ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/224927543

Human-Computer Interaction

Chapter · January 2004

CITATIONS
958

READS
179,659

4 authors, including:

Alan Dix
University of Birmingham
444 PUBLICATIONS **11,363** CITATIONS

SEE PROFILE

Gregory Abowd
Georgia Institute of Technology
404 PUBLICATIONS **34,925** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project Inferring Mood Instability on Social Media by Leveraging Ecological Momentary Assessments View project

Project S'Cool View project

All content following this page was uploaded by Alan Dix on 23 May 2014.

The user has requested enhancement of the downloaded file.

Human-Computer Interaction

Alan Dix, Janet Finlay, Gregory D. Abowd, Russell Beale

February 13th, 2005

Contents

Preface

A summary by Luuk van der Knaap. This summary is meant to be used beside the original text, not instead of it. It may freely be downloaded for personal use. However, if you intent to use the summary for other means than stricktly personal, you are kindly asked to contact me in advance. Download-file and contact information are available through my website: http://www.luukvanderknaap.tk.

The structure of the summary corresponds 1:1 to the structure of the book, except for the original summaries being left out.

Chapter 1

The human

1.1 Introduction

In 1983, Card, Moran and Newell described the Model Human Processor: a simplified view of the human processing involved in interacting with computer systems. MHP comprises 3 subsystems: the perceptual system, the motor system and the cognitive system. Each of them has a processor and memory. MHP also includes a number of Principles of operation which dictate the behavior of the system under certain conditions.

1.2 Input-output channels

In interaction with a computer, the human input is the data output by the computer vice versa. Input in humans occurs mainly through the senses and output through the motor controls of the effectors. Vision, hearing and touch are the most important senses in HCI. The fingers, voice, eyes, head and body position are the primary effectors.

1.2.1 Vision

Visual perception can be divided in 2 stages: the physical reception of the stimulus from the outside world, and the processing and interpretation of that stimulus.

The eye is a mechanism for receiving light and transforming it into electrical energy. Light is reflected from objects in the visual field and their image is focussed on the back of the eye, where it is transformed into an electrical signal and passed to the brain. The most important components are the cornea and lens and the retina with the blind spot and photoreceptors: rods and cones, located on the fovea. Rod are highly sensitive to light and usable under low illumination, but do not distinguish fine details. The cones are less sensible to light and can distinguish color.

The eye can perceive size and depth using the visual angle. If two objects are at the same distance from the eye, the larger one will have a larger visual angle. Similarly, if two objects of the same size are at different distances from the eye, the furthest one will have the smaller visual angle. The visual angle is given in degrees or minutes of arc (1 degree = 60 minutes of arc, see also figure 1.2).Visual acuity is the ability of a person to perceive small details. If the visual angle is to small, the detail will not be perceived. The minimum visual angle is approximately .5 seconds of arc. However, according to the law of size constancy, our perception of size relies on more factors than the visual angle, for example, the perception of depth. Depth can be perceived through various cues, e.g. indications in the visual context about an object's distance and familiarity with the size of the object. Perception of size and depth are highly intertwined.

Perception of brightness is a subjective reaction to levels of light emitted by an object: luminance Contrast is related to luminance, since it is the function of the luminance of the object and the background. Visual acuity increases with increased luminance. However, on screen, the flicker also increases with the luminance. The eye perceives color because the cones are sensitive to light of different wavelengths. It should be reminded that 3-4% of the fovea is sensitive to blue, making blue acuity lower.

The context in which an object appears allows our expectations to clearly disambiguate the interpretation of the object. In class, the example B/13 is used to illustrate this. however, it can also create optical illusions, for example in the Muller-Lyer illusion (fig 1.6).

Reading, finally, consists of several stages. First, the visual pattern of the word is perceived. Second, it is decoded with reference to an internal representation of language. Finally, the word is processed as part of the sentence or phrase using syntactic and semantic analysis. During the first two stages, the eyes make saccades (jerky movements), followed by fixations. The eye moves both forwards and backwards over the text, called regressions, which is increased when the text is more complex.

1.2.2 Hearing

The ear receives vibrations on the air and transmits them through various stages to the auditory nerves. The ear compromises 3 sections, the outer ear (pinna and auditory canal), middle ear (tympanic membrane (with ossicles) and cochlea) and inner ear (with cilia). The inner ear is filled with cochlean liquid. The sound waves are transmitted to the liquid using the ossicles. The vibrations, now in the liquid, bend the cilia which releases a chemical transmitter. The transmitter causes impulses in the auditory nerves. The human ear can hear frequencies from 20 Hz to 15 kHz. The sound we perceive is (selectively) filtered, which is illustrated by the cocktail party effect: we can notice our name spoken out in a noisy room.

Sound (vibrations) have a number of characteristics. The pitch is the frequency of the sound. The higher the frequency, the higher the sound. The loudness corresponds to the amplitude of the sound. Timbre relates to the type of the sound, independent of frequency and amplitude.

1.2.3 Touch

The apparatus of touch (haptic perception) is not localized. Stimuli are received through the skin, which contains various types of sensory receptors. Mechanoreceptors, responding to pressure, are important in HCI. There are 2 kinds of

MR's: Rapidly adapting mechanoreceptors, responding to immediate pressure as the skin is intended. They stop responding as continuous pressure is applied, to which slowly adapting mechanoreceptors respond. Some areas of the body have greater sensitivity/acuity than others. This can be measured using the two-point threshold test.

A second aspect of haptic perception is kinesthesis: awareness of the position of the body and limbs, due to receptors in the joints. There are 3 types: rapidly adapting (respond when moving limb in direction), slowly adapting (respond to movement and static position) and positional receptors (only responding to static positions).

1.2.4 Movement

When making movements, a stimulus is received through the sensory receptors and transmitted to the brain. After processing, the brain 'tells' the appropriate muscle to respond. The movement time is dependent on the physical characteristics of the subjects. The reaction time varies according to the sensory channel through which the stimulus is received.

Accuracy is a second measure of motor skill. A fast respond does not always mean a less accurate response. The time taken to hit a target is a function of the size of the target and the distance that has to be moved. This is formalized in Fitts' law, which is commonly written as:

$$Movementtime = a + b\log_2(distance/size + 1)$$

where a and b are empirically constants.

1.3 Human memory

We can distinguish 3 types of memory: sensory buffers, short-term memory (or working memory) and long-term memory.

1.3.1 Sensory memory

The sensory memories act as buffers for stimuli received through each of the senses: iconic memory for vision, echoic memory for sounds and haptic memory for touch. These memories are constantly overwritten by new information coming in on these channels. Information is passed from the sensory memory into short-term memory by attention, filtering the stimuli to those that are at that moment of interest (arousal, or shift of attention).

1.3.2 Short-term memory

STM is used to store information which is only required fleetingly. STM can be accessed rapidly, however, also decays rapidly. It has a limited capacity. Miller stated the 7+/-2 rule, which means that humans can store 5-9 chunks of information. Chunks can be single items or groups of items, like 2 digits of a telephone number grouped together. Patterns can be useful as aids to memory.

The recency effect suggests that STM recall is damaged by interferences of other information. Long-term memory is not effected.

1.3.3 Long-term memory

LTM differs from STM in various ways. It has an unlimited capacity, a slow access time and forgetting occurs more slowly or not at all. Information is stored here from the STM through rehearsal. There are 2 types of LTM: episodic memory and semantic memory. Episodic memory represents our memory of event and experiences in a serial form. Semantic memory is a structured record of facts, concepts and skills that we have acquired, derived from the episodic memory.

According to the semantic network model, the semantic memory is structured as a network (fig 1.11). The more general the information is, the higher is the level on which it is stored. This allows us to generalize about specific cases. The connections in the network are made using associations. There are other models about the organization of our LTM. Structured representations like frames and scripts, for example, organize information into data structures. Frames have slots to add attribute values. A script comprises a number of elements, which, like slots, can be filled with appropriate information. Another model is the production system, which holds IF-THEN rules: if information coming into the STM matches one of the conditions in the LTM, the appropriate action is executed.

There are 3 main activities related to LTM: storage of information, forgetting and information retrieval.

- **Storage:** The rehearsal of a piece of information from the STM stores it in the LTM. If the total learning time is increased, information is remembered better (total time hypothesis). However, the learning time should be well spread (distribution of practice effect). But repetition alone is not enough: information should be meaningful and familiar, so it can be related to existing structures and more easily incorporated into memory.

- **Forgetting:** There are 2 main theories of forgetting: decay and interference. Decay suggests that information held in LTM may eventually be forgotten. Jost's Law states that if 2 memory traces are equally strong at the same time, the older one will be more durable. Information, however, can also be lost through interference: if we acquire new information, it causes the loss of old information: retroactive interference. It is also possible that the older information interferes with the newly acquired information: proactive inhibition. Forgetting is affected by emotional factors too.

- **Retrieval:** There are 2 types of information retrieval: recall and recognition. In recall the information is produced from memory. It can be facilitated by providing cues, e.g. the category in which the information may be placed. In recognition, the presentation of the information provides the knowledge that the information has been seen before.

1.4 Thinking: reasoning and problem solving

Thinking can require different amounts of knowledge. Some thinking activities are very directed and the knowledge required is constrained. Others require vast amounts of knowledge from different domains. Thinking can be divided in reasoning and problem solving.

1.4.1 Reasoning

Reasoning is the process by which we use the knowledge we have to draw conclusions or infer something new about the domain of interest. There are different types of reasoning: deductive, inductive and abductive.

- **Deduction:** Deductive reasoning derives the logically necessary conclusion from the given premises. The logical conclusion does not have to correspond to our notion of truth. The human deduction is weak at the points where truth and validity clash.
- **Induction:** Inductive reasoning is generalizing from cases we have seen to infer information about cases we have not seen. In practise, induction is used to fill in missing details while reasoning.
- **Abduction:** Abduction reasons from a fact to the action or state that caused it. Abduction is used to derive explanations for the events we observe.

1.4.2 Problem solving

Problem solving is the process of finding a solution to an unfamiliar taste, using (adapting) the knowledge we have. There are different views on problem solving:

- **Gestalt theory:** The Gestalt theory states that problem solving is both productive and reproductive; insight is needed to solve a problem. However, this theory has not been accepted as 'sufficient'.
- **Problem space theory:** The problem space comprises problem states and problem solving involves generating these states using legal state transition operators. People use these to move from the initial state to the goal state. Heuristics (e.g. Means-end analysis) are employed to select the right operators.
- **Use of analogy:** Problems are solved by mapping knowledge relating to a similar known domain to the new problem: analogical mapping.

1.4.3 Skill acquisition

Experts often have a better encoding of knowledge: information structures are fine tuned at a deep level to enable efficient and accurate retrieval. According to the ATC model, these skills are acquired through 3 levels:

- The learner uses general-purpose rules which interpret facts about a problem. (slow, memory-demanding)

- The learner develops rules specific to the task, using proceduralization.
- The rules are tuned to speed up performance, using generalization.

1.4.4 Errors and mental models

There are different types of errors: changes in context of skilled behavior can cause errors. An incorrect understanding/model of a situation can cause errors too, because humans tend to create mental models, based on experience, which may differ from the actual situation.

1.5 Emotion

Emotion involves both physical and cognitive events. Our body responds biologically to an external stimulus and we interpret that in some way as a particular emotion. That biological response (affect) changes the way we deal with different situations and this has an impact on the way we interact with computer systems.

1.6 Individual differences

The principles and properties discussed apply to the majority of people, but humans are not all the same. Differences should be taken into account in the designs: divide the users in target groups, for example.

1.7 Psychology and the design of interactive systems

1.7.1 Guidelines

General design principles and guidelines (straightforward or complex) can be and have been derived from the above discussed theories. See chapter 7.

1.7.2 Models to support design

Psychological analysis has led to the development of analytic and predictive models of user behavior. See chapter 12.

1.7.3 Techniques for evaluation

Psychology provides a range of empirical techniques which we can employ to evaluate designs and systems. See chapter 9.

Chapter 2

The computer

2.1 Introduction

Interaction (with or without computer) is a process of information transfer.

The diversity of devices reflects the fact that there are many different types of data that may be entered into and obtained from a system, as there are many different users. In the early days, batch processing was common: a large mass of information was dumped into and processed by the computer. Nowadays, computers respond within milliseconds and computer systems are integrated in many different devices.

2.2 Text entry devices

2.2.1 The alphanumeric keyboard

The vast majority of keyboards have a standardized layout, known by the first six letters on the top row: QWERTY. The non-alphanumeric keys are not standardized. This layout is not optimal for typing, but dates from the time of mechanical limitations of the typewriter. Today, the keys can also be arranged in alphabetic order (the alphabetic keyboard), but this does not improve typing performance. The DVORAK keyboard does, placing the keys in a different order on a similar layout as found on the QWERTY keyboards. The layout minimized the stretch of fingers and the use of weak fingers, reducing fatigue and increasing typing speed (10-15%).

2.2.2 Chord keyboards

On chord keyboards, only a few keys are used. Letters are produces pressing multiple keys at once. They are smaller than conventional keyboards and have a short learning time.

2.2.3 Phone pad and T9 entry

The numeric keys on a cellphone can be pressed more than once to enter letters. Most phones have 2 keypad modes: a numeric and an alphabetic mode. Most phones have additional modes for entering (initial) capitals. On modern phones you can also find the T9-algorithm. This uses a large dictionary to disambiguate words by typing the relevant letters keys once.

2.2.4 Handwriting recognition

Current technology is still fairly inaccurate and makes a lot of mistakes, partly due to the enormous differences between people's handwriting. HR deals mostly worth stroke information: the way in which the letter is drawn, not the letter itself. Therefore, online recognition is most accurate. HR has the advantage of size and accuracy over small keyboards and are therefore often used in mobile computing.

2.2.5 Speech recognition

The performance of speech recognition is still relatively low, even for a restricted vocabulary. Adjusting the system for use with natural language gives birth to even more problems: the 'errors' in natural language use, different voices, emotions and accents etc. This means the system has to be tuned for each different user. SR can be used in 3 scenarios: as an alternative text entry device, replacing the keyboard in the current software, with new software especially designed for SR and in situations where the use of keyboards is impractical or impossible.

2.3 Positioning, pointing and drawing

2.3.1 The mouse

The mouse is an indirect input device, because a transformation is required to map from the horizontal nature of the desktop to the vertical alignment of the screen. Invented in 1964 by Engelbart, his mouse used 2 wheels that slid across the desktop and transmitted x, y-coordinates to the computer. There have been experiments with foot-controlled mice.

2.3.2 Touchpad

Touchpads are touch-sensitive tablets, operated by sliding the finger over it and are mostly used in notebook computers. Performance can be increased using accelerators.

2.3.3 Trackball and thumbwheel

A trackball is an upside-down mouse: instead of moving the device itself, the ball is rolled to move the cursor. Trackballs are often used by RSI users. Thumbwheels (in 2 dimensions) offer less usability because they can only manipulate the horizontal and vertical movement of the cursor. 1-dimensional thumbwheels are often included on the normal mice the enhance the scrolling.

2.3.4 Joystick and keyboard nipple

There are two types of joysticks: absolute sticks, in which the position of the cursor corresponds to the position of the joystick in its base, and isometric sticks, in which the pressure on the stick (in a certain direction) controls the velocity of the cursor in that direction. Keyboard nipples are tiny joysticks that are sometimes used on notebook computers.

2.3.5 Touch-sensitive screens (touchscreens)

Touchscreens detect the position of the user's finger or stylus on the screen itself and are therefore very direct. They work by having the finger/stylus interrupting a matrix of light beams, making capacitance changes on a grid overlaying the screen or by ultrasonic reflections. It is a direct device: no mapping is required. However the selection of small area's is difficult and intensive use can be tiring.

2.3.6 Stylus and lightpen

For more accurate positioning, systems with touch-sensitive surfaces often emply a stylus. An older technology for the same purpose is the lightpen, which emits radiation detected by the screen. A difficulty of this and other direct devices is that pointing obscures the display, making it more difficult to use in rapid successions.

2.3.7 Digitizing tablet

A device used for freehand drawing. A resistive tablet detects point contact between two separated conducting sheets. Magnetic, capacitive and electrostatic tablets use special pens. The sonic tablet requires no pad: an ultrasonic sound emitted by the pen is detected by 2 microphones.

2.3.8 Eyegaze

Eyegaze allows you to control the computer by looking at it, while wearing special glasses, head-mounted boxes etc. By tracking a laser beam's reflection in the eye, the direction in which the eye is looking is determined. The system needs to be tuned and is very expensive, but also very accurate.

2.3.9 Cursor keys and discrete positioning

For 2D-navigation, cursor keys can sometimes be preferable. The same goes for remote-controls and cellphones.

2.4 Display devices

2.4.1 Bitmap displays, resolution and color

A bitmap-base means that the display is made of a fixed number of dots or pixels in a rectangular grid. The color or intensity at each pixel is held by the computer's video card. The more bits per pixel, the more colors/intensities are possible.

Also is the resolution of the screen: the total number of pixels (in a 4:3-ratio) and the density of the pixels.

Anti-alinsing: softening the edges of line segments, blurring the discontinuity and making the juggles less obvious.

2.4.2 Technologies

In a CRT-monitor a stream of electrons is emitted from an electron gun, which is than focussed and directed by magnetic fields. As the beam hits the phosphor-coated screen, the phosphor is excited by the electrons and glows. Flicker can be reduced by increasing the scanning rate or by interlacing, in which odd lines are scanned first, followed by even lines.

In LCD's a thin layer of liquid crystals is sandwiched between two glass plates. External light passes through the top plate and is polarized. This passes through the crystal and is reflected back to the user's eye by the bottom plate. The polarization of each single crystal can be turned electronically.

2.4.3 Large displays and situated displays

There are several types of large displays. Some use gas-plasma technology and usually have a 16:9-ratio. Several smaller screens can also be places together in a video wall. Projectors are possible too, in two variants: projectors with 3 lenses (red, green and blue) can build a full-color image. LCD-projectors have a small screen, through which light is projected on a screen.

2.4.4 Digital paper

Thin flexible material that can be written to electronically, but keeps it's contents when removed from the power supply.

2.5 Devices for virtual reality and 3D interaction

2.5.1 Positioning in 3D

Changing from 2D to VR does not mean going to 3 degrees of freedom, but (sometimes) to 6, because except for moving in 3 dimensions, you can also roll, turn, twist etc.

Humans can use a 3D-environment with a 2D-device (mouse). The human mind is therefore capable of handling multiple degrees of indirection. A 3D-input device is the 3D-mouse, which has 6 degrees of freedom: 3 for position (x,y,z), 1 for pitch, yawn and roll. However, sometimes its better to use a dataglove: a lycra glove with fibers laid around the fingers, detecting the joint angles of the fingers and thumb.

The position of the head can be tracked using a VR-helmed, which can also display the 3D-world to each eye. With other devices, e.g. special clothing or a modified trampoline, the position and movement of the whole body can be tracked.

2.5.2 3D displays

3D can be displayed on normal screens using shadows, depth etc. It is also possible to generate the natural stereoscopic images for both eye positions and have them delivered to he eyes using a VR-helmed. Finally, users can enter a VR cave, where the VR world is projected around them. If the VR-system performances too slow, and there is a delay between movement and image, disorientation and sickness may occur.

2.6 Physical controls, sensors and special devices

2.6.1 Special displays

Except for CRT and LCD, there are numerous other display devices, e.g. LED's, ganges, dials and head-up displays.

2.6.2 Sound output

We do not yet know how to utilize sound in a sensible way to achieve maximum effects and information transference in HCI. However, by having sounds confirm a right action, we can speed up interaction.

2.6.3 Touch, feel and smell

Force feedback gives different amounts of resistance to an input device depending on the state of the virtual operation. Haptic devices are various forms of force, resistance and texture influencing our physical senses.

2.6.4 Physical controls

Not only the function of controls, but also the physical design is important and needs to suit the situation in which it is used: kitchen equipment, for example, needs controls that can be cleaned easily.

2.6.5 Environment and bio-sensing

There are many sensors in our environment monitoring our behavior. Their measurements range from temperature and movement to the user's emotional state.

2.7 Paper: printing and scanning

2.7.1 Printing

The most common printers nowadays are dot-based. In order of increasing resolution, familiar types are dot-matrix printers, ink-jet printers and laser printers.

2.7.2 Fonts and page description languages

Some printers print ASCII-characters and bitmaps 'by itself'. Many more complex documents are translated into suitable bitmaps by the computer. More sophisticated printers can accept a page description language, e.g. PostScript. The programming-language for printing includes standard-programming constructs, which means that less data has to be send to the printer in comparison to using a bitmap.

2.7.3 Screen and page

There are many differences (e.g. size, color depth, resolution etc.) between a paper print and a computer monitor, which causes problems when designing WYSIWYG-software. Especially the correct alignment of text (in different fonts) is difficult.

2.7.4 Scanners and optical character recognition

Scanners produce a bitmap image from a 'hard' original and can, using optical character recognition, transfer a page of text directly into a txt-file. There are 2 kinds of scanners: flat-bed (as in a copie machine) and hand-held (as in a fax machine, however the scanner has to be manually pulled over the paper). Scanners shine a beam of light at the page and record the intensity and color of the reflection. The resolution of the scanner can differ highly between different types.

2.8 Memory

2.8.1 RAM and short-term memory (STM)

Most current active information is held in the random access memory (RAM). RAM is volatile: contents are lost when the power is turned off. However, there are more expensive or low-power consuming memory techniques that can hold their contents when the power is off.

2.8.2 Disks and long-term memory (LTM)

There are 2 main techniques used in disks: magnetic disks (floppy, harddisk, tape) and optical disks. (CD-ROM/DVD). In comparison to RAM, the computers LTM is rather slow.

2.8.3 Understanding speed and capacity

The capacity of RAM is limited and therefore multitask-systems tend to swap background-running programs from RAM to the harddisk. When the program is fully activated it has to be swapped back, which can cause delays (von Neumann bottleneck).

2.8.4 Compression

Compression techniques can be used ti reduce the amount of storage required for text, bitmaps and video. In text, logical contructions in the sentence can be replaced by a short code. In video, differences between frames can be recorded instead of the whole frames. If fractal compression is used, the quality can even improve in the process.

2.8.5 Storage format and standards

The basic standard for text storage is the ASCII character codes, which assign to each standard printable character and several control characters an internationally recognized 7 bit code. UNICODE is an extended version of this system and can also code for foreign characters. However, this is all unformatted text. All editors which produce formatted texts have their own file format. Also for images there exists a wide range of formats.

2.8.6 Methods of access

Standard database access is by special key fields with an associated index. The user has to know the key before the system can find the information. Indices on databases are limited due to the storage costs, privacy and security. The user's mistakes in searching can be compensated by using forgiving systems, for example by matching a key to a database index which corresponds closely.

2.9 Processing and networks

2.9.1 Effects of finite processor speed

The processing speed of an interactive system can effect the user by being too slow (which can be avoided by using buffers) or too fast. The faults can be functional, in which the program does the wrong action. Slow responses from the system can also cause the so called cursor tracking and icon wars. If the system is too fast, the user will not have enough time to interpret the system's output.

2.9.2 Limitations on interactive performance

Several factors that can limit the speed of an interactive system. They can be:

- Computation bound: Make sure the user has an indication of the system's progress.
- Storage channel bound: Select the best fitting kind of memory and access technique.
- Graphics bound: The actual time of graphic operations can differ much from the estimates.
- Network capacity

2.9.3 Network computing

Networked systems have an effect on interactivity, because the large distances may cause a noticeable delay in response from the system. The actions of other users may also influence your own interaction with the connected computers.

Chapter 3

The interaction

3.1 Introduction

There are a number of ways in which the user can communicate with the system: batch input, direct manipulation, virtual reality etc.

3.2 Models of interaction

3.2.1 The terms of interaction

- Purpose of an interactive system: Aid the user in accomplishing goals from some application domain.
- Domain: An area of expertise and knowledge in some real-world activity.
- Tasks: Operations to manipulate the concepts of a domain.
- Goal: Desired output from a performed task.
- Intention: Specific action required to meet the goal.
- Task analyses: Identification of the problem space for the user of an interactive system in terms of domain, goals, intention and tasks.
- System's language: Core language, describes computational attributes of the domain relevant to the System state.
- User's language: Task language, describes psychological attributes of the domain relevant to the User state.
- System: Computerized application.

3.2.2 The execution-evaluation cycle

The plan formulated by the user is executed by the computer. When finished, the user evaluates the results and determines the further actions. Both execution and evaluation can be divided into the following subsections:

1. Establishing the goal

2. Forming the intention (more specific than goal)

3. Specifying the action sequence (based on intention)

4. Executing the action

5. Perceiving the system state

6. Interpreting the system state

7. Evaluating the system state with respect to the goals and intentions

- Gulf of execution: Difference between the user's formalization of the actions and the actions allowed by the system.
- Gulf of evaluation: Distance between the physical presentation of the system state and the expectation of the user.

3.2.3 The interaction framework

(Figure 3.1 & 3.2). On the user-side, communication is in task-language and on the system side, in core language. The user's formulation of the desired task needs to be articulated in the input-language. The task is phrased in terms of certain psychological attributes that highlight the important features of the domain for the user which, if mapped clearly onto the input language, simplify the articulation of a task. Direct manipulation can also facilitate the articulation.

The responses of the input are translated to stimuli for the system. Once a state transition has occurred within the system, the execution phase is completed and the evaluation begins by translating the system's responses into stimuli for the output component. Finally, the response from output is translated to stimuli for the user.

3.3 Frameworks and HCI (figure 3.3)

- Ergonomics: The user side of the interface, covering both input and output and the user's immediate context.
- Dialog design and interface styles.
- Presentation and screen design

3.4 Ergonomics

The study of the physical characteristics of the interaction.

3.4.1 Arrangement of controls and displays

Inappropriate placement of controls and displays can lead to inefficiency, frustration and sometimes dangerous situations.

Organization of controls:

- Functional: functionally related controls are grouped together.
- Sequential: Controls are organized to reflect the order of their use in a typical interaction.
- Frequency: The most often used controls can be accessed most easily.

3.4.2 The physical environment of the interaction

The system's design needs to fit the users size, position (sitting/standing), comfort and safety.

3.4.3 Health issues

- Physical position.
- Temperature.
- Lighting.
- Noise.
- Time.

3.4.4 The use of color (guidelines)

Color used in displays should be as distinct as possible and the distinction should not be affected by changes in contrast. Blue should not be used to display critical information. If color is used as an indicator, it should not be the only cue: additional coding information should be included. The colors should correspond to common conventions and user expectations. Color conventions are culture-determined.

3.4.5 Ergonomics and HCI

Ergonomics contribution to HCI is in determining constraints on the way we design systems and suggesting detailed and specific guidelines and standards. Ergonomic factors are in general well established and understood and are therefore used as the basis for standardizing hardware designs.

3.5 Interaction styles

3.5.1 Command line interface

CLI provides a means of expressing instructions to the computer directly, using function keys, single characters, abbreviations or whole-word commands. They are flexible (parameters) and can be combined to apply a number of tools to the same data. Commands should be remembered by the user, the CLI offers no ques.

3.5.2 Menus

A set of menu options available for the user is displayed on the screen. The user can select an option (recognition!) using either mouse or keyboard. The menus can be presented text-based and graphical.

3.5.3 Natural language

The ambiguity of natural language makes it very hard for a machine to understand. However, systems can be built to understand restricted subsets of a language, which is relatively successful.

3.5.4 Question/answer and query dialog

The user is asked a series of questions and so is led through the interaction step by step. These interfaces are easy to learn and use, but are limited in their functionality and power.

Query languages are used to construct queries to retrieve information from a database. They require specifications from the user in a strict syntax.

3.5.5 Form-fills and spreadsheets

Primarily used for data entry but can also be useful in data retrieval applications. Most form-filling interfaces assist the user during the interaction. Spreadsheets are a sophisticated variation of form filling. The user can enter and alter values and formulae in any order and the system will maintain consistency amongst the values displayed, ensuring that all formulae are obeyed.

3.5.6 The WIMP interface

Windows, icons, menus and pointers: the default interface style for the majority of computer systems today.

3.5.7 Point-and-click interfaces

The PCI is closely related to the WIMP-style: pointing and clicking are the only actions required to access information.

3.5.8 Three-dimensional interfaces

The simplest technique is where ordinary WIMP elements are given a 3D appearance. A more complex technique uses interfaces with 3D workspaces. The objects displayed are flat, but are displayed in perspective: they shrink when they are further away. The most complex 3D-workspace is virtual reality.

3.6 Elements of the WIMP-interface

The elements of the WIMP interfaces are called widgets: the toolkit for interaction between user and system.

3.6.1 Windows

Windows are areas on the screen that behave as if they were independent terminals in their own right: it can contain any information and can be resized or moved around. Some systems allow windows within windows.

3.6.2 Icons

An icon is a small picture used to represent a closed window.

3.6.3 Pointers

The different shapes of the cursor are often used to distinguish modes. Cursors are also used to give information about the systems activity (hour-glass). In essence pointers are nothing more than small bitmap images with a hotspot: the locatin to which they point.

3.6.4 Menus

A menu presents a choice of operations or services that can be performed by the system at a given time. Menus provide information cues in the form of an ordered list of operations that can be scanned and selected by using the pointer. There are two types: pop-up menus, that represent context-dependent options, and pull-down menus, that are always visible. The right grouping of the menu-items is the most difficult part of designing a menu.

3.6.5 Buttons

Buttons are individual and isolated regions within a display that can be selected by the user to invoke a specific action. Radio buttons are used for selecting one option from a group. When there are multiple options selectable, check boxes are more common.

3.6.6 Toolbars

Mostly equivalent to menus, except for that a toolbar can also hold buttons.

3.6.7 Palettes

Palettes are mechanisms for making the set of possible modes and the active mode visible to the user (collection of icons).

3.6.8 Dialog boxes

Dialog boxes are information windows used by the system to bring the user's attention to some important information, possibly an error or a warning used to prevent a possible error, or as a subdialog for a very specific task.

3.7 Interactivity

Interactivity is essential in determining the 'feel' of a WIMP environment. In WINP environments, the user takes the initiative, with many options and many applications simultaneously available. The exceptions to this are the preemptive parts of the interface, where the system can take the initiative for various reasons (e.g. the need for specific information). In modern systems, preemptive parts should be avoided as much as possible.

3.8 The context of the interaction

The presence of other people in a work environment affects the performance of the worker in any task, for example, by 'competition-behaviour'. However, when it comes to acquisition of new skills, the presence of others can inhibit performance (fear of failure). In order to perform well, users must be motivated. If the (computer) system makes it difficult for the user to perform a certain task, he might get frustrated and his productivity could drop. The user may also lose motivation if a system is introduced that does not match the actual requirements of the job to be done. In that case the user will reject the system, be resentful and unmotivated or adapt the intended interaction to his own requirements. A well designed system, however, may also work motivating on the user.

3.9 Experience engagement and fun

It is no longer sufficient that users can use a system, they have to want to use it as well.

3.9.1 Understanding experience

The sense of flow occurs when there is a balance between anxiety and boredom. In education, there is the zone of proximal development, in which you do things with some support that you cannot do yourself. Learning is optimal in this zone.

3.9.2 Designing experience

Nothing interesting in this subsection ;).

3.9.3 Physical design and engagement

Designers' constraints:

- Ergonomic
- Physical
- Legal and safety
- Context and environment
- Aesthetic

- Economic
- Fluidity: The extent to which the physical structure and manipulation of the device naturally relate to the logical functions it supports.

3.9.4 Managing value

If we want people to want to use a device or application, we need to understand their personal values. In the development of software we should take into account that the user wants to see the gains from the new technique as soon as possible and not after a long time of using it.

Chapter 4

Paradigms

4.1 Introduction

Two questions for the designer:

- How can an interactive system be developed to ensure its usability?
- How can the usability of an interactive system be demonstrated or measured?

One approach to answer these questions is by means of example, in which successful interactive systems are commonly believed to enhance usability and, therefore, serve as paradigms for the development of future products.

4.2 Paradigms for interaction

4.2.1 Time sharing

Time sharing means that a single computer could support multiple users. The introduction of time sharing meant the end of batch-processing, in which complete jobs processed individually.

4.2.2 Video display units

The earliest applications of display screen images were developed in military applications. However, it took until 1962 to develop Sketchpad, a simulation language for visual models. It demonstrated that computers could be used to create visual models of abstractions.

4.2.3 Programming toolkits

The idea of building components of a computer system that will allow you to rebuild a more complex system is called bootstrapping and has been used to a great extent in all of computing. The power of programming toolkits is that small, well understood components can be composed in fixed ways in order to create larger tools. Once these larger tools become understood, they can continue to be composed with other tools, and the process continues.

4.2.4 Personal computing

As technology progresses, it is now becoming more difficult to distinguish between what constitutes a personal computer or workstation and what constitutes a mainframe. Some examples of the first personal-computing applications are LOGO and NLS.

4.2.5 Window systems and the WIMP interface

Humans are capable of doing multiple tasks at the same time and therefore frequently change their 'train of thoughts'. The personal computer needs to be just as flexible in order to be an effective dialog partner. The modern PC is, and by using windows it can present messages to the user in the context of their task, so the user is able to distinguish the messages from different tasks.

4.2.6 The metaphor

Metaphors are used quite successful to teach new concepts in terms of ones which are already understood. This also works with computers: many of the tasks on a computer are presented as metaphors of tasks in an office environment. However, the metaphor is inadequate for promoting (and even gets in line with) a full understanding of the computer. Furthermore, metaphors portray a cultural bias and therefore it is difficult to create a metaphor that is internationally understood.

4.2.7 Direct manipulation

Features:

- Visibility of the objects of interest.
- Incremental action at the interface with rapid feedback on all actions.
- Reversibility of all actions, so that users are encouraged to explore without severe penalties.
- Syntactic correctness of all actions, so that every user action is a legal operation.
- Replacement of complex command languages with actions to manipulate directly the visible objects.

Psychological approach: model-world metaphor (direct engagement):
In a system built on the model-world metaphor, the interface is itself a world where the user can act, and which changes state in response to user actions. The world of interest is explicitly represented and there is no intermediary between user and world. Appropriate use of the model-world metaphor can create the sensation in the user of acting upon the objects and task domains themselves: direct engagement.

From the user's perspective, the interface is the system. A consequence of DM is that there is no longer a clear distinction between input and output. Widgets become interaction objects, with input and output. A way to bring DM into practice is through WYSIWYG-interfaces.

4.2.8 Language versus action

The interface can be seen as the mediator between the user and the system. The user gives instructions to the interface and it is then the responsibility of the interface to see that those instructions are carried out. The user-system communication is by means of indirect language instead of direct actions.

Two meaningful interpretations to this:

- Users are required to understand how the underlying system functions and the interface as interlocutor need not perform much translation.
- Users are not required to understand the underlying system: the interface serves a more active role by translating and interpreting the user's input to correct system commands.

4.2.9 Hypertext

Hypertext is based on the memex-technique: a storage and retrieval apparatus used to link different texts together. The name hypertext points to the nonlinear structure of the information.

4.2.10 Multi-modality

A multi-modal interactive system is a system that relies on the use of multiple human communication channels. Each different channel for the user is referred to as a modality of interaction. not all systems are multi-modal, however. Genuine multi-modal systems rely to a greater extent on simultaneous use of multiple communication channels for both input and output.

4.2.11 Computer-supported cooperative work

The main distinction between CSCW systems and interactive systems designed for a single user is that designers can no longer neglect the society within which many users operate. CSCW systems are built to allow interaction between humans via the computer and so the needs of the many must be represented in the one product.

CSCW can be synchronous (users have to be online at the same time) and asynchronous (users don't have to be online at the same time).

4.2.12 The world wide web

WWW is not the same as internet. Internet is just the connection between different computers. WWW is the graphic top-layer which is very popular for exchanging information in the HTML-markup notation. It even took the introduction of the WWW to make the internet popular and currently the web is one of the major reasons for buying computers.

4.2.13 Agent-based interfaces

Software agents perform actions for the user. The major problem is to specify the users task correctly to the user in a suitable language. Some agents use AI to learn from the user. Some agents have an embodiment: a representation in the interface (e.g. an icon).

4.2.14 Ubiquitous computing

The intention of UC is to create a computing infrastructure that permeates our physical environment so much that we do not notice the computer any longer. On a small scale, this is already put into practice (watches, PDAs etc.), however a major breakthrough will still take some time.

4.2.15 Sensor-based and context-aware interaction

Sensor based interaction is simply the future-idea of the computer adjusting to our behavior and performing on background using the information gathered from sensors.

In context-aware computing the interaction is implicit than in ordinary interface use. The computer or sensor-enhanced environment is using heuristics and other semi-intelligent means to predict what would be useful for the user. CA-applications should follow the principles of appropriate intelligence:

- Be right as often as possible, and useful when acting on these correct predictions.
- Do not cause inordinate problems in the event of an action resulting from a wrong prediction.

Chapter 5

Interaction design basics

5.1 Introduction

Interaction design is about how the artifact produced is going to affect the way people work: the design of interventions.

5.2 What is design?

- Design: achieving goals within constraints.
- Goals: the purpose of the design we are intending to produce
- Constrain: the limitations on the design process by external factors
- Trade-off: choosing which goals or constraints can be relaxed so that others can be met.

5.2.1 The golden rule of design

Understand your material: computers (limitations, capacities, tools, platforms) and people (psychological, social aspects, human error)

5.2.2 To err is human

It is the nature of humans to make mistakes and systems should be designed to reduce the likelihood of those mistakes and to minimize the consequences when mistakes happen.

5.2.3 The central message: the user

During design, always concentrate on the user.

5.3 The process of design (see also fig. 5.1)

- Requirements: Through observations and interviews, the features of the system to be designed are mapped.

- Analysis: Through various methods, the gathered requirements are ordered to bring out key issues.
- Design: Various design guidelines help you to move from what you want to how to do it. They are discussed in other chapters and sections.
- Iteration and prototyping: Try out early versions of the system with real users.
- Implementation and deployment: writing code, documentation and make hardware.

5.4 User focus

Once more: gather as much information as possible about the future users of the system. Terminology:

- Stakeholders: people affected directly or indirectly by a system
- Participatory design: bringing a potential user fully into the design process
- Persona: rich picture of an imaginary person who represents your core user group

5.5 Scenarios

Scenarios are stories for design: rich stories of interaction sometimes illustrated with storyboards.

5.6 Navigation design

5.6.1 Local structure

Much of interaction involves goal-seeking behavior, because users do not know the system entirely. Therefore, the interface should always make clear:

- where you are
- what you can do
- where you are going/what will happen in terms of the interaction or state of the system. Furthermore:
- Icons are not self-explanatory: they should be explained!
- The different meaning of the same command in different modes should be clear.
- The system should give feedback about the effect of an action. In most information systems, it is as essential to know where you have been.

5.6.2 Global structure - hierarchical organization

Overall structure of an application: the way the various screens, pages or physical device states link to one another. This can be done using hierarchy: humans tend to be better at using this structure, as long as the hierarchy does not go to deep.

5.6.3 Global structure - dialog

Dialog: the pattern of non-hierarchical interaction occurring when the user performs a certain action, e.g. deleting a file.

5.6.4 Wider still

- Style issues: we should conform to platform standards
- Functionality issues: the program should conform to standard functions.
- Navigation issues: we may need to support linkages between applications

5.7 Screen design and layout

5.7.1 Tools for layout

- Grouping and structure: if things logically belong together, then we should normally visually group them together.
- Order of groups and items: the order on the screen should follow the natural order for the user.
- Decoration: decorations can be used to emphasize grouping.
- Alignment: the proper use of alignment can help the user to find information in lists and columns quickly.
- White space: white space can be used to separate blocks, highlight structures etc.

5.7.2 User actions and control

For entering information, the same criteria dictate the layout. It is also very important that the interface gives a clear clue what to do. A uniform layout is then helpful. Affordance (things may (by their shape for example) suggest what to do with them) is, sometimes, helpful as well. It is, however, not appropriate to depict a real-world object in a context where its normal affordances do not work.

5.7.3 Appropriate appearance

The way of presenting information on screen depends on the kind of information, the technologies available to present it and the purpose for which it is used. We have an advantage when presenting information in an interactive system in that it is easy to allow the user to choose among several representations, thus making it possible to achieve different goals.

In an ideal design, the interface is both usable and aesthetically pleasing. However, the looks of the interface should never come to the disadvantage of the usability. This is mostly the case with the excessive use of color and 3D.

Localization/internationalization: the process of making software suitable for different cultures and languages.

5.8 Iteration and prototyping(hill-climbing approach, local & global maxima, see also fig 5.14)

- Formative evaluation: intended to improve designs.
- Summative evaluation: verify whether the product is good enough.

In order for prototyping methods to work, you need to understand what is wrong and how to improve it, and you also need a good starting point. If the design is very complex, it is sometimes wise to start with various alternatives and to drop them one by one during the design process.

Chapter 6

HCI in the software process

6.1 Introduction

- Software engineering: the subdiscipline that adressses the management and technical issues of the development of software systems.
- Software life cycle: the activities that take place form the initial concept for a software system up until its eventual phasing out and replacement.

HCI aspects are relevant within all the activities of the software life cycle.

6.2 The software life cycle

6.2.1 Activities in the life cycle (fig 6.1)

- Requirements specification: capture a description of what the eventual system will be expected to provide. Requirements, formulated in natural language, are translated to a more formal and unambigious language.
- Architectural design: how does the system provide the services expected from it. In this part, the system is decomposed into components that can be brought in from existing products or that can be developed from scratch
- Detailed design: a refinement of the component description provided by the architectural design, made for each component seperately.
- Coding and unit testing: implementing the detailed design in an executable programming language and testing the different components.
- Integration and testing: integrating the different components into a complete system and testing it as a whole. Sometimes also certify the system according to ISO-standards.
- Maintenance: all the work on the system after the system is released.

6.2.2 Validation and verification

Verfification (designing the thing right) will most often occur within a single life-cycle activity or between two adjacent activities. Validation of a design (designing the right thing) demonstrates that within the various activities the customer's requirements are satisfied. Because verification proofs are between rather formal languages, the proofs are rather formal too. The validation proof, however, is not: there is a gap between the real world and structured design, known as the formality gap. The consequence is, that there is always a certain subjectivity involved with validation.

6.2.3 Management and contractual issues

In management, the technical view on the software lifecycle is sometime insufficient: a much wider perspective must be adopted which takes into account the marketability of a system, it training needs, the availability of skilled personnel or possible subcontractors, and other topics outside the activities for the development if the isolated system.

In managing the development process, the temporal relationship between the various activities is more important, as are the intermediate deliverables which represent the technical content, as the designer must demonstrate to the customer that progress is being made. The technical perspective of the life cycle is described in stages of activity, whereas the managerial perspective is described in temporally bound phrases: input and output of documentation.

6.2.4 Interactive systems and the software life cycle

The life cycle for development described above presents the process of design in a somewhat pipeline order. In reality, the actual process is iterative: work in one design activity affects work in any other activity both before or after it in the life cycle. All of the requirements for an inteactive system cannot be determined from the start. During the design process, the system is made 'more usable' by having the potential user test the prototypes and observe his behaviour. In order to do this, clear understanding of human task performance and cognitive processes is very important.

6.3 Usability engineering

The emphasis for usability engineering is in knowing exactly what criteria will be used to judge a product for its usability. In relation to the software life cycle, one of the important features of usability engineering is the inclusion of a usability specification, forming part of the requirement specification, that concentrates on features of the user-system interaction which contribute to the usability of the product. Various attribute of the systemare suggested as gauges for testing the usability. For each attribute, six items are defined to form the usability specification of that attribute:

- Measuring concept: makes the abstract attribute more concrete by describing it in terms of the actual product.
- Measuring method: states how the attribute will be measured.

- Now level: indicates the value for the measurement with the existing system.
- Worst case: the lowest acceptable measurement for the task.
- Planned level: the target for the design.
- Best case: the level which is agreed to be the best possible measurement given the current state of development tools and technology.

6.3.1 Problems with usability engineering

The major feature of usability engineering is the assertion of explicit usability metrics early on in the design process which can be used to judge a system once it is delivered. The problem with usability metrics (see also table 6.4) is that they rely on measurements of very specific user actions in very specific situations. At early stages of design, the designers do not yet have the information to set goals for measured observations. Another problem is that usability engineering provides a means of satisfying usability specifiations and not necessarily usability: the usability metrices must be interpreted correctly.

6.4 Iterative design and prototyping

Iterative design: a purposeful design process which tries to overcome the inherent problems of incomplete requirement spcification by cycling through several designs, incrementally improving upon the final product with each pass. On the technical side, this is described by the use of prototypes. There are 3 main approaches of prototyping:

- Throw-away: the knowledge gained from the prototype is used in the final design, bu the prototype is discarted (fig 6.5).
- Incremental: the final product is released as a series of components that have been prototyped seperately (fig 6.6).
- Evolutionary: the prototype is not discarted but serves as a basis for the next iteration of the design (fig 6.7).

Prototypes differ according to the amount of functionality and performance they provide relative to the final product. The importance lies in its projected realism, since they are tested on real users. Since providing realism in prototypes is costly, there are several problems on the management side:

- Time: prototyping costs time which is taken away from the real design. Therefore, there are rapid-prototyping techniques.
- Planning
- Non-functional features: some of the most important features, as safety and reliability, cannot be tested using a prototype.
- Contracts: Prototyping cannot form the basis for a legal contract and must be supported with documentation.

6.4.1 Techniques for prototyping

- Storyboards: a graphical depiction of the outward appearance of the intended system, without any accompanying system functionality.
- Limited functionality simulations: Programming support for simulations means a designer can rapidly build graphical and textual interaction objects and attach some behaviour to those objects, which mimics the system's functionality. There are many techniques to build these prototypes. A special one is the Wizard of Oz technique, in which the system is controlled by human intervention.
- High-level programming support: High-level programming languages allow the programmer to abstract away from the hardware specifics and thinkin terms that are closer to the way the input and output devices are perceived as interaction devices. This technique can also be provided by a user interface management system, in which features of the interface can be designed apart from the underlying functionality

6.4.2 Warning about iterative design

First, design decisions made at the beginning of the prototyping process are often wrong and design initia can be so great as never to overcome an initial bad decision. Second, if a potential usability problem is discovered, it is important to understand and solve the reason for the problem, and not the symptoms of it.

6.5 Design Rationale

DR is the information that explains why a computer system is the way it is, including its structural and functional description. The benefits of DR:

- DR provides a communication mechanism among the members of the design team.
- DR can capture the context of a design decision in order that a different design team can determine if a similar rationale is appropriate for their product.
- producing a DR forces the designer deliberate more carefully about design decisions.
- since there are mostly alternatives for a 'best design', the DR clearifies the decisions.It also orders the, sometimes many, possible alternatives.
- capturing the context of a decision (eg. the hardware) in the DR will help when using the current design in future designs.

6.5.1 Process-oriented design rationale

DR is often represented using the IBIS (issue-based information system), in which a hierarchical process oriented structure is used: A root issue identifies the main problem, and various descendent positions are puth forth as potential solutions. The relationship between issue en position is refuted by arguments. The IBBIS can be notated textual and graphical.

6.5.2 Design space analysis

In this representation, the design space is initially structured by a set of questions representing the major issues of the design. Options provide alternative solutions to the question.Options can evoke criteria and new quentions and therefore the entire representation can also be hierarchically visualised in a tree-graph.

6.5.3 Psychological design rationale

The purpose of PDR is to design the natural task-artifact cycle of design activity. When a new system becomes an artifact, further observation reveales that in addition to the required tasks it also supports tasks the designer never intended. Once these new tasks have been understood, they can serve as requirements for future artifacts.

The first step in PDR is to identify the tasks that the proposed system will adress and to characterize those tasks by questions that the user tries to answer in accomplishing them. For each question, a set of scenarios of user-system behavior is suggested to support the user in addressing the question. The initial system can then be implemented to provide the functionality suggested by the scenarios. Once this system is running, observations of its use and some designer reflextion is used to produce the actual DR for that version of the system. By forcing the designer to document the PDR, it is hoped that he will become more aware of the natural evolution of user tasks and the artifact, taking advantage of how consequences of one design can be used to improve later designs.

Chapter 7

Summary chapter 9: Evaluation techniques

7.1 Introduction

Evaluation should occur throughout the design life cycle, with the results feeding back into modifications of the design. A distinction is made between evaluation by the designer or a usability expert and evaluation that studies actual use of the system.

7.2 Goals of evaluation

Evaluation has 3 main goals: to assess the extent and accessibility of the system's functionality, to assess the users' experience of the interaction and to identify any specific problems with the system.

7.3 Evaluation through expert analysis

The basic intention of expert analysis is to identify any areas that are likely to cause difficulties because they violate known cognitive principles, or ignore accepted empirical results. 4 approaches are considered here: cognitive walkthrough, heuristic evaluation, the use of models and use of previous work.

7.3.1 Cognitive walkthrough

CW is a detailed review of a sequence of actions, in this case, the steps that an interface will require the user to perform in order to accomplish some known task. The evaluators go through each step and provide a story about why that step is not good for new users. To do a CW, you need four things: a specification or prototype of the system, a description of the task the user is to perform on the system, a complete written list of the actions needed to complete the task with the system and an indication of who the users are and what kind of experience and knowledge the evaluators can assume about them.

For each step, the evaluators try to answer the following questions: Is the effect of the action the same as then users goal at that point? Will the users see that the action is available? Once the users have fount the correct action, will they know it is the one they need? After the action is taken, will users understand the feedback they get?

7.3.2 Heuristic evaluation

A heuristic is a guideline or general principle or rule of thumb that can guide a design decision or be used to critique a decision that has already been made. HE is a method for structuring the critique of a system using a set of relatively simple and general heuristics. Several evaluators independently critique a system to come up with potential usability problems. Each evaluator assesses the system and notes violations of any of the following heuristics and the severity of each of these violations based on four factors: how common is the problem, how easy is it for users to overcome, will it be a one-off problem or a persistent one, and, how seriously will the problem be perceived. The overal result is a severity rating on a scale of 0-4 (see also pg 325).

The 10 heuristics (Nielsen, see also pg 325): Visibility of the system status, match between system and real world, user control and freedom, consistency and standards, error prevention, recognition rather than recall, flexibility and efficiency of use, aesthetic and minimalist design, help users recognize, diagnosize and recover from errors, and help and documentation.

7.3.3 Model-based evaluation

Cetain cognitive and design models provide a means of combining design specification and evaluation into the same framework.

7.3.4 Using previous studies in evaluation

A similar experiment conducted earlier can cut some of the costs of a new design evaluation by reusing the data gained from it.

7.4 evaluation through user participation

7.4.1 Styles of evaluation

Labaratory studies In LS, users take part in controlled tests, often in a specialist usability laboratorium. The advantages are the advanced laboratory equipment and the interruption-free environment. The disadvantage is the lack of context, which may result in unnatural situations.

Field studies in FS, the user is observed using the system in its own work environment. The advantage is the 'natural' use of the system that can hardly be achieved in the lab. However, the interruptions that come with this natural situation may make the observations more difficult.

7.4.2 Emperical methods: experimental evaluation

Any experiment has the same basic forms: the evaluator chooses a hypothesis to test, which can be determined by measuring some attribute of participant behavior. A number of experimental conditions are considered which differ only in the values of certain controlled variables. Any changes in the behavioral measures are attributed to the different conditions. Some factors in the experiment must be considered carefully: the participants chosen, the variables tested and manipulated and the hypothesis tested.

Participants should be chosen to match the expected user population as closely as possible: they must be representative of the intended user population. The sample size must also be large enough to be representative of the intended user population.

Variables come in two main types: those manipulated (independent) and those measured (dependent). The values of the independent variable are known as levels. More complex experiments may have more than one independent variable.

Hypotheses are predictions of the outcome of an experiment, framed in terms of dependent and independent variables, stating that a variation in the independent variable will cause a difference in the dependent variable. The aim of the experiment is proving the hypothesis, which is done by disproving the opposite null-hypothesis.

Experimental design consists of different phases: the first stage is to choose the hypothesis and define the dependent and independent variable. The second step is to select the experimental method: between-subjects, in which each participant is assigned to a different condition, and within-subject, in which each user perfoms under each condition.

Statistical measures: the data should first of all be save to enable performing multiple analysis on the same data. The choice of statistical analysis depends on the type of data and the questions we want to answer. Variables can be classified as dicrete(which can take a finite number of values and levels) and continuous variables (which can take any value between a lower and upper limit) A number of tests can be applied on this data, which are described on pg 333-334.

7.4.3 Obsevatinal techniques

Think aloud and cooperative evaluation

Think aloud is a form fo observation where the user is asked to talk through what he is doing as he is being observed. It has the advantage of simplicity, but the information provided is often subjective and may be selective. A variation is cooperative evaluation, in which the user and evaluator work together to evaluate the system.

Protocol analysis

Methods for recording user actions include paper and pencil, audio recording, video recording, computer logging and user notebooks. In practice, a mixture of the different methods is used. With recordings, the problem is transcription.

Automatic protocol analysis tools

Using Experimental Video Annotator, an evaluator can use predefined tags to write an audio ot video transcription in real time. Using Workplace Project, this can be done while supporting the analysis and synchronization of information from different data streams. DRUM supports the same facilities.

Post-task walkthrough

A walkthrough after the observation reflects the participants' actions back to them after the event. The participant is asked to comment it and to answer questions by the evaluator in order to collect missing information.

7.4.4 Query techniques

Queries provide direct answers from the user about usability questions, but the information is often subjective.

Interviews provide a direct and structured way of gathering information and can be varied to suit the situation. They should be planned in advance with a basic set of questions, and may then be adapted to the specific user. Questionnaires are less flexible than interviews: they are planned in advance. However, it can be used to reach a wider group and takes less time to administer. The styles of questions that can be included are: general background questions, open ended questions, scalars, multi-choice questions and ranked questions. It is always wise to perform a pilot study to test the questionnaire.

7.4.5 Evaluation through monitoring physiological responses

The physiological response monitors receiving currently most attention are eye tracking and physiological measurement.

Eye movements are believed to reflect the amount of cognitive processing a display requires and, therefore, hjow easy or difficult it is to process. Eye movements are based on fixations and saccades (movements between points of interest). Possible measurements are the number of fixations (more –> less efficient search), fixation duration (longer –> more difficult display) and scan path (indicating areas of interest, search strategy and cognitive load). Physiological measurements may be useful in determining the user's emotional response to an interface. It involves attaching various probes and sensors to the user, measuring hearth activity, sweat glands activity, muscle activity and brain activity. The disadvantage is that the readings are hard to interpret.

7.5 Choosing an evaluation method

Factors that distinquish different techniques:

- Design vs implementation: the earlier in the proces, the cheaper and quicker the evaluation must be.
- Labaratory vs field studies

- Subjective vs objective: subjective evaluations require the interpretation of the evaluator and are easily used incorrectly. Objective evaluations provide repeatable results, but sometimes less information.
- Qualitative vs quantitative measurements
- Information provided: the level of information required depends on the state of the design process and influences the required method: the evaluation may concern a certain part of the system or the system as a whole.
- Immediacy of response: some methods record the user's behavior at the time of the interaction itself, others rely on the users recollection of events, which may be incomplete or biased.
- Intrusiveness: the more obvious the evaluation method is to the user, the more it may influence the user's behavior.
- Resources: the limit on resources and other practical restrictions may have their effects on the user's design.

7.5.1 A classification of evaluation techniques

See pg. 360-362.

Chapter 8

Summary chapter 10: Universal design

8.1 Introduction

Universal design is the proces of designing products so that they can be used by as many people as possible in as many situations as possible. Applied to HCI, this means designing interactive systems that are usable by anyone, with any range of abilities, using any technology platform. This can be achieved by designing systems either to have built in redundancy or to be compatible with assistive technologies.

8.2 Universal design principles

In the late 1990's a group at North Carolina State Uiversity proposed seven general principles of universal design, which give us a framework in which to develop interactive systems.

1. Equitable use: the design is useful to people with a range of abilities and appealing to all. No user is excluded or stigmatized. Wherever possible, access should be the same for all. Where appropraite, security, privacy and safety provision should be available to all.

2. Flexibility in use: the design allows for a range of ability and preference, through choise of methods of use and adaptivity to the user's pace, precision and custom.

3. Simple and intuitive to use, regardless of the users (intellectual/physical) properties. It should provide prompting and feedback as far as possible.

4. Perceptive information: the design should provide effective communication of information regardless of the environmental conditions or the user's abilities.

5. Tolerance for error: minimizing the impact and damage caused by mistakes or unintended behavior.

6. Low physical effort: systems should be designed t be comfortable to use, minimizing physical effort and fatigue.

7. Size an space for approach and use: the placement of the system should be such that it can be reached and used by any user regardless of body size, posture or mobility.

8.3 Multi-modal interaction

Since our daily interaction with the world around us is multi-modal, interaction channels hat use more than 1 sensory channel also provide a richer interactive experience. The use of multiple sensory channels increases the bandwith of the interaction between human and computer and also makes the interaction look more like a natural human-human interaction.

8.3.1 Sound in the interface

There is experimental evidence that the addition of adio confirmation of modes reduces errors. There are 2 types of sound available: speech and non-speech.

Speech in the interface

- **Structure of speech** The English language is made up of 40 phonemes: atomic elements of speech that represent a distinct sound. However, the sounds do not make the language entirely: the alteration of tone and quality in phonemes, prosody, gives additional emotion and meaning to a sentence. Also, the sound of a phoneme is influenced by its preceding phoneme, which is called co-articulation. The result of prosody and co-articulation on phonemes can be used to construct a set of allophones, which represent all the different sounds of a language. These can be combined into morphemes: the smallest still meaningful elements of a language, being either words or part of words.

- **Speech recognition** Speech recognition has not yet been very succesful due to the complexity of language, but also because background noise interferes with the input, the user's provide gap-fillers in their speech and different speakers produce different sounds. However, despite it's limitations, speech recognition is becoming available in commercial products.

- **Speech synthesis** Speech synthesis has also not yet been very succesful, mostly because we are sensitive to variation and intonation in speech which can barely be accomplished by the computer. Also, being, transient, spoken output cannot be reviewed or browsed easily. However, for users with certain viual or speech disabilities, the current techniques allready work well.

- **Uninterpreted speech** Speech does not have to be interpreted by a computer to be useful in the interface: recordings of speech can be a useful output.

non-speech sound

Non-speech sounds can often be assimilated more quickly than speech sounds, and are language-independent. It also requires less of the users attention. A disadvantage is that the meaning of the sounds has to be learned.

There are two kinds of usable non-speech sounds: sounds that occur naturally in the world (example: SonicFinder) and using more abstract generated sounds (example: Earcons).

8.3.2 Touch in the interface

The use of touch in the interface is known as haptic interaction (cutaneous perception [tactile sensations through the skin] and kinesthetics [the perception of movement and position]). Touch can provide a primary source of information for users with visual impairments and a richer multi-modal experience for sighted users. The main devices are the electronic braille and the force feedback device.

8.3.3 Handwriting recognition

Handwriting is mostly captured using a digitizer tablet or electronic paper. Recognition is difficult due to the differences between various person's handwriting. Individually written characters are better recognized than longer strings.

8.3.4 Gesture recognition

Gesture is user-dependent, subject to variation and co-articulation and therefore difficult to recognize by a computer. The current systems mostly use data-gloves to capture the gestures.

8.4 Designing for diversity

- Visual impairment: there are two main approaches: the use of sound and the use of touch.
- Hearing impairment: this does not have much influence on the use of a certain interface.
- Physical impairment: for most of this kind of users, the precision required for mouse control is very difficult. This can sometimes be solved by applying speech input, an eyegaze system or a keyboard driver attached to the head.
- Speech impairment: the use of a normal interface is not a problem. To assist the disabled, multimedia systems provide synthetic speech and text-based communication- and conferencesystems.
- Dyslexia: to minimize the amount of text the user needs to process, speech input and output can replace reading and writing. Specially designed spelling correction programs can check the user's input.

- Autism: Communication and social interaction are major areas of difficulty for people with autism. Because computers interact rather impersonal, people with autism can use them well as a communications medium.

8.4.1 Designing for different age groups

- Older people: the lack of mobility of many elderly might catch their interest in e-mail and instant-messaging. However, due to the disabilities lots of these users have, designs must be clear, simple and forgiving of errors.
- Children: Especially younger children have special design-needs, not only in software, but also in documentation and in hardware, since they do not often have a well-developed hand-eye coordination and may have trouble using a keyboard.

8.4.2 Designing for cultural differences

Different area's of misunderstanding include the different meaning of symbols, the use of gestures and the direction of reading and the universal meaning of colours.

Chapter 9

Summary chapter 11: User support

9.1 Introduction

Four main types of assistance that users require are quick reference, task-specific help, full explanation and a tutorial. A distinction is made between help systems and documentation: help systems are problem oriented and specific, and documentation is system-oriented and generic.

9.2 Requirements of user support

- Availability: the user needs to be able to access help at any time during his interaction with the system.
- Accuracy and completeness: due to the frequent software updates, accuracy and completeness are difficult aspects of support. The help-function should cover the whole system.
- Consistency: different parts and versions of the help system should be consistent in terms of content, terminology and style of presentation.
- Robustness: since the help function is mostly used when the user is experiencing system problems, it should be predictable and not easily influenced by errors.
- Flexibility: the ideal help system should adapt to the properties of it's user and it's environment.
- Unobtrusiveness: the help system should not prevent the user from continuing with normal work, nor should it interfere with the user's application.

9.3 Approaches to user support

- Command assistance: the user requests help on a particular command and is presented with a help screen of manual page describing it. In order to use it, the user has to know what he is looking for.

- Command prompts: in command line interfaces CP provides help when the user encounters an error, usually in the form of correct usage prompts. These prompts are useful but assume knowledge of the command.

- Context-sensitive help: these range from those that have specific knowledge of the particular user to those that provide a simple help key or function that is interpreted according to the context in which it is called and will present help accordingly.

- Online tutorials: allow the user to work through the basics of an application within a test environment. An alternative to the traditional online tutorial is to allow the user to learn the system by exploring and experimenting with a version with limited usability.

- Online documentation: makes the exisiting paper documentation available on the computer for a larger number of users. Documentation is designed to provide a full description of the system's functionallity and behavior in a systematic manner: a high amount of generic information. Minimal manuals should provide enough information for less experienced users.

- Wizzards and assistants:

  - Wizzard: a task-specific tool that leas the user through the task, using information supplied by the user in response to questions along the way. They are common in application as they offer the user the possibility to perform a complex task safely, quickly and efficiently. They can, however, be unnecessarily constraining.
  - Assistants: software tools that monitor user behavior and offer suggestions or hits when they recognize familiar sequences. They should most of all be inobtrusive.

9.4 Adaptive help systems

Adaptive help is a special case of a general class of interactive systems, known as intelligent systems. They operate by monitoring the activity of the user and constructing a model of him. Using this model, together with knowledge of the working-domain and general information, the adaptive system will present help relevant to the task and suited to the user's experience.

9.4.1 Knowledge representation: user modelling

Adaptable systems allow the user to provide a model of himself around which the system will be configured by adjusting preferences. However, a model can also be provided by the designer of can be generated by the system itself out of observations. Approaches:

- Quantification: the system recognized a number of different levels of expertise, to which it will respond differently. The user's level of expertised as perceived by the system is adjusted during the interaction.

- Stereotypes: the system cathegorizes the user as a member of a known group of users or stereotype, based on user characteristics.
- Overlay models: the user's behavior is compared to the behavior of an idealized model. The differences indicate the level of expertise of the user.

9.4.2 Knowledge representation: domain and task modeling

Some help systems build a model of the user's current task or plan, which can be accomplished by representing user tasks in terms of the used command sequences.

9.4.3 Knowledge representation: modeling advisory strategy

Providing help with a system that includes modeling advisory strategy allows it not only to select approppriate advice for the user but also to use an appropriate method of giving advice.

9.4.4 Techniques for knowledge representation

Four main groups of representation systems, that are often combined:

- Rule-based techniques: knowledge is represented as a set of rules and facts, which are interpreted using some inference mechanism.
- Frame-based techniques: used to represent commonly occuring situations and default knowledge, a frame is a structure that contains labeled slots, representing related features.
- Network-based techniques: represent knowledge about the user and system in terms of relationships between facts (semantic network).
- Example-based techniques: represent knowledge implicitly within a decision structure of a classification system. Items are matched to the example.

9.4.5 Problems with knowledge representation and modeling

Knowledge is often difficult to elicit, and it is hard to ensure completeness and correctness. The amount of knowledge required is substantial: adaptive help is expensive. Interpreting the information appropriate is also difficult.

9.4.6 Other issues

- Initiative: who should direct the interaction? Mixed initiative is the best solution, but the user must always be alble to override the system.
- Effect: which part should you make adaptive and how much information do you really need to gather? Most often, to detailed information is gathered.

- Scope: is the help to be offered at an application level or system wide? System wide is much more complex.

9.5 Designing user support systems

The design of user support should not be an add-on but should be fully integrated in the system. The content of the help and context in which it will be used should be considered before the technology that it will require.

9.5.1 Presentation issues

- Requesting help: is the help function accessed through a command, a button or a seperate application?
- Displaying help: in a new window, pop-up boxes or at command line level?
- Effective presentation: besides the normal design guidelines, the right style of language use is also very important, as are matters like indexing and readability.

9.5.2 Implementation issues

Physical constraints like speed, memory capacity and screen size, or software-issues like programming-languages and command types influence the implementation, as well as the structure of the help: a file, hierarchie, database etc. Also the authors of the help material should be involved in the design process.

Chapter 10

Summary chapter 19: Groupware

10.1 Introduction

CSCW: computer-supported cooperative work: how to design systems to support work as a group and how to understand the effect of technology on the group's work pattern. The computer products that support CSCW are called groupware.

10.2 Groupware systems

Groupware can be classified by where and when the participants are performing the cooperative work, or by the function of the meeting. The first is summarized in a time/space matrix (pg 665). The framework used to organize the information in this chapter is based on the entities involved in cooperative work: the participants and the artifacts on which they work (see pg 666). From this point of view, there are different functions the groupware can support in the framework: computer-mediated communication (supporting the direct communications between participants), meeting and decision support systems (capturing common understanding) and shared applications and artifacts (supporting the participants' interaction with shared objects - the artifacts of work.

10.3 Computer-mediated communication

Asynchronous remote Communication: E-mail/bulletin boards. Synchronous remote communication: instant messaging, sms.

10.3.1 E-mail and bulletin boards

Stages of sending a simple E-mail message: preparation –> dispatch –> delivery –> notification –> receipt. Other important aspects of E-mail are attachments, cc's and distribution lists.

E-mail and electronic conference systems have differences: they vary in terms of who controls the distribution list: in E-mail, the sender or system administrator edits the list. In electronic newsgroups or bulletin boards, the user decides which groups he joins. Also, fast E-mail can serve as a kind of synchronous communications: a chat program. There is a difference in granularity between different programs: some transmit each character the sender types, others only send complete contributions.

10.3.2 Structured message systems

E-mail en electronic conferences provide an overload of messages. Various forms of structured message systems have been developed to help deal with this overload: the Information Lens, which filters the messages. This works best if the sender keeps to a template for his messages. Another approach is the use of conversation-structures.

10.3.3 txt is gr8

In the last years, the use of instant-messaging services and sms has increased rapidly.

10.3.4 Video conferences and communication

Synchronous remote facilities: video conferences, pervasive video for enhancing social communication and video integrated with another shared application. To make the interaction more social, video wall's can be used to communicate with different offices, but this gives rise to a lot more usability problems, like camera range.

10.3.5 Virtual collaborative environments

VR-techniques allow participants to meet in a virtual world. The representation of a participant is an embodiment.

10.4 Meeting and decision support systems

Three types of systems where the generation and recording of ideas and decisions is the primary focus: Argumentation tools (record argument to arrive at a decision, supporting asynchronous co-located design teams), Meeting rooms (supporting face-to-face groups [synchronous co-located] in brainstorming and management meetings) and shared drawing surfaces (used for synchronous remote design meetings).

10.4.1 Argumentation tools

From a CSCW viewpoint, a group of workers should be able to work on the same document or program. In the simplest form, this can be done one at the time. Sophisticated tools also have facilities to allow several designers to use the system simultaneously. It therefore needs concurrency control: different people's work should not interfere. Those systems also have notification mechanisms to let participants know which parts have been changed. The system may allow a range of interaction styles, from asynchronous to synchronous.

10.4.2 Meeting rooms

MR's contain a large viewscreen around which all the participants are seated. Each participant has his own terminal, which he can use for private applications. The screen takers the form of an electronic whiteboard, on which all participants can write. Using floor control policies, only one participant can write at a certain time, for example using locking mechanisms. However, due to the influence of the technology on the social aspects of the meeting and the problems concerning deictic references, these meeting rooms are not yet very common. Digital techniques are, however, used to capture the drawings made during a meeting.

10.4.3 Shared work surfaces

The synchronous co-located meeting room software can be used for synchronous remote meetings. This causes additional problems: person-to-person communication, computer networks, delays, etc. To make the systems more realistic, most of them support free hand drawing. There are different variations: some of them use camera's to film a whiteboard, others film a piece of paper and sometimes you can write directly on the touchscreen.

10.5 Shared applications and artifacts

10.5.1 Shared PCs and shared window systems

Shared PCs and shared window systems allow ordinary applications to be the focus of cooperative work. Two or more computers function as if they were one. The difference with the meeting room is the absence of the specialized software. Using a kind of locking protocol, the users work with ordinary programs. The focus of the activities is on document processing and technical support.

10.5.2 Shared editors

A shared editor is an editor for text or graphics which is collaboration aware: it knows that it is being shared. Due to the large amount of options that can be customized, these editors become more adaptable. People are mostly allowed to edit different parts of the same document. The problem is that indexical expressions often do no longer have meaning: both users may be looking at different parts of the document. WYSISIS-views can prevent these misunderstandings.

10.5.3 Co-authoring systems

Co-authoring is largely asynchronous. There are a lot of systems supporting this type of work, in which comments are linked to the text. Co-authoring systems support concurrency control, but can also allow participants to work synchronously.

10.5.4 Shared diaries

Each person uses a shared electronic diary and the system tries to find a free spot when you want to plan a meeting. This evokes both technical and social problems: privacy and access rights, for example.

10.5.5 Communication through the artifact

The awareness of the actions of other participants on the artifact they work on is a form of communication through the artifact. Sometimes, this is enough for effective cooperative working.

Chapter 11

Summary chapter 20: Ubiquitous computing and augmented realities

Remark 1 *Due to the abstract contents of chapter 20, it is absolutely recommended to study the original text intensively because this summary may not include all the important aspects.*

11.1 Introduction

Nothing interesting in this section ;-)

11.2 Ubiquitous computing applications research

The defining characteristic of ubiquitous computing is the attempt to break away from the traditional desktop interaction paradigm and move computational power into the environment that surrounds the user. Rather than force the user to search out and find the computer's interface, ubiquitous computing suggests that the interface itself can take on the responsibility of locating and serving the user. The technique used is any computing technology that permits human interaction away from a single workstation.

11.2.1 Defining the appropriate physical interaction experience.

The interaction using ubiquitous computing will be like the way humans interact with the physical world. The drive for a ubiquitous computing experience has resulted in a variety of important changes to the input, output and interactions that define the human experience with computing.

First, input has moved beyond the explicit nature of textual input from keyboards and selection from pointing devices to a greater variety of data types, resulting in a shift to more implicit forms of input. Sometimes as implicit as gathering physical information from sensors: no active human intervention is necessary.

Second, the integration of ubiquitous computing capabilities into everyday life also requires new output technologies and techniques. We do no longer need a desktop screen to display the system output: a variety of displays can be placed across our environment. Two important trends have emerged: first, we want to move information between separate displays easily and coordinate the interaction between multiple displays. Secondly, we desire them to be less demanding of our attention. These trend toward peripheral output has been explored for a particular class of displays, called ambient. Ambient displays require minimal attention and cognitive effort, and are thus more easily integrated into a persistent physical space. These displays do not have to be visual, but can also produce (for example) motoric output.

Third, an important factor of ubicomp is that it attempts to merge computational artifacts smoothly with the world of physical artifacts. By overlaying electronic information over the real world, an augmented reality is produced.

11.2.2 Application themes for ubicomp

The brief history of ubicomp demonstrates some emergent features that appear across many applications: the ability to use implicitly sensed context from the physical and electronic environment to determine the correct behavior of any given device. Another feature is the ability to easily capture and store memories of live experiences and serve them up for later use. The trajectory of these two applications themes coupled with the increasing exploration of ubiquitous computing into novel, non-work environments, points to the changing relationship between people and computing, and thus the changing purpose of ubicomp applications. This newer trajectory is called everyday computing.

Context-aware computing

Location of identifiable entities (eg humans) is a very common piece of context used in ubicomp application development (mostly gps-based). However, there is more context than position and identity. In addition, context awareness involves 'when' (time-awareness, for example to analyze routines), 'what' (perceiving and interpreting human activity), 'why' (understand why people are doing what they are doing)

Automated capture and access

We define capture and access as the task of preserving a record of some live experience that is then reviewed at some point in the future. Tool that support this activity can remove the burden of doing something humans are not good at, so they can focus attention on activities they are good at.

Toward continuous interaction

The majority of computer applications support well-defined tasks that have a marked beginning and end with multiple subtasks in between. These applications are not well suited to the more general activities of ubicomp. Therefore, the emphasis on designing for continuously available interaction requires addressing these features of informal, daily activities: no clear beginning and end, interruption is expected, multiple activities operate concurrently, time is important and associative models of information are needed.

11.2.3 Understanding interaction in ubicomp

As the application of computers has broadened, designers have turned to models that consider the nature of the relationship between the internal cognitive processes and the outside world. The design-focus is therefore currently on finding the balance in this relationship, using three main theories:

- Activity theory: AT recognizes goals, actions and operations. Both goals and actions are fluid based on the physical state of the world instead of more fixed, a priori plans. Due to a change in the circumstances, an operation can require more attention than usual. AT also emphasizes the transformational properties of artifacts that implicitly carry knowledge and traditions.
- Situated action and distributed cognition: SA emphasizes the improvisational aspects of human behavior and de-emphasizes a priori plans that are simply executed by the person. Knowledge of the world continually changes the shape and execution of a task. Ubiquitous computing should adapt to this and should not require the user to follow a predefined script. DC also de-emphasizes internal human cognition, but in this case, it turns to a system perspective where humans are just part of a larger system. Ubiquitous computing efforts information by distributed cognition, focus on designing for a larger system goal, in contrast to the use of an individual appliance and emphasizes how information is encoded on objects and how that information is translated, and perhaps transmitted, by different users.
- Understanding human practice: ethnography and cultural probes: The challenge for ubicomp designers is to uncover the very practices through which people live, and to make these invisible practices visible and available to the developers has emerged as a primary approach to address the need to gain rich understandings of a particular setting and the everyday practices that encompass these settings. In the context of ubicomp, the goal of an ethnographic investigation is to provide so that ubicomp environments seamlessy mesh with everyday practices that encapsulate the goals, attitudes, social relationships, knowledge and language of the intended setting. Moreover, cultural probes have been used to collect information from settings on order to inspire the development of new digital devices.

11.2.4 Evaluation challenges for ubicomp

The shift away from the desktop also means that ubicomp needs to be operational in strange environments. However, this means uncertainty in how to apply qualitative and quantitative evaluation methods. Although many researchers have investigated the possibilities, the lack of employment of ubiquitous environments has hampered many of these activities.

11.3 Virtual and augmented reality

VR is more than just an entire, immersive, VR-world. There are different aspects of virtual reality.

11.3.1 VR technology

Usually a headset is used. VR headsets are becoming lighter and easier to handle. The computer power necessary to display a smooth, fully detailed VR-world is not yet available, therefore the VR-world often looks 'blocky'. Input is usually through data gloves and speech recognition.

11.3.2 Immersive VR

Immersive VR takes the user completely into a VR-world, usually through helmed and data glove. In the VR setup, color and shading are used in a primitive form to give dimension and depth to images, but work is continuing on developing efficient algorithms on dedicated machines to allow more detailed imaging.

11.3.3 VR on the desktop and in the home

In desktop VR, 3D images are presented on a normal computer screen and manipulated using mouse and keyboard. An example are interactive video games like flight simulators. Using the special programming language VRML, this technology is widely available.

11.3.4 Command and control

VR is intensively used in training military and emergency operations, for example in flight simulators.

11.3.5 Augmented reality

In augmented reality systems electronic images are projected over the real word. An example is the head-up display in aircrafts. The link with ubiquitous computing is obvious when you attach semi-transparent goggles to a wearable computer. The main difficulty is the registration: the alignment of the virtual and physical world.

11.3.6 Current and future applications of VR

Today, most VR is used for military simulations and games. Of course, a lot of different applications are imaginable, including medical ones.

11.4 Information and data visualization

11.4.1 Scientific and technical data

Three-dimensional representations of scientific and technical data can be classified by the number of dimensions in the physical world that correspond to physical spatial dimensions, as opposed to those that correspond to more abstract parameters. The next step away for 3D is when two of the dimensions represent a physical plane and the third is used to represent some data for each point. In any such representation it is hard to choose a viewing point, as it is likely that tall structures in the foreground will hide important features in the background. Finally, we have the case where only one or none of the dimensions represent a spatial dimension. Using our imagination, we can understand these representations.

11.4.2 Structured information

Data sets that arise in information systems typically have many discrete attributes and structures. One common approach is to convert the discrete structure into some measure of similarity. A range of techniques can than be applied to map the data points into two or three dimensions, preserving as well as possible the similarity measures (similar points closer). For the representation of networks and hierarchies, standard techniques are available.

11.4.3 Time and interactivity

Many data sets include temporal values and the passage of time itself can be used in order to visualize other types of data. The time in the data can be mapped directly onto time at the interface: the time-varying data are replayed. Alternatively, a spatial dimension may be mapped onto the passage of time at the interface to gain a 3D object. Special toolsets are available to analyze temporal information in 3D-models.