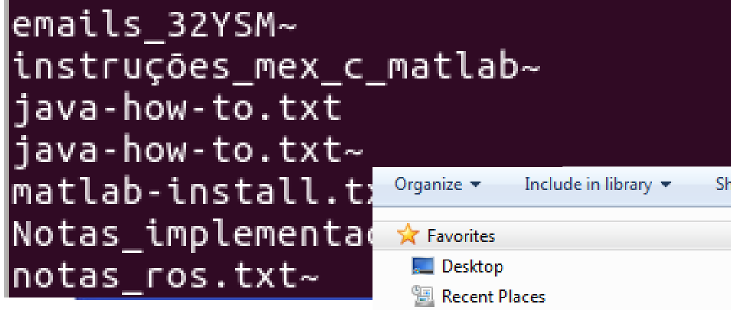
List information about the FILEs (the current directory by default). Sort entries alphabetically if none of **-cftuvSUX** nor **--sort**.

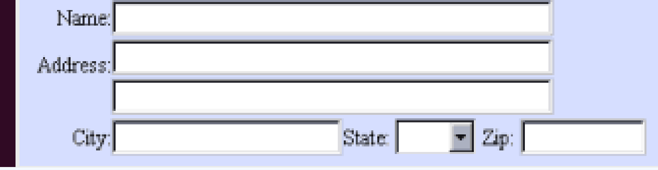
Mandatory arguments to long options are mandatory for short options too.

-a, **--all**
  do not ignore entries starting with .
-A, **--almost-all**
  do not list implied . and ..
--author
  with **-l**, print the author of each file
-b, **--escape**
  print octal escapes for nongraphic characters
-d, **--directory**
  list directory entries instead of contents, and do not dereference symbo
-D, **--dired**
  generate output designed for Emacs' dired mode
-f
  do not sort, enable **-aU**, disable **-ls --color**
-F, **--classify**
  append indicator (one of */=>@|) to entries
--file-type
  likewise, except do not append '*'
--format=*WORD*
  across **-x**, commas **-m**, horizontal **-x**, long **-l**, single-column **-1**, verbose
--full-time
  like **-l --time-style**=*full-iso*
-g
  like **-l**, but do not list owner
--group-directories-first
  group directories before files.
augment with a **--sort** option, but any
  use of **--sort**=*none* (**-U**) disables grouping
-G, **--no-group**
  in a long listing, don't print group names
-h, **--human-readable**
  with **-l**, print sizes in human readable format (e.g., 1K 234M 2G)
--si
  likewise, but use powers of 1000 not 1024
-H, **--dereference-command-line**
  follow symbolic links listed on the command line

Etc.., etc., etc.

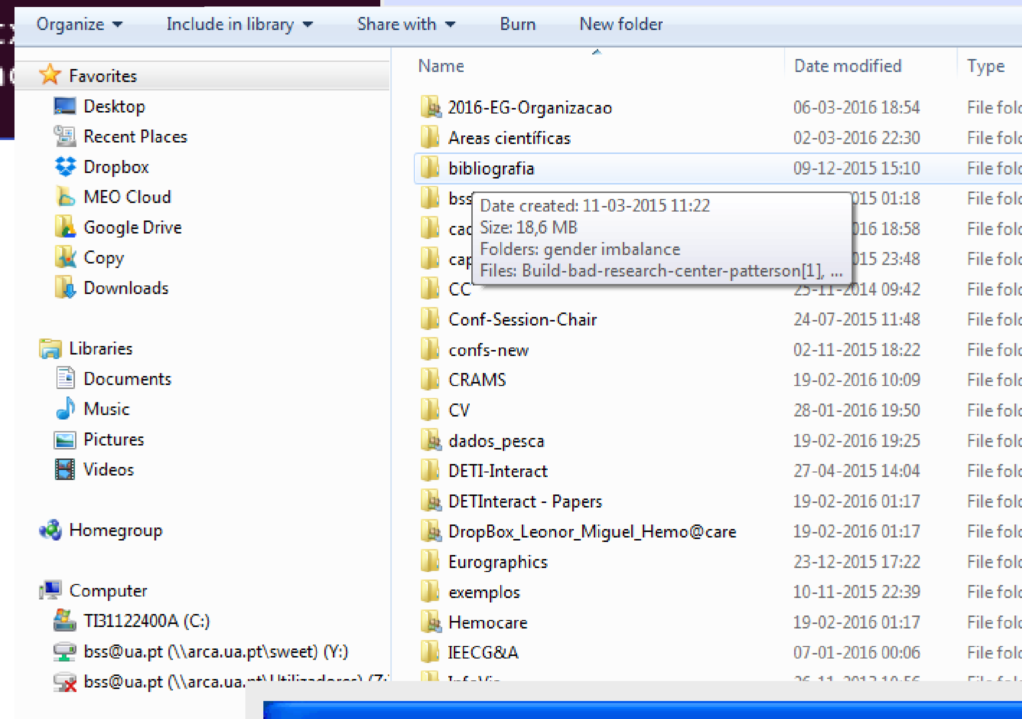
Interaction/ Dialog styles

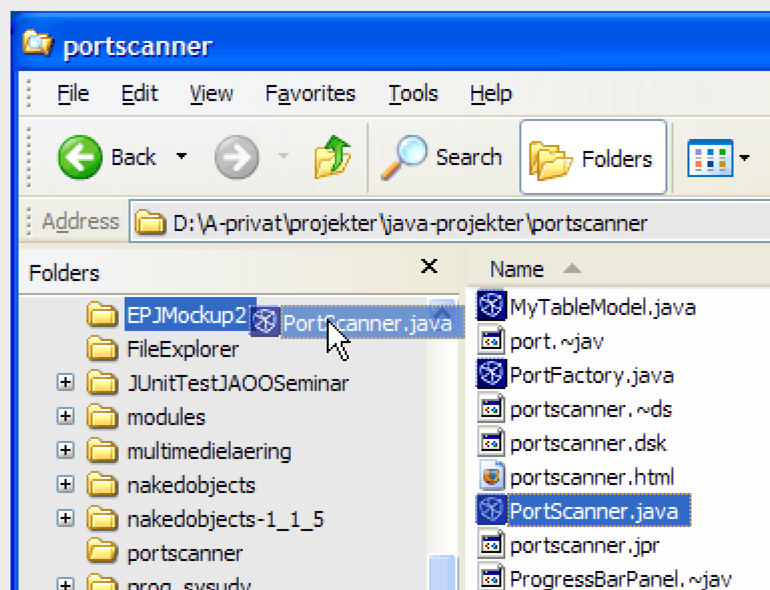


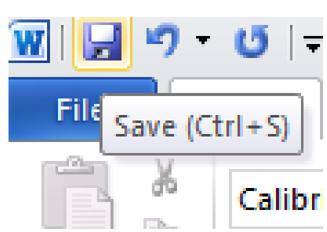


A possible classification:

- Menus
- Direct manipulation
- Fill-in-forms
- Function keys
- Question and answer
- Command languages
- Natural languages



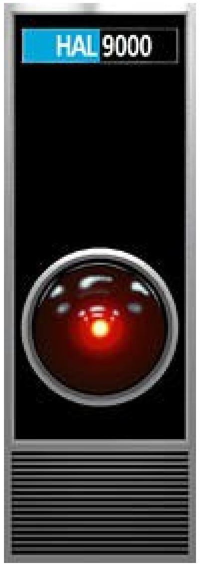




Often two or more styles are used simultaneously

Natural language

Note: NLP has evolved a lot, yet It still is not possible to maintain a conversation with a computer as in 2001 A Space Odyssey …



- Communication between humans and computers through natural language involves:
  - recognition
  - generation
- Natural languages as interaction style are not full blown natural languages, they are restricted natural languages
- Natural languages (as interaction style) differ in “habitability” (how easy and natural is it for users)

Note:
natural language as a dialog style and voice interaction are different things! Why?

- Habitability (mismatch between the users' expectations and the capabilities of a natural language) is related to the language domains:
  - Conceptual - the set of objects and actions provided by the language
  - Functional – what may be directly expressed by the language
  - Syntactic – syntactic forms that may be understood
  - Lexical - the variety of words that may be understood
- Conceptual model limitations are not very disturbing; however, limitations in any other domain make the language less habitable

Example:

- Imagine an information system of a University including a data base with information about employees that may be accessed using a natural language:

  - Conceptual domain: information about employees

  - The question “What is the salary of the University Restaurant manager?” may be out of the functional domain and imply two questions due to functional domain limitations:
    - “Who is the University Restaurant manager?” (answer: Mr. XXX)
    - “What is the salary of Mr. XXX?”

  - “What is the salary of Mr. XXX?” may not be recognized (due to syntactic domain limitations) even if the information is stored in the DB

  - “What are the wages of Mr. XXX?” may not be recognized due to lexical domain limitations if wages does not belong to the lenguage

User profile to whom Natural languages are adequate

Knowledge and experience
High task experience
Low application experience
Low computer literacy
High typing skill (if written)

Task characteristics
Low frequency of use
No or little training
Optional use

Main advantages and disadvantages of Natural Language dialog style

Advantages (potential)

- Powerful
- Flexible
- Efficient

(second to command languages)

Disadvantages

- Assume problem domain knowledge
- Imply clarification dialogs
- Imply typing skills (if written)
- Improvements are not visible
- May create unrealistic expectations, and generate negative reactions
- Difficult and expensive to implement

A few Natural Language design guidelines

- Provide a (restrict) natural language habitable in all domains

- Define a subset of a (real) natural language using the Wizard of Oz method

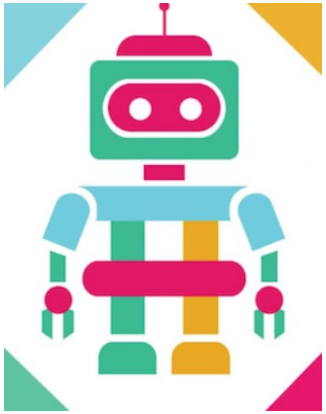
- Generate valid outputs concerning the four domains (e.g. always use words that the system recognizes)

Conversational User interfaces (CUIs)

Think of the potential advantages and disadvantages of CUIs:

- Chatbots

- Voice assistants



"Just like the touch interface, not everything will become conversational"

What doesn't fit the principles of Conversational UI well?
Products where the use case involves a technical user who wants fine grain control over the interface, …. e.g. CAD software, or a programming IDE…."

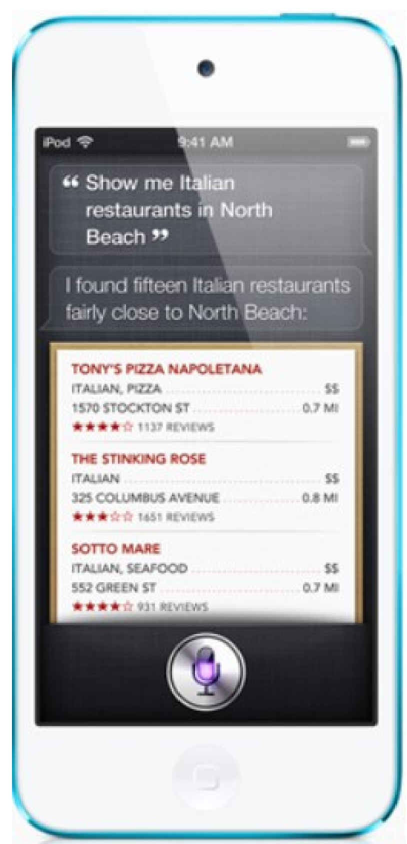
https://uxdesign.cc/conversational-ui-its-not-just-chat-bots-and-voice-assistants-case-study-cb1865da306a

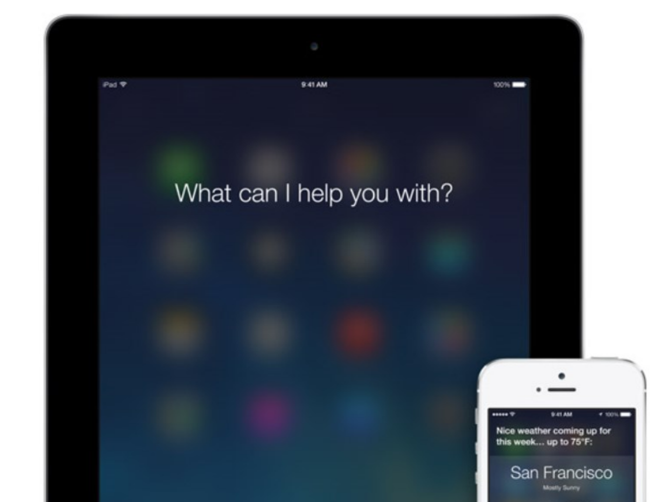
Current examples of Natural language interaction (mostly via voice)

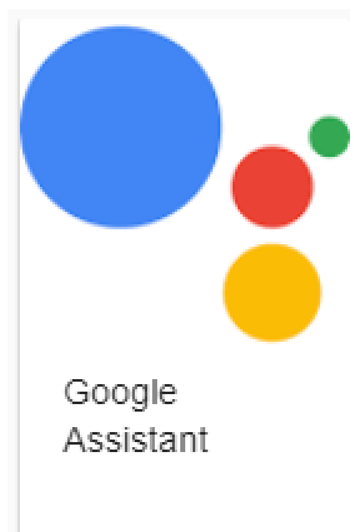
Mobile phone personal assistants:

- Siri for Apple’s iOS

- Google assistant

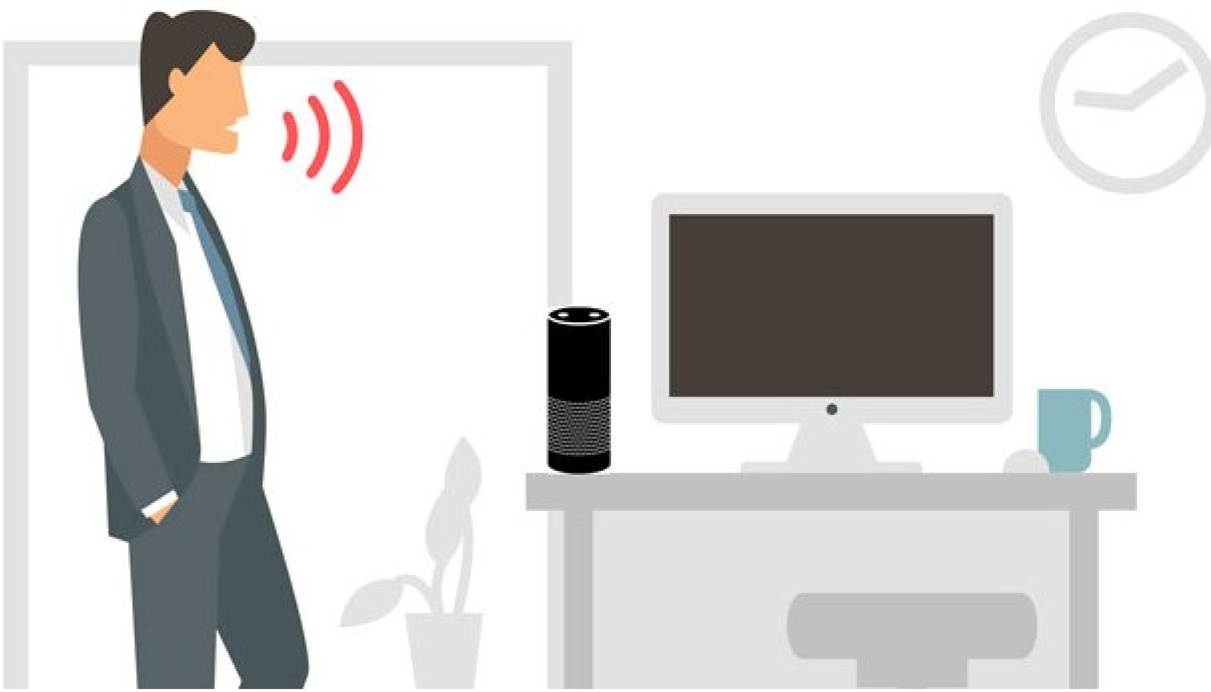






Another example
(natural language via voice)

- Amazon Alexa



https://www.nngroup.com/articles/voice-interaction-ux/

Wizard of Oz prototyping

- A prototype that only works by having someone behind-the-scenes "pulling the levers and flipping the switches" (named after the classical film)

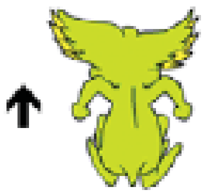
- A user interacts with an interface without knowing that the responses are given by someone

The "wizard" was a "man behind-the-scene"

https://en.wikipedia.org/wiki/Wizard_of_Oz_experiment

Example of using the Wizard of Oz method in other situations

- Definition of a set of gestures to use in a game



Höysniemi, J., Hämäläinen, P., Turkki, L., and Rouvi, T. 2005. “Children's intuitive gestures in vision-based action games”. *Commun. ACM* 48, 1, Jan. 2005, 44-50

Example of using the Wizard of Oz method in other situations



- Haptic Wizard of Oz Prototyping aids designers in rapidly designing and testing interactive hardware like this above car cockpit

D. Leithinger, C. Zheng, and E. Y. Do, “Haptic Wizard of Oz Prototyping in VR,” in *VR/MR Workshop*, 2018.

Wizard of Oz @ HCI-UA-2013

Paulo Dias, T. Sousa, J. Parracho, I. Cardoso, A. Monteiro, Beatriz Sousa Santos
“Student Projects Involving Novel Interaction with Large Displays”, IEEE Computer Graphics and Applications, vol.34, no.2, Mar.-Apr. 2014, pp.80-86

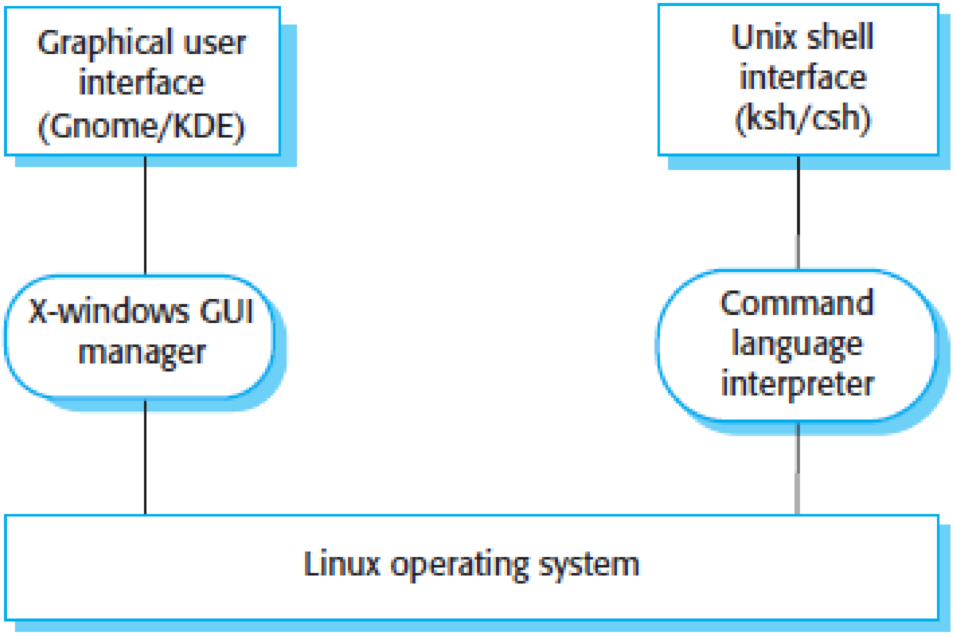
Used to get insight on which gestures might be more intuitive to control a Pac-Man game

Main advantages and disadvantages of interaction styles

Interaction style	Main advantages	Main disadvantages	Application examples
Direct manipulation	Fast and intuitive interaction Easy to learn	May be hard to implement Only suitable where there is a visual metaphor for tasks and objects	Video games CAD systems
Menu selection	Avoids user error Little typing required	Slow for experienced users Can become complex if many menu options	Most general-purpose systems
Form fill-in	Simple data entry Easy to learn Checkable	Takes up a lot of screen space Causes problems where user options do not match the form fields	Stock control Personal loan processing
Command language	Powerful and flexible	Hard to learn Poor error management	Operating systems Command and control systems
Natural language	Accessible to casual users Easily extended	Requires more typing Natural language understanding systems are unreliable	Information retrieval systems

(Sommerville, 2010, chap.29)

Multiple user interfaces example



(Sommerville, 2010, chap.29)

3D User Interfaces

- User interfaces involving 3D interaction (i.e. interaction in which the user's tasks are performed directly in a 3D spatial context).

- Are more and more used:

  - Virtual and augmented reality
  - 3D workspaces
  - Data Visualization …

- But have some issues:
  – User disorientation
  (in the real world we have more information)

Applications of virtual and augmented reality

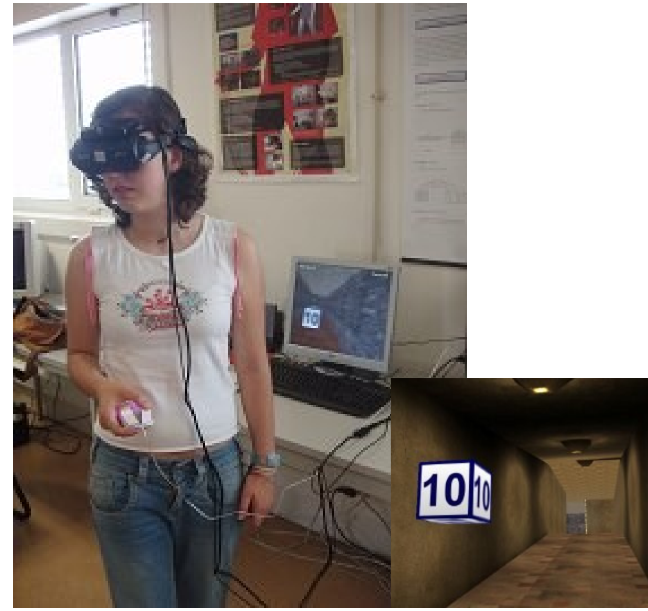
- Training and simulation

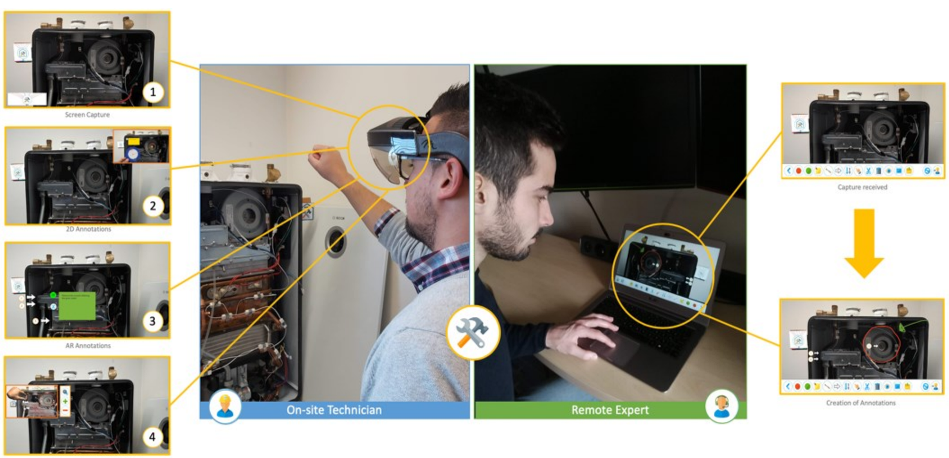
- Project review

- Therapy

- Entertainment

…





Main bibliography

- B. Shneiderman et al., *Designing the User Interface- Strategies for Effective Humaman–Computer Interaction*, 5th ed., Addison Wesley, 2009

- Soegaard, Mads. Interaction Styles

  http://www.interactiondesign.org/encyclopedia/interaction_styles.html

  https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/3d-user-interfaces

- Ian Sommerville, Software Engineering, 9 ed, Addison Wesley , 2010
  https://ifs.host.cs.st-andrews.ac.uk/Books/SE9/WebChapters/PDF/Ch_29%20Interaction_design.pdf

Universidade de Aveiro
Departamento de Electrónica,
Telecomunicações e Informática

Output devices

Interfaces Humano-Computador

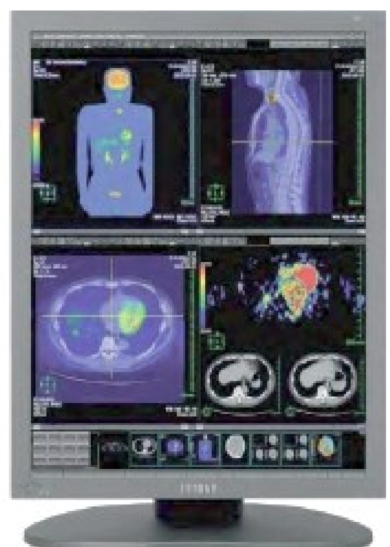
Beatriz Sousa Santos

The ultimate display?

"The ultimate display would, of course, be a room within which the computer can control the existence of matter. A chair displayed in such a room would be good enough to sit in. Handcuffs displayed in such a room would be confining, and a bullet displayed in such a room would be fatal." (Ivan Sutherland, 1965)

We are not yet there …

There are a lot output devices for a lot of different applications
Visual displays:



Less conventional visual displays

What future?

https://vrscout.com/news/snapchat-reveals-next-gen-ar-glasses/
(May, 2021)

And not only to produce visual displays…

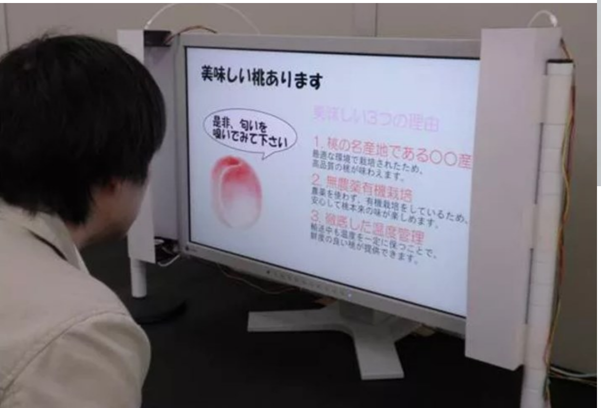
http://www.geomagic.com/en/products/phantom-omni/overview

sound

Touch and force feedback

http://www.cyberglovesystems.com/cyberglove-ii/

smell



https://vrscout.com/news/olfactory-engineering-scent-based-vr/

Graphics/visual Displays

Are computer interfaces that present images to one or several users

A possible taxonomy:

- Personal displays:
  - monitors
  - HMDs (VR/AR)
  - Binoculars
  - Monitor-based displays/active glasses
  - Autostereoscopic displays

- Large volume displays:
  – Caves
  – Walls
  – Domes
  –…

Personal Displays

The images may be monoscopic or stereoscopic, monocular (for a single eye) or binocular (displayed on both eyes).

- Screens of various sizes

- Head Mounted Displays (HMDs)

- 3-D Binoculars (hand supported)

- Auto-stereoscopic displays (desk supported)

Large-volume displays

- CAVE type displays
- Wall-type displays
- Domes
- ,,,

http://www.mechdyne.com/cave2.aspx

- Main technologies:
  - LED displays (several types)
  - LCD displays
  - Autostereoscopic displays: lenticular/barrier
  - …
- Other technologies:
  electrophorectic,…

https://en.wikipedia.org/wiki/E_Ink

- **Images provided by computer monitors are poor when compared to the real world**

- **It is amazing what we get from such simple devices**

- Monitors have several limitations:
  - Small range of intensities and colors
  - Lack of focusing distance
  - Small field of view
  - ...

Voice synthesizers

- There are several types:

  Digitized - concatenates recorded basic sounds

  Synthesised – concatenates sounds generated with models

- There are several technical challenges due to the nature of human voice:

  different pronunciation rules

  meaning may be changed by intonation

  differences in intonation reflect different moods

- **The quality of a synthesizer implies much more than intelligibility**

Advantages of using voice output:

When the user has:

- physical deficiency
- to move around
- hands and eyes busy
- Adverse conditions: low visibility, low O_2, high Gs

Disadvantages:

- Is tiresome and uncomfortable for long periods
- Is transient (taxes STM)
- May have privacy issues
- May disturb other people

Examples of using voice input/output and natural language interaction style:

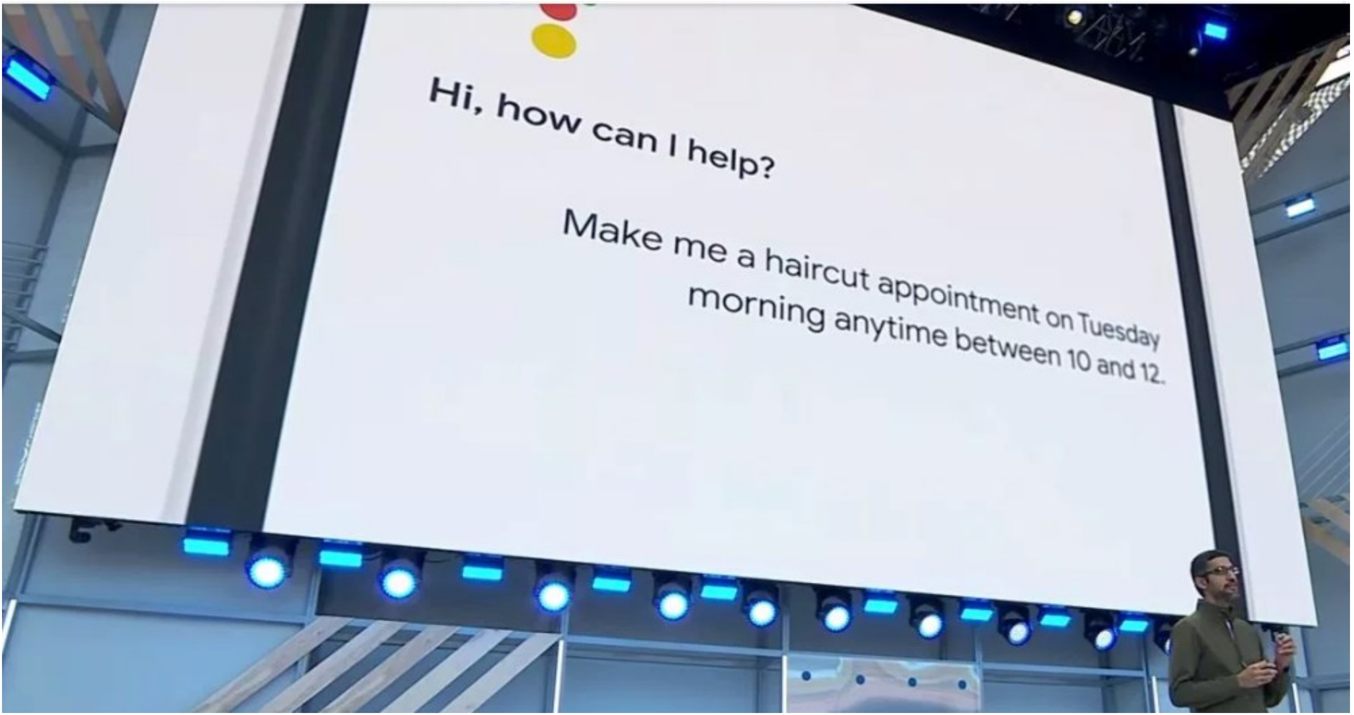
- Siri
- Alexa
- Google Home
- Google Duplex

https://en.wikipedia.org/wiki/Amazon_Alexa

Google Duplex

Is a culmination of various efforts over the years in deep learning, natural language understanding, speech recognition, and text-to-speech.

“In the domain of making appointments, it passes the Turing test … which is an extraordinary breakthrough. It doesn’t pass it in the general terms …”



https://9to5google.com/2018/05/21/google-duplex-explained-turing-test/

Some guidelines to use voice output

- Consider voice output as an alternative when the user must move around, has hands and eyes busy

- Avoid voice output in open environments, when the privacy and security are important issues and frequency of usage is high

- Use approx. 180 words per minute

- When messages are not expected, start with non-critical words that provide context

- Say first the goal and then the solutions

- Allow messages to be repeated

Example: Medical VR Total Knee Surgical Simulator Demo

https://ghostproductions.com/medical-vr-virtual-reality/surgical-training/

A glimpse of the future? Interactive live holography

http://realviewimaging.com/technology/

Conclusion

- **Technology shall not be used only because it is new!**

- **Independently from the type or state of the art of the input / output devices it is necessary to understand their usability for different types of users, tasks and context**

Universidade de Aveiro
Departamento de Electrónica,
Telecomunicações e Informática

Screen Layout Design and Color

Beatriz Sousa Santos 2020

- The screen design is an important part of the UI development
- A poor screen design may degrade user performance
- Screen layout must be carefully designed
- There are numerous guidelines (we have seen already some of them)

Screen Layout Guidelines

- Several types:

General layout of information

messages

Text

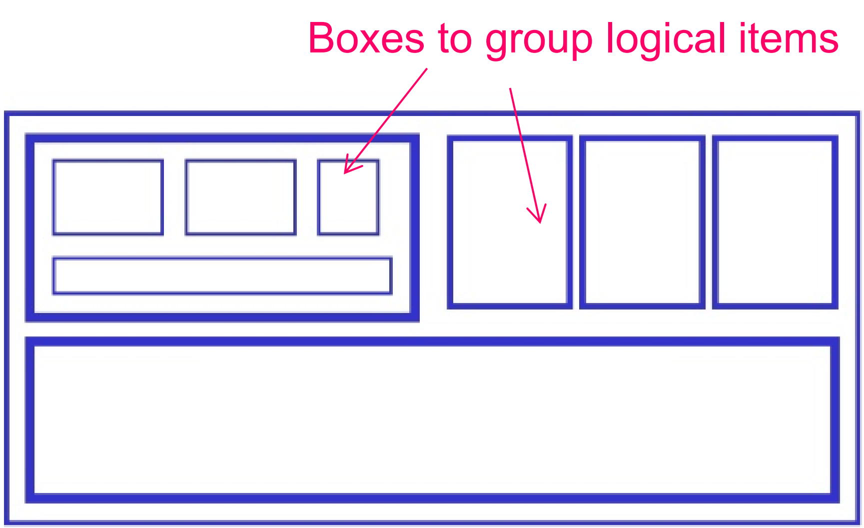
instructions

Numbers

Coding techniques (color and others)

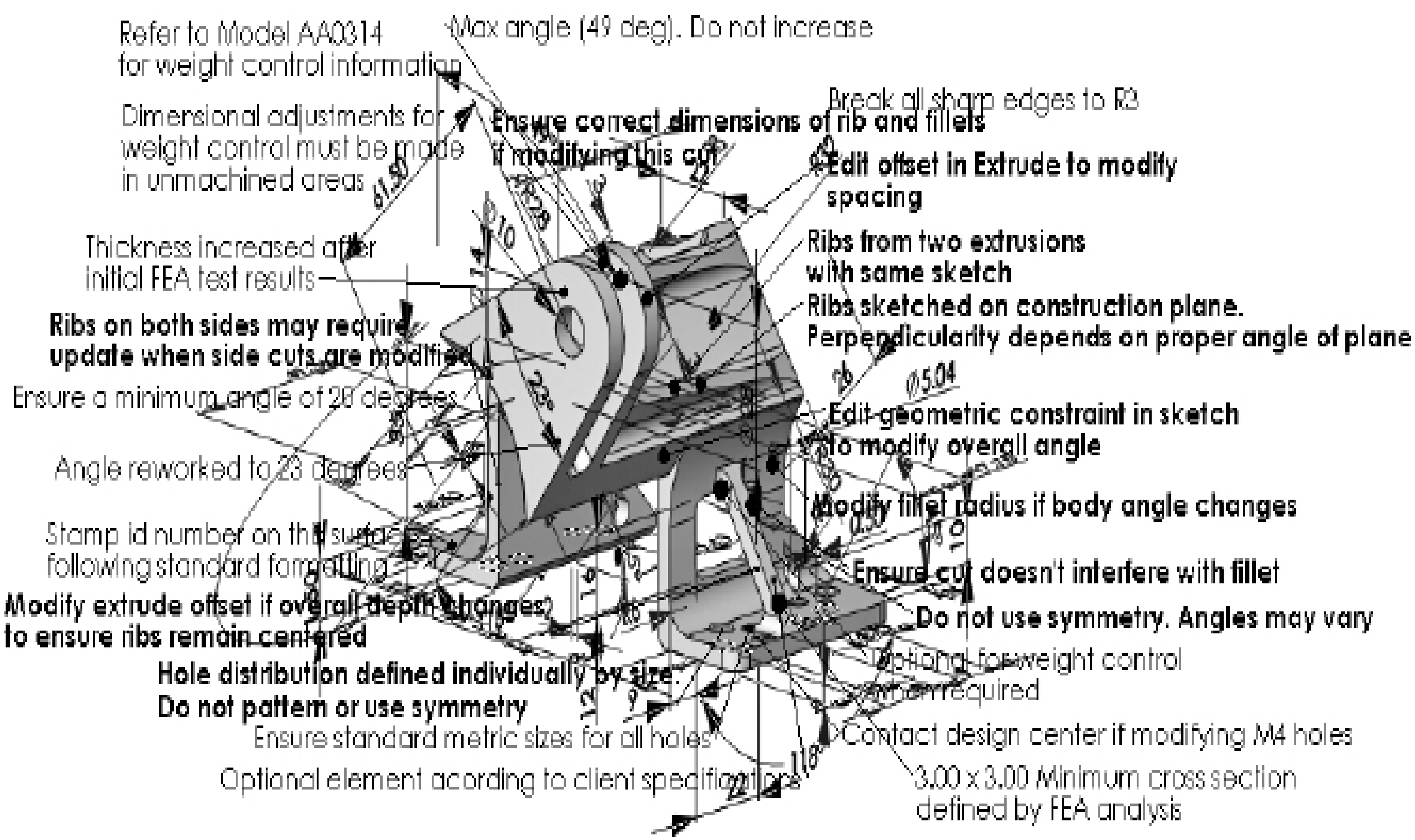
Information layout

- Include only the needed information
- Include all needed information
- Begin at the top left corner and align left (in Western culture)
- Group items according to type
- Leave plenty of white space
- Use leaders in multiple columns



Include only the needed information

Avoid Visual Clutter



Text

- Avoid using only capital letters (are more difficult to read)
- Avoid text with many capital letters
- Do not use too many fonts for emphasis
- In multiple columns use leaders or greying

use fonts for emphasis
(but not too many)

ABCDEF HIJKLM
NOPQRSTUVWXYZ

Alcântara - Terra			12:36			13:06			13:36	
Campolide	12:15		12:41	12:45		13:11	13:15		13:41	13:45
Rossio	12:19			12:49			13:19			13:49
Sete Rios		12:19	12:43		12:49	13:13		13:19	13:43	
Entrecampos		12:22	12:47		12:52	13:17		13:22	13:47	
Roma - Areeiro		12:24	12:49		12:54	13:19		13:24	13:49	

Use greying

Willy Wonka and the Chocolate Factory
Winston Churchill - A Biography
Wizard of Oz
Xena - Warrior Princess

Left aligned: more readable

Willy Wonka and the Chocolate Factory
Winston Churchill - A Biography
Wizard of Oz
Xena - Warrior Princess

Right aligned: fine for effects
But more difficult to read

In multiple columns it is difficult to read across gaps:

sherbert	75
toffee	120
chocolate	35
fruit gums	27
coconut dreams	85

use leaders

sherbert ________	75
toffee ________	120
chocolate ________	35
fruit gums ________	27
coconut dreams ________	85

or greying

sherbert	75
toffee	120
chocolate	35
fruit gums	27
coconut dreams	85

- Messages shall:
  - Have a detail level adequate to user knowledge and experience
  - Be specific and understandable
  - Be brief and concise
  - Be positive
  - Be helpful

Error messages

Too verbose

better

The processing of the text editor yielded 23 pages of output	Output 23 pages
Error in SIZE field	Error: SIZE range is 4 to 16
Cannot exit before saving file	Save file before exiting
Bad/illegal file name	Maximum file name length is 8 chars
Syntax error 1542	Unmatched left parenthesis in line 210

Too vague

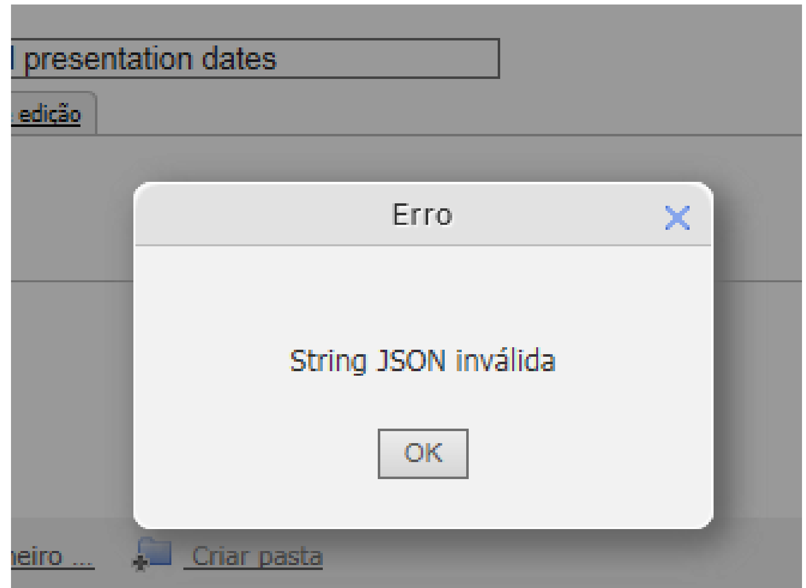
Negative

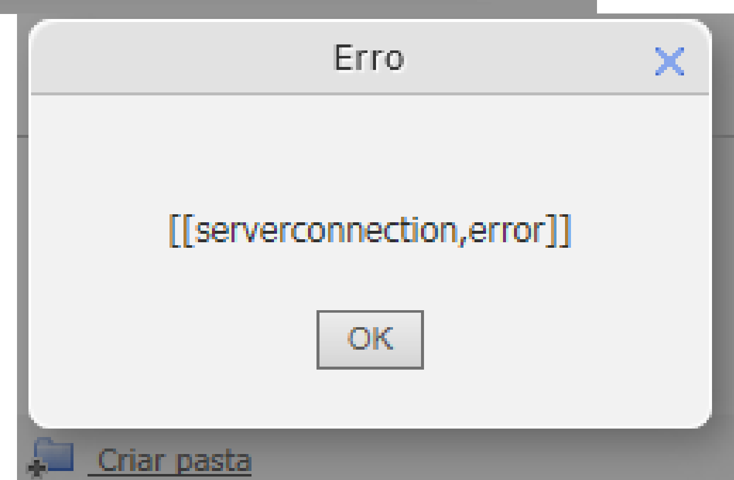
Not helpful

Examples of useless messages for users

Except (maybe) for Chinese people!

Moodle:



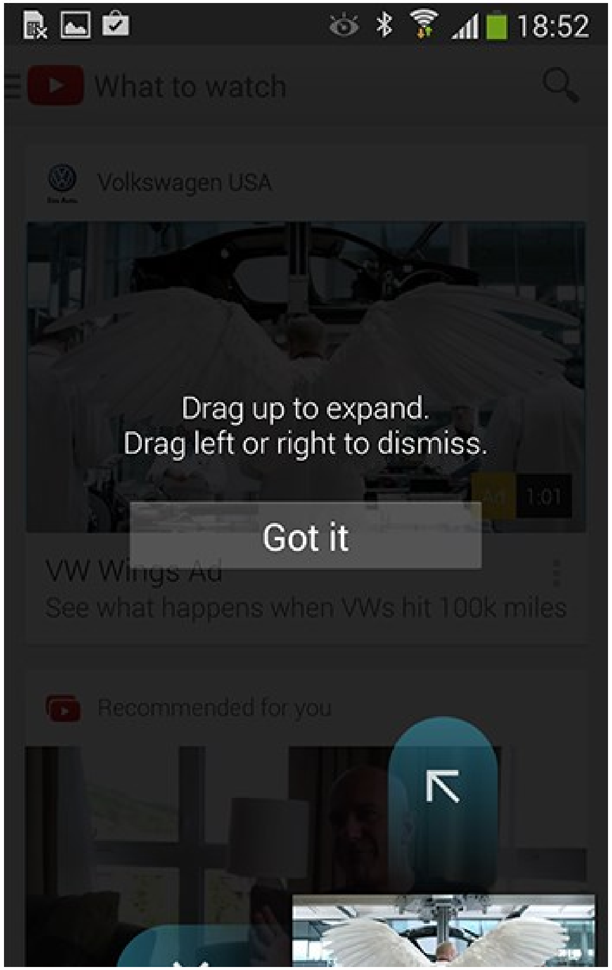


Instructional Overlays and Coach Marks for Mobile Apps

- Instructions in mobile applications must be designed for optimal scannability, as users tend to dismiss them quickly and do not read thoroughly

Main guidelines:

- Short, Focused Tips
- Avoid Chains of Tips
- Use Visuals When Possible
- Keep Tips Sparse



https://www.nngroup.com/articles/mobile-instructional-overlay/

Numbers

- Integers shall be right justified
- Real numbers shall be aligned by the decimal point
- Avoid unnecessary zeros (at left)
- Long numbers shall be divided in groups of 3 or 4

Which is the largest?

532.56	627.865	
179.3	1.005763	75
256.317	382.583	120
15	2502.56	35
73.948	432.935	27
1035	2.0175	85
3.142	652.87	
497.6256	56.34	

Right align integers

Align decimal points

Numbers

	Better
10 100 1000 10000	10 100 1000 10000
100.00 25.365 5432.01 1.45591	100.00 25.365 5432.01 1.45591
10:1 p.m. 002	10:02 p.m. 2
6173954686	617-395-4686

Coding techniques

- Blinking
- Bold
- Size
- Font
- Underlining
- Shape
- Special characters and icons
- Proximity
- Borders
- Sound
- Colour

Main guideline: use parsimoniously any coding technique!

Specific problems for different platforms: mobile

- Many guidelines are similar for mobile and desktop design, but their mobile interpretation is much more unforgiving
  - Context of use
  - Size of screen
  - Platform limitations

http://www.nngroup.com/articles/mobile-sharpens-usability-guidelines/

https://developer.android.com/design/index.html

Links on tablet and mobile usability

Raluca Budiu, The State of Mobile User Experience, NNGroup, March, 2015
http://www.nngroup.com/articles/mobile-usability-update/

https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/

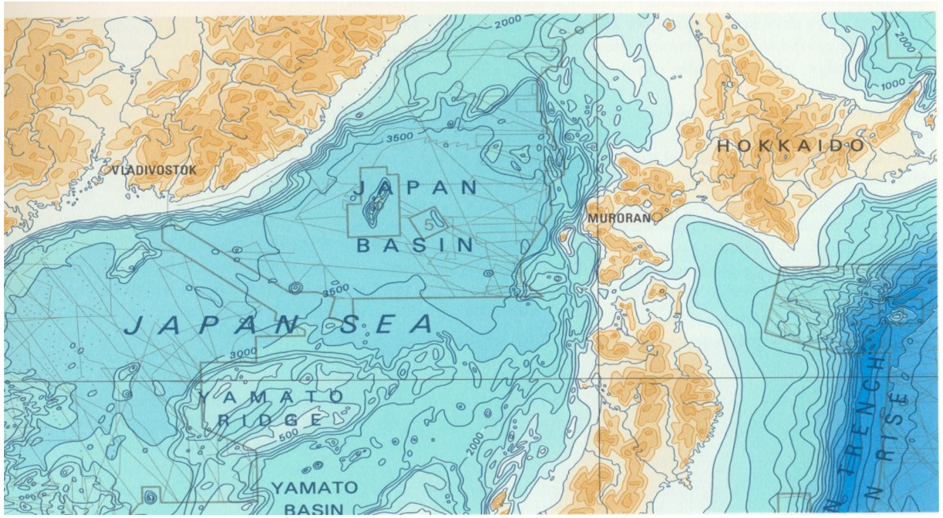
https://developer.android.com/design/index.html

A critical situation: automotive dashboards

https://www.scientificamerican.com/article/automobile-dashboard-technology-is-simply-awful/
(April 2018)

Universidade de Aveiro
Departamento de Electrónica,
Telecomunicações e Informática

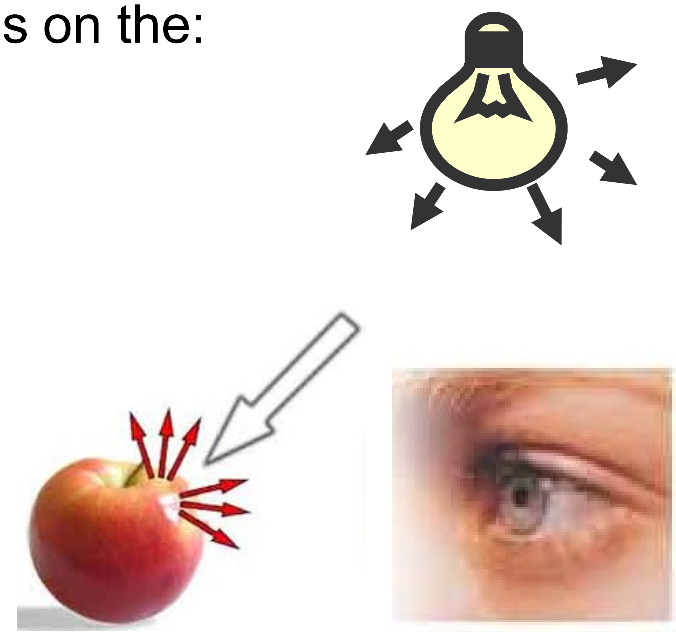
Color usage



Color

- Color is a complex and multidisciplinary subject:
  - Physics
  - Physiology and psychology
  - Art and graphic design

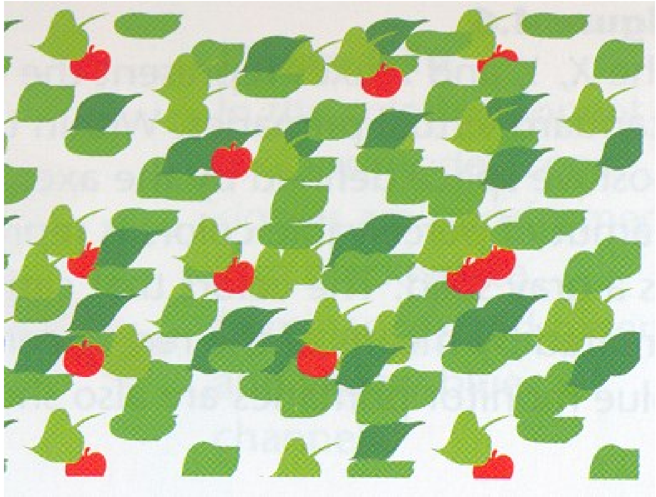
- The perceived color of an object depends on the:
  - Material characteristics
  - Illumination
  - Ambient color
  - Human visual system



How many cherries?

(Ware, 2004)

How many cherries?



Color may support users in many tasks!
(yet, if not properly used may make them more difficult!)

Using color

Besides increasing realism, it may have the following **advantages**:

It may:

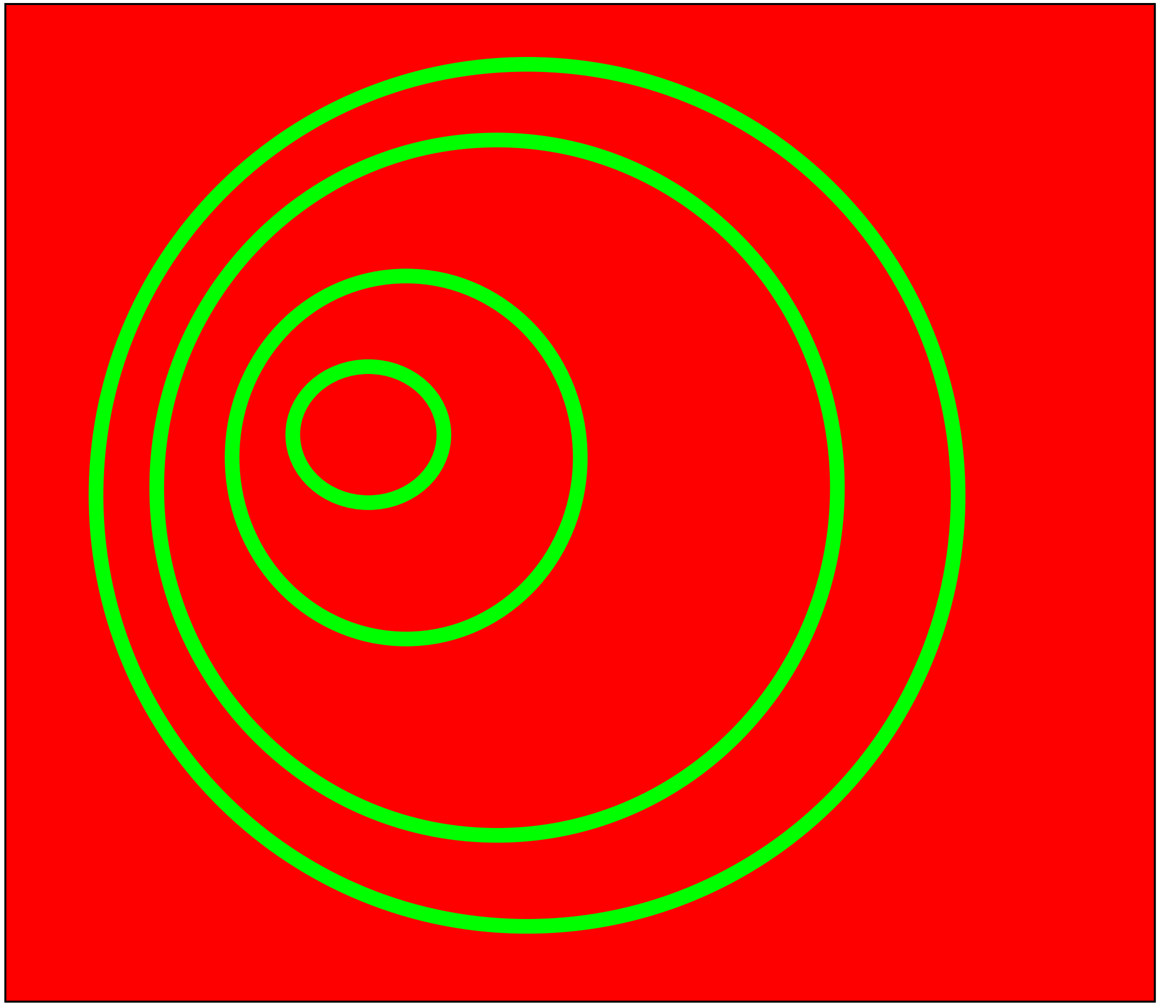
- Show the logical organization of the information displayed
- Represent values
- Catch the attention
- Increase satisfaction
- Ease the search in complex displays
- Trigger emotions

…

However, **it may degrade user's performance** if not used properly

Guidelines for using color

- **Use color parsimoniously**
- Use a limited number of colors
- Firstly make it work without color
- Use color coherently
- Avoid using simultaneously several saturated colors
- Do not convey information solely through color
- Make color coding support the user task
- Make the color coding as obvious as possible
- Allow the user to control the color code
- Take into account the cultural meaning of colors …



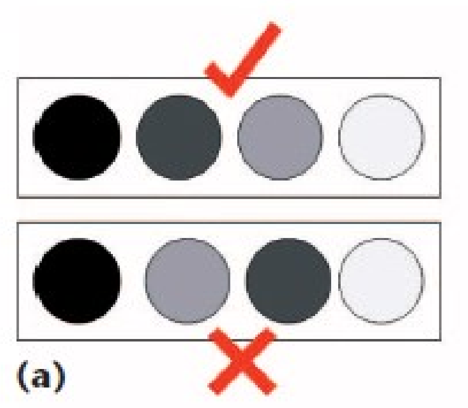
Saturated complementary colors should not be used simultaneously

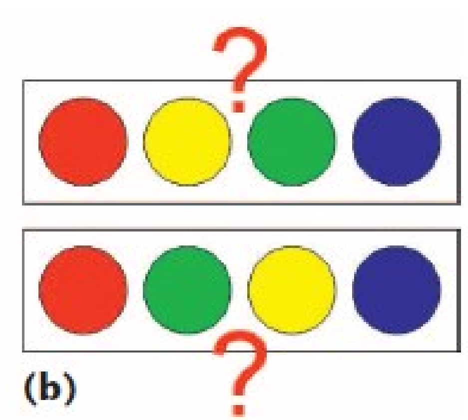
Small spots of color on a neutral background enhance relevant information

(Tufte, 1990)

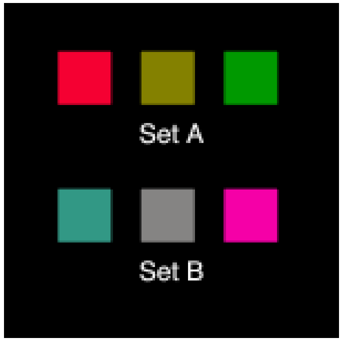
- Do not expect to easily perceive order from color

(Borland, Taylor II, 2007)

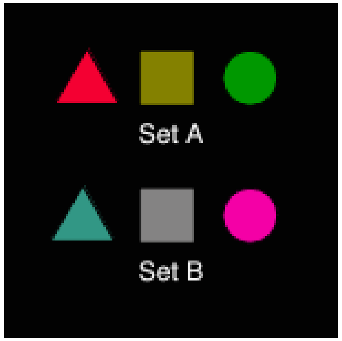


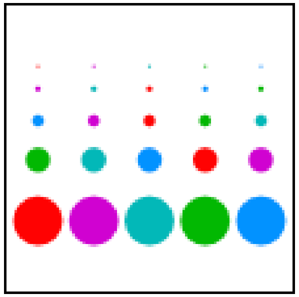


The elements within these sets look identical to deuteranopes, the most common kind of dichromat:

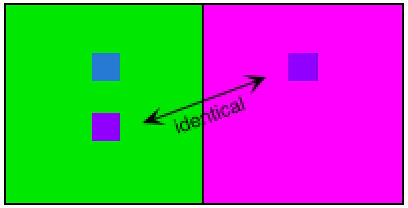


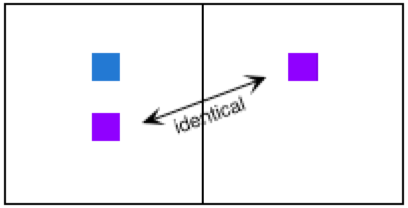
These can be discriminated on the basis of non-color differences:





Don’t use colour coding on small elements





Use neutral gray surrounds where color judgments are critical.

Color Vision deficiencies

- ≈ 8% of men and 1% of women have some type of color vision deficiency
- Generally it is genetic (associated to the X chromosome)
- Common deficiencies are explained by the lack of cones (color sensor cells in the retina) sensitive to the long and medium λ (dicromacies)):
  - Protanopia (LW – "Red" cone)
  - Deuteranopia (MW – "Green" cone) (Daltonism)
- There are three types of inherited deficiencies:
  - Monocromacy (disorder or lack of all color sensitivity)
  - Dicromacy (disorder or lack of one type of cone)
  - Anomalous Tricromacy (disorder in cones)

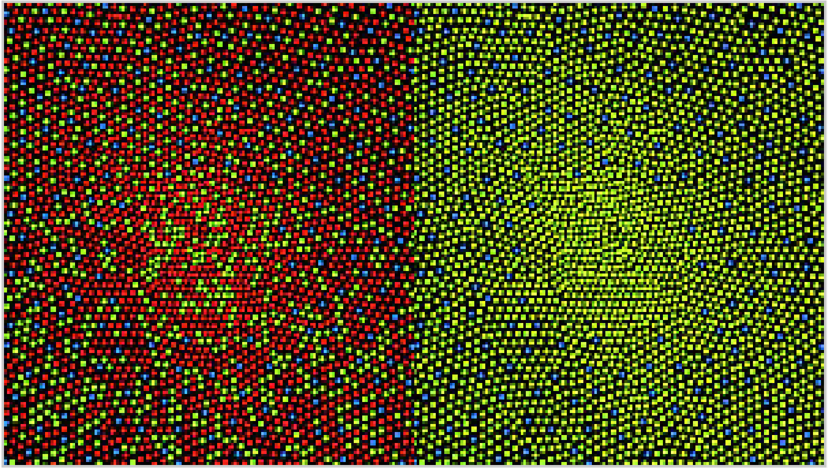


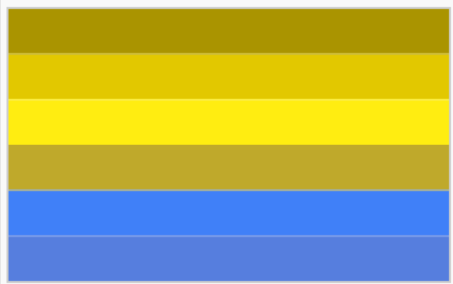
Illustration of the distribution of cone cells in the fovea of an individual with normal color vision (left), and a color blind (protanopic) retina. Note that the center of the fovea holds very few blue-sensitive cones.

http://en.wikipedia.org/wiki/Photoreceptor_cell

Rainbow colors as viewed by people suffering from color vision deficeiencies

The colors of the rainbow as viewed by a person with no color vision deficiencies.

The colors of the rainbow as viewed by a person with protanopia.



The colors of the rainbow as viewed by a person with deuteranopia.

The colors of the rainbow as viewed by a person with tritanopia.

http://en.wikipedia.org/wiki/Color_blindness

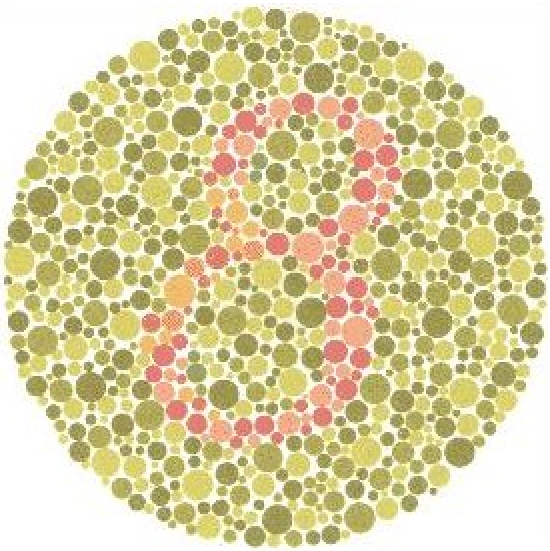
Simulating color vision deficiencies

Original image as seen by a normal observer

As seen by a deuteranope (daltonic)

http://www.daltonize.org/

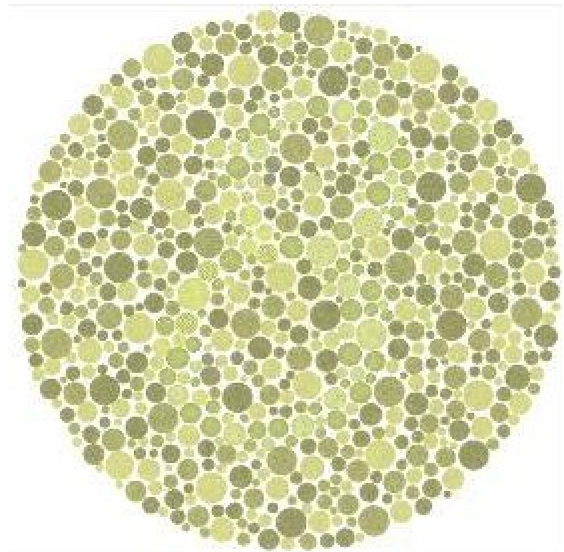
Simulating color vision deficiencies
Ishihara-2



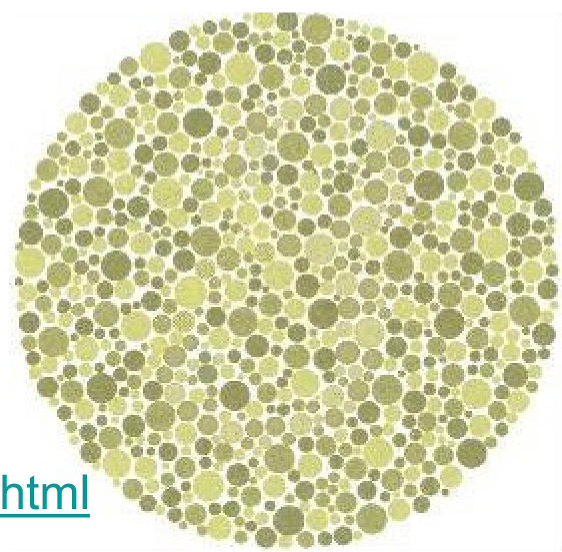
Original image as seen
by a normal observer

As seen by an observer
with a color vision
deficiency:

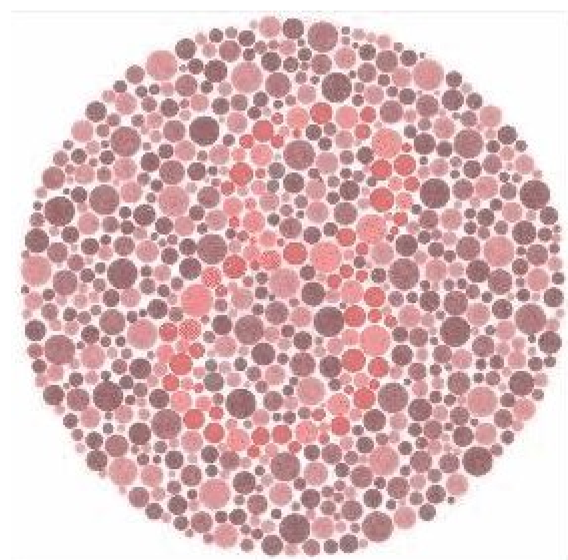
Green-Blind/Deuteranopia



Red-Blind/Protanopia



Blue-
Blind/Tritanopia

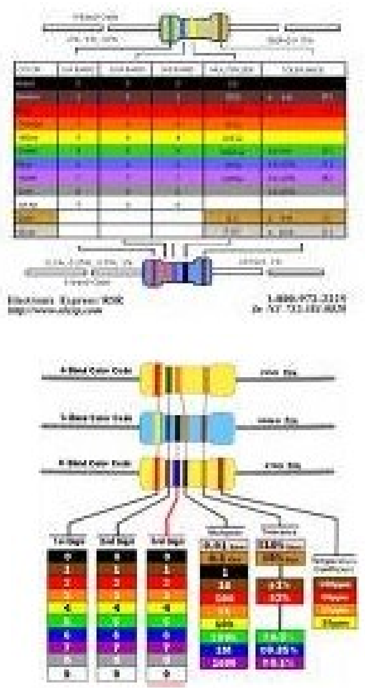


http://www.color-blindness.com/coblis/coblis.html

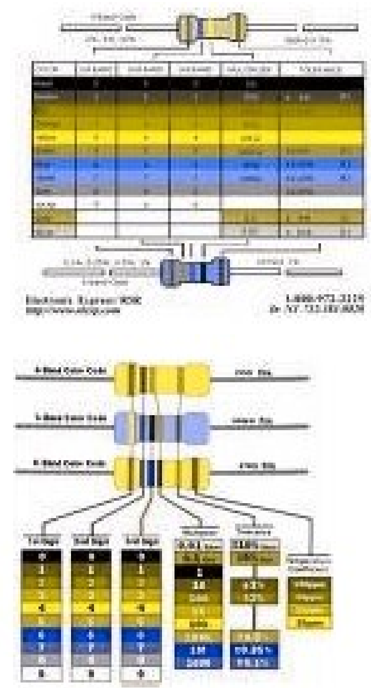
A commonly used color code

(that does not use color redundantly)

Original Image



as seen by a deuteranope



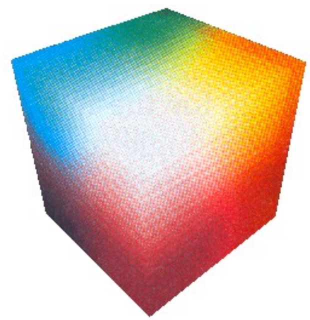
It does not work for deuteranopes !!

How can we describe color experience?

- Color perception happens in the mind due to light properties
- Different color descriptions are necessary for:

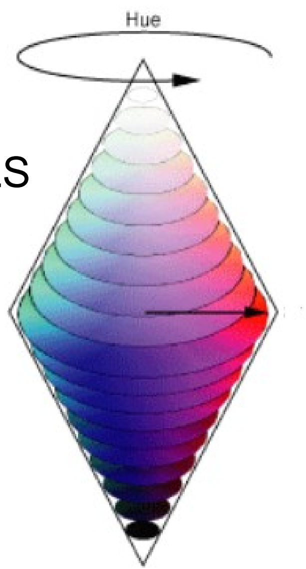
light stimuli		**color sensations**
external (physical)	→	subjective (mental)
photometry, colorimetry		color attributes

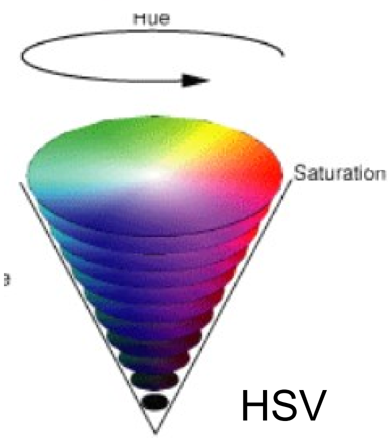
Color Models



RGB

HLS



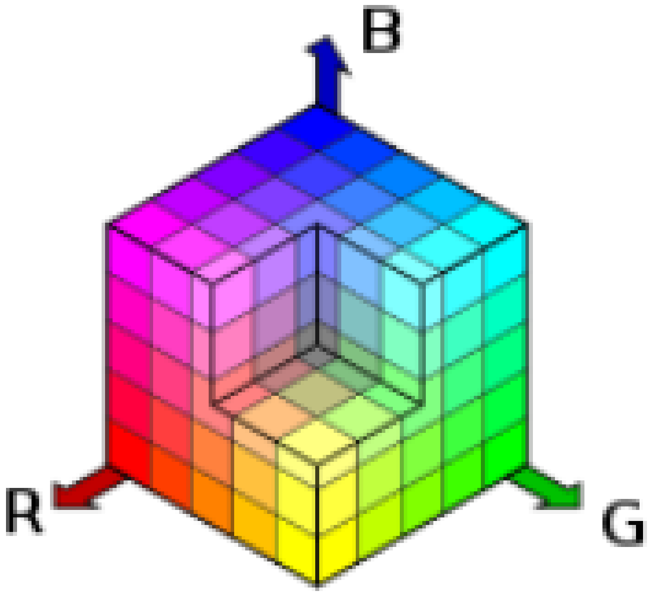


Munsell

HSV

- Objects are perceived as having a color depending on the spectrum of the reflected light (or emitted)
- But different spectra may induce similar color sensations
- It is important to be able to describe color objectively
- There are to types of color production systems:
    - Additive (eg.: monitors, TV sets, projectors) $\longrightarrow$ RGB
    - Subtractive (e.g.: printers) $\longrightarrow$ CMY
- RGB and CMY are H/W oriented color models not adequate for users
- There are more color models …

The RGB color model:



The **RGB color model** is an additive color model in which red, green, and blue light (the primary colors) are added to reproduce a broad array of colors.

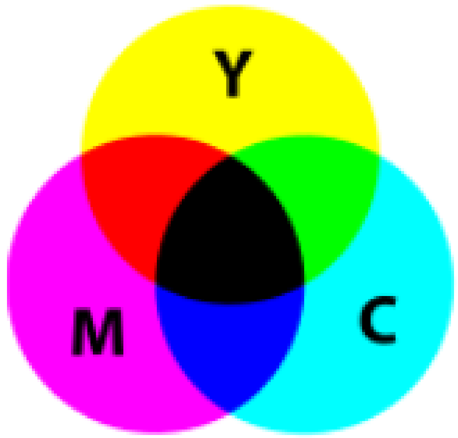
The color space is a cube in a Cartesian coordinate system

White -> 1, 1, 1

Black -> 0, 0, 0

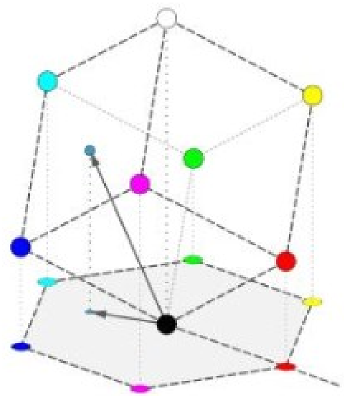
https://en.wikipedia.org/wiki/RGB_color_model

The **CMY color model** is a subtractive color model in which cyan, magenta, and yellow (the primary colors) are subtracted from white to reproduce a broad array of colors.



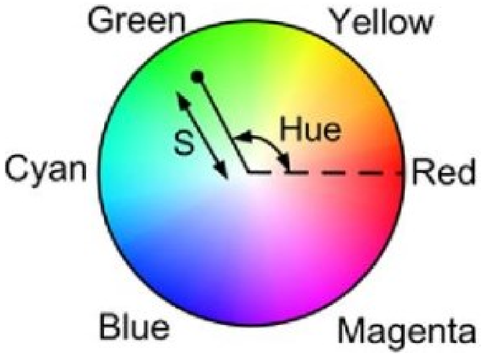
The color space is also a cube in a Cartesian coordinate system

White -> 0, 0, 0
Black -> 1, 1, 1

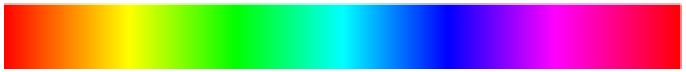


There are other models more adequate to color specification by the users:

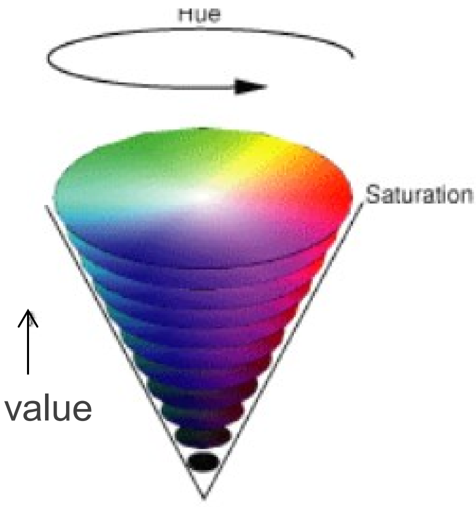
- HSV
- HLS



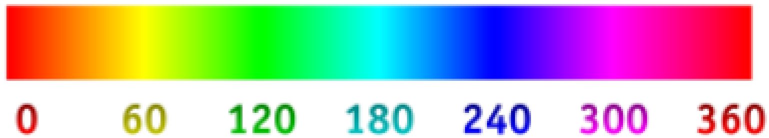
- Humans describe color based on 4 psychophysical variables related to physical variables:
  - Hue – the degree to which is similar to or different from stimuli that are described as red, green, blue, and yellow
  - Saturation – related to the amount of achromatic light
  - Lightness – related to the objects reflectance (for reflecting objects)
  - Brightness – for light emitting objects



https://en.wikipedia.org/wiki/Hue



- HSV color model:
  - Hue

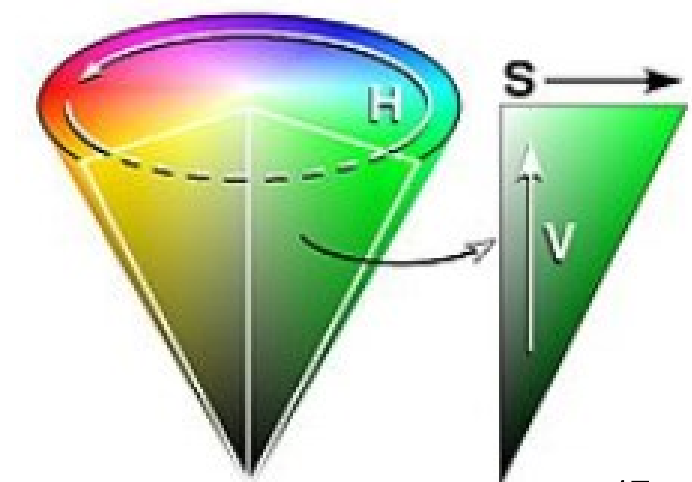


  - Saturation – related to the amount of achromatic light
  - Value - controls the brightness: 0% - pure black   100% - pure white

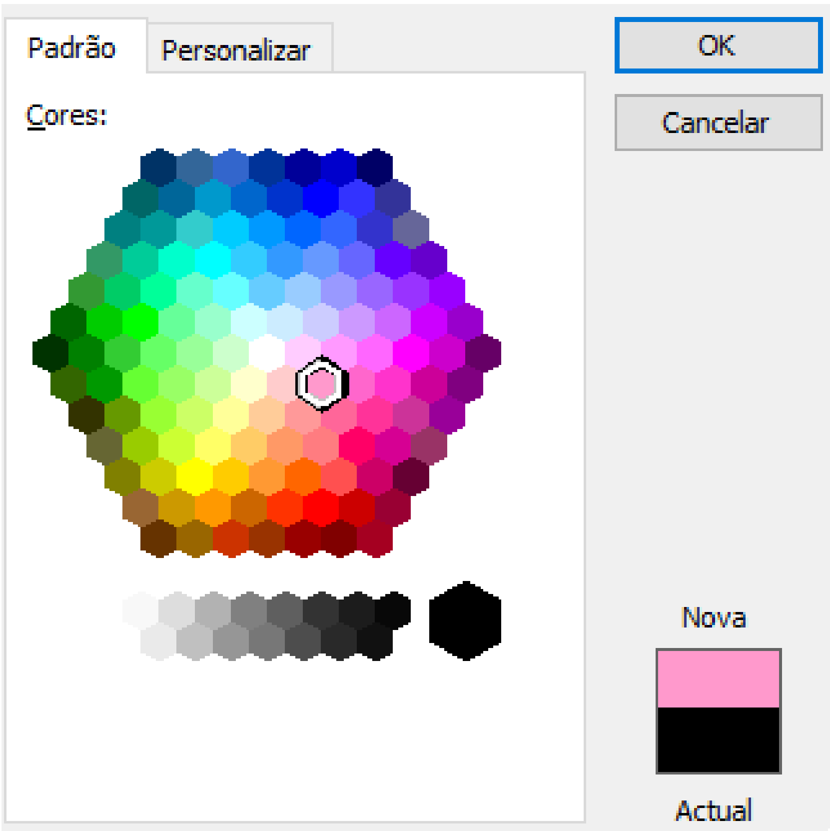
Uses cylindrical coordinates

https://www.khanacademy.org/partner-content/pixar/color/color-101/v/color-3

https://programmingdesignsystems.com/color/color-models-and-color-spaces/index.html



- Let the user select a color:



Interesting Links

- Introduction to color guidelines and standards (NASA)

http://colorusage.arc.nasa.gov/guidelines_0.php

- Effective Visual Communication for Graphical User Interfaces

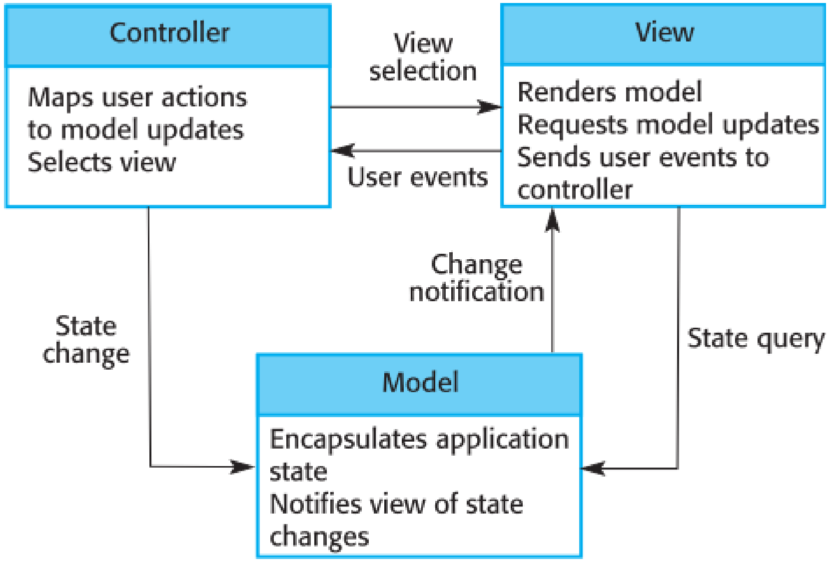
http://web.cs.wpi.edu/~matt/courses/cs563/talks/smartin/int_design.html

- Ergonomic design for human interface design, Cornell University Ergonomics Web

http://ergo.human.cornell.edu/ahtutorials/interface.html

Universidade de Aveiro
Departamento de Electrónica,
Telecomunicações e Informática

S/W Patterns for Interactive Systems



- A complete application consists of:
  - User interface (UI)
  - Core functions of the application

- How can we effectively develop interactive applications with these two parts?

- Follow an established development methodology suited for interactive systems:

Architectural patterns

S/W Patterns – historical perspective

- 1987

  First small pattern language for **designing user interfaces**
  by K. Beck and W. Cunningham
  (inspired by building and urban architect C. Alexander)

- 1993

  Pattern Languages of Programming (PLoP) conference series
  by The Hillside Group
  https://www.hillside.net/conferences

- 1994

  "Design Patterns: Elements of Reusable Object-Oriented Software"
  by Erich Gamma et al. (aka the Gang of Four (GoF))
  https://hillside.net/patterns/about-patterns

S/W Patterns

- Proven **solution to a problem in a context** (Gamma et al., 1994)

- Each pattern documents a reusable solution, encapsulates knowledge about successful practices, and provides information about its usefulness and tradeoffs

- Many companies have written pattern collections:

  Amazon,

  Google,

  IBM,

  Microsoft,

  Oracle,

  Siemens, etc.

  https://hillside.net/patterns/about-patterns

Content of a Pattern

- Different formats are used for describing patterns, generally including:
  - **Name**
  - **Problem**
  - **Context**
  - **Solution**
  - Forces
  - Resulting Context
  - Examples
  - Rationale
  - Related Patterns
  - Known Uses
- May include an Abstract providing an overview of the pattern and indicating the types of problems it addresses and the target audience

https://pubs.opengroup.org/architecture/togaf9-doc/arch/chap22.html

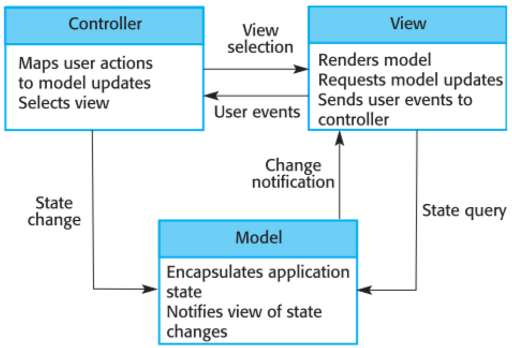
Forces of a Pattern

- Security, robustness, reliability, fault-tolerance
- Efficiency, performance, throughput, bandwidth requirements, space utilization
- Ease-of-use
- Ease-of-construction
- Completeness and correctness
- Scalability (incremental growth on-demand)
- Extensibility, evolvability, maintainability
- Modularity, independence, re-usability, openness, composability (plug-and-play), portability
- etc.

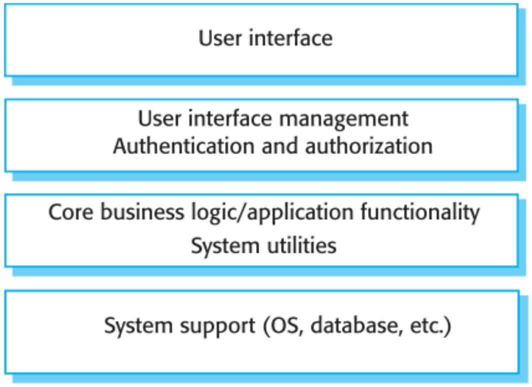
https://pubs.opengroup.org/architecture/togaf9-doc/arch/chap22.html

Architectural patterns often used in interactive s/w

– Model View Control (MVC)



– Layered architecture



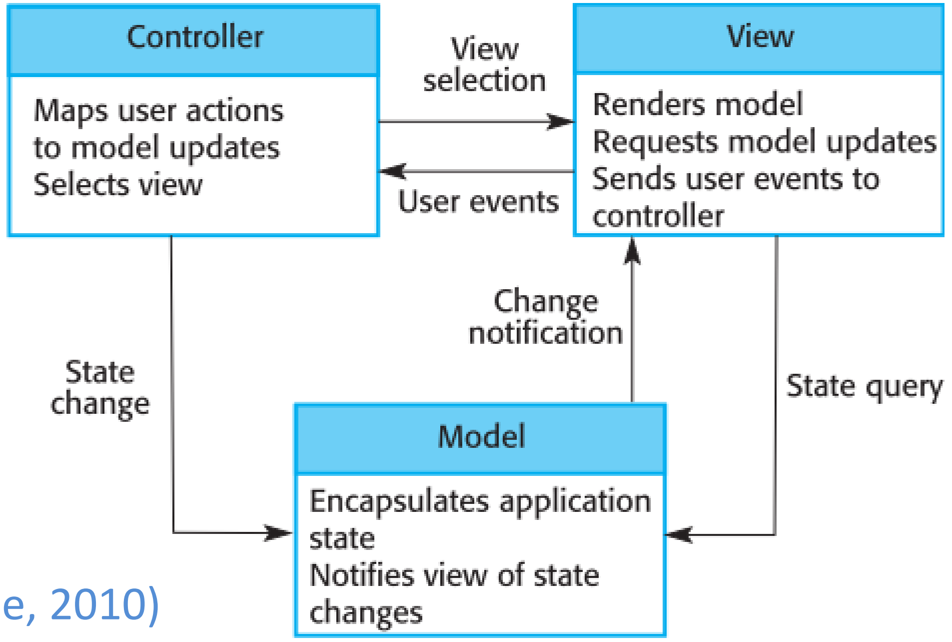
(Sommerville, 2010)

Model–view–controller (MVC)

- MVC is an architectural pattern/paradigm commonly used for interactive systems
- Was proposed in the 70's as a computational architecture for interactive programs by the designers of SmallTalk (one of the first object-oriented and modular languages)
- It expresses the "core of the solution" to a problem while allowing it to be adapted for each system

Model–view–controller (MVC)

- The application is divided in three parts:
  - Model
  - View
  - Controller



(Sommerville, 2010)

- Separating internal representations of information from the ways it is presented to and accepted from the user
- It decouples these major components allowing for efficient code reuse and parallel development.

- Originally developed for desktop computing, MVC has been widely adopted as an architecture for Web applications (and others…)
- Frameworks vary in their interpretations in the way that the MVC responsibilities are divided between the client and server
- Some web MVC frameworks take a thin client approach that places almost the entire model, view and controller logic on the server: the model exists entirely on the server
- Other frameworks allow the MVC components to execute partly on the client

MVC Components

- The ***model*** is the central component of the pattern:

It expresses the application's behavior in terms of the problem domain, independent of the UI. It manages the data, logic and rules of the application

- A ***view*** can be any output representation of information:

Multiple views of the same information are possible, such as a bar chart for management and a tabular view for accountants

- The ***controller*** accepts input and converts it to commands for the model or view

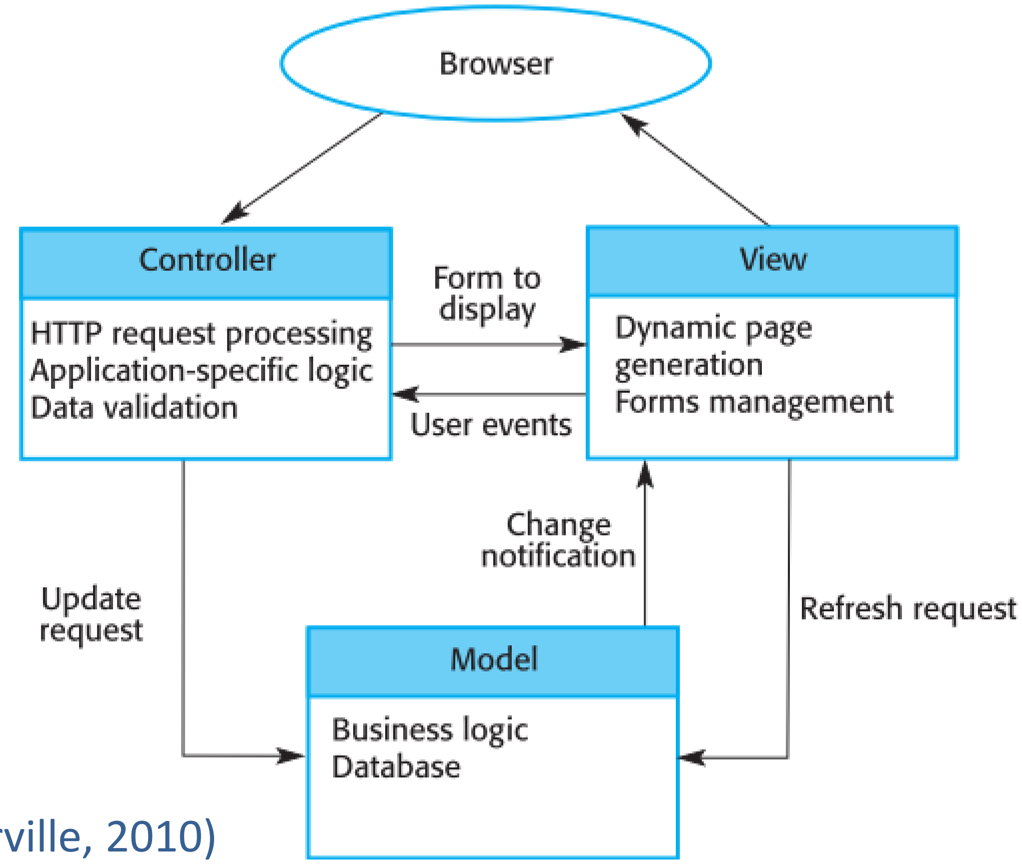
MVC Interactions

- This design defines the interactions among the three components
- The model is responsible for managing the data of the application. It receives user input from the controller
- The view means presentation of the model in a particular format
- The controller is responsible for responding to the user input and perform interactions on the data model objects. The controller receives the input, optionally validates the input and then passes it to the model

Name	MVC (Model-View-Controller)
Description	Separates presentation and interaction from the system data. The system is structured into three logical components that interact with each other. The Model component manages the system data and associated operations on that data. The View component defines and manages how the data is presented to the user. The Controller component manages user interaction (e.g., key presses, mouse clicks, etc.) and passes these interactions to the View and the Model. See Figure 6.5.
Example	Figure 6.6 shows the architecture of a web-based application system organized using the MVC pattern.
When used	Used when there are multiple ways to view and interact with data. Also used when the future requirements for interaction and presentation of data are unknown.
Advantages	Allows the data to change independently of its representation and vice versa. Supports presentation of the same data in different ways, with changes made in one representation shown in all of them.
Disadvantages	May involve additional code and code complexity when the data model and interactions are simple.

(Sommerville, 2010)

Example:
Web application architecture using MVC



(Sommerville, 2010)

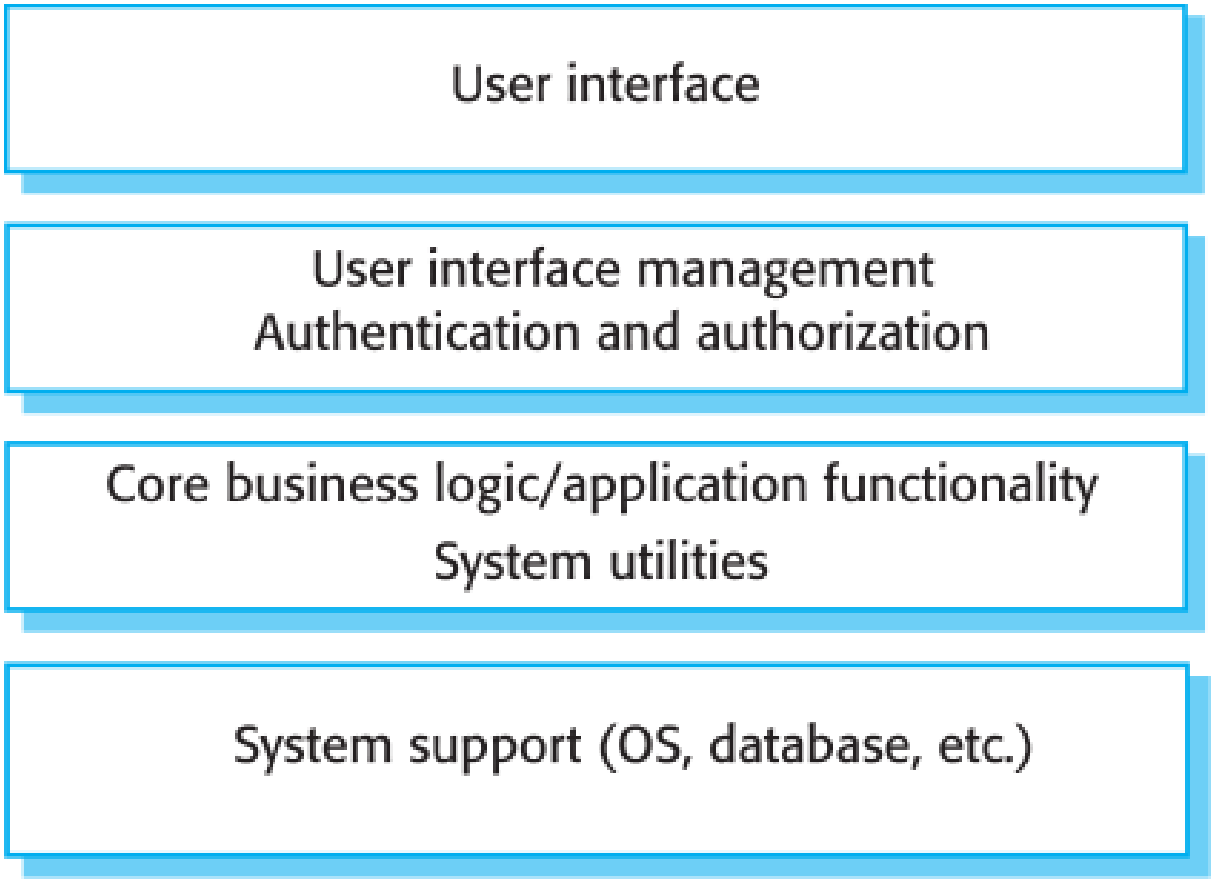
Layered Architecture

- Separation and independence are fundamental to S/W architectural design because they allow changes to be localized
- The Layered Architecture pattern is another way of achieving separation and independence
- The system functionality is organized into separate layers
- Each layer only relies on the facilities and services offered by the layer immediately beneath it
- It supports the incremental development of systems.
- The architecture is also changeable and portable.

Name	Layered architecture
Description	Organizes the system into layers, with related functionality associated with each layer. A layer provides services to the layer above it, so the lowest level layers represent core services that are likely to be used throughout the system. See Figure 6.8.
Example	A layered model of a digital learning system to support learning of all subjects in schools (Figure 6.9).
When used	Used when building new facilities on top of existing systems; when the development is spread across several teams with each team responsibility for a layer of functionality; when there is a requirement for multilevel security.
Advantages	Allows replacement of entire layers as long as the interface is maintained. Redundant facilities (e.g., authentication) can be provided in each layer to increase the dependability of the system.
Disadvantages	In practice, providing a clean separation between layers is often difficult, and a high-level layer may have to interact directly with lower-level layers rather than through the layer immediately below it. Performance can be a problem because of multiple levels of interpretation of a service request as it is processed at each layer.

(Sommerville, 2010)

Example:
A generic layered architecture



(Sommerville, 2010)

Main bibliography

- Gamma, R., Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison Wesley, 1994
- https://www.amazon.com/Design-Patterns-Elements-Reusable-Object-Oriented/dp/0201633612
- Kim, G., *Human–Computer Interaction- Fundamentals and Practice*. CRC Press, 2015
- Sommerville, I., *Software Engineering*, 10th ed., Pearson Education, 2016

  https://www.amazon.com/Software-Engineering-10th-Ian-Sommerville/dp/0133943038

Universidade de Aveiro
Departamento de Electrónica,
Telecomunicações e Informática

Usability
Evaluation Methods

Beatriz Sousa Santos

- Usability is, according to ISO 9241-11:

"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"

- How to measure it??

(Cokton, 2013):

"Put simply, **usability evaluation assesses the extent to which an interactive system is easy and pleasant to use"**.

Things aren't this simple at all though, but …:

- Usability is a measurable property of all interactive digital technologies

- Evaluation methods determine if an interactive system or device is usable

- And the extent of its usability, through robust, and reliable metrics

- Evaluation methods and metrics are thoroughly documented …

http://www.interaction-design.org/encyclopedia/usability_evaluation.html
https://www.nngroup.com/articles/which-ux-research-methods/

Evaluation Methods

- **Analytical** (without users)
  - Heuristic Evaluation ✓
  - Cognitive Walkthrough ✓
  - Model based methods
  - Review methods

- **Empirical** (involving users)
  - Observation
  - Query
  - Controlled Experiments

  Observation, Query — usability tests ✓

(✓ - have used in Lab classes)

Heuristic Evaluation (Nielsen and Molich 1990)

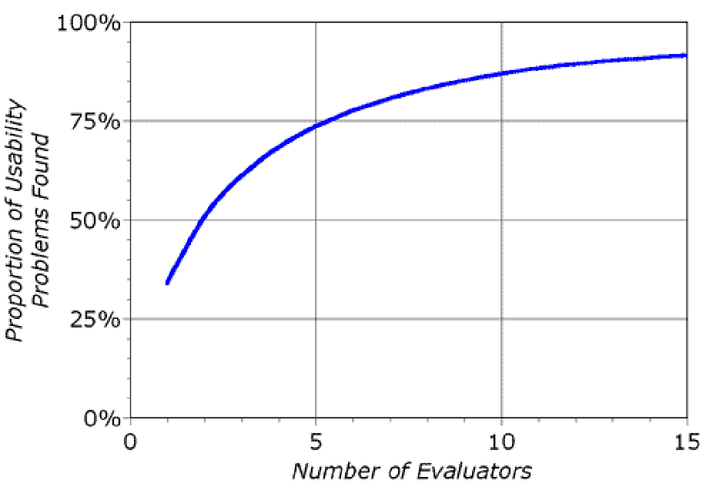
- A "**discount usability engineering method**" for quick, cheap, and easy evaluation of a UI design
- The most popular of the usability inspection methods
- It is a systematic inspection of a design for usability
- Meant to find the usability problems in the design so that they can be attended to as part of an iterative design process.
- Involves a small set of analysts judging the UI against a list of usability principles ("heuristics").

- Is difficult for a single individual to do; one person will never be able to find all the problems

- Involving multiple evaluators improves the effectiveness of the method significantly

- Nielsen generally recommends to use three to five evaluators

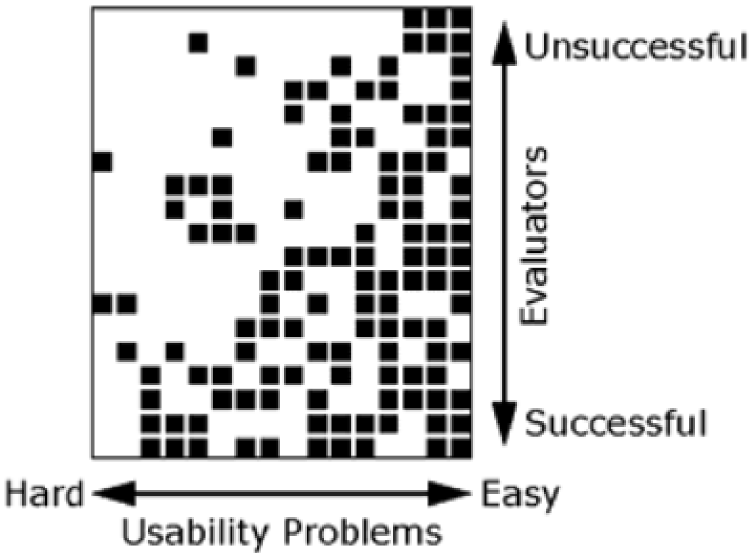
- not much gain by using larger numbers



https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/

Example:

- Heuristic evaluation of a banking system:
  - 19 evaluators
  - 16 usability problems



black square - problem found

white square – not found

http://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/

This suggests that in general 3 to 5 evaluators may be reasonable!

How to select the number of evaluators for a specific case?

- Consider the following criteria:
  - **Complexity** of the user interface
  - **Experience** of the evaluators
  - **Expected costs** /benefits
  - **Criticality** of the system (cost of user errors)
  - …

How to perform HE

- Should be performed by **several evaluators**
  (one person will never be able to find all the problems)

- **Evaluators should work independently**:
  - First get **a general idea** of the UI
  - Then perform a **detailed inspection using a set of heuristics**
  - List usability problems (**heuristics not followed and severity degree**)

- Findings of all evaluators should be **integrated in the same report**

The report should help the development team to prioritize problem fixing!

https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/

- Nielsen proposed **10 general usability heuristics**,

- yet **there are other sets, e.g. for**

  - different types of applications

    (web, mobile, visualization, … applications)

  - different types of users

    (for seniors, children…)

http://www.interaction-design.org/encyclopedia/usability_evaluation.html

How to perform Heuristic Evaluation

Each evaluator:

- First **make a general analysis** to get to know the UI
- Then, make a **systematic analysis** having in mind **the heuristics**
- **Take note of each potential problem**, the heuristic and the severity grade

Finally, compile all the potential problems and discuss with other evaluators

http://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation

Ten Nielsen’s heuristics

- **Visibility of system status**
- **Match between system and the 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, and recover from errors**
- **Help and documentation**

https://www.nngroup.com/articles/ten-usability-heuristics/

Example:
Heuristic #6 - Recognition rather than recall

Example of Usability Heuristic #6:
It's easier for most people to recognize the capitals of countries, instead of having to remember them. People are more likely to correctly answer the question Is Lisbon the capital of Portugal? *rather than* What's the capital of Portugal?

Tips

- Let people recognize information in the interface, rather than having to remember ("recall") it.
- Offer help in context, instead of giving users a long tutorial to memorize.
- Reduce the information that users have to remember.

Learn more:

https://www.nngroup.com/articles/recognition-and-recall/

https://www.nngroup.com/articles/ten-usability-heuristics/

Severity rating of usability problems

Is a combination of **three factors**:

- The **frequency** with which the problem occurs
- The **impact** of the problem if it occurs
- The **persistence** of the problem

The following 0 to 4 **rating scale** can be used to rate the severity of usability problems:

0 = I don't agree that this is a usability problem at all (to be used in the discussion)

1 = **Cosmetic problem**

2 = **Minor usability problem**

3 = **Major usability problem**

4 = **Usability catastrophe**

http://www.nngroup.com/articles/how-to-rate-the-severity-of-usability-problems/

- **Main advantages of heuristic evaluation**:
  - May produce **useful results with modest investment**
  - **Simple to apply** even by not very experienced evaluators
  - May be **used along the development process from early phases**

- **Main limitations**:
  - **Subjective** (partially overcome with more and more experienced evaluators)
  - **Tends to find many small problems** which may not be very important
  - **Can't find all usability problems**

-> **evaluation involving users is needed!**

Cognitive Walkthrough (Wharton, et al., 1992)

- Usability inspection method **not involving users** (analytical)
- Based on the fact that users usually prefer to learn a system by using it (e.g., instead of studying a manual)
- **Focused on assessing learnability** (i.e., how easy it is for new users to accomplish tasks with the system)
- **Applicable at early phases**, before any coding

How to perform a cognitive walkthrough

1- **Task analysis**: sequence of steps or actions required by a user to accomplish a task, and the system responses

2- Designers and developers **walkthrough as a group**, asking themselves a set of questions at each step

3- Data gathering during the walkthrough: **answering the questions** for each subtask usability problems are detected

4- **Report of potential issues**

5- **UI redesign to address the issues identified**

CW Four questions:

- **Will the user try to achieve the effect that the subtask has?**
  (Does the user understand this subtask is needed to reach the goal?)

- **Will the user notice that the correct action is available?**
  (E.g. is the button visible?)

- **Will the user understand that the wanted subtask can be achieved by the action?**
  (E.g. the button is visible but the user doesn’t understand the text and will not click on it)

- **Does the user get feedback?**
  Will the user know that they have done the right thing?

Common issues

- The evaluator may not know the optimal way to perform the task; the method involves the optimal sequence of actions
- Involves an extensive analysis and documentation and often too many potential issues are detected, resulting very **time consuming**

Thus:

Lighter variants of Cognitive Walkthrough were proposed to make it **more applicable** in S/W development companies

Streamlined Cognitive Walkthrough (Spencer, 2000)

- Only two questions:

  - Will the user know what to do at this step?

  comprises the 3 first questions of CW

  - If the user does the right thing, will they know that they did the right thing, and are making progress towards their goal?

- And a set of rules to streamlining the walkthrough and trade-off granularity for coverage

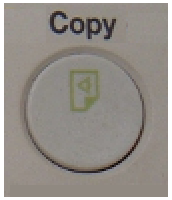
According to Spencer the method can be applied successfully if the usability specialist:

- takes care to prepare the team for the walkthrough,
- avoids design discussions during the walkthrough,
- explicitly neutralizes defensiveness among team members,
- streamlines the procedure by collapsing the first three questions into one question,
- and captures data selectively

Example: Evaluation of a desktop photocopier UI

- Machine UI:
  - numeric keypad,
  - "Copy" button,
  - push button on the back to turn on the power

The machine automatically turns itself off after 5 min inactivity



- Task: copy a single page
- User: any office worker
- Actions needed: turn on the power,
  put the original on the machine,
  press the "Copy" button

http://hcibib.org/tcuid/chap-4.html#4-1

- Story for action number one:
  “the user wants to make a copy and knows that the machine
  has to be turned on. So she pushes the power button. Then she
  goes on to the next action”

Not convincing!

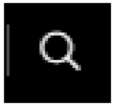
- why shouldn't the user assume that the machine is already on? That is often the case

- Will the user figure out that the machine is off, and find the power switch?

  etc. etc.

Another example: Look for a person’s phone number and email address at the University of Aveiro Web site
User: any student from the University

Task analysis:

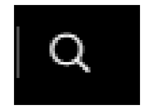
- find the icon  (search);
- input part of the person’s name and search in “Pessoas“
- get the phone number

But the defined user profile (any student from the University) includes foreign students, thus a previous action is needed:

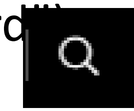
- select the English version 

For each action we need to ask the two questions and put ourselves in the shoes of the user!

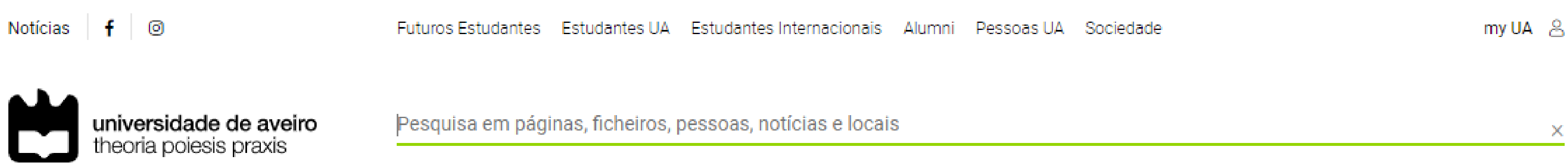
Previous action for foreign students: Select the English version seems easy (it is a “standard” way to do it in sites)

First action in the Portuguese version: find the icon 

Q1 - Will the user know what to do at this step?

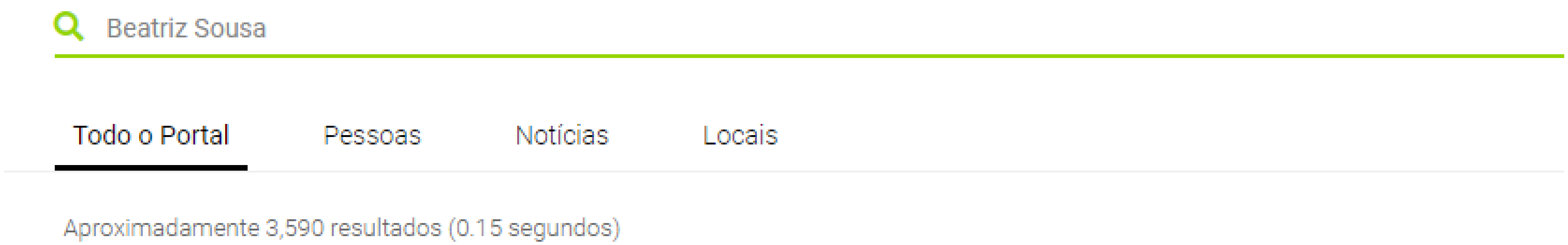
Even without tooltip the correct icon seems recognizable (it is “standard”) 

Q2 - If the user does the right thing (selects the icon), will they know that they did the right thing, and are making progress towards their goal?



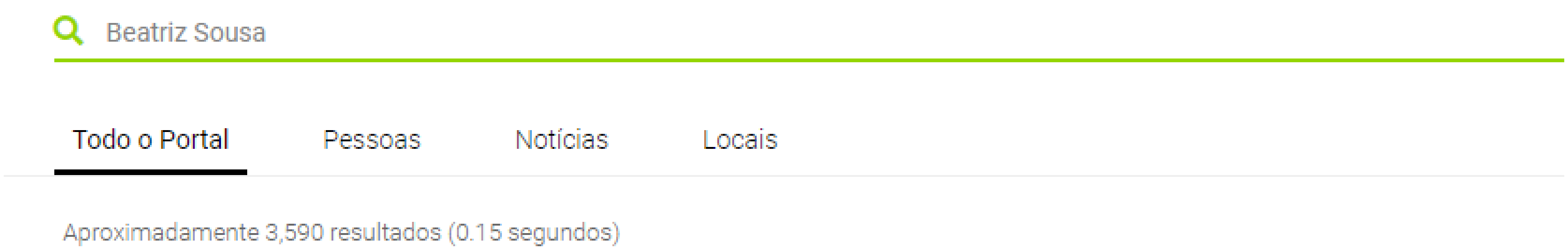
Probably yes; while it may not look a search bar, it is adequately labeled (Pesquisa em páginas, ...)

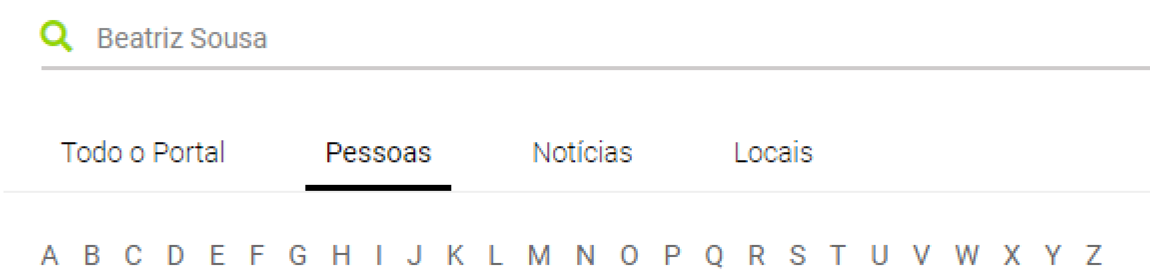
Second action: input part of the person's name and search in "Pessoas"



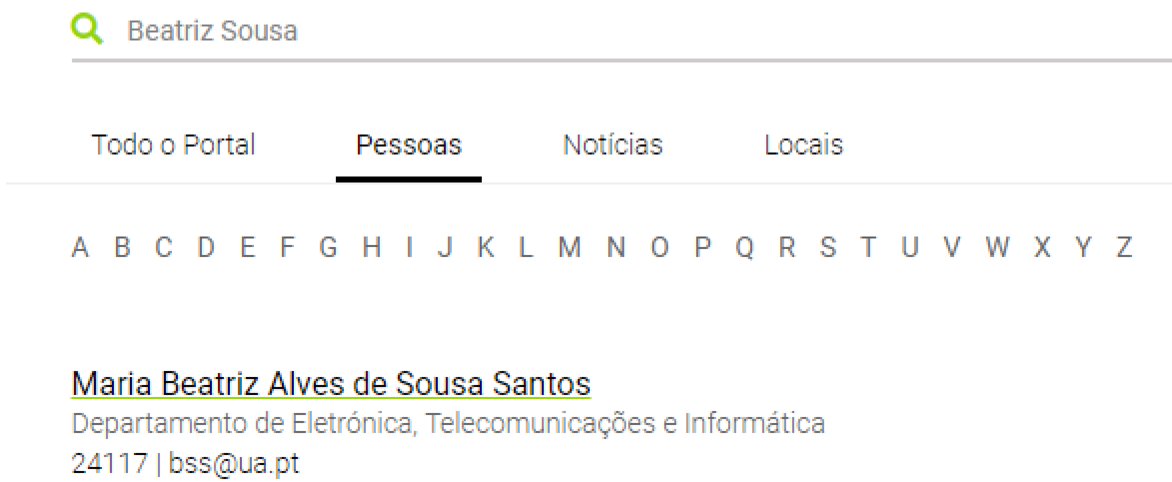
Q1 - Will the user know what to do at this step?

Probably yes; it is easy to recognize that s/he should input the person's name and select "Pessoas"





Q2 - If the user does the right thing (inputs the name and selects “Pessoas”), will they know that they did the right thing, and are making progress towards their goal?



Probably yes; however, some users may not recognize 24117 as a phone number (it only has 5 digits, as it is internal, and not 9 as possibly expected)

In conclusion:

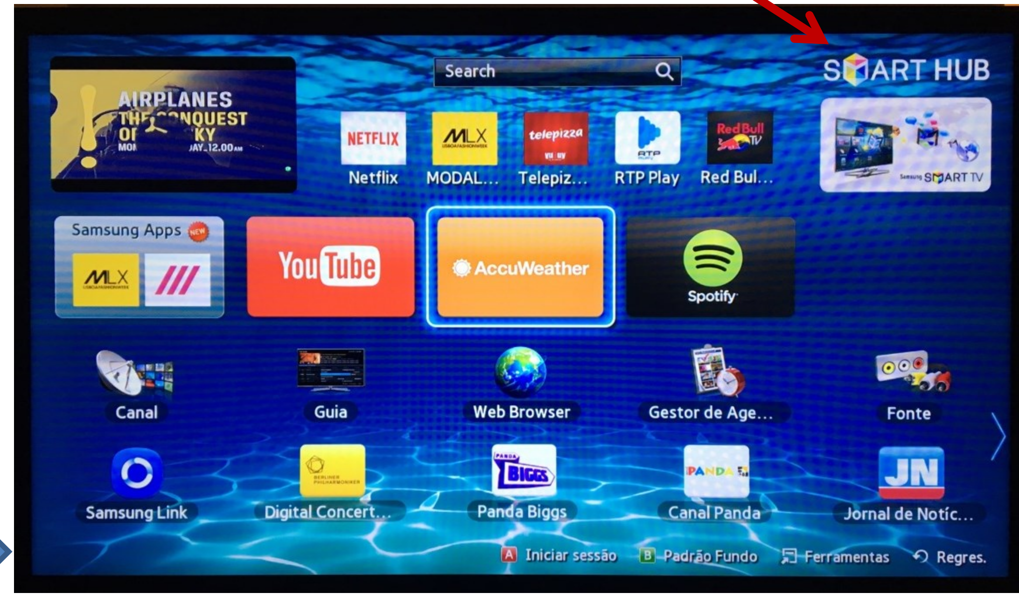
- it seems easy for the target users to reach the phone number and email address;
- however, the phone number may be not recognized as such

Another example: Smart TV

How to access the Internet?

(before reading the manual?)

(we see the symbol at the screen only after pressing it on the control!)



Practice the Streamlined Cognitive Walkthrough:

Analyzing interactive systems/applications that should be very intuitive (e.g. consumer electronics):

- Turn on and off the video projector in your Lab using the remote control or directly on the projector
  user: any student from the University

- Change the Channel using the box of your TV service (not the remote control)
  user: anyone having a TV box

Limitations of Analytical Methods

– Are subjective
– Involve several usability experts
– Cannot find all usability problems

Thus, empirical methods (involving users) are needed !!

observation
query

Usability test (engineering approach)

controlled experiments (scientific approach)

Evaluation Methods

- Analytical (without users)
  - Heuristic Evaluation ✓
  - Cognitive Walkthrough ✓
  - Model based methods
  - Review methods

- Empirical (involving users)
  - Observation
  - Query
  - Controlled Experiments

usability tests ✓

(✓ - have used in Lab classes
  - have seen in papers)

Ethics in applying empirical methods

Involving users implies **specific cautions**:

- Asking for explicit consent
- Confidentiality
- Security (avoid any risk)
- Freedom (users may give up at any time)
- Limit stress

It's the system that is under evaluation not the user!

https://www.nngroup.com/articles/user-research-ethics/

Empirical evaluation styles

These methods may be performed:

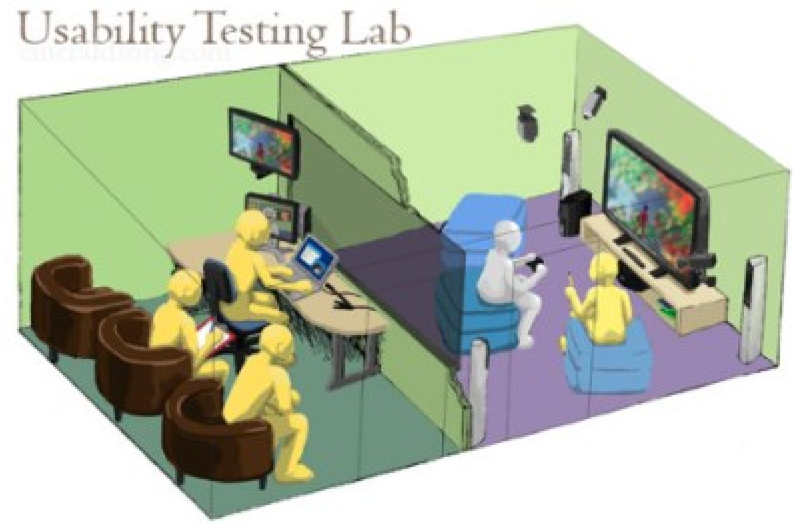
- In the laboratory (more controlled)
- In the field (more realistic)

They produce complementary information;

if possible use both!

https://www.nngroup.com/articles/field-studies/

Observation



Has many variants from very simple to very complex and expensive:

- Direct: observer takes notes
- Undirect: through audio/ vídeo – more complex and time consuming
- Think Aloud: users are asked to explain what they are doing
- Logging: users activity is logged by the system
- Combinations of the previous, etc

https://www.usabilitybok.org/usability-testing-methods

Think aloud Observation

Participants are asked to use the system while continuously thinking out loud (verbalizing their thoughts as they use the system)

Benefits:

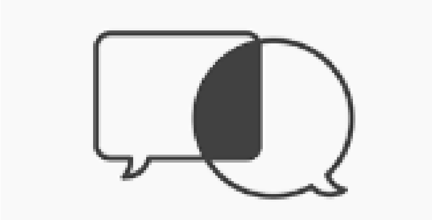
- Inexpensive
- Flexible
- Easy to learn and apply

Limitations:

- Unnatural situation
- Filtered statements
- Changing user behavior

https://www.usabilitybok.org/usability-testing-methods
https://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool/

Query



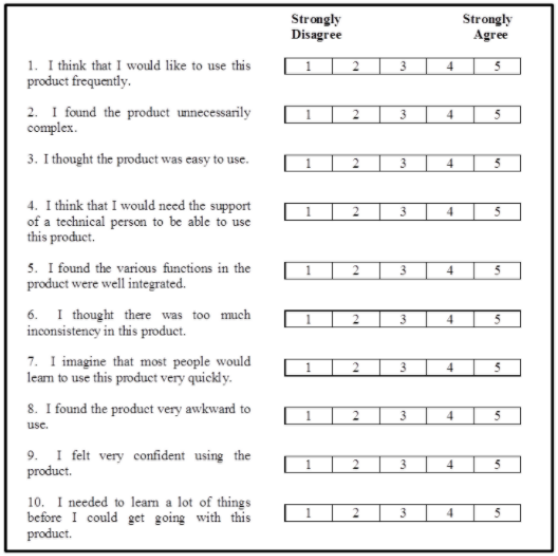
- Two main variants:
  - Questionnaire
    (reach more people; less flexible)
  - Interview
- **Should always be carefully prepared and tested**
- Collected data should be carefully analyzed

https://www.interaction-design.org/literature/article/useful-survey-questions-for-user-feedback-surveys
https://www.interaction-design.org/literature/article/how-to-conduct-user-interviews

Well-known usability questionnaires

- **System Usability Scale** (SUS)

- Questionnaire for User Interface Satisfaction (QUIS)



	Strongly Disagree				Strongly Agree
1. I think that I would like to use this product frequently.	1	2	3	4	5
2. I found the product unnecessarily complex.	1	2	3	4	5
3. I thought the product was easy to use.	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this product.	1	2	3	4	5
5. I found the various functions in the product were well integrated.	1	2	3	4	5
6. I thought there was too much inconsistency in this product.	1	2	3	4	5
7. I imagine that most people would learn to use this product very quickly.	1	2	3	4	5
8. I found the product very awkward to use.	1	2	3	4	5
9. I felt very confident using the product.	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this product.	1	2	3	4	5

- SUS provides a "quick and dirty", reliable tool for measuring the usability
- It includes 10 questions with five response options

- QUIS is a measurement tool designed to assess a computer user's subjective satisfaction with the UI
- It is designed to be configured according to the needs of each UI analysis by including only the sections that are of interest to the user
- It includes questions with ten response options

- Both questionnaires should be completed following use of the UI in question

System Usability Scale (SUS)

- Provides a "**quick and dirty**", reliable tool for measuring the usability
- It includes 10 questions with five response options
- It allows to **evaluate a wide variety of products and services** (H/W, S/W, mobile devices, websites and applications)
- Has become an **industry standard**, with references in over 1300 publications

Benefits of using a SUS

- Is a **very easy** scale to administer to participants
- Can be used on **small sample sizes** with reliable results
- **Is valid** – it can differentiate between usable and unusable systems

https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html

SUS Questions

- I think that I would like to use this system frequently.
- I found the system unnecessarily complex.
- I thought the system was easy to use.
- I think that I would need the support of a technical person to be able to use this system.
- I found the various functions in this system were well integrated.
- I thought there was too much inconsistency in this system.
- I would imagine that most people would learn to use this system very quickly.
- I found the system very cumbersome to use.
- I felt very confident using the system.
- I needed to learn a lot of things before I could get going with this system.

https://www.usability.gov/how-to-and-tools/resources/templates/system-usability-scale-sus.html

Scoring SUS

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	1	2	3	4	5
2. I found the system unnecessarily complex	1	2	3	4	5
3. I thought the system was easy to use	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	1	2	3	4	5

…

Let R(n) be the answer to Question n:

$$SUS = \left(\sum_{n=1}^{5} R(n)-1 + 5- R(n*2) \right) * 2.5$$

0… 100; SUS > 68 would be considered above average

QUIS - Questionnaire for User Interface Satisfaction

- The QUIS contains:

  - a demographic questionnaire,
  - a measure of overall system satisfaction,
  - a measure of specific UI factors (e.g. screen visibility, terminology and system information, learning factors, and system capabilities)

- QUIS has pen and paper and PC software versions for administration
- Uses a 10-point scale to rate 21 items relating to the system's usability
- These ratings produce data for the overall reaction to a system's usability on 6 factors.
- It is easy to use and analyse.

https://ext.eurocontrol.int/ehp/?q=node/1611

Example questions of QUIS

OVERALL REACTIONS TO THE SOFTWARE

terrible 0 1 2 3 4 5 6 7 8 9 wonderful

difficult 0 1 2 3 4 5 6 7 8 9 easy

frustrating 0 1 2 3 4 5 6 7 8 9 satisfying

inadequate power 0 1 2 3 4 5 6 7 8 9 adequate power

dull 0 1 2 3 4 5 6 7 8 9 stimulating

rigid 0 1 2 3 4 5 6 7 8 9 flexible

SCREEN

Characters on the computer screen

hard to read 0 1 2 3 4 5 6 7 8 9 easy to read

Highlighting on the screen simplifies task

not at all 0 1 2 3 4 5 6 7 8 9 very much

Organization of information on screen

confusing 0 1 2 3 4 5 6 7 8 9 very clear

Sequence of screens

confusing 0 1 2 3 4 5 6 7 8 9 very clear

USABILITY AND USER INTERFACE

Use of colors and sounds

poor 0 1 2 3 4 5 6 7 8 9 good

System feedback

poor 0 1 2 3 4 5 6 7 8 9 good

System response to errors

awkward 0 1 2 3 4 5 6 7 8 9 gracious

System messages and reports

poor 0 1 2 3 4 5 6 7 8 9 good

System clutter and UI “noise”

poor 0 1 2 3 4 5 6 7 8 9 good

Usability tests

- Involve **observation and query**

- Main aspects:
  - Participants
  - Tasks
  - Test facilities and systems
  - Protocol
  - Usability measures
  - Data analysis

- May have a **complex logistics**

- Standard: **Common Industry Format (CIF)** for usability test reports

https://www.usability.gov/how-to-and-tools/methods/planning-usability-testing.html
https://www.interaction-design.org/literature/topics/usability-testing

Participants

- The total number of participants to be tested
  (a valid statistical analysis implies a sufficient number of subjects)
- Segmentation of user groups tested, if more than one
- Key characteristics and capabilities of user group
  (user profile: age, gender, profession, computing experience, product experience, etc.)
- How to select participants
- Differences between the participant sample and the user population
  (e.g. actual users might have training whereas test subjects were untrained)

Tasks

- The task scenarios for testing
- Why these tasks were selected

  (e.g. the most frequent tasks, the most troublesome tasks)
- The source of these tasks

  (e.g. observation of users using similar products, product specifications)
- Any task data given to the participants
- Completion or performance criteria established for each task

  (e.g. n. of clicks < N, time limit)

Test Facilities and equipment

- The setting and type of space in which the evaluation will be done

  (e.g. usability lab, cubicle office, meeting room, home office, home family room, manufacturing floor, etc.)

- Any relevant features or circumstances that can affect the results

  (e.g. video and audio recording equipment, one-way mirrors, or automatic data collection equipment)

- Participant’s computing environment

  (e.g. computer configuration, including model, OS version, required libraries or settings, browser name and version; relevant plug-in, etc.)

- Display and input devices characteristics

- Any questionnaires to be used

Protocol

- Procedure: the logical design of the test

- Participant general instructions and task instructions

- The usability measures to be used:
  a) for **effectiveness** (completeness rate, errors, assists)
  b) for **efficiency** (times)
  c) for **satisfaction**

Common Industry Format (CIF) for usability test reports ISO/IEC 25062:2006

- Specifies the format for reporting the results of a **summative** evaluation
- The most common type of usability evaluation is **formative**, (i.e. designed to identify problems that can be fixed)
- A summative evaluation produces usability metrics that describe how usable a product is when used in a particular context of use
- The CIF report format and metrics are consistent with the ISO 9241-11

https://www.iso.org/standard/43046.html

https://www.userfocus.co.uk/articles/cif.html

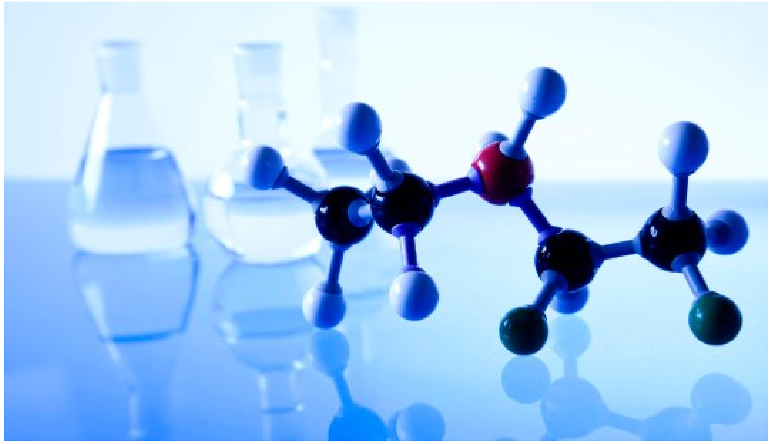
Software engineering -- Software product Quality Requirements and Evaluation (SQuaRE) -- **Common Industry Format (CIF) for usability test reports**

This standard was last reviewed and confirmed in 2019.

The format includes the following elements:

- the description of the product,
- the goals of the test,
- the test participants,
- the tasks the users were asked to perform,
- the experimental design of the test,
- the method or process by which the test was conducted,
- the usability measures and data collection methods, and
- the numerical results.

Controlled experiments



- The **“work horse” of experimental science** ...

- Important issues to consider:
  - Hypothesis
  - Variables (input or independent; output or dependent, secondary)
  - Experimental design (within groups; between groups)
  - Protocol
  - Participants (number, profile)
  - Statistics

https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/experimental-methods-in-human-computer-interaction

Controlled experiment

- Define **hypotheses**
- Define input (independent), output (dependent) and secondary **variables**
- Define **experimental design** (within-groups / between groups)
- Define **protocol**
- Select the **participants**
- Prepare all the **documentation**:
  - list of tasks and perceived difficulty
  - final questionnaire

  To the user

  - list of tasks for the experimenter to take notes

  To the experimenter
- Run a **pilot test**
- Take care of the **logistics** … and after the experiment **analyze data**

Controlled experiment

Variables:

- **Independent or input variables** – what is controlled
  (e.g. interaction method)
- **Dependent or output variables** – what is measured
  (e.g. times and errors)
- **Secondary variables –** not controlled but may influence the result
  (e.g. age, previous experience)

Controlled experiment

Experimental design:

- **Within-groups or within-subjects** – all participants use the same conditions (usually in randomized order to avoid bias)

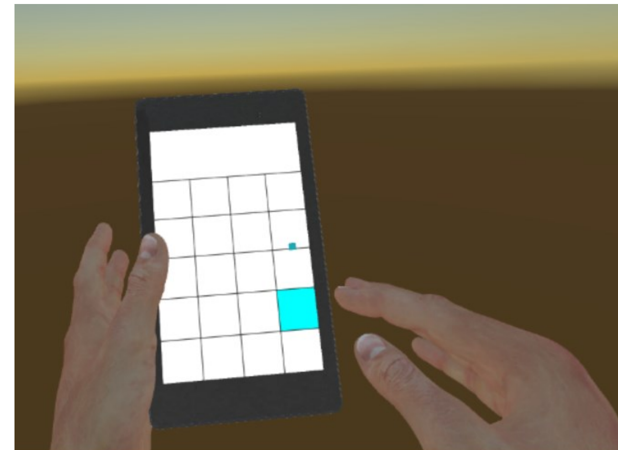
  advantages – a smaller number of participants
  same profile
  disadvantages – prone to fatigue or learning bias

- **Between-groups or between-subjects** – each participant uses only one condition

  advantages – less fatigue or learning bias
  disadvantages – higher number of participants needed
  different participants' profile

Examples of Controlled Experiments performed @ HCI - DETI

- Study of the Effect of Hand-Avatar in a Selection Task using a Tablet as Input Device in an Immersive Virtual Environment



- Comparing two alternative versions of Meo Go

"Effect of Hand-Avatar in a Selection Task using a Tablet as Input Device in an Immersive Virtual Environment"

L. Afonso, P. Dias, C. Ferreira, B. Sousa Santos
IEEE 3D UI, Los Angeles, March 2017

- Research question: How does the virtual representation of the user's hands influence the performance on a button selection task performed in a tablet-based interaction within an immersive virtual environment?

- Method: Controlled experiment

- 55 participants used three conditions:
- no-hand avatar,
- realistic avatar,
- translucent avatar.

- Participants were faster but made slightly more errors with no-avatar

- Considered easier to perform the task with the translucent avatar

Experimental Design

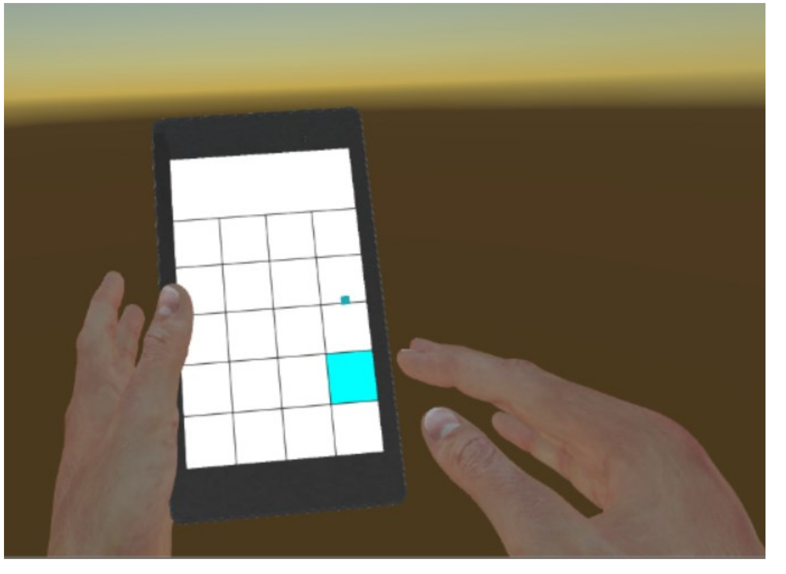
Null Hypothesis: usability is independent of the hands representation

Independent (input) variable (with 3 levels): representation of the hands

Dependent (output) variable: usability (performance + satisfaction)

Within-groups: all participants used all experimental conditions (in different sequences to avoid learning or fatigue bias)

Task: selecting as fast as possible a highlighted button from a group of twenty buttons (repeated measures)

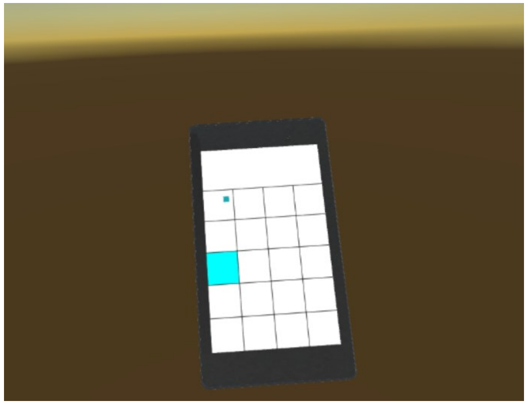


Experimental Conditions

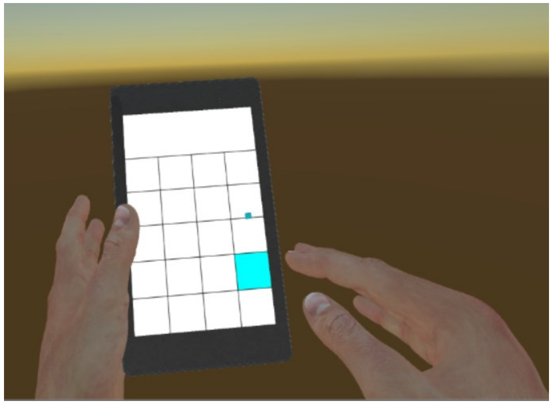
1- **No avatar**: the user only sees the virtual tablet;

2- **Realistic avatar**: a realistic representation of the hands movement is shown

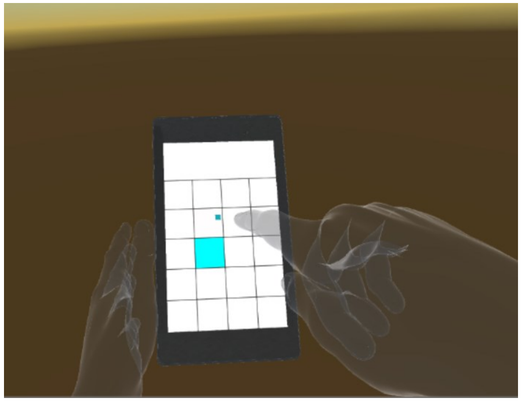
3- **Translucent avatar**: a translucent hand model is used (to alleviate occlusion)



No-avatar



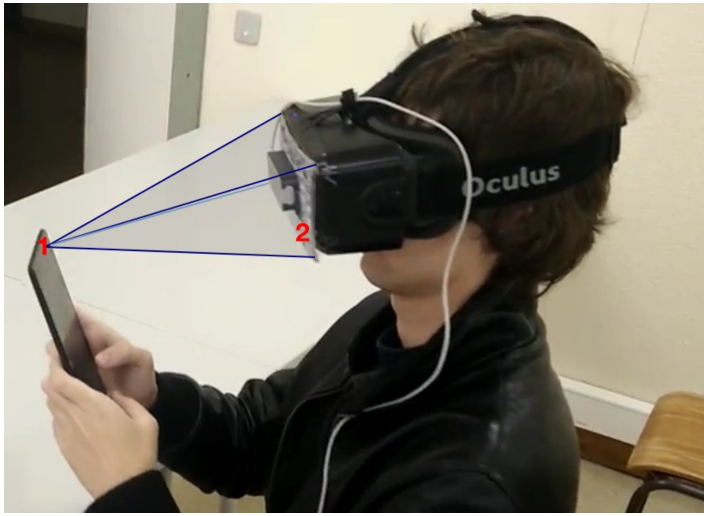
Realistic avatar

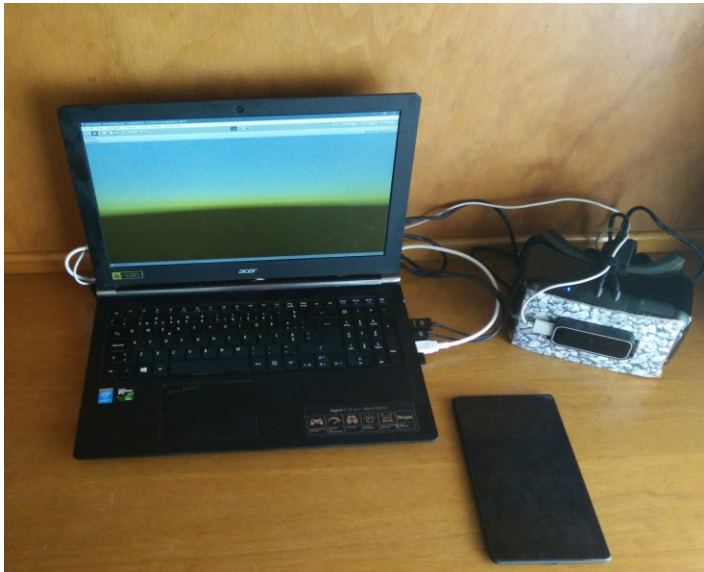


Translucent-avatar

Experimental Set-up

- Laptop running the main application (in Unity)
- HMD (Oculus Rift DK2) providing head tracking
- Tablet (Google Nexus 7) as input device running the controller application (in Unity)
- Leap Motion (mounted on the HMD) to track the user's hands
- Tablet camera tracking the position and orientation of an AR marker on the HMD to map tablet position in the virtual world (using Vuforia)





Main Results

Selection time:

Participants completed the button selections in average **faster with no-avatar (statistically significant)**

Selection errors:

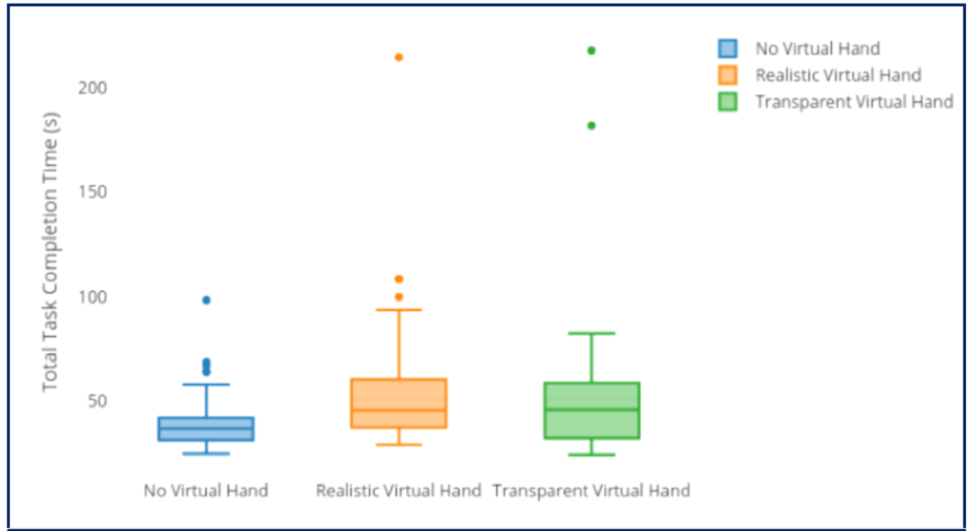
Participants made slightly less errors with avatar - realistic or translucent- **(statistically significant)**

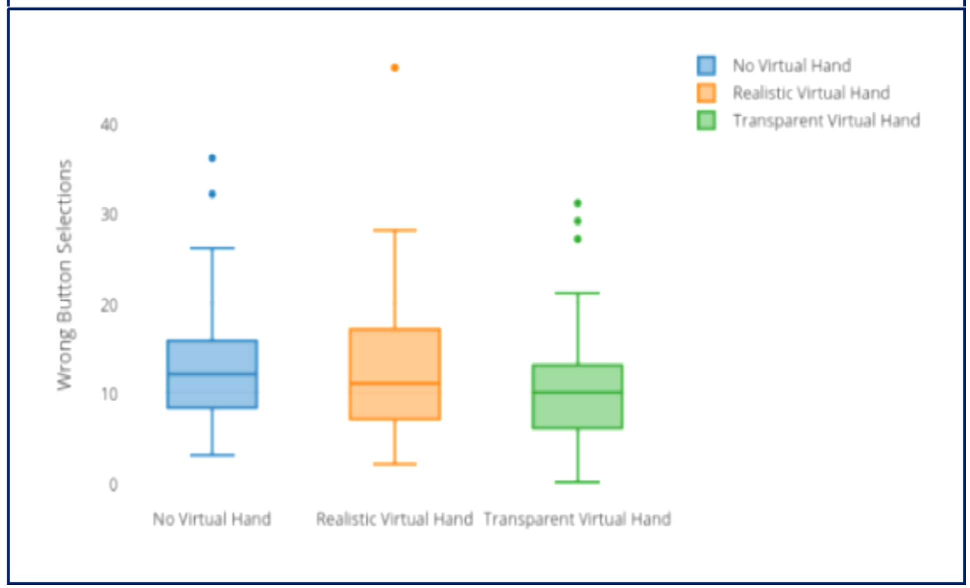
Participants' opinion:

The translucent avatar:

- was more often preferred

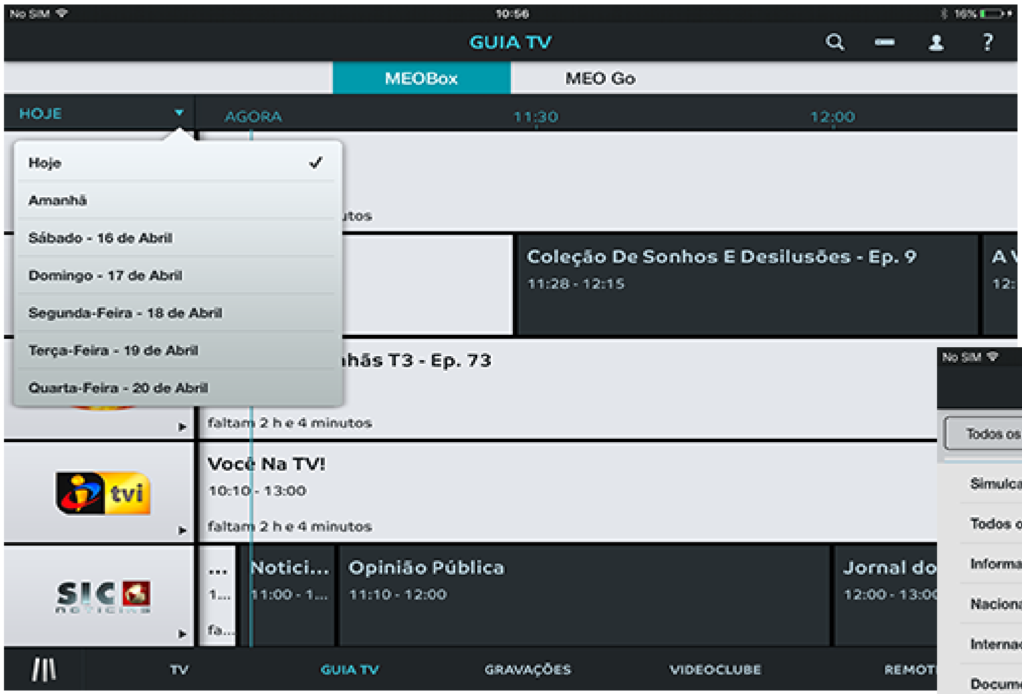
- was considered as better than the realistic avatar (statistically significant)





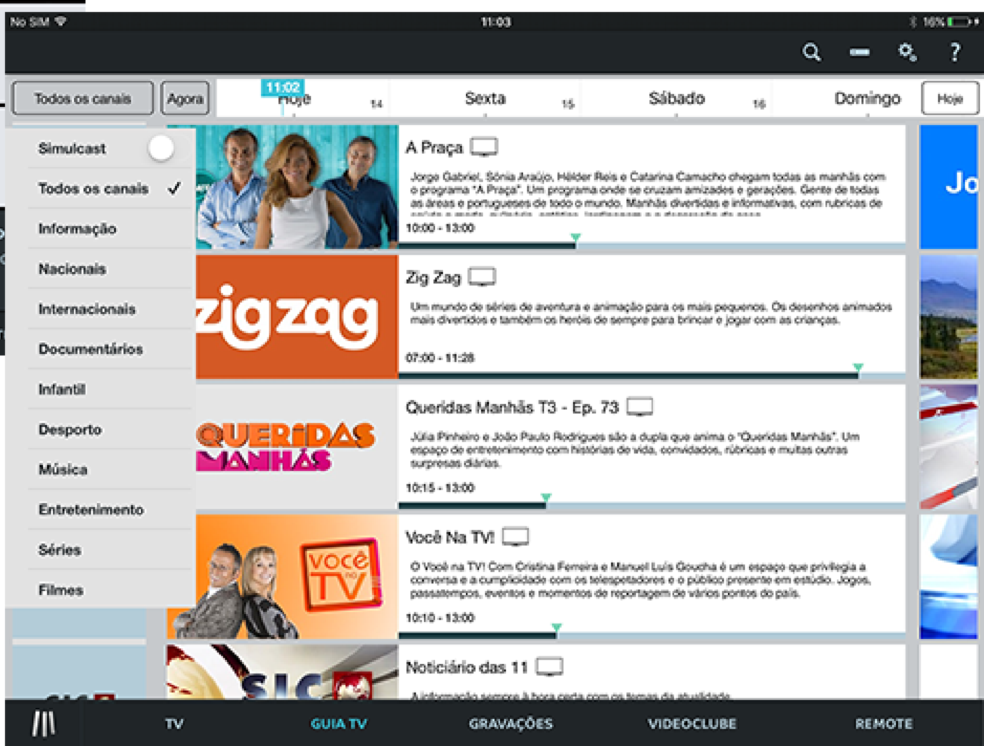
Comparing two versions of Meo Go

Altisse Labs User - UX Group
DETI – HCI course
April/2016



Current version

New version



Controlled experiment: comparing two versions of Meo Go: Current version *vs* Version to be deployed

- Null Hypothesis: both are equally usable
- Input (independent) variable: version
- Output (dependent) variables: performance, opinion and satisfaction
- Secondary variable: participant profile
- Participants: 66 volunteer HCI -2016 students (12 female)
- Experimental design: between groups (one version per participant)
- Exploratory Data Analysis and non-parametric tests (ordinal variables)

Summary of results

- Participants were satisfied with both versions
  no significant difference between versions was observed

- Both versions got a good classification concerning application usage
  no significant difference between versions was observed

- Both versions got a good classification concerning difficulty
  no significant difference between versions was observed

- Both versions got a good classification concerning specific aspects of the application
  two aspects improved significantly in version 2

- **No significant difference** between the two versions is observed concerning **total task time**

Study main limitations

- Participants' profile - students are not representative of all target users
- Tasks – are simple and performed in a controlled context - limit ecological validity
- Think aloud - total task time must be considered with caution

However:

HCI students are aware of usability issues and provided valuable feedback

- This data analysis was complemented by the analysis of comments and suggestions

- **The combined analysis provided more insight concerning usability issues**

- Allowed **identify easy to implement improvements** in the UI with potential **positive impact on the UX** of the final version

Bibliography for Usability evaluation – Books and links

- **Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale, *Human-Computer Interaction*, 3rd edition, Prentice Hall, 2004**

- Jackob Nielsen, Usability Engineering, Morgan Kaufmann, 1993

- **Peter Mitchell, *A Step-by-step Guide to Usability Testing*, iUniverse, 2007**

- Gilbert Cockton, Usability Evaluation. In: Soegaard, Mads and Dam, Rikke Friis (eds.), The Encyclopedia of Human-Computer Interaction, 2nd Ed, 2013, Aarhus, Denmark: The Interaction Design Foundation. (2018)

http://www.interaction-design.org/encyclopedia/usability_evaluation.html

- Norman/ Nielsen Group site - http://www.nngroup.com/articles/
- Usability.gov site - https://www.usability.gov/index.html
- User focus site - https://www.userfocus.co.uk/articles/

Epilogue of this course

- We will have the Human in the loop for long in most situations …
- And even when/if they are no longer in the loop

Technology shall serve the Human

(and not the other way around…)

Preparing the Exam

- Study the Slides
- Study the mandatory readings
- Answer the Exam preparation questions available in Moodle

Mandatory Readings for the Exam

- Slides available at Moodle and at the course web page:

  http://sweet.ua.pt/bss/disciplinas/IHC-ECT/IHC-ECT-home.htm

- Alan Dix t al., *Human-Computer Interaction*, 3rd ed., Prentice Hall, 2004 (at the Library)
  - Chapters: 1 to 4 (for the topics addressed in the slides)
  - Chapter 9 (for the topics addressed in the slides)
  - Chapters 12, 14 and 16 (for the topics addressed in the slides)

- Ian Sommerville, *Software Engineering*, 9. Ed., Addison Wesley, 2009
  (Chapter 29, available at Moodle)

Universidade de Aveiro
Departamento de Electrónica,
Telecomunicações e Informática

Usability Principles and Paradigms

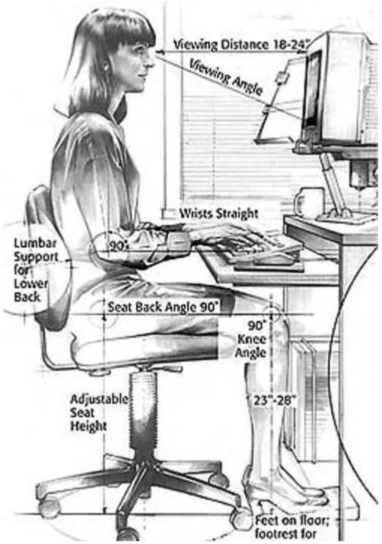
(Donald Norman, Design of everyday things)

Beatriz Sousa Santos, 2021

- Introduction
- History
- Usability and standards
- Principles
- Paradigms

- During and after the World War II the following disciplines emerged:

Ergonomics – physical aspects **Human factors** – also cognitive aspects



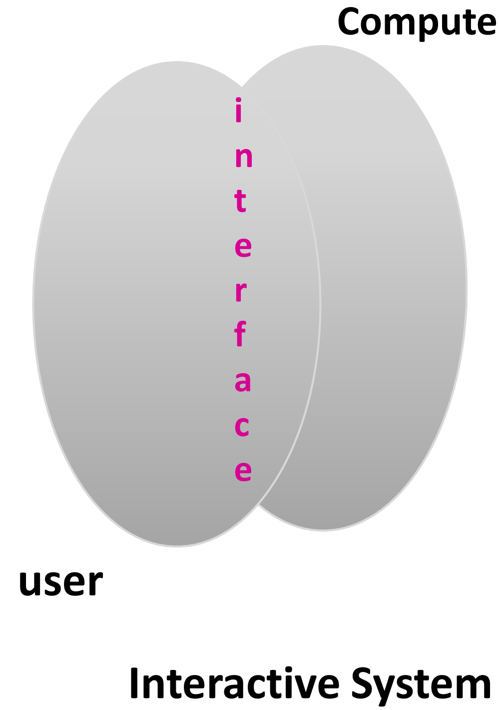
Ergonomics and Human Factors

- Interaction emerged as new independent field within Computing in the 80s, mainly due to:
  - Lower price of technology
  - Technology migration
  - Need to increase users' productivity

Man-Machine Interaction → **Human-Computer Interaction**
(nighties)

- It expanded rapidly
- It is currently an interdisciplinary field
- Human Centered Computing is an ACM scientific area within Computing (also at the University of Aveiro)

Interactive systems design



- Interactive systems include a “module” which we don´t control:

  The user, who:

  - is very complex
  - not well known
  - we cannot control

(and users may be very different)

This makes design difficult

- User Interface (UI) is the means by which the user and a computer system interact
- To the user **“the interface is the system”**
- The user interface design involves a considerable effort

Interactive system design – Human-Centered design

- Involves knowing:

  Usability principles (independent from technology)

  Usability paradigms (more technology dependent)

  \+ **Evaluation**

  \+ **Methods**

- We must know the **success examples** (usability paradigms)
- Understand **why they work** (usability principles)
- Use the **adequate methods** (user-centered approach)
- And **test**, re-design,

  test, redesign

  ...

  until we attain the usability goals

- **Usability** is, according to **ISO 9241-11**:

“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**”

- Effectiveness + efficiency -> **ease of use**

- **Satisfaction** is also very important

- **User Experience (UX)** is:

" person's perceptions and responses resulting from the use and/or anticipated use of a product, system or service"

- UX includes all the users' emotions, preferences, perceptions, physical and psychological responses, … that occur before, during and after use

- UX is **broader than usability**, it includes other aspects…

- Usability criteria can be used to assess aspects of user experience.

https://www.iso.org/obp/ui/#iso:std:iso:9241:-210:ed-1:v1:en

Usability

- Is directly related to the system capacity to allow users **attaining their goals through its usage**

- Fundamental aspects:

  - **easy to use** (fast and with few errors) (efficiency, efficacy-> performance)

  - **satisfaction**

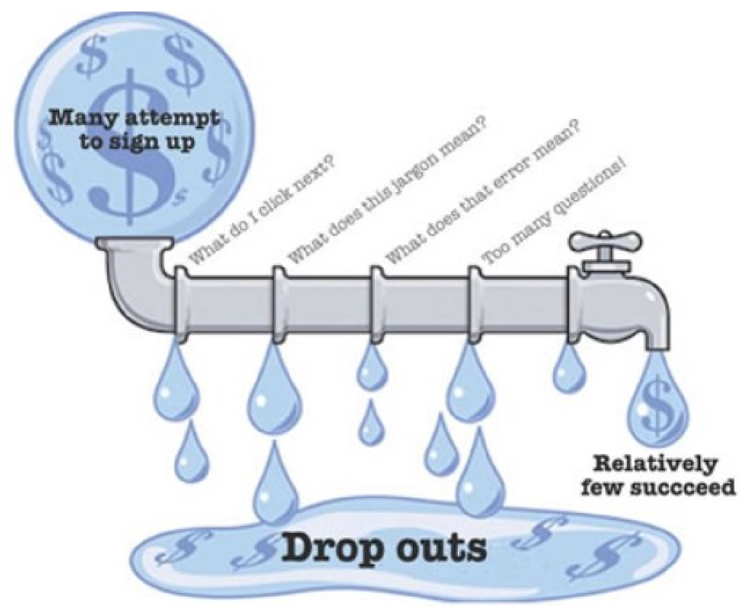
  Is defined in a **context of use**: is a system property of allowing specific users to perform specific tasks efficiently with efficacy and satisfaction

- Easy to learn and remember (learnability, memorability) is a related aspect

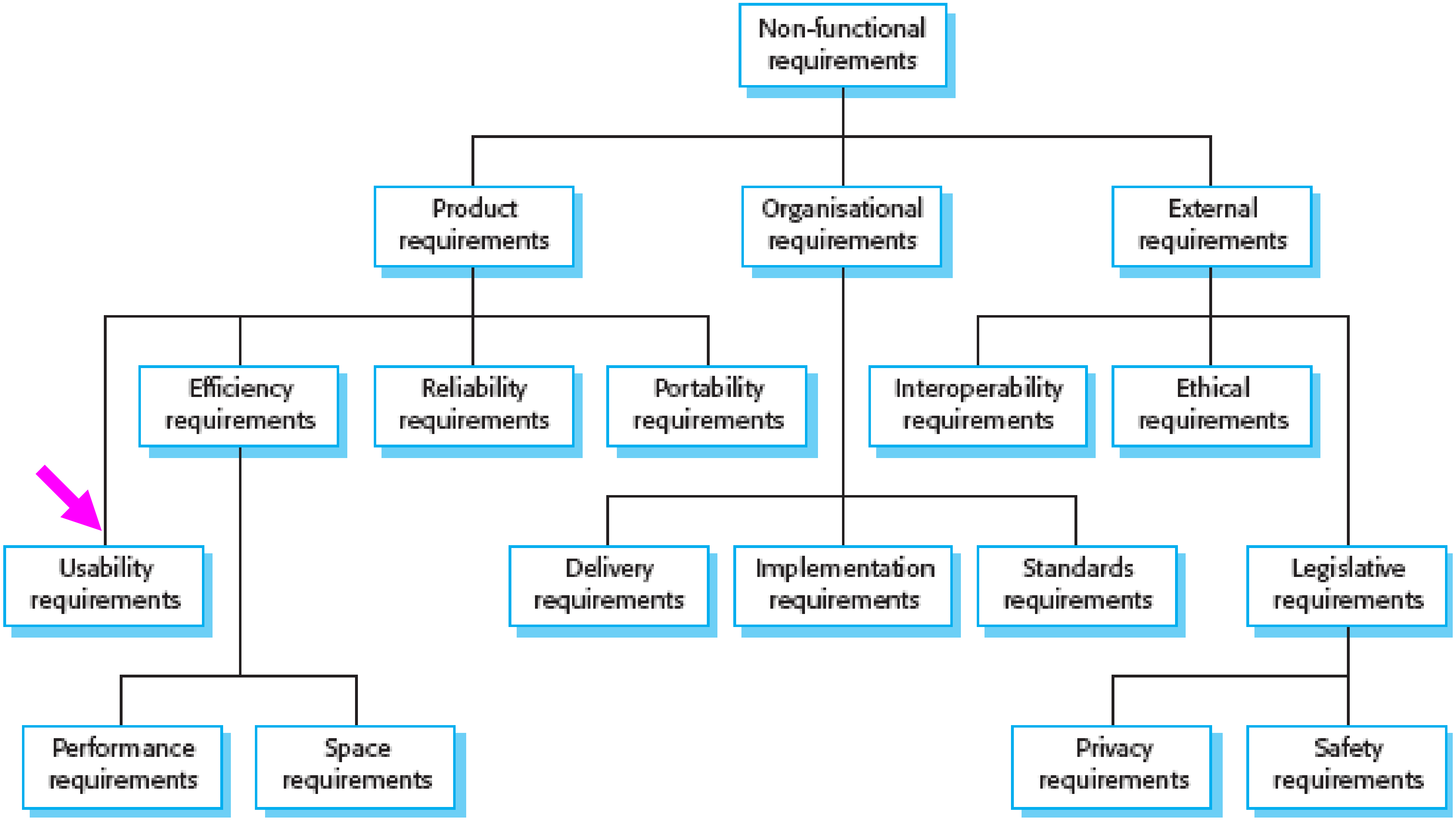
- Main usability benefits:

  - Higher user performance and satisfaction
  - Lower development costs
  - Lower support costs …

  – **Higher profits for everyone!**

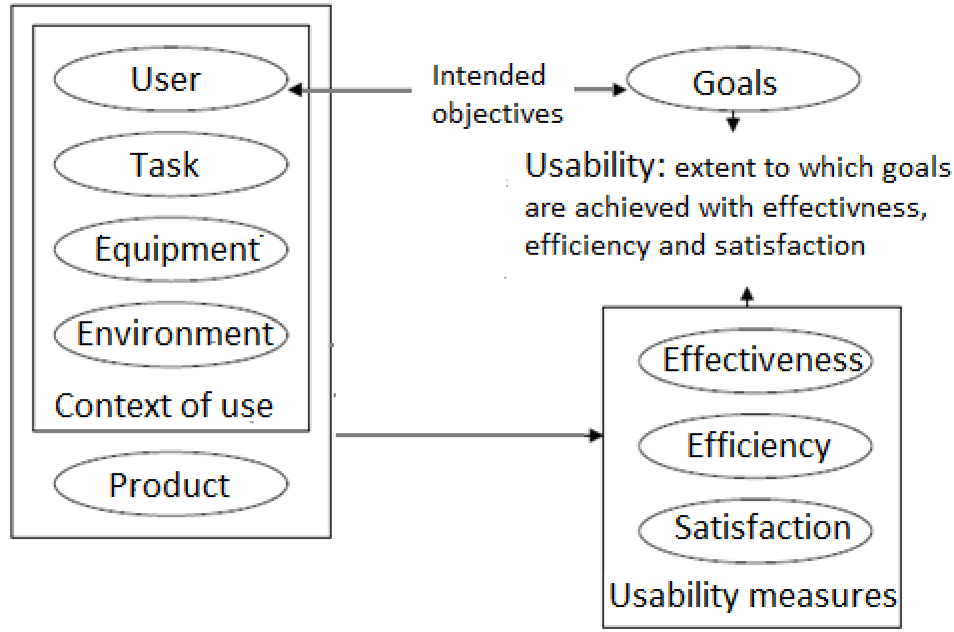


Usability is a non-functional requirement



Ian Sommerville, *Software Engineering*, 9th ed., Addison Wesley, 2011

Usability standards



- ISO 9241-11 (1998)

Ergonomic requirements for office work with visual display terminals Part 11 : Guidance on usability

Explains how to identify the information needed to specify or evaluate usability in terms of measures of:
- performance
- satisfaction

- **ISO 13407 -> ISO 9241-210 (2010)**

Human-centred design processes for interactive systems

- And others related ISO 13.180 **Ergonomics**

- **ISO 13407 adresses:**

- **… Four Principles of Human-Centered Design:**

  - active involvement of users
  - appropriate allocation of function to system and to user
  - iteration of design solutions
  - multi-disciplinary design

- **... and Four Human-Centered Design Activities:**

  - understand and specify the context of use
  - specify user and organizational requirements
  - produce more than one candidate design solution
  - evaluate designs against requirements

- **ISO 9241-112:2017**
- **Ergonomics of human-system interaction — Part 112:**
- **Principles for the presentation of information**

- **…** establishes ergonomic design principles for interactive systems related to the software-controlled presentation of information by user interfaces.

- It applies to the three main modalities
  visual, auditory, tactile/haptic

- These principles apply to the perception and understanding of presented information
- are applicable in analysis, design, and evaluation of interactive systems
- **…**

https://www.iso.org/standard/64840.html

Paradigms

- Inspirations for a conceptual model
- General approach adopted by a community for carrying out research
  - Shared assumptions, concepts, values, and practices
  - For example, desktop, ubiquitous computing, in the wild

Some usability paradigms
(along the history of computing)

(VDUs)

Video Display Unites (VDUs) (1950s)

Time sharing (1960s)

WIMP (Windows, Icons, Menus, Pointers) (1980s)

Direct manipulation (1980s)

WWW (1990s)

(WIMP)

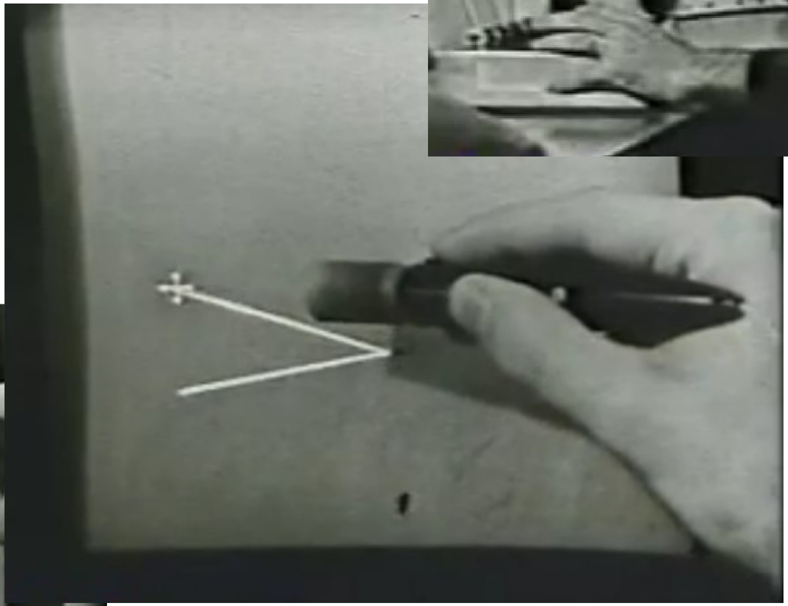
Ubiquitous computing (1990s)

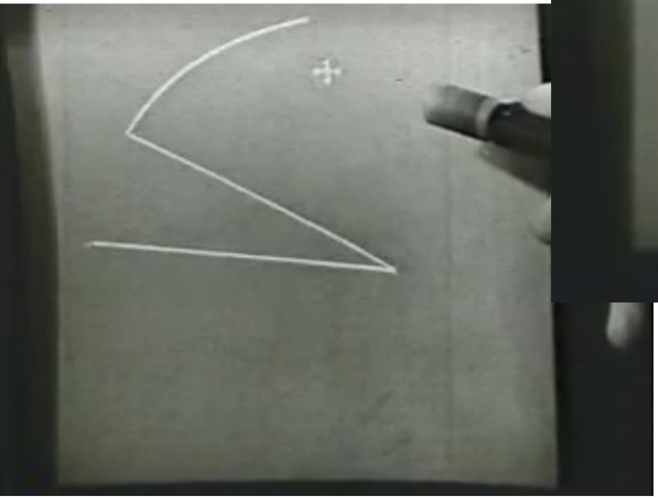
(anytime, anywhere…)

(networking everything ...)

Wearable Computing

Sketchpad
(Ivan Sutherland, 1963)





https://www.youtube.com/watch?v=6orsmFndx_o

Alto and Macintosh

^

Apple Macintosh 512KB, 1984

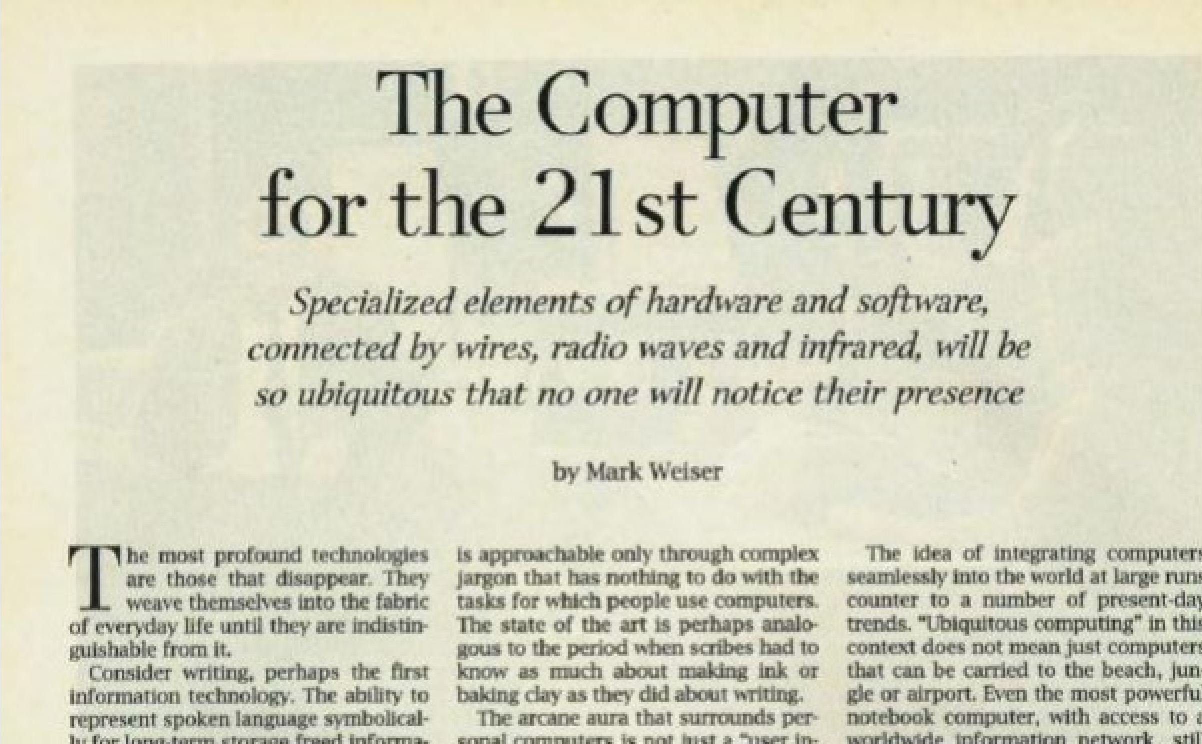
< Xerox PARC, 1973

Ubiquitous computing (Ubicomp)

Mark Wiser, “The Computer for the 21st Century”, Scientific American, Sept 1991, pp. 94-104 (http://wiki.daimi.au.dk/pca/_files/weiser-orig.pdf)

- Computing everywhere and anywhere

- Related concepts:
  - Pervasive computing
  - Ambient intelligence
  - Cyber-physical computing
  - Internet of things
  - Haptic computing



The Computer for the 21st Century

Specialized elements of hardware and software, connected by wires, radio waves and infrared, will be so ubiquitous that no one will notice their presence

by Mark Weiser

The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.

Consider writing, perhaps the first information technology. The ability to represent spoken language symbolically for long-term storage freed informa-

is approachable only through complex jargon that has nothing to do with the tasks for which people use computers. The state of the art is perhaps analogous to the period when scribes had to know as much about making ink or baking clay as they did about writing.

The arcane aura that surrounds personal computers is not just a "user in-

The idea of integrating computers seamlessly into the world at large runs counter to a number of present-day trends. "Ubiquitous computing" in this context does not mean just computers that can be carried to the beach, jungle or airport. Even the most powerful notebook computer, with access to a worldwide information network, still

- Ubiquitous computing involves:

  - small, inexpensive, robust networked processing devices
  - distributed at all scales throughout everyday life

- Examples:
  - refrigerators "aware" of their suitably tagged contents
  - domestic control illumination and heating, continuously and imperceptibly considering the occupants

- Ubiquitous computing presents challenges across computer science: in systems design and engineering, in systems modelling , in user interfaces

http://www.youtube.com/watch?v=CbGw1fX9tMk

https://www.youtube.com/watch?v=JrWQtYAUD8w

Wearable computing

Steve Mann's 'GlassEye™' (aka EyeTap)

“the study or practice of inventing, designing, building, or using miniature body-borne computational and sensory devices. Wearable computers may be worn under, over, or in clothing, or may also be themselves clothes, i.e. "Smart Clothing" (Mann, 1996a).

Other terms: "Body-Borne Computing" or "Bearable Computing"

https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/wearable-computing

Usability principles (a possible list)

User compatibility

Task compatibility

Work-flow compatibility

Product compatibility

Feedback

Coherence

Familiarity

Simplicity

Flexibility

Control

Technology invisibility

Robustness

Error protection

Usability goals:
Easy to learn and memorise
Easy to use
Satisfaction

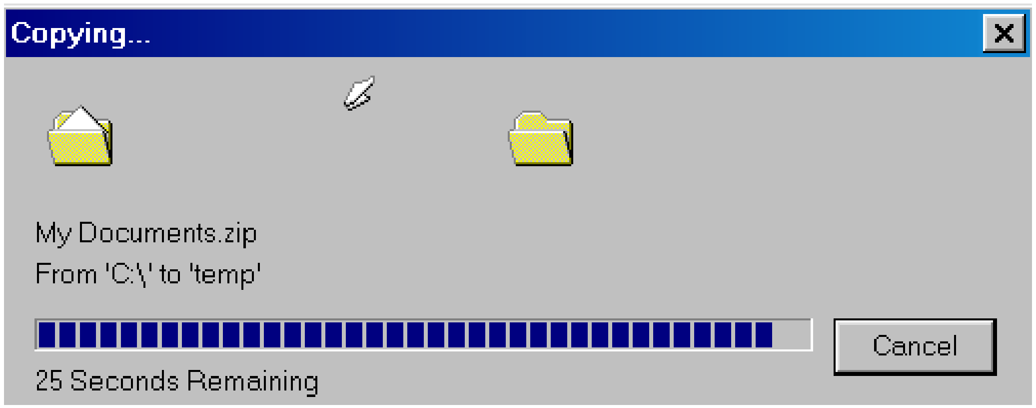
Principles should be used in interactive computing systems…

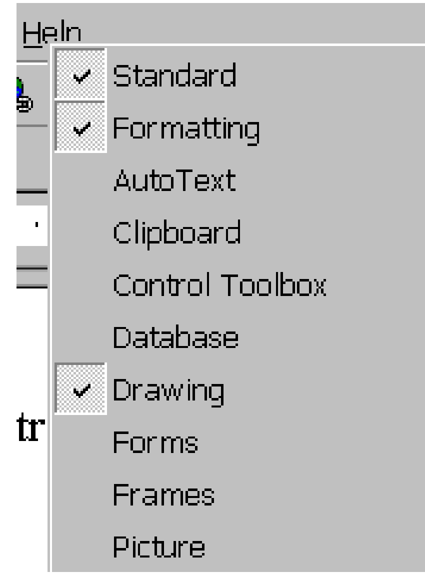
More conventional …

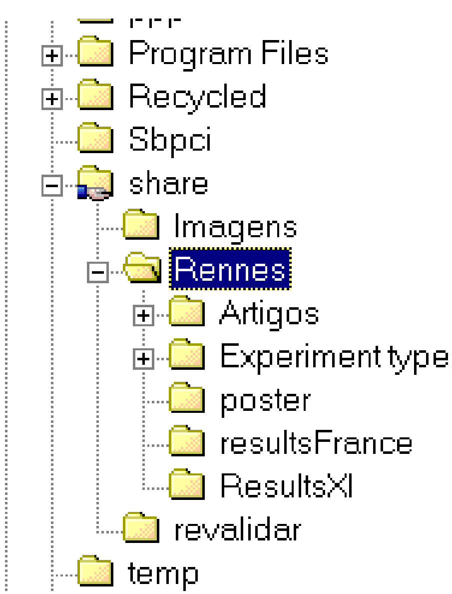
other devices …

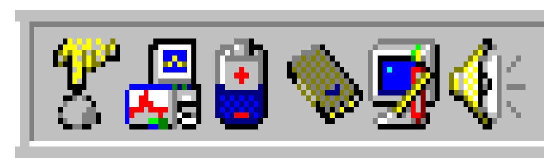
And less conventional interactive computing systems…

Feedback (the past)
Visibility of the system status

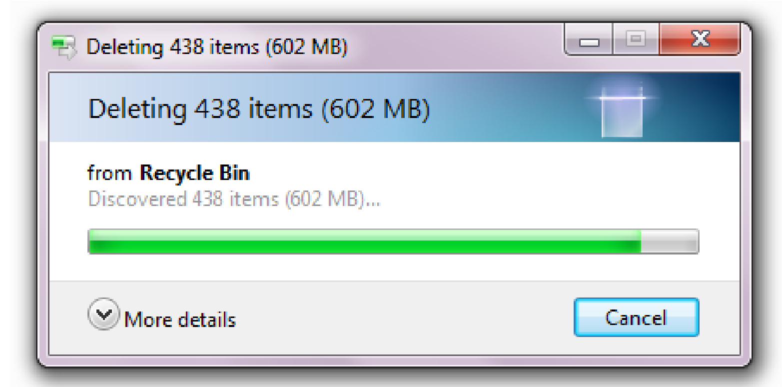


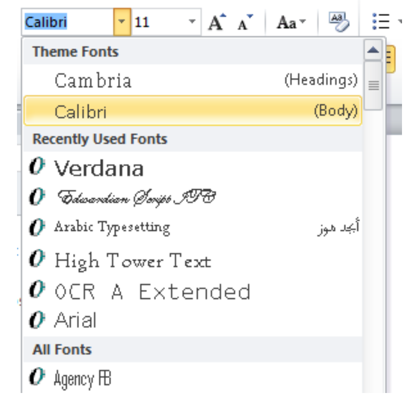


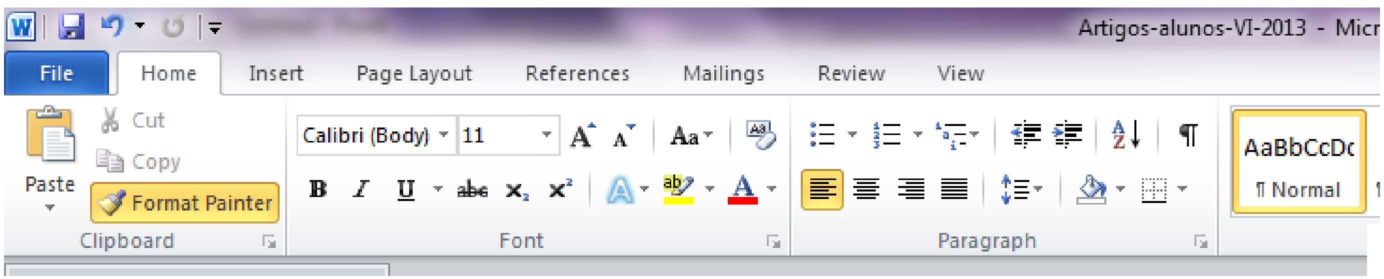


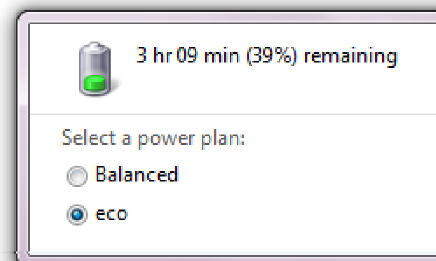


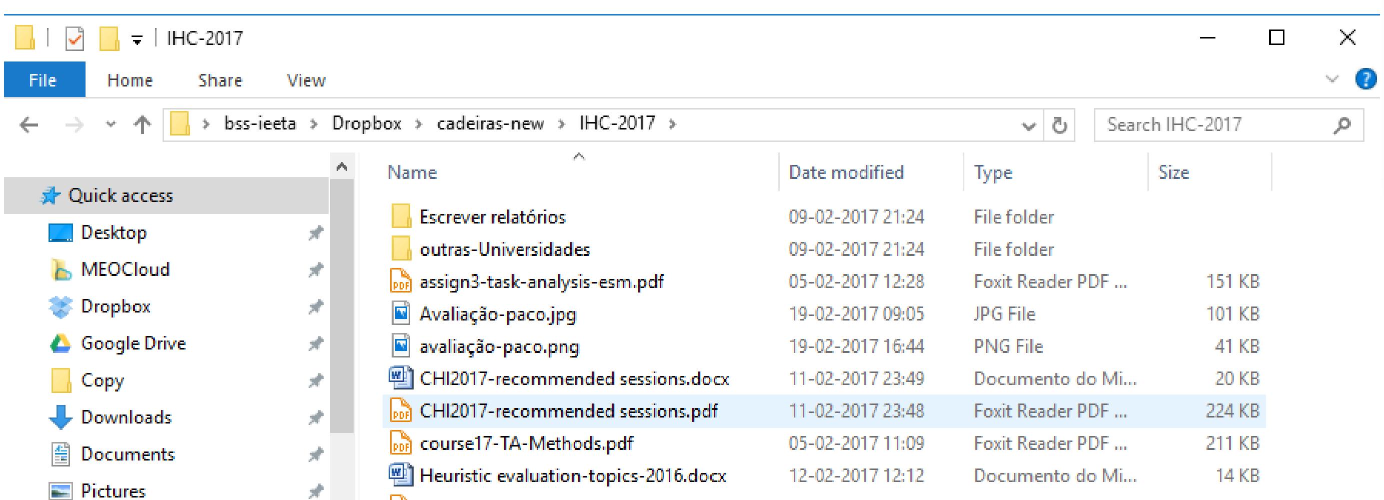
Visibility of the system status, Feedback (in more conventional platforms)

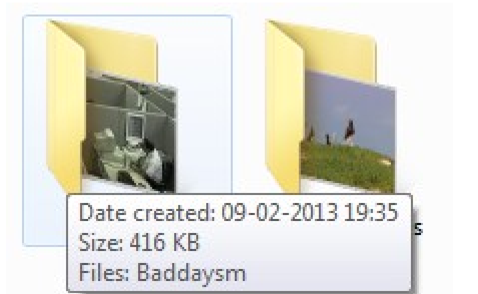








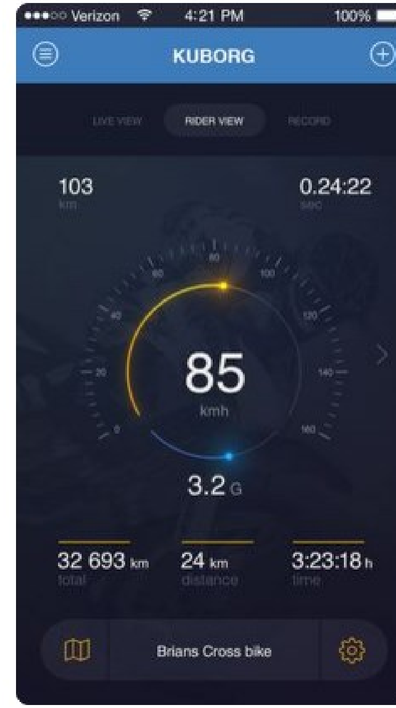




Feedback
Visibility of the system status

TV off

ON



ON

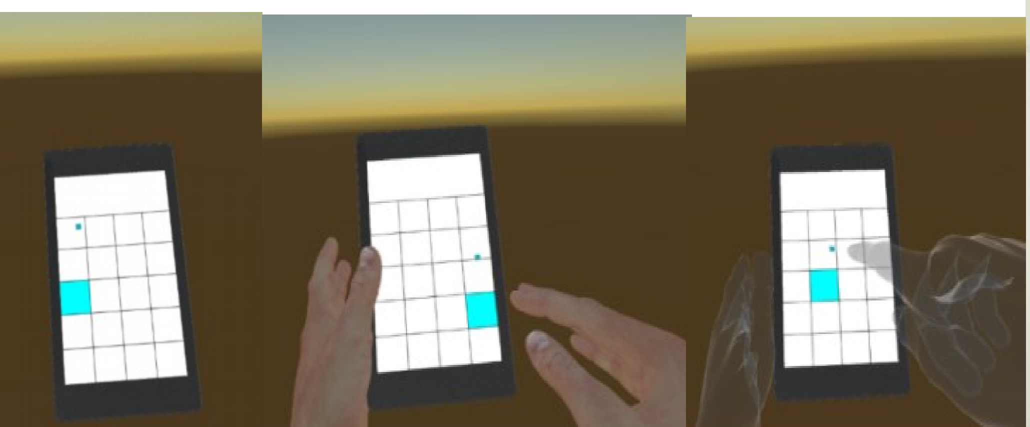
TV on

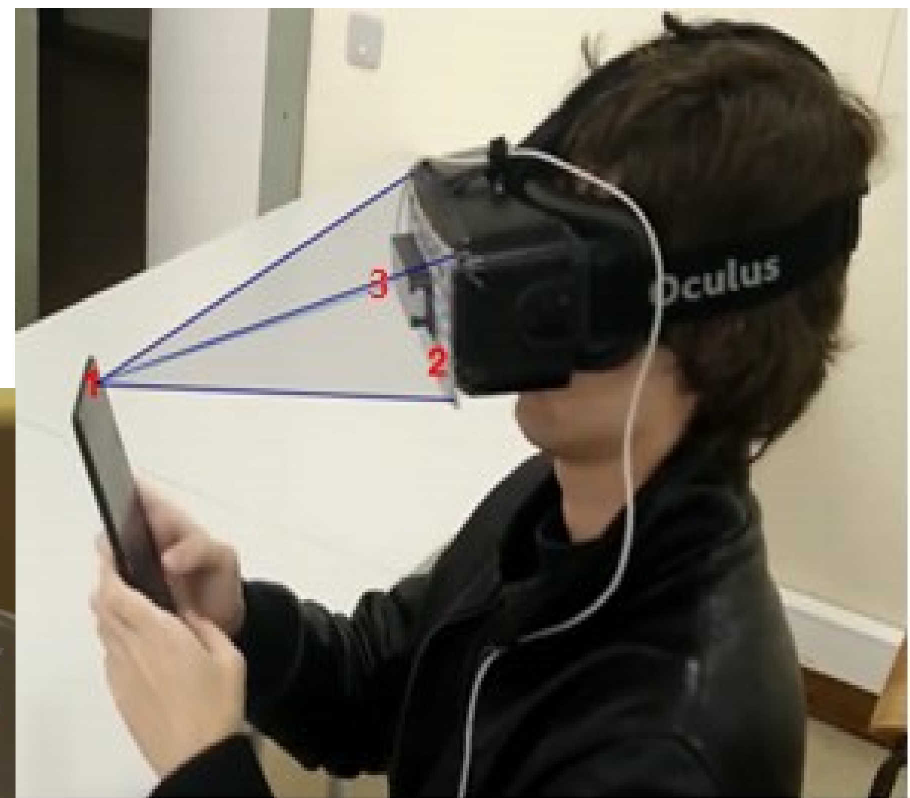
Feedback
Visibility of the system status

In a virtual reality system it is important to have:

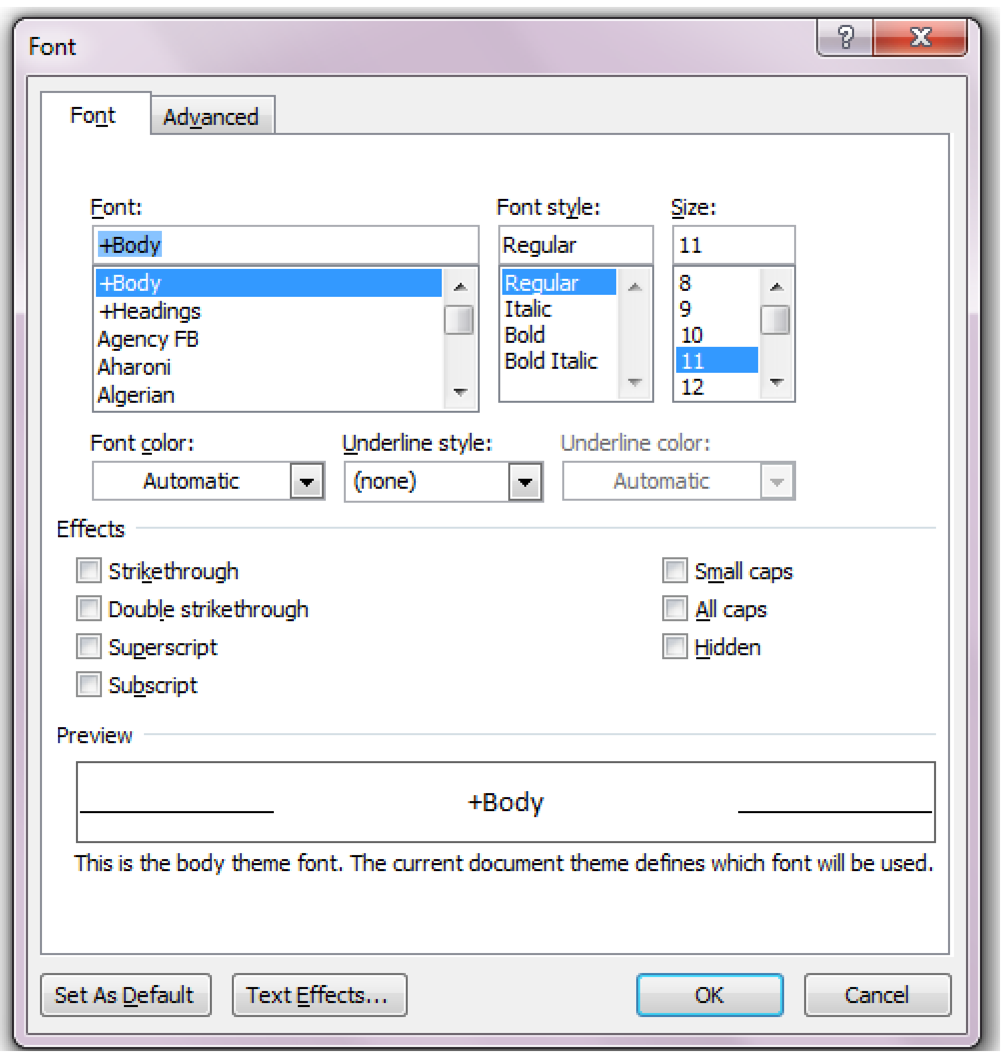
- Feedback in tasks (navigation, manipulation, selection …)
- Visibility concerning body position (avatar)
- …

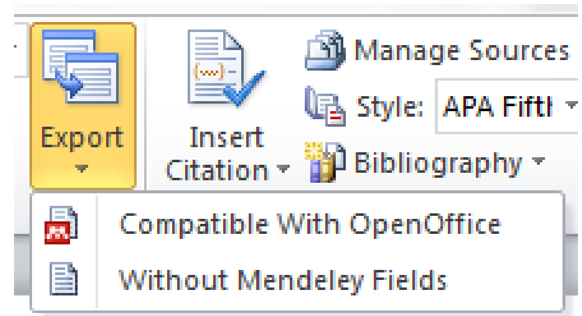
No avatar Realistic avatar Translucent avatar

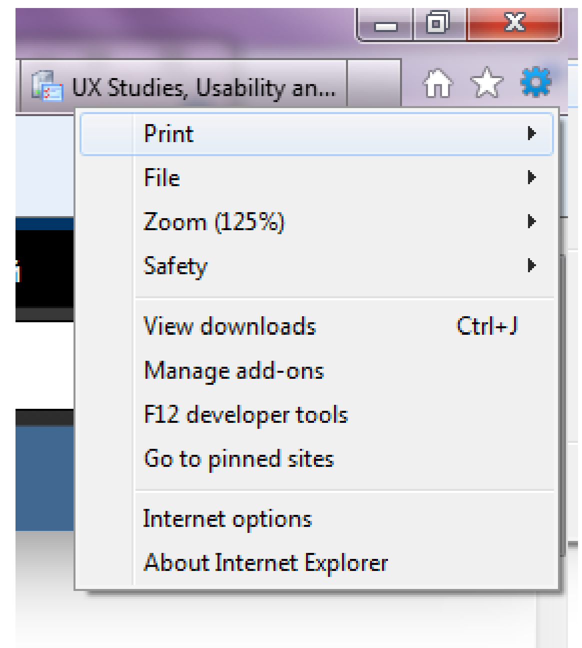




Simplicity
(defaults hide complexity)



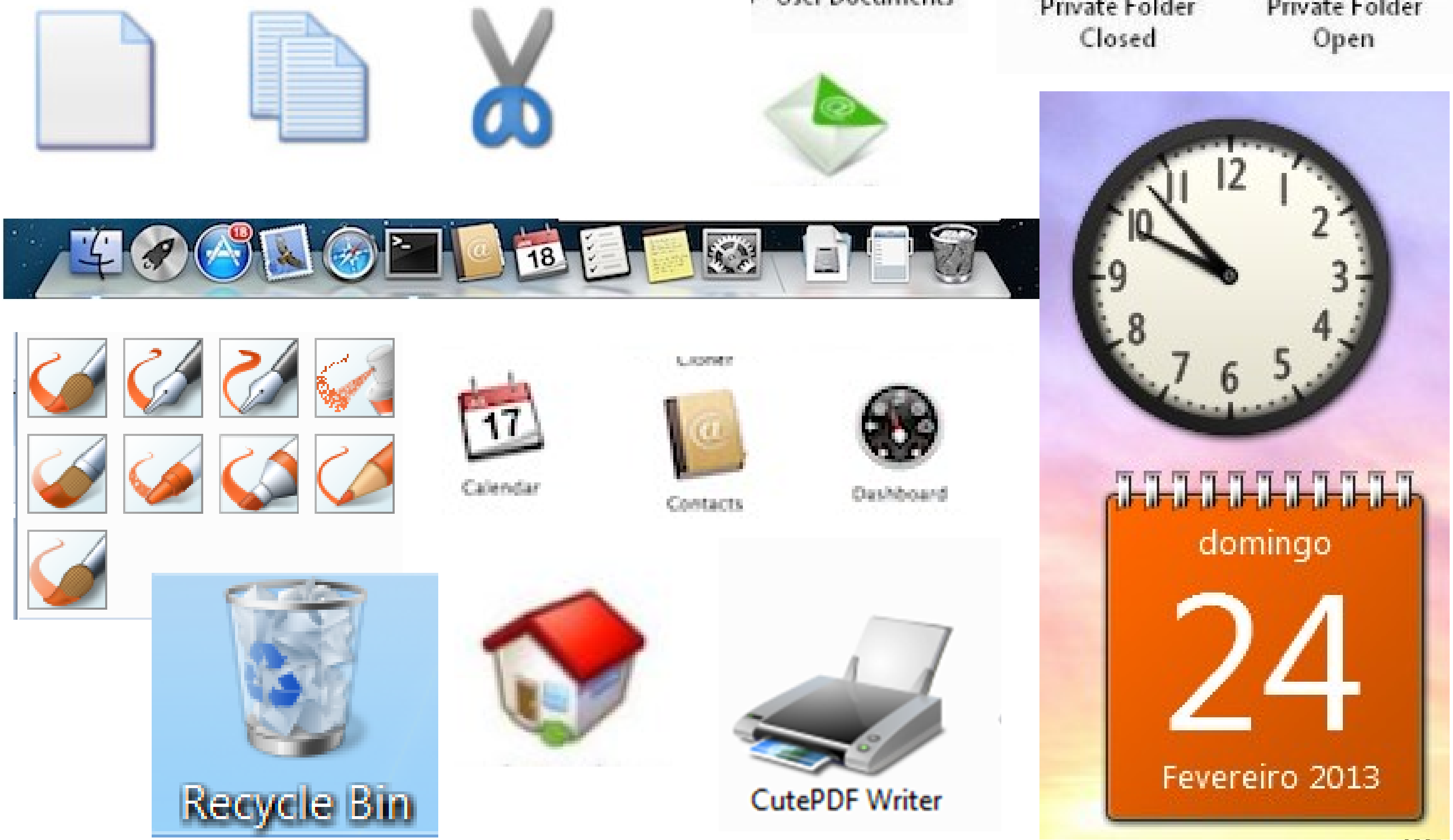




Much of the functionality is not directly accessible

Familiarity

(profit from the user's experience)



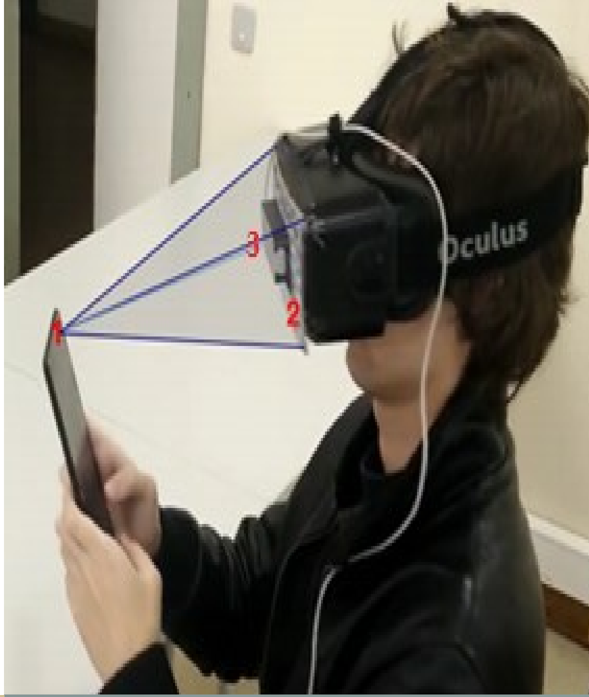
Familiarity

Familiar Icons

Familiarity

In less conventional interactive systems it is important to have:

- Familiar gestures to perform tasks (navigation, manipulation, selection …)



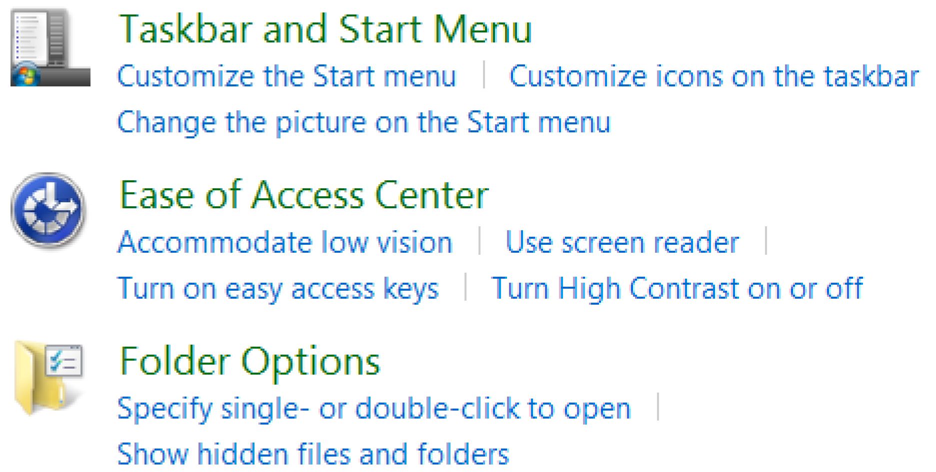

Manipulation method

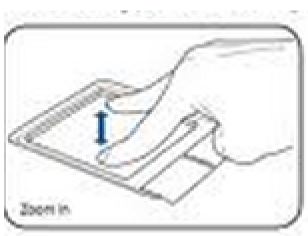

Bike navigation method

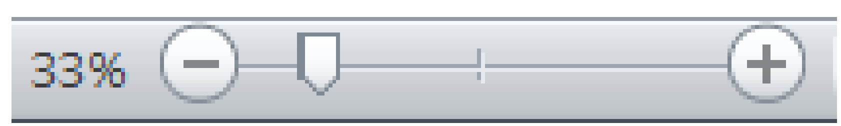

Selection method

Flexibility

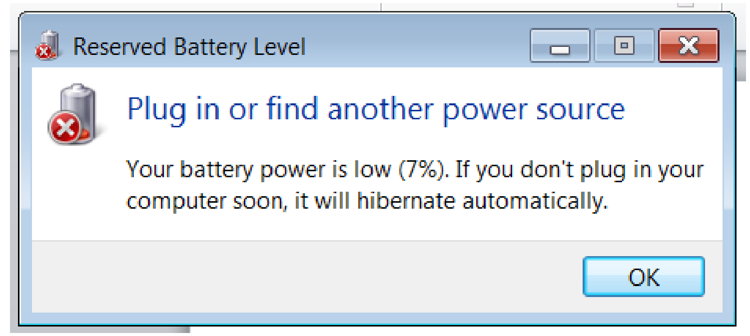
(let the user choose)

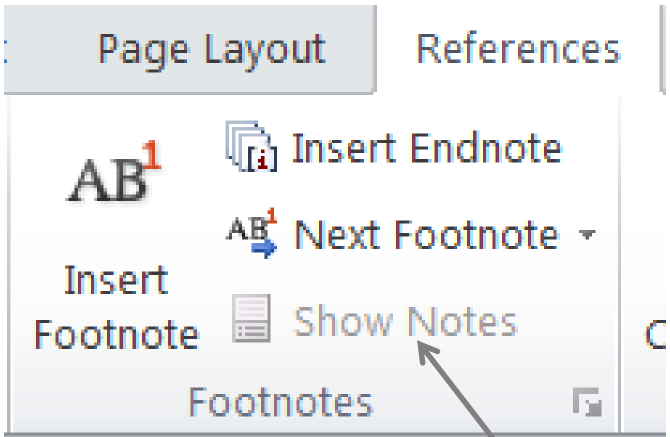




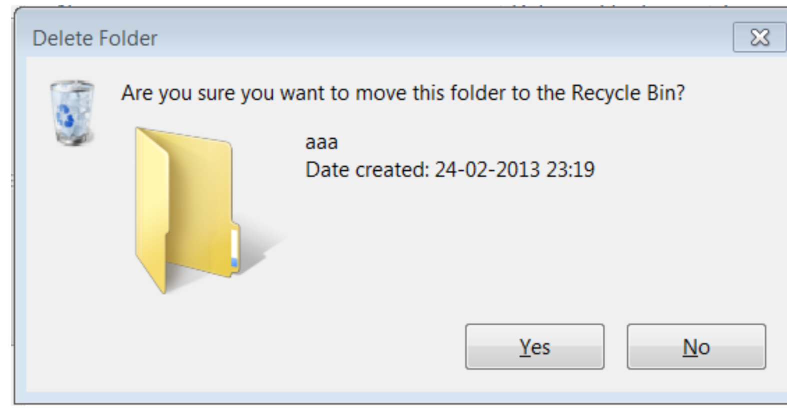


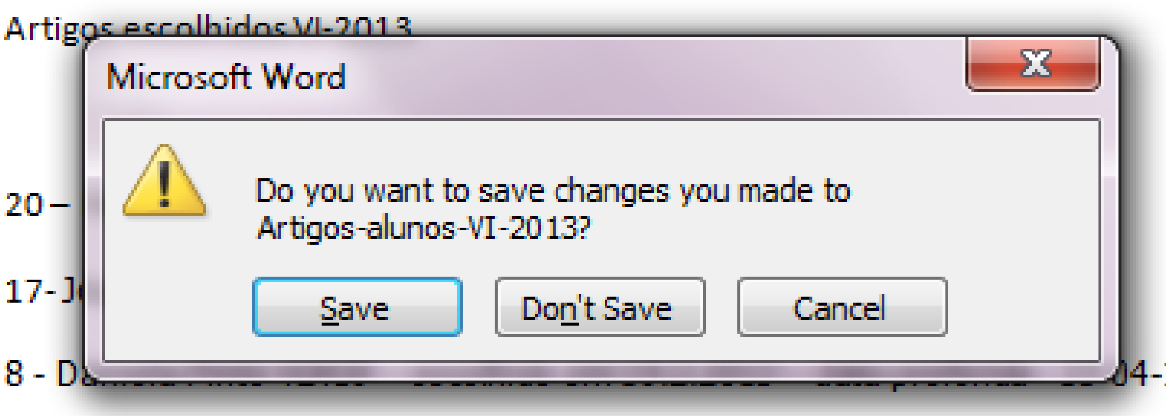
Robustness and error prevention





Not accessible (in grey)





Old usability problems @ DETI (already solved!)

Solved : lights control @ room 4.1.02

Usability problems @ home

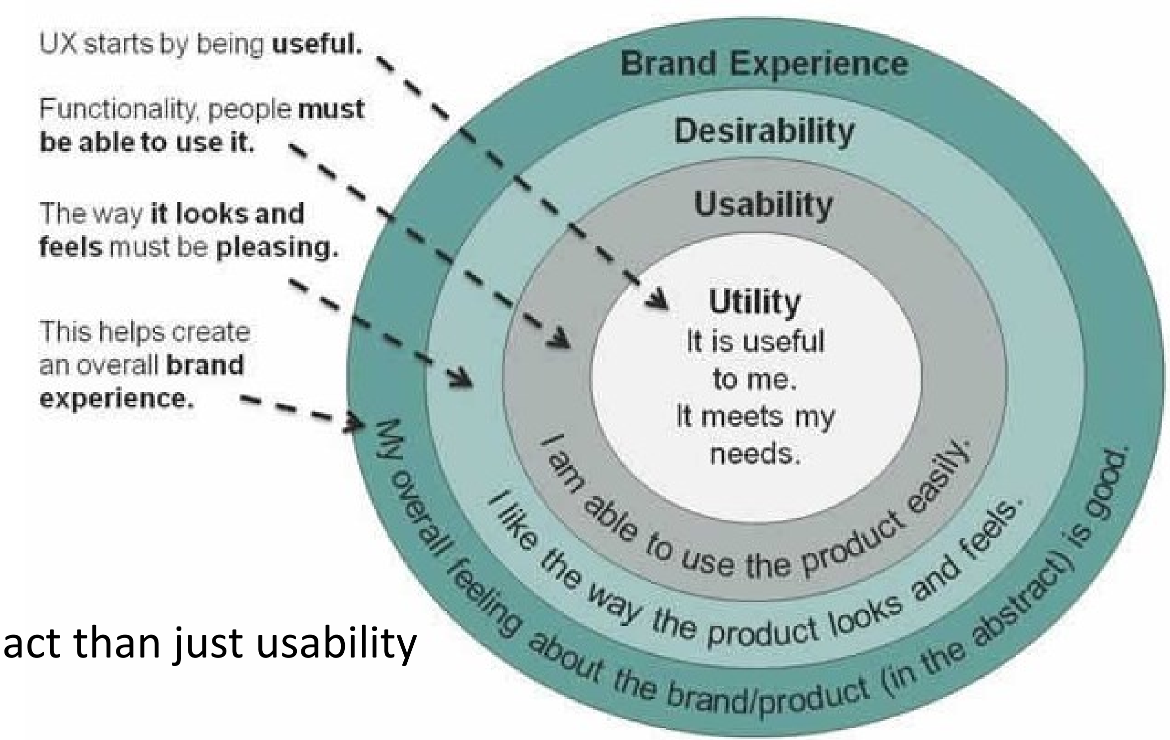
How does it open?

Wrong affordance!

User Experience (UX)

- The ease in which people interact with a system to achieve specific goals
- The experience a person has when he/she interacts with a product (encompasses all aspects)

Usability -> function



A positive UX has a greater impact than just usability

https://www.nngroup.com/articles/ux-research-cheat-sheet/

http://uxpa.org/resources/definitions-user-experience-and-usability

- **Usability** is concerned with the “effectiveness, efficiency and satisfaction with which specified **users** achieve specified goals in particular environments”
- **User experience** is concerned with “all aspects of the **user's experience** when interacting with the product, service,

- User experience (UX) involves a person's:
  - – behaviors,
  - – attitudes,
  - – and emotions about using a particular product, system or service

- It includes the practical, experiential, affective, meaningful and valuable aspects of human-computer interaction and product ownership

- and also a person’s perceptions of system aspects such as utility, ease of use and efficiency

- may be considered subjective and is dynamic as it is constantly modified over time

Main bibliography

- Dix, A., J. Finley, G. Abowd, B. Russell, *Human Computer Interaction*, 3rd ed., Prentice Hall, 2003

- Shneiderman, B., Plaisant, C., Cohen, M., and Jacobs, S., *Designing the User Interface: Strategies for Effective Human-Computer Interaction* , 5th ed., Addison-Wesley, 2009

  or previous editions

- The Encyclopedia of Human Computer Interaction, 2nd ed., Interaction Design Foundation. https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed