Nové, teda, staré otázky, ku ktorým zopár otázok pribudlo a zopár odbudlo. Vypracované narýchlo, viac menej podľa slajdov a starých vypracovaných otázok. Za kvalitu angličtiny neručím.

# Introduction

**What is human-computer interaction? What are the three paradigms of HCI? What are the mediums of advanced HCI?**

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

***Design, implementation, evaluation***

Three paradigms of HCI are **human factors**, **classical cognitivism information** and **embodied cognition interaction.**

Mediums of advanced HCI are… VR goggles? Movement sensors, eyes tracking sensors etc. VA reality ubiquitous computing (Internet of Things); tangible computing brain-computer interfaces.

**Why is it important to consider product usability in a project? Which informatics professions make use of usability engineering?**

Usability engineering is software engineering. Web/Desktop developers, UI/UX designers.

It’s important to consider product usability, because doing so **we reduce risk of making product that doesn’t meet given requirements** and won’t be usable.

**What are typical errors in completed design that did not take into account the principles of usability? Comment, give an example from practice.**

* Design is based on **incorrect requirements**; the product isn’t doing what it’s supposed to do
* Product is **not intuitive,** navigation **doesn’t seem natural**. Dialog in product is in wrong order
* It’s **not easy to work with**. Users have struggle using it
* It’s **not tested** properly (product is too slow, processes in product contain logical mistakes)
* Nonstandard GUI controls
* **Inconsistency**
* No/Wrong default values; bad error messages

**UI design is based on the following principles:**

* **understanding users and their tasks**
* **participatory design**
* **design of visual interfaces**
* **design principles based on heuristics and recommendations**

**Explain these issues and sketch related techniques and theoretical foundations.**

* Understanding users and their tasks – design centred on tasks, **task centred system** design; how to develop task examples; how to evaluate design through a task centred walk-through
* Participatory design **–** design centred on **user a prototyping;** user evaluation interface; how to watch user with goal to find problems in interface
* Design of visual interfaces – design of everyday use type of things (when is element functional); background of designing interface (representations and metaphors); graphic interface of display (position of interface elements on screen)
* Design principles based on heuristics – using **guidelines** to design and discover usability problems

**What are the human characteristics and how the signal can be interpreted? What are the types of memory function that exist? Explain briefly how they work.**

Human characteristics. Each person is different, each person is processing input information in **different way.** (visual, auditory, haptic, movement). Each person has different information stored in **memory**, different level of **reasoning, problem solving, skill** and of course **emotions** influences human behaviour and capabilities.

First we have to physically receipt a stimulus (via eyes) then process and interpret. Signal can be interpreted by:

* **Size and depth;** visual angle indicates how much of view object occupies, relates to the size and distance from eye; familiar objects are perceived as constant size, despite changes in visual angle when far away; cues like overlapping help perception of size and depth
* **Brightness;** subjective reaction to levels of light
* **Colour;** made up of hue, intensity and saturation; cones in eye are sensible to colour wavelengths; blue acuity is lowest; 8% males and 1% females are colour-blind

Visual system compensates for movement and changes in luminance (optical illusions); context is used to resolve ambiguity; optical illusions occur due to over compensation.

Signal can by interpreted by **hearing, reading, touching.**

Types of memory.

**Sensory memory; buffers** for stimuli received through senses; **Iconic** memory (visual stimuli); **Echoic** memory (aural stimuli); **Haptic** memory (tactile stimuli). For example, stereo sound. Its continuously overwritten.

**Short-Term Memory;** rapid access, rapid decay, limited capacity.

**Long-Term Memory;** repository for all our knowledge; slow access, slow decay if any; huge capacity. There are two types. Episodic (serial memory of events) and Semantic (structured memory of facts, concepts and skill). Semantic memory provides access to the information.

# Task and user centred design

**Task centred system design concerns with end-user perspective. Explain the phases: identification, requirements, design, walkthrough evaluation. State the goals of each phase and what techniques are suitable to achieve them.**

* **Identification –** learning about users and their real task; detailed examples of tasks they perform or want to perform (routine, infrequent but important, infrequent and incidental). We can identify tasks by **immersing yourself (**dive in) in their actual work context; by **observing** people in work context; **interviewing** users as they do their work etc…
* **Requirements –** typical users of system (why they are typical users); **set of tasks** which will be addressed by the interface. Requirements are **listed in terms** how they address tasks (absolutely must include, should include, could include, exclude). Then we should discuss why we put task in their category.
* **Design -** With knowledge of tasks, we create interface and its **logical order,** flow. We create dialog. Consider **the real-world context** for users and how design **features work together**.
* **Walkthrough evaluation – testing of design;** it’s a process. First, we select one of scenarios, then for each user’s step/action in the task we evaluate if user’s actions are believable, if we can rely on users knowledge and training. If there is problem, note it, comment it, then assume it’s repaired and continue to the next task.

Good for **debugging in interface**.

**How would you identify real users of a system and their tasks? Discuss approaches when users and tasks already exist and when not (will arise in future)**

**If they exist –** we **contact potential users** of our system and **note which tasks** are important to do (**sort it by importance** in categories). Then we could **observe users** doing their work in their **environment;** we can ask and **interview** then as they work, we should **serve** their requests.

**If they not –** we **create, synthesize** our potential users by creating **expected set of users and tasks.** Later, we should **verify and modify** if necessary.

**What properties must have good task examples? What facts are recorded and what is described?**

Good tasks say what user wants to do but **doesn’t say how to do it**. There should not be any assumptions about interface. Good tasks are **very specific** and **describe complete job**. It forces the designer to consider how interface features work together. Contrasts how information i/o flows through dialog. (where does it come from; where does it go, what happens next). They are **evaluated.**

We record what **users know** before testing a task, we ask users for omissions, corrections, clarifications and **suggestions.**

**Characterize evaluation of design using walk-throughs. What is input for the walk-through, what properties are tested and evaluated? Describe the process of evaluation.**

First, we choose one task to evaluate. Then for **each step/action** in the task we **observe user** if they know what to do, how to navigate, if they feel good using interface. If they don’t we have a problem. We should **note the problem**, ask for **some comment** from a user and note it. Then assume we fixed it and carry on to the next task. **Repeat for every task** we want to evaluate.

**Explain and argument the main differences between goal oriented design and task oriented design.**

**Goal oriented design vs. task oriented design.**

In GCD we articulate, **express user goals** instead of task. Goal is **desired end situation** which tend to be **stable**. On the other hand**, task is an intermediate process** needed to achieve the goal and may change as technology or work patterns change.

GCD – designer is looking for solutions that satisfy these goals, task sequence may differ substantially from current process. Approach in GCD; developing a persona. Precise, specific description of the user and the goal they accomplish; develop a cast of characters (3 - 12), one will be primary persona – the main focus of the design.

**For a given application find 3 typical users. Create their description and name their typical (different) tasks. Classify the priority.**

Improvise…

# Computer

**What is a computer and what are the various elements?**

A computer system is made up of various elements

* + each of these elements affects the interaction
  + input devices – text entry and pointing
  + output devices
  + screen (small & large), digital paper
  + virtual reality
  + special interaction and display devices
  + physical interaction – e.g. sound, haptic, bio-sensing
  + paper – as output (print) and input (scan)
  + memory – RAM & permanent media, capacity & access
  + processing – speed of processing, networks

**Give examples of text entry devices. Explain how positioning, pointing and drawing works. What is the difference between trackball and joystick? What is eye gaze and how it works?**

*Keyboard, phone pad, handwriting recognition, speech recognition.*

Positioning, pointing and drawing. There are two methods for detecting motion.

**Mechanical –** ball on underside of mouse turns as mouse is moved; rotates **orthogonal** potentiometers; can be used on almost any flat surface.

**Optical –** light emitting diode on underside of mouse; less resistant to dust and dirt. Detects alterations in reflected light intensity to calculate relative motion.

There are some experiments with the footmouse… not very common ☺.

**Touchpad –** small touch sensible tablets; stroke to move mouse pointer; mainly used in laptop computers. Important to have good “acceleration” settings.

**Trackball -** ball is rotated inside static housing(like a upside down mouse); relative motion moves cursor; **indirect** device; fairly accurate, very fast for gaming. Used in some portable laptops.

**Thumbwheels –** for accurate CAD; for fast scrolling.

**Joystic – indirect device,** pressure of stick = velocity of movement. Buttons for selection, often used for computer games.

Keyboard nipple, touchscreen, stylus and light pen, graphic tablet… cursor keys, discrete positioning controls.

**Eye gaze –** control interface by eye gaze direction (look at menu to select it). Uses very low power laser beam reflected off retina. Mainly **used for evaluation**, but there is potential for **hands-free control**. High accuracy requires headset, cheaper devices sit under the screen like a webcam.

**What are the most characteristic display devices? Are there health hazards?**

Bitmap displays, CRT, LCD. There are health hazards regarding CRT. X-rays are largely absorbed by the screen, but not at rear! UV- and IR radiation from phosphors. Electrostatic fields can cause rashes.

**How VR and 3D Interaction work? What are the main technologies? Explain VR motion sickness. What are Burdea’s 3 I’s of VR?**

VR and 3D interaction are based on positioning in 3D space and seeing in 3D. For positioning there are used virtual controls, 3D mouse, data glove, VR helmets and whole body tracking.

VR headsets are small TV for each eye with slightly different angle creating 3D effect.

Oculus rift, HTV Vive…

VR motion sickness, it’s caused by time delay between head movement and eyes. (lags, slow software) or by depth perception. Headset gives different stereo distance.

Burdea’s I’s of VR are:

* **Interactivity** – user impacts world
* **Immersion** (diving in) – believing you are there
* **Imagination** – user “buying” into the experience

**What is haptics? Give example of how it can be used. Mention the different human factors that exist in VR. What are the advantages and disadvantages of VR/games?**

From Greek “*Haphe”*, means pertaining (relating) to the sense of touch. Haptic technology refers to technology which interfaces the user via the **sense of touch by applying forces**, vibrations and motions **to the user**. This mechanical stimulation can be used to assist in the crating of virtual objects and to **enhance** (improve) the remote control of machines and devices **(teleoperators**). Some low-end haptic devices are already common – joysticks and steering wheels in gaming.

Human factors in VR.

* **Human performance** efficiency
* **Societal implications**
* **Health and safety**

Will the user get sick, which task are most suitable for users etc..

Of course, there are some potential dangers in VR/games. Excessive gameplay can be fatal. In Korea one man died after playing 50hours. Games can lead **to isolation** and potential suicide.

Advantages of VR/Games. People regularly exposed to videogames have improved visual and spatial attention, memory, mental rotation abilities, movement control brain networks, visuomotor skills.

**What are the characteristic physical controls and sensors? What is Moore’s law?**

Buttons, dials, lights, screens, sounds, control panel etc…   
  
**Moore’s law;** computers are **getting faster** and faster every year. Moore noticed a pattern, processor **speed doubles every 18 months** and it’s similar for memory.

# High level models of human-computer interaction

**Schneiderman model of interaction distinguishes semantic and syntactic knowledge. Which usability problems are related to syntactic knowledge?**

**Syntactic knowledge** **– rules** or **combinations of command and signals** seen as device-dependent detail of how to use system. For example (backspace – deletes previous character; right mouse button – raise menu…etc.). That’s nice, but there are **user problems** with syntactic knowledge. Such as, syntactic details differ between (and within) system. That means **little consistency** (arbitrary, kind of random). Example leaving mail reading in gmacs. (Different commands to do similar task.). Hard to learn and easily forgotten.

**Schneiderman model of interaction distinguishes semantic and syntactic knowledge. Which usability problems are related to the semantic knowledge of computer concepts and task concepts? Characterize mapping problems.**

**Semantic knowledge –** the meaningbehind computer concepts. Usually follows hierarchical structure. People learn computer concepts by meaningful learning, demonstrations, explanations, trial by error etc. There are some **problems** too. Many people now using computers aren’t computer scientists, so they must be trained in **computer literacy**. How to use mouse, scrollbar, shortcuts, closing and opening windows. People tends to prefer working on task, not on computer knowledge.

**Mapping is extremely important.** Task semantics to computer semantics to computer syntax. (write a letter – open a file, use editor, save it to disk – select menu items, key strokes for formatting…). Bad mapping is using latex for writing a letter, because aside from task semantics, user must also know semantics/syntax of text editor, latex, unix compiling. Relatively good mapping is trashcan used to throe away files. User must know only how to use mouse (mouse syntax of selecting and dragging) and computer semantics is almost analogous to task semantics.

**In a simplified Norman model, we consider 4 stages of interaction (intention, selection, execution, evaluation). Explain these stages and their interrelations when performing a task.**

* **Intention** – **“what we want to happen”;** internal mental characterisation of goal; may compromise goals and sub goals (but rarely are well planned); similar to **task semantics** (*e.g. “begin letter to Aunt Harriet”*)
* **Selection** – selecting an action; review possible actions and select most appropriate; similar to mapping between **task and compute semantics**; *(e. g. :use the emacs editor to create file harriet.letter)*
* **Execution** – carry out the action using the computer; similar to **mapping between semantics and computer syntax;** *(e. g. type “emacs -nw harriet.letter”)*
* **Evaluation – check the results** of executing the action and compare it with expectations. Requires perception, interpretation and incremental evaluation.

**4-stages model Norman identifies gulfs of execution and evaluation. Explain these issues.**

Checking if executed action results meet expectations.

**Gulf of Execution –** do actions provided by system correspond to the **intentions** of the user? Gulf is amount of effort exerted *(put into)* to **transform intentions** into selected and executed actions. Good system provides **direct mapping** between intentions and selections. *(printing a letter)*

**Gulf of Evaluation –** can **feedback be interpreted** in terms of intentions and expectations? Feedback should be readable and transparent. Gulf is amount of effort exerted to **interpret feedback**. Good system is where feedback is easily interpreted as task expectations. *(graphical simulation of text before printing).* Bad system is when there is no feedback or it’s difficult to interpret. *(e. g. some Unix commands.)*

**4-stages Norman model supports design solution by raising and considering questions on visibility, quality of conceptual model, good mappings a high-quality feedback. Explain, comment, give examples.**

* **Visibility –** it’s possible to see current state of application and alternatives
* **Quality of conceptual model –** constituent model, same rules through whole system, comman visual coherence, similarity.
* **Good mappings –** link between actions and results, state of system, commands and everything visible
* **High-quality feedback -**  whole and coherent feedback about results of actions, user knows what happened, what he accomplished.

# User centred design

**Explain the difference between system-centred and user-centred design.**

**System-centred design.** Design is based upon system and its requirements. Upon what is easy to build on given platform or what is easy to recreate from the available tools and of course what programmes finds interesting. These designs are complex and precise, but harder to learn how to use them.

**User-centred design.** Design is based upon a **user’s abilities and real needs, context, work, real tasks**. User needs **usable and useful product**. Design should **be comfortable to use for a user**, remove as many obstacles as possible. User centred system is based on **understating** the domain of work or play in which people are engaged and in which they interact with computers. The result of user centred system should be **satisfied costumer.** The process is collaboration between designers and costumers.

**What is participatory design, positives and negatives.**

Participatory design is when **end** users are members in process of designing a system. They are subject of **collaboration** and they offer **huge feedback** to designers.

**Upsides:**

* Users are **excellent at reacting** to suggested system designs (designs must be concrete and visible)
* Users bring important “**folk**, slang” **knowledge** of work context which may be inaccessible to the design team.
* Often results in greater but-in for the system

**Downsides:**

* **Hard to get** good pool of end users (expensive, reluctance). **Users do not want to be tested**, feel like laboratory rats
* Users are **not expert designers**, you can’t expect them to come up with design ideas
* Users have **problem forming meaningful feedback**
* The user is **not always right**

**Methods for involving a user - explaining design, visualization, sketching and prototyping. Describe these techniques and their usage in early and late design stages.**

We have to communicate with user, at least talk to them, context interviews, interviewing users in their workspace as they are dong their job. Discover their culture, expectations, requirements.

* Explaining design – **explain** what you are going to do, get input at all design stages and do revisions
* Visualization – people react far differently with verbal explanations than visual. This is why **prototypes are critical**.
* Sketching – first sketches are made on paper, later we have to make more detailed, exact and formal sketches. It’s purpose is to **brainstorm**, get **user reactions,** modifications and **suggestions.** They are easy and fast to make, but hard to envision a dialog progression.
* Prototyping – **getting sketches together**, make them interactive

**Explain the following attributes of sketches in the design process: quick, timely, disposable, plentiful, clear vocabulary, constrained resolution, consistency with state, suggesting and exploring.**

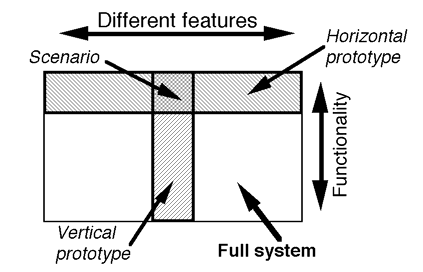
They are

* quick to make;
* timely provided when needed;
* disposable (single-use) investment in the concept, not the execution;
* plentiful, they make sense in a collection of series of ideas;
* clear vocabulary, rendering and style indicates it’s a sketch, not and implementation;
* constrained (limited) resolution doesn’t inhibit concept exploration
* consistency with state, refinement of rendering matches the actual state of development of the concept
* suggesting and exploring – value lies in suggesting and provoking what could be, they are catalyst, **starting point to conversation** and interaction

**When designing we use prototypes with a constrained functionality (horizontal, vertical, scenarios). Explain the differences and usage.**

**Horizontal.** Designed whole system, entire surface interface and interactions inside, but there is no functionality, it’s just a simulation, no real work can be performed. To show **many different features**.

**Vertical.** Includes **in-depth functionality** for only **few selected features** (sometimes just one). Common design ideas **can be tested** in depth.

**Scenario. Scripts** of particular fixed uses of the system, **no deviation** allowed.   
  


# Brain Computer Interfaces

# 

**What are brain computer interfaces?**

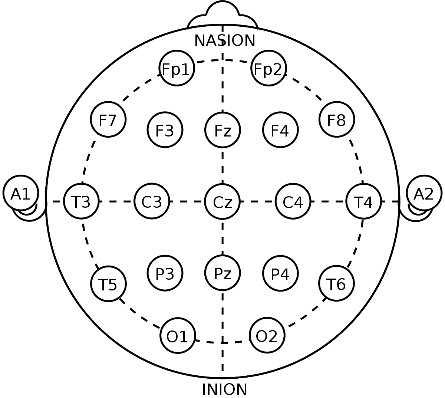
Brain-Computer Interface (BCI) or Brain– Machine Interface (BMI), is a direct way of **communication** between the **brain** and a **computer system**. BCI uses signals from brain to control computers or send then some information.

**What is the electroencephalogram? What are the main principles of EEG? What is the 10-20 system?**

EEG is **electrophysiological** monitoring method to record **electrical activity** of the brain. It’s **non-invasive** (usually) with electrodes placed **on scalp**, head.

The brain’s electrical charge is maintained by billions of neurons. Neurons pass signals via **action potential**. When the wave of ions reaches electrodes on scalp, the difference in push is measured by voltmeter. The activity is noted. Different wavelengths of different frequencies depending in location on scalp mean different things. For example, 9-11Hz frequency in front (motor cortex) means intention of movement.

**10-20 system is international system** describing electrode placement on the scalp for EEG tests or experiments.



**What are the three main types of EEG-based BCIs? What is BCI illiteracy and how to improve it?**

* **Event related potential** (P300)
* **Sensorimotor rhythms** (SMR)
* **Steady state visually evoked potentials (SSVEP)**

**BCI illiteracy.** Obtaining reliable BCI control. (20% of BCI do not obtain reliable BCI control).

BCI illiteracy can be improved via improving **classification accuracy**, **changing paradigm**, changing **neuroimaging techniques**, combining neuroimaging techniques, **combining** paradigms

**Mention some case studies that BCIs can be used for HCI. What is bio-feedback?**

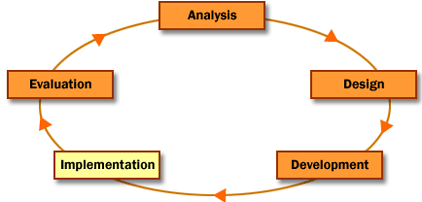
Multimodal games. Motor imagery. **Neurogaming.** BCI is used as primary input, excludes usage of traditional controllers. Examining brain activity while playing computer games. **User profiling** for BCI and Games.

Biofeedback is process of **gaining awareness** of many physiological functions. Process that enables an individual to learn how to change physiological activity for the purposes **of improving health and performance.** Biofeedback in HCI is very experimental at this stage and we won’t see many commercial applications soon.

# Evaluating interfaces with users

**Evaluating interfaces with users is applied in all stages of project lifecycle. What are the main goals in following stages: pre-design, initial design, iterative design, acceptance testing?**

* **Pre-design** – investing in new expensive system **requires proof of viability** (ability to survive), proof that system is needed, **required**
* **Initial design** – develop and evaluate initial design ideas with the user



* **Iterative design – improving** design. Does system match user’s tasks requirements? Are there specific problems? Finding **solutions that work.**
* **Acceptance testing –** Verify that system **meets expected** user performance **criteria.** *(80% of first time customers will take 1-3 minutes to withdraw 50€ from automatic teller)*

**What are the differences between naturalistic and experimental approaches to evaluation? Explain trade-offs and external and internal validity of testing.**

**Naturalistic approach.** Observations occurs in **real life,** realistic setting. Problems with naturalistic approach: **hard to arrange** and do, **time consuming**, **may not generalize**.

**Experimental approach.** Experimenter controls all environmental factors. Study relations by manipulating **independent variables**. **Observe effects** on **one** or more dependent variables. Nothing else changes. It’s **targeted** **on** one or more **variables**. *There is no difference in user performance (time and error rate) when selecting an item from pull down or pull right menu of 4 items.*

**External validity.** Confidence that results applies to real situations. Usually good in natural situations.

**Internal validity.** Confidence in our experimental results. Usually good in experimental settings.

**Trade-off.** Natural vs. Experimental. **Precise** and direct control over experimental versus desire for maximum generalizability (maximum generalization) in real life situations.

**Usability evaluation applies low cost methods, such as inspection, extracting the conceptual model, observation, query techniques and continuous evaluation. Characterize these techniques and discuss pros and cons.**

Low cost methods capture **most large** and **many minor** problems. Qualitative and quantitative.

* **Inspection** – designer tries system or prototype if it “**feels right’**. Benefits are that you can catch soma major problems in **early versions**. Problems; it’s not reliable as it is completely **subjective**, inspector is non-typical user and intuition and introspection are often wrong.
* **Extracting the conceptual model** – (take out conceptual model) showing user static images of the prototype or screens during use. Then asking user to explain the function of each screen element and ow they would perform a particular task. We note how person **perceives** very first time it’s viewed and how person perceives a screen after its been used for a while. Value of extracting the conceptual model is its good for eliciting (evoking) peoples understanding before and after. But it’s poor for examining system exploration and learning. (Good for finding presentation mistakes, not logical)
* **Observation** – we observe user **interacting** with system. In lab – user is asked to complete set of predefined tasks or in field – user goes through normal duties. Excellent at identifying gross design/interface problems. Validity **depends on how much controlled the situation is.**
* **Query techniques** – interviews (audio, video), questionnaires, surveys; good for finding specific issues. But it’s time consuming accounts are subjective.
* **Continuous evaluation** – Monitor systems in actual use, used in late stages of development. Fixing problem in next release. Users give feedback via **gripe lines.** Good for seeing real life use.

**Explain the basic ethic principles before testing: user's time, comfort, privacy, informing, volunteering.**

Use pilot tests, have everything ready before user shows up. Emphasize that system is being tested, not user. Acknowledge that SW may have some issues. Let users know they can stop at any time. Tell users individual test will be completely confidential. Inform the user about monitoring and answer objectively all users questions. Only use volunteers. They must sign paper.

**Explain the basic ethic principles during testing: user's time, comfort, privacy.**

Don’t waste users time by never having user performing unnecessary tasks. Make user comfortable by giving him early success experience, keeping relaxed atmosphere at room, coffee, breaks, stop if test is unpleasant… User is not tested, the system is. Maintain privacy. Do not allow user’s management to observe test.

**Explain the basic ethic principles after testing: comfort, privacy, informing.**

State that **they helped** you find areas of improvement, inform them, answer questions about experiment that could’ve biased results before. Never report results in a way user can be identified. Only show tapes outside the research group with the user’s permission.

# Evaluation - Controlled experiments

**Four scales of measurement are nominal, ordinal, interval, ratio. Explain briefly, give examples and consider the possible sources of errors.**

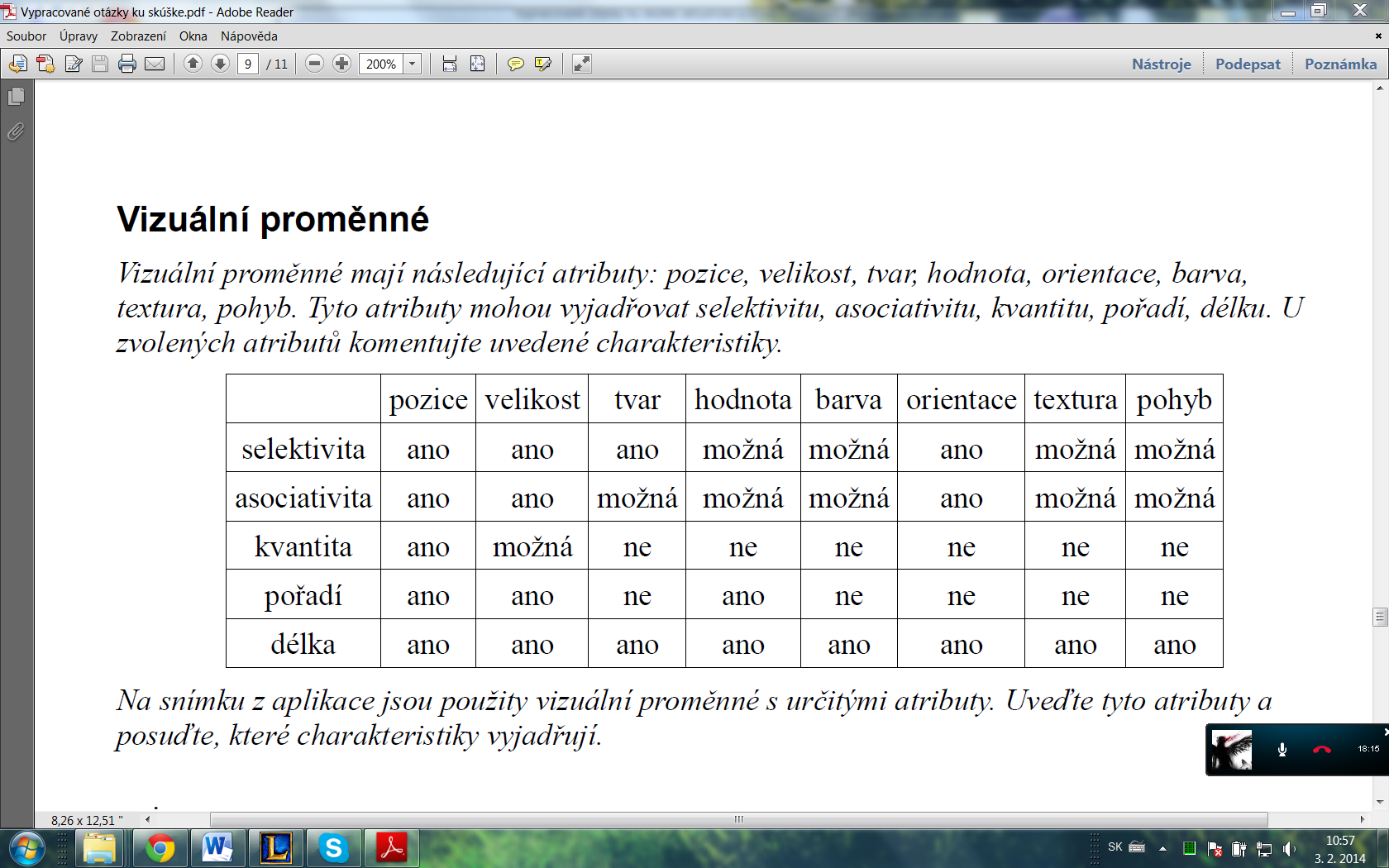
* **Nominal.** Classification into named or numbered **unordered categories.** (Country of birth, user groups, gender). **Agreement in labelling**, vague labels, **vague** diffs in objects may be sources of error.
* **Ordinal.** Classification into named or numbered categories. No information on magnitude of differences between categories. Same as in nominal, agreement in labelling, vague labels…
* **Interval.** Classification into ordered categories with equal diff between categories. (Temperature `C or `F). Sources of error are instrument calibration, readability, human error, skill.
* **Ratio.** Interval scale with absolute, non-arbitrary (fixed) zero. *(temperature in Kelvin, length, weight, time periods)*

**The following concepts are important for designing: perceived affordances, causality, visible constraints, mapping, transfer effects, idioms & population stereotypes, conceptual models, individual differences, difficulty of design. Explain the selected concepts, provide positive and negative examples. !!!**

* **perceived affordances, perceived** possibilities of object which it’s offering to be used
* **causality,** thing, what happens right after action is perceived by people like it was caused by the action – interpretation of feedback
* **visible constraints,** constrain of possible actions of object perceived by its looks
* **mapping,** set of all relations between objects
* **transfer effects,** people transfer their knowledge from similar objects they know to current objects
* **idioms & population stereotypes,** idioms of interface, standard looking interface we know, we remember and know how to use it, it may define different behaviour.   
  Population stereotypes. Idioms differ depending on culture.
* **conceptual models,** people use mental models of way how things work created on these concepts. Models offer people mental simulation of running a device
* **individual differences,** people differ. Rule of thumb, care about 95% of audience.
* **difficulty of design.**

# Information representation and visualization

**Visual variable attributes are: position, size, shape, value, orientation, colour, texture, motion. The attributes may be selective, associative, quantitative, order, length. Comment these characteristics for the selected attributes.**



**The snapshots of a given application contain visual variables with specific attributes. Mark these attributes and asses the characteristics they express.**

# Metaphors

**What are interface metaphors, theirs purpose and possible problems? Comment emotional colouring, possible restrictions, typical errors of metaphors' usage.**

Represents a system object as if it were another type of object. *(Disc/network file structure represented as file folders).* It uses our **knowledge of familiar**, concrete objects to understand abstract computer and task concepts. Problem is that metaphor portrays **inaccurate/naïve conceptual model** of the system. (spreadsheet is a sheet, games imitate real world). Use metaphor **that match user’s task**. Desktop for office workers, paintbrush for artists.

**Emotional tone**/colour should be a**ppropriate** to users. (deletion metaphors – trashcan, black hole, nuclear explosion).

**Errors**. Too literal, too cute (novelty quickly fades), mismatched metaphor doesn’t match user thinking.

**Explain "interface with direct manipulation" and related issues visibility, reversible and incremental actions, pointing and moving, continuous display. Provide an example of direct manipulation interface and comment where, and in what form are these notions used.**

Interface behaves as if interaction was **through real-world object**, not an abstract system. It gives **feeling working directly** on the task. They are almost always **based on a metaphor.** Visibility of the objects of interest. Rapid, reversible, incremental actions, manipulation by pointing and moving. Immediate and continuous display of results. I see objects I can manipulate. I move objects by pointing and moving. Displaying immediate result. (moving bar of changing brightness changes brightness immediately, up and down, up and down…) Another example is Solitaire. Dragging cards to move them, clicking them to flip. On the “table’ there are only card, objects I can manipulate, nothing more.

**Screen design can be guided by CRAP (contrast, repetition, alignment, proximity). What these principles enforce and guarantee? Show their application on a given screen snapshot.**

* **Contrast –** different things look differently, it’s based-on hierarchy and logic
* **Repetition –** using same style creates unified look between screens and within screen itself; using rules for placement of objects in whole system
* **Alignment –** visual alignment of elements to create order and visual flow. It’s easy for user to look for something, because it’s ordered.
* **Proximity –** closeness; grouping elements with similar meaning or type together.

They guarantee bigger visual order, it’s easier for user to look and find things on the screen. They create more pleasant viewing experience.

**CRAP (contrast, repetition, alignment, proximity) may be supported by grids. Compare given screen snapshots and criticize the design using CRAP&grid principles.**

Principles of grid is similar to the alignment. TO DO.

# Cognitive models

**Describe a general model of human processing of external information. Explain the cooperation of perceptual, cognitive and motor systems during interaction. Discuss how the parameters memory capacity, decay, representation and processing cycle time influence interaction. Explain the model with a given example.**

Sensory register – perception well…

There are important parameters as **memory** capacity, **decay**, **representation, processing** cycle time.

**Reaction time is closely related to decision-making. What are the factors influencing decision time? Explain Hick-Hyman law of reaction time.**

Decision time is influenced by **number of possible alternative actions** that could be selected.

Hick-Hyman law shows a logarithmic increase in reaction time **RT** as the number of possible stimulus-response alternatives **N** increases. Humans process information at a constant rate.

RT = a + b\*logn\_2N

**Explain Fitt's law, give examples.**

Time = a + b \* log\_2(D/S+1)

Movement time **increases** as the **distance** to the target **increases.**Movement time **decreases** as the **size** of target increases.  
It’s usually used for determining **best case** for new kinds od **input methods**.

**Compare KLM and GOMS models. Comment their usage in practice.**

**KLM**. How to make a KLM. Listing specific actions user does to perform task.

* **k**eystrokes and **b**utton presses
* mouse movements/ **p**ointing
* hand movements between keyboard and mouse / **h**oming
* **d**rawing
* system **r**esponse time (if it makes user wait)

Add **m**ental operators. Assign execution time to steps and then sum the times.

KLM only provides execution time and operator sequence. KLM is **cheap** method for evaluation of usability. It counts how many actions it takes to accomplish given task; low level model;

**GOMS.**

* **Goal** – what the user wants to achieve
* **Operator –** elementary perceptual, motor or cognitive act
* **Method –** a **series** of operators that forms a procedure for doing something
* **Selection rule –** how the user decides between methods

We use GOMS analysis **for comparing UI** designs, **profiling** and building a **help system**. (GOMS modelling makes user tasks and goals **explicit,** can suggest questions user will ask and answers)

GOMS is higher level than KLM. Input is detailed description of UI and tasks. (tasks should be **goal** centred). Output is many quantitative and qualitative measurements. It creates hierarchy in evaluated tasks. Disadvantage of GOMS: not so good at predicting errors; takes a long time to conduct analysis; whole may not be the sum of the parts; not as easy as heuristic analysis. It’s not used often.

# Heuristic evaluation

**For evaluating interfaces various criteria may be used, such as following heuristics:**

* **simple and natural dialog**
* **user's language**
* **minimization of memory load**
* **consistency**
* **feedback**
* **clearly marked exits**
* **shortcuts**
* **positive error solution**
* **offered help**

**Evaluate a selected application given several screen snapshots.**

****