DOWNLOADED FROM STUCOR APP

HUMAN COMPUTER INTERACTION NOTES
STUCOR APP

REGULATION 2013
CS6008 NOTES

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SYLLABUS (THEORY)

Sub. Code	:CS6008	**Branch / Year / Sem**	: CSE/IV/VIII
Sub.Name	:**Human Computer Interaction**	**Batch**	: 2015-2019
	:	**Academic Year**	: 2018-2019

L T P C
3 0 0 3

UNIT I FOUNDATIONS OF HCI **9**

The Human: I/O channels – Memory – Reasoning and problem solving; The computer: Devices – Memory – processing and networks; Interaction: Models – frameworks – Ergonomics – styles – elements – interactivity- Paradigms.

UNIT II DESIGN & SOFTWARE PROCESS **9**

Interactive Design basics – process – scenarios – navigation – screen design – Iteration and prototyping. HCI in software process – software life cycle – usability engineering – Prototyping in practice – design rationale. Design rules – principles, standards, guidelines, rules. Evaluation Techniques – Universal Design.

UNIT III MODELS AND THEORIES **9**

Cognitive models –Socio-Organizational issues and stake holder requirements –Communication and collaboration models-Hypertext, Multimedia and WWW.

UNIT IV MOBILE HCI **9**

Mobile Ecosystem: Platforms, Application frameworks- Types of Mobile Applications: Widgets, Applications, Games- Mobile Information Architecture, Mobile 2.0, Mobile Design: Elements of Mobile Design, Tools.

UNIT V WEB INTERFACE DESIGN **9**

Designing Web Interfaces – Drag & Drop, Direct Selection, Contextual Tools, Overlays, Inlays and Virtual Pages, Process Flow. Case Studies.

TOTAL: 45 PERIODS

TEXT BOOKS

T1. Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale, "Human Computer Interaction", 3rd Edition, Pearson Education, 2004 (UNIT I , II & III).

T2. Brian Fling, "Mobile Design and Development", First Edition , O'Reilly Media Inc., 2009 (UNIT –IV).

T3. Bill Scott and Theresa Neil, "Designing Web Interfaces", First Edition, O'Reilly, 2009.(UNIT-V).

SUBJECT IN-CHARGE **HOD**

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

M.I.E.T. ENGINEERING COLLEGE
(Approved by AICTE and Affiliated to Anna University Chennai)
TRICHY – PUDUKKOTTAI ROAD, TIRUCHIRAPPALLI – 620 007

UNIT I

FOUNDATIONS OF HCI

The Human: I/O channels – Memory – Reasoning and problem solving; The computer: Devices – Memory – processing and networks; Interaction: Models – frameworks – Ergonomics – styles – elements – interactivity- Paradigms.

1. HUMAN

1.1 Introduction

Human-computer interaction (commonly referred to as HCI) researches the design and use of computer technology, focused on the interfaces between people (users) and computers. Researchers in the field of HCI both observe the ways in which humans interact with computers and design technologies that let humans interact with computers in novel ways.

User

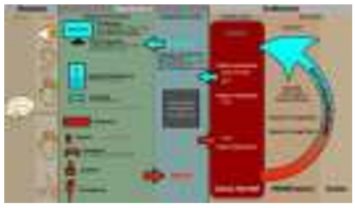
By "user", we may mean an individual user, a group of users working together. An appreciation of the way people's sensory systems (sight, hearing, touch) relay information is vital. Also, different users form different conceptions or mental models about their interactions and have different ways of learning and keeping knowledge and. In addition, cultural and national differences play a part.

Computer

When we talk about the computer, we're referring to any technology ranging from desktop computers, to large scale computer systems. For example, if we were discussing the design of a Website, then the Website itself would be referred to as "the computer". Devices such as mobile phones or VCRs can also be considered to be —computers‖.

Interaction

There are obvious differences between humans and machines. In spite of these, HCI attempts to ensure that they both get on with each other and interact successfully. In order to achieve a usable system, you need to apply what you know about humans and computers, and consult with likely users throughout the design process. In real systems, the schedule and the budget are important, and it is vital to find a balance between what would be ideal for the users and what is feasible in reality.



STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

The Goals of HCI

The goals of HCI are to produce usable and safe systems, as well as functional systems. In order o produce computer systems with good usability, developers must attempt to: understand the factors that determine how people use technology, develop tools and techniques to enable building suitable systems, achieve efficient, effective, and safe interaction put people first.

Underlying the whole theme of HCI is the belief that people using a computer system should come first. Their needs, capabilities and preferences for conducting various tasks should direct developers in the way that they design systems. People should not have to change the way that they use a system in order to fit in with it. Instead, the system should be designed to match their requirements.

Usability

Usability is one of the key concepts in HCI. It is concerned with making systems easy to learn and use. A usable system is:

- easy to learn
- easy to remember how to use
- effective to use
- efficient to use
- safe to use
- enjoyable to use

Factors in HCI

There are a large number of factors which should be considered in the analysis and design of a system using HCI principles. Many of these factors interact with each other, making the analysis even more complex. The main factors are listed in the table below:

Organisation Factors

- Training, job design, politics, roles, work organisation
- Environmental Factors
- Noise, heating, lighting, ventilation
- Health and Safety Factors

The User

- Cognitive processes and capabilities
- Motivation, enjoyment, satisfaction, personality, experience
- Comfort Factors
- Seating, equipment, layout.

User Interface

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

Input devices, output devices, dialogue structures, use of colour, icons, commands, navigation, graphics, natural language, user support, multimedia,

Task Factors **:** Easy, complex, novel, task allocation, monitoring, skills

Constraints **:** Cost, timescales, budgets, staff, equipment, buildings

System Functionality: Hardware, software, application

Productivity Factors : Increase output, increase quality, decrease costs, decrease errors, increase innovation

Disciplines contributing to HCI

The field of HCI covers a wide range of topics, and its development has relied on contributions from many disciplines. Some of the main disciplines which have contributed to HCI are:

Computer Science

- technology
- software design, development & maintenance
- User Interface Management Systems (UIMS) & User Interface Development Environments (UIDE)
- prototyping tools
- graphics

Cognitive Psychology

- information processing
- capabilities
- limitations
- cooperative working
- performance prediction

Social Psychology

- social & organizational structures

Ergonomics/Human Factors

- hardware design
- display readability

Linguistics

- natural language interfaces

Artificial Intelligence

- **intelligent software**

Engineering & Design

- graphic design

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

- engineering principles

1.2 INPUT–OUTPUT CHANNELS

A person‘s interaction with the outside world occurs through information being received and sent: input and output. In an interaction with a computer the user receives information that is output by the computer, and responds by providing input to the computer – the user‘s output becomes the computer‘s input and vice versa.

For example, sight may be used primarily in receiving information from the computer, but it can also be used to provide information to the computer, for example by fixating on a particular screen point when using an eyegaze system. Input in the human occurs mainly through the senses and output through the motor control of the effectors.

There are five major senses: **sight, hearing, touch, taste and smell**. Of these, the first three are the most important to HCI. **Taste and smell** do not currently play a significant role in HCI, and it is not clear whether they could be exploited at all in general computer systems, although they could have a role to play in more specialized systems (smells to give warning of malfunction, for example) or in augmented reality systems. **vision, hearing** and touch are central.

There are a number of effectors, including the limbs, fingers, eyes, head and vocal system. In the interaction with the computer, the fingers play the primary role, through typing or mouse control, with some use of voice, and eye, head and body position.

Imagine using a personal computer (PC) with a mouse and a keyboard. The application you are using has a graphical interface, with menus, icons and windows. In your interaction with this system you receive information primarily by sight, from what appears on the screen.

1.2.1 Vision

Human vision is a highly complex activity with a range of physical and perceptual limitations, We can roughly divide visual perception into two stages: the physical reception of the stimulus from the outside world, and the processing and interpretation of that stimulus. On the one hand the physical properties of the eye and the visual system mean that there are certain things that cannot be seen by the human; on the other the interpretative capabilities of visual processing allow images to be constructed from incomplete information. We need to understand both stages as both influence what can and cannot be perceived visually by a human being, which in turn directly affects the way that we design computer systems. We will begin by looking at the eye as a physical receptor, and then go on to consider the processing involved in basic vision.

The human eye

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

Vision begins with light. The eye is a mechanism for receiving light and transforming it into electrical energy. Light is reflected from objects in the world and their image is focused upside down on the back of the eye. The receptors in the eye transform it into electrical signals which are passed to the brain.

The eye has a number of important components. The cornea and lens at the front of the eye focus the light into a sharp image on the back of the eye, the retina. The retina is light sensitive and contains two types of photoreceptor: rods and cones.

Rods are highly sensitive to light and therefore allow us to see under a low level of illumination. They are unable to resolve fine detail and are subject to light saturation. This is the reason for the temporary blindness we get when moving from a darkened room into sunlight: the rods have been active and are saturated by the sudden light. The **cones** do not operate either as they are suppressed by the rods. We are therefore temporarily unable to see at all. There are approximately 120 million rods per eye which are mainly situated towards the edges of the retina. Rods therefore dominate peripheral vision.

Cones are the second type of receptor in the eye. They are less sensitive to light than the rods and can therefore tolerate more light. There are three types of cone, each sensitive to a different wavelength of light. This allows color vision. The eye has approximately 6 million cones, mainly concentrated on the fovea, a small area of the retina on which images are fixated.

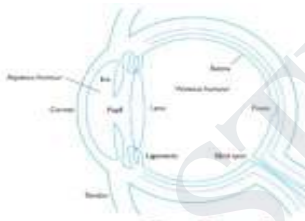


Figure: Human Eye

The retina is mainly covered with photoreceptors there is one blind spot where the optic nerve enters the eye. The blind spot has no rods or cones, our visual system compensates for this so that in normal circumstances we are unaware of it.

The retina also has specialized nerve cells called ganglion cells. There are two types:

X-cells, which are concentrated in the fovea and are responsible for the early detection of pattern; and Y-cells which are more widely distributed in the retina and are responsible for the early detection of movement. The distribution of these cells means that, while we may not be able to detect changes in pattern in peripheral vision, we can perceive movement.

Visual perception

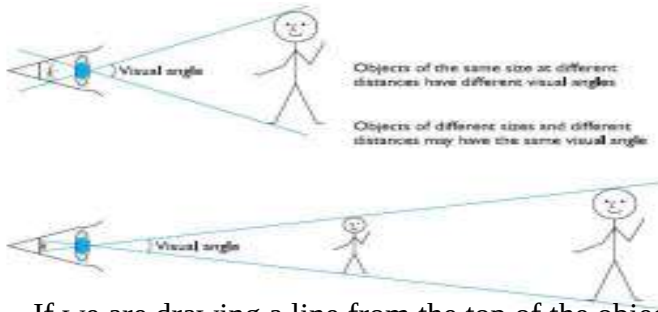
STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

The information received by the visual apparatus must be filtered and passed to processing elements which allow us to recognize coherent scenes, disambiguate relative distances and differentiate colour.

How does the eye perceive size, depth and relative distances? To understand this we must consider how the image appears on the retina. Reflected light from the object forms an upside-down image on the retina. The size of that image is specified as a visual angle. Figure illustrates how the visual angle is calculated.



If we are drawing a line from the top of the object to a central point on the front of the eye and a second line from the bottom of the object to the same point, the visual angle of the object is the angle between these two lines. Visual angle is affected by both the size of the object and its distance from the eye. Therefore if two objects are at the same distance, the larger one will have the larger visual angle. Similarly, if two objects of the same size are placed at different distances from the eye, the furthest one will have the smaller visual angle. The visual angle indicates how much of the field of view is taken by the object. The visual angle measurement is given in either degrees or minutes of arc, where 1 degree is equivalent to 60 minutes of arc, and 1 minute of arc to 60 seconds of arc.

Perceiving brightness

An aspect of visual perception is the perception of brightness. Brightness is in fact a subjective reaction to levels of light. It is affected by luminance which is the amount of light emitted by an object. The luminance of an object is dependent on the amount of light falling on the object‘s surface and its reflective properties. Luminance is a physical characteristic and can be measured using a photometer. Contrast is related to luminance: it is a function of the luminance of an object and the luminance of its background.

Perceiving colour A third factor that we need to consider is perception of colour. Colour is usually regarded as being made up of three components: **hue, intensity and saturation**. Hue is determined by the spectral wavelength of the light. Blues have short wavelengths, greens medium and reds long. Approximately 150 different hues can be discriminated by the average person. Intensity is the brightness of the color, and saturation is the amount of whiteness in the color. By varying these two, we can perceive in the region of 7 million different colors.

The capabilities and limitations of visual processing

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

Visual processing involves the transformation and interpretation of a complete image, from the light that is thrown onto the retina. Visual processing compensates for the movement of the image on the retina which occurs as we move around and as the object which we see moves. Although the retinal image is moving, the image that we perceive is stable. Similarly, colour and brightness of objects are perceived as constant, in spite of changes in luminance.

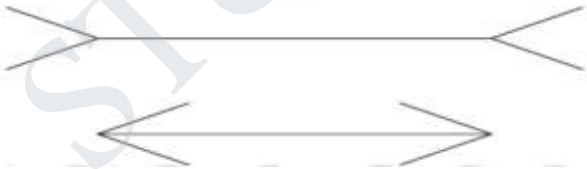
This ability to interpret and exploit our expectations can be used to resolve ambiguity. For example, consider the image shown in Figure is an ambiguous shape

Now consider Figure_s below. The context in which the object appears allows our expectations to clearly disambiguate the interpretation of the object, as either a B or a 13.

ABC

12 13 14

Consider Figure below, which line is longer? Most people when presented with this will say that the top line is longer than the bottom. In fact, the two lines are the same length. This may be due to a false application of the law of size constancy: the top line appears like a concave edge, the bottom like a convex edge.



Reading

There are several stages in the reading process. First, the visual pattern of the word on the page is perceived. It is then decoded with reference to an internal representation of language. The final stages of language processing include syntactic and semantic analysis and operate on phrases or sentences.

During reading, the eye makes jerky movements called saccades followed by fixations. Perception occurs during the fixation periods, which account for approximately 94% of the time elapsed. The eye moves backwards over the text as well as forwards, in what are known as regressions. If the text is complex there will be more regressions.

Adults read approximately 250 words a minute. It is unlikely that words are scanned serially, character by character, since experiments have shown that words can be recognized as quickly as single characters. Instead, familiar words are recognized using word shape. This

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

means that removing the word shape clues (for example, by capitalizing words) is detrimental to reading speed and accuracy. The speed at which text can be read is a measure of its legibility. Experiments have shown that standard font sizes of 9 to 12 points are equally legible, given proportional spacing between lines . Similarly line lengths of between 2.3 and 5.2 inches (58 and 132 mm) are equally legible. However, there is evidence that reading from a computer screen is slower than from a book . This is thought to be due to a number of factors including a longer line length, fewer words to a page, orientation and the familiarity of the medium of the page. These factors can of course be reduced by careful design of textual interfaces. a negative contrast (dark, characters on a light screen) provides higher luminance and, therefore, increased acuity, than a positive contrast. This will in turn increase legibility. Experimental evidence suggests that in practice negative contrast displays are preferred and result in more accurate performance.

1.2.2 Hearing

The sense of hearing is often considered secondary to sight, but we tend to underestimate the amount of information that we receive through our ears. hearing begins with vibrations in the air or sound waves. The ear receives these vibrations and transmits them, through various stages, to the auditory nerves. The ear comprises three sections, commonly known as the

Outer ear, middle ear and inner ear.

The outer ear is the visible part of the ear. It has two parts: the pinna, which is the structure that is attached to the sides of the head, and the auditory canal, along which sound waves are passed to the middle ear. The outer ear serves two purposes. First, it protects the sensitive middle ear from damage. The auditory canal contains wax which prevents dust, dirt and over-inquisitive insects reaching the middle ear. It also maintains the middle ear at a constant temperature. Secondly, the pinna and auditory canal serve to amplify some sounds.

The middle ear is a small cavity connected to the outer ear by the tympanic membrane, or ear drum, and to the inner ear by the cochlea. Within the cavity are the ossicles, the smallest bones in the body. Sound waves pass along the auditory canal and vibrate the ear drum which in turn vibrates the ossicles, which transmit the vibrations to the cochlea, and so into the inner ear. This _relay‘ is required because, unlike the air-filled outer and middle ears, the inner ear is filled with a denser cochlean liquid. If passed directly from the air to the liquid, the transmission of the sound waves would be poor. By transmitting them via the ossicles the sound waves are concentrated and amplified.

Processing sound

Processing sound has a number of characteristics which we can differentiate. Pitch is the frequency of the sound. A low frequency produces a low pitch, a high frequency, a high pitch. Loudness is proportional to the amplitude of the sound; the frequency remains constant.

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

Timbre relates to the type of the sound: sounds may have the same pitch and loudness but be made by different instruments and so vary in timbre. We can also identify a sound‘s location, since the two ears receive slightly different sounds, owing to the time difference between the sound reaching the two ears and the reduction in intensity caused by the sound waves reflecting from the head.

The human ear can hear frequencies from about 20 Hz to 15 kHz. It can distinguish frequency changes of less than 1.5 Hz at low frequencies but is less accurate at high frequencies. Different frequencies trigger activity in neurons in different parts of the auditory system, and cause different rates of firing of nerve impulses. The auditory system performs some filtering of the sounds received, allowing us to ignore background noise and concentrate on important information. The exception is multimedia, which may include music, voice commentary and sound effects. However, the ear can differentiate quite subtle sound changes and can recognize familiar sounds without concentrating attention on the sound source.

1.2.3 Touch

Touch provides us with vital information about our environment. It tells us when we touch something hot or cold, and can therefore act as a warning. It also provides us with feedback when we attempt to lift an object, for example. Consider the act of picking up a glass of water. If we could only see the glass and not feel when our hand made contact with it or feel its shape, the speed and accuracy of the action would be reduced. This is the experience of users of certain virtual reality games: they can see the computer-generated objects which they need to manipulate but they have no physical sensation of touching them. Watching such users can be an informative and amusing experience! Touch is therefore an important means of feedback, and this is no less so in using computer systems. Feeling buttons depress is an important part of the task of pressing the button. Also, we should be aware that, although for the average person, haptic perception is a secondary source of information, for those whose other senses are impaired, it may be vitally important. For such users, interfaces such as braille may be the primary source of information in the interaction. The apparatus of touch differs from that of sight and hearing in that it is not localized. The skin contains three types of sensory receptor: thermo receptors respond to heat and cold, nociceptors respond to intense pressure, heat and pain, and mechanoreceptors respond to pressure.

1.2.4 Movement

A simple action such as hitting a button in response to a question involves a number of processing stages. The stimulus (of the question) is received through the sensory receptors and transmitted to the brain. The question is processed and a valid response generated. The

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

brain then tells the appropriate muscles to respond. Each of these stages takes time, which can be roughly divided into reaction time and movement time.

Movement time is dependent largely on the physical characteristics of the subjects: their age and fitness, for example. Reaction time varies according to the sensory channel through which the stimulus is received. A person can react to an auditory signal in approximately 150 ms, to a visual signal in 200 ms and to pain in 700 ms.

A second measure of motor skill is accuracy. One question that we should ask is whether speed of reaction results in reduced accuracy. This is dependent on the task and the user. In some cases, requiring increased reaction time reduces accuracy. This is the premise behind many arcade and video games where less skilled users fail at levels of play that require faster responses. Speed and accuracy of movement are important considerations in the design of interactive systems, primarily in terms of the time taken to move to a particular target on a screen. The target may be a button, a menu item or an icon, for example. 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 . There are many variations of this formula, which have varying constants, but they are all very similar. One common form is

Movement time = a + b log2(distance/size +1)

where a and b are empirically determined constants.

1.3 HUMAN MEMORY

Our memory contains our knowledge of actions or procedures. It allows us to repeat actions, to use language, and to use new information received via our senses. It also gives us our sense of identity, by preserving information from our past experiences.

Memory is the second part of our model of the human as an information-processing system. Memory is associated with each level of processing. Bearing this in mind, we will consider the way in which memory is structured and the activities that take place within the system. It is generally agreed that there are three types of memory or memory function: sensory buffers, short-term memory or working memory, and long-term memory. There is some disagreement as to whether these are three separate systems or different functions of the same system. It is sufficient to note three separate types of memory. These memories interact, with information being processed and passed between memory stores.

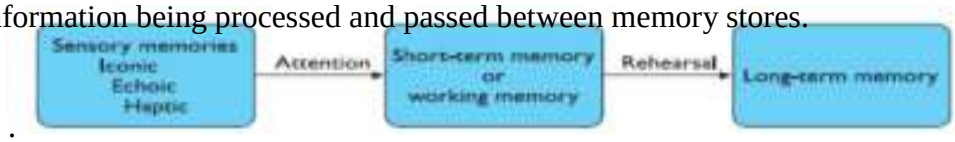


Figure: A model of the structure of memory

1.3.1 Sensory memory

The sensory memories act as buffers for stimuli received through the senses. A sensory memory exists for each sensory channel: iconic memory for visual stimuli, echoic

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

memory for aural stimuli and haptic memory for touch. These memories are constantly overwritten by new information coming in on these channels.

The existence of echoic memory is evidenced by our ability to ascertain the direction from which a sound originates. This is due to information being received by both ears. Since this information is received at different times, we must store the stimulus in the meantime. Echoic memory allows brief _play-back _of information. Information is passed from sensory memory into short-term memory by attention, thereby filtering the stimuli to only those which are of interest at a given time.

Attention is the concentration of the mind on one out of a number of competing stimuli or thoughts. It is clear that we are able to focus our attention selectively, choosing to attend to one thing rather than another. This is due to the limited capacity of our sensory and mental processes.

1.3.2 Short-term memory

Short-term memory or working memory acts as a _scratch-pad‘for temporary recall of information. It is used to store information which is only required fleetingly. Short-term memory can be accessed rapidly, in the order of 70 ms. It also decays rapidly, meaning that information can only be held there temporarily, in the order of 200 ms. Short-term memory also has a limited capacity. There are two basic methods for measuring memory capacity. The first involves determining the length of a sequence which can be remembered in order. The second allows items to be freely recalled in any order.

1.3.3 Long-term memory

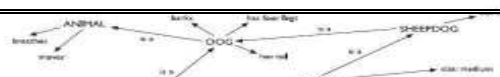
If short-term memory is our working memory or _scratch-pad‘, long-term memory is our main resource. Here we store factual information, experiential knowledge, procedural rules of behavior – in fact, everything that we _know‘. It differs from short-term memory in a number of significant ways. First, it has a huge, if not unlimited, capacity. Secondly, it has a relatively slow access time of approximately a tenth of a second. Thirdly, forgetting occurs more slowly in long-term memory, if at all.

Long-term memory is intended for the long-term storage of information. Information is placed there from working memory through rehearsal. Unlike working memory there is little decay: long-term recall after minutes is the same as that after hours or days.

Long-term memory structure

There are two types of long-term memory: episodic memory and semantic memory. Episodic memory represents our memory of events and experiences in a serial form. It is from this memory that we can reconstruct the actual events that took place at a given point in our lives. Semantic memory, on the other hand, is a structured record of facts, concepts and skills

STUCOR APP



DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

that we have acquired. The information in semantic memory is derived from that in our episodic memory.

Long-term memory processes

This process can be optimized in a number of ways. Ebbinghaus performed numerous experiments on memory, using himself as a subject. In these experiments he tested his ability to learn and repeat nonsense syllables, comparing his recall minutes, hours and days after the learning process. He discovered that the amount learned was directly proportional to the amount of time spent learning. This is known as the total time hypothesis. However, experiments by Baddeley and others suggest that learning time is most effective if it is distributed over time.

There are two main theories of forgetting: decay and interference. The first theory suggests that the information held in long-term memory may eventually be forgotten. Ebbinghaus concluded from his experiments with nonsense syllables that information in memory decayed logarithmically, that is that it was lost rapidly to begin with, and then more slowly. Jost‘s law, which follows from this, states that if two memory traces are equally strong at a given time the older one will be more durable.

The second theory is that information is lost from memory through interference. If we acquire new information it causes the loss of old information. This is termed retroactive interference. A common example of this is the fact that if you change telephone numbers, learning your new number makes it more difficult to remember your old number. This is because the new association masks the old. However, sometimes the old memory trace breaks through and interferes with new information. This is called proactive inhibition.

Forgetting is also affected by emotional factors. In experiments, subjects given emotive words and non-emotive words found the former harder to remember in the short term but easier in the long term.

First, proactive inhibition demonstrates the recovery of old information even after it has been _lost‘ by interference. Secondly, there is the _tip of the tongue‘ experience, which indicates that some information is present but cannot be satisfactorily accessed. Thirdly, information may not be recalled but may be recognized, or may be recalled only with prompting. This leads us to the third process of memory: information retrieval. Here we need to distinguish between two types of information retrieval, recall and recognition. In recall the information is reproduced from memory. In recognition, the presentation of the information provides the knowledge that the information has been seen before. Recognition is the less complex cognitive activity since the information is provided as a cue.

1.4 THINKING: REASONING AND PROBLEM SOLVING

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

Humans, on the other hand, are able to use information to reason and solve problems, and indeed do these activities when the information is partial or unavailable. Human thought is conscious and self-aware: while we may not always be able to identify the processes we use, we can identify the products of these processes, our thoughts. In addition, we are able to think about things of which we have no experience, and solve problems which we have never seen before.

Thinking can require different amounts of knowledge. Some thinking activities are much directed and the knowledge required is constrained. Others require vast amounts of knowledge from different domains. For example, performing a subtraction calculation requires a relatively small amount of knowledge, from a constrained domain, whereas understanding newspaper headlines demands.

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 a number of different Types of reasoning: deductive, inductive and abductive. We use each of these types of reasoning in everyday life, but they differ in significant ways.

Deductive reasoning

Deductive reasoning derives the logically necessary conclusion from the given premises.

For example,

If it is Friday then she will go to work

It is Friday

Therefore she will go to work.

Inductive reasoning

Induction is generalizing from cases we have seen to infer information about cases we have not seen. Induction is a useful process, which we use constantly in learning about our environment. We can never see all the elephants that have ever lived or will ever live, but we have certain knowledge about elephants which we are prepared to trust for all practical purposes, which has largely been inferred by induction. Even if we saw an elephant without a trunk, we would be unlikely to move from our position that _All elephants have trunks‘, since we are better at using positive than negative evidence.

Abductive reasoning

The third type of reasoning is abduction. Abduction reasons from a fact to the action or state that caused it. This is the method we use to derive explanations for the events we observe. For example, suppose we know that Sam always drives too fast when she has been drinking. If we see Sam driving too fast we may infer that she has been drinking. Of course, this too is

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

unreliable since there may be another reason why she is driving fast: she may have been called to an emergency.

Problem solving

Human problem solving is characterized by the ability to adapt the information we have to deal with new situations often solutions seem to be original and creative. There are a number of different views of how people solve problems.

The Gestalt view that problem solving involves both reuse of knowledge and insight. This has been largely superseded but the questions it was trying to address remain and its influence can be seen in later research. In the 1970s by Newell and Simon, was the problem space theory, which takes the view that the mind is a limited information processor.

Gestalt theory

Gestalt psychologists were answering the claim, made by behaviorists, that problem solving is a matter of reproducing known responses or trial and error. This explanation was considered by the Gestalt school to be insufficient to account for human problem-solving behavior. Instead, they claimed, problem solving is both productive and reproductive. Reproductive problem solving draws on previous experience as the behaviorists claimed, but productive problem solving involves insight and restructuring of the problem. Indeed, reproductive problem solving could be a hindrance to finding a solution, since a person may _fixate‘on the known aspects of the problem and so be unable to see novel interpretations that might lead to a solution. Gestalt psychologists backed up their claims with experimental evidence.

Problem space theory

Newell and Simon proposed that problem solving centers on the problem space. The problem space comprises problem states, and problem solving involves generating these states using legal state transition operators. The problem has an initial state and a goal state and people use the operators to move from the former to the latter. Such problem spaces may be huge, and so heuristics are employed to select appropriate operators to reach the goal. One such heuristic is means–ends analysis. In means–ends analysis the initial state is compared with the goal state and an operator chosen to reduce the difference between the two.

Newell and Simon‘s theory, and their General Problem Solver model which is based on it, have largely been applied to problem solving in well-defined domains, for example solving puzzles. These problems may be unfamiliar but the knowledge that is required to solve them is present in the statement of the problem and the expected solution is clear. In real-world problems finding the knowledge required to solve the problem may be part of the problem, or specifying the goal may be difficult.

STUCOR APP

Analogy in problem solving

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

A third element of problem solving is the use of analogy. Similarities between the known domain and the new one are noted and operators from the known domain are transferred to the new one.

Skill acquisition

The entire problem solving that we have considered so far has concentrated on handling unfamiliar problems. A commonly studied domain is chess playing. It is particularly suitable since it lends itself easily to representation in terms of problem space theory. The initial state is the opening board position; the goal state is one player checkmating the other; operators to move states are legal moves of chess. It is therefore possible to examine skilled behavior within the context of the problem space theory of problem solving.

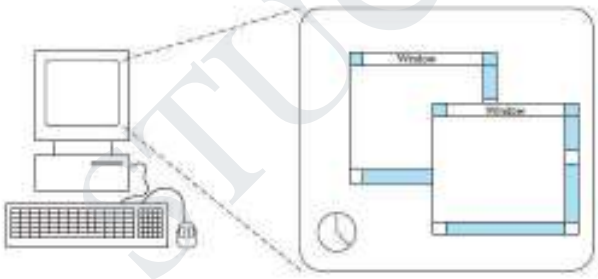
Errors and mental models

Human capability for interpreting and manipulating information is quite impressive. Some are trivial, resulting in no more than temporary inconvenience or annoyance. Others may be more serious, requiring substantial effort to correct.

1.5 THE COMPUTER

A typical computer system

There is the computer _box'itself, a keyboard, a mouse and a colour screen. The screen layout is shown alongside it. Data have to be entered into and obtained from a system, and there are also many different types of user, each with their own unique requirements.



Levels of interaction – batch processing

There was minimal interaction with the machine: the user would simply dump a pile of punched cards onto a reader, press the start button, and then return a few hours later. This still continues today although now with pre-prepared electronic files or possibly machine-read forms. With batch processing the interactions take place over hours or days. In contrast the typical desktop computer system has interactions taking seconds or fractions of a second. The field of Human– Computer Interaction largely grew due to this change in interactive pace.

Richer interaction – everywhere, everyway

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

Information appliances are putting internet access or dedicated systems onto the fridge, microwave and washing machine: to automate shopping, give you email in your kitchen or simply call for maintenance when needed. We carry with us WAP phones and smartcards, have security systems that monitor us and web cams that show our homes to the world.

1.5.1 Elements

A computer system comprises various elements, each of which affects the user of the system.

- Input devices for interactive use, allowing text entry, drawing and selection from the screen:
    - text entry: traditional keyboard, phone text entry, speech and handwriting
    - pointing: principally the mouse, but also touchpad, stylus and others
    - 3D interaction devices.
- Output display devices for interactive use:
    - different types of screen mostly using some form of bitmap display
    - large displays and situated displays for shared and public use
    - digital paper may be usable in the near future.
- Virtual reality systems and 3D visualization which have special interaction and display devices.
- Various devices in the physical world:
    - physical controls and dedicated displays
    - sound, smell and haptic feedback
    - sensors for nearly everything including movement, temperature, bio-signs.
- Paper output and input: the paperless office and the less-paper office:
    - different types of printers and their characteristics, character styles and fonts
    - Scanners and optical character recognition.

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

1.6 Memory

1.6.1 Short term memory (RAM)

At the lowest level of computer memory are the registers on the computer chip, but these having little impact on the user except in so far as they affect the general speed of the computer? Most currently active information is held in silicon-chip random access memory (RAM). Different forms of RAM differ as to their precise access times, power consumption and characteristics. Typical access times are of the order of 10 nanoseconds, that is a hundred-millionth of a second, and information can be accessed at a rate of around 100 Mbytes (million bytes) per second. Typical storage in modern personal computers is between 64 and 256 Mbytes. Most RAM is volatile, that is its contents are lost when the power is turned off. However, many computers have small amount of non-volatile RAM, which retains its contents, perhaps with the aid of a small battery. This may be used to store setup information in a large computer, but in a pocket organizer will be the whole memory. Non-volatile RAM is more expensive so is only used where necessary, but with many notebook computers using very low-power static RAM, the divide is shrinking. By strict analogy, non-volatile RAM ought to be classed as LTM, but the important thing we want to emphasize is the gulf between STM and LTM in a traditional computer system. In PDAs the distinctions become more confused as the battery power means that the system is never completely off, so RAM memory effectively lasts forever. Some also use flash memory, which is a form of silicon memory that sits between fixed content ROM (read-only memory) chips and normal RAM. Flash memory is relatively slow to write, but once written retains its content even with no power whatsoever. These are sometimes called silicon disks on PDAs. Digital cameras typically store photographs in some form of flash media and small flash-based devices are used to plug into a laptop or desktop's USB port to transfer data.

1.6.2 Disks and long-term memory (LTM)

For most computer users the LTM consists of disks, possibly with small tapes for backup. The existence of backups, and appropriate software to generate and retrieve them, is an important area for user security. However, we will deal mainly with those forms of storage that impact the interactive computer user. There are two main kinds of technology used in disks: magnetic disks and optical disks. The most common storage media, floppy disks and hard (or fixed) disks, are coated with magnetic material, like that found on an audio tape, on which the information is stored. Typical capacities of floppy disks lie between 300 kbytes and 1.4 Mbytes, but as they are removable, you can have as many as you have room for on your desk. Hard disks may store from under 40 Mbytes to several gigabytes (Gbytes) that is several thousand million bytes. With disks there are two access times to consider, the time taken to find the right track on the disk, and the time to read the track. The former dominates

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

random reads, and is typically of the order of 10 ms for hard disks. The transfer rate once the track is found is then very high, perhaps several hundred kilobytes per second. Various forms of large removable media are also available, fitting somewhere between floppy disks and removable hard disks, and are especially important for multimedia storage. Optical disks use laser light to read and (sometimes) write the information on the disk. There are various high capacity specialist optical devices, but the most common is the CD-ROM, using the same technology as audio compact discs. CD-ROMs have a capacity of around 650 megabytes, but cannot be written to at all. They are useful for published material such as online reference books, multimedia and software distribution. Recordable CDs are a form of WORM device (write-once read-many) and are more flexible in that information can be written, but (as the name suggests) only once at any location – more like a piece of paper than a blackboard. They are obviously very useful for backups and for producing very secure audit information. Finally, there are fully rewritable optical disks, but the rewrite time is typically much slower than the read time, so they are still primarily for archival not dynamic storage. Many CD-ROM reader/writers can also read DVD format, originally developed for storing movies. Optical media are more robust than magnetic disks and so it is easier to use a jukebox arrangement, whereby many optical disks can be brought online automatically as required. This can give an online capacity of many hundreds of gigabytes. However, as magnetic disk capacities have grown faster than the fixed standard of CD-ROMs, some massive capacity stores are moving to large disk arrays.

1.6.3 Compression

In fact, things are not quite so bad, since compression techniques can be used to reduce the amount of storage required for text, bitmaps and video. All of these things are highly redundant. Consider text for a moment. In English, we know that if we use the letter ‘q’ then ‘u’ is almost bound to follow. At the level of words, some words like ‘the’ and ‘and’ appear frequently in text in general, and for any particular work one can find other common terms (this book mentions ‘user’ and ‘computer’ rather frequently). Similarly, in a bitmap, if one bit is white, there is a good chance the next will be as well. Compression algorithms take advantage of this redundancy. For example, Huffman encoding gives short codes to frequent words and runlength encoding represents long runs of the same value by length value pairs. Text can easily be reduced by a factor of five and bitmaps often compress to 1% of their original size. For video, in addition to compressing each frame, we can take advantage of the fact that successive frames are often similar. We can compute the difference between successive frames and then store only this – compressed, of course. More sophisticated algorithms detect when the camera pans and use this information also. These differencing methods fail when the scene changes, and so the process periodically has to restart and send a

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

new, complete (but compressed) image. For storage purposes this is not a problem, but when used for transmission over telephone lines or networks it can mean glitches in the video as the system catches up. With these reductions it is certainly possible to store low-quality video at 64 Kbyte/s; that is, we can store five hours of highly compressed video on our 1 Gbyte hard disk. However, it still makes the humble video cassette look very good value. Probably the leading edge of video still and photographic compression is fractal compression. Fractals have been popularized by the images of the Mandelbrot set (that swirling pattern of computer-generated colors seen on many T-shirts and posters).Fractals refer to any image that contains parts which, when suitably scaled, are similar to the whole. If we look at an image, it is possible to find parts which are approximately self-similar, and these parts can be stored as a fractal with only a few numeric parameters. Fractal compression is especially good for textured features, which cause problems for other compression techniques. The decompression of the image can be performed to any degree of accuracy, from a very rough soft-focus image, to one more detailed than the original. The former is very useful as one can produce poor-quality output quickly, and better quality given more time. The latter is rather remarkable – the fractal compression actually fills in details that are not in the original.

1.6.4 Storage format and standards

The most common data types stored by interactive programs are text and bitmap images, with increasing use of video and audio, and this subsection looks at the ridiculous range of file storage standards. We will consider database retrieval in the next subsection. The basic standard for text storage is the ASCII (American standard code for information interchange) character codes, which assign to each standard printable character and several control characters an internationally recognized 7 bit code (decimal values 0–127), which can therefore be stored in an 8 bit byte, or be transmitted as 8 bits including parity. Many systems extend the codes to the values 128–255, including line-drawing characters, mathematical symbols and international letters such as ‘æ’. There is a 16 bit extension, the UNICODE standard, which has enough room for a much larger range of characters including the Japanese Kanji character set. As we have already discussed, modern documents consist of more than just characters. The text is in different fonts and includes formatting information such as centering, page headers and footers. On the whole, the storage of formatted text is vendor specific, since virtually every application has its own file format. This is not helped by the fact that many suppliers attempt to keep their file formats secret, or update them frequently to stop others’ products being compatible. With the exception of bare ASCII, the most common shared format is rich text format (RTF), which encodes formatting information including style sheets. However, even where an application will import or export RTF, it may represent a cut-down version of the full document style. RTF regards the document as

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

formatted text, that is it concentrates on the appearance. Documents can also be regarded as structured objects: this book has chapters containing sections, subsections . . . paragraphs, sentences, words and characters. There are ISO standards for document structure and interchange, which in theory could be used for transfer between packages and sites, but these are rarely used in practice. Just as the PostScript language is used to describe the printed page, SGML (standard generalized markup language) can be used to store structured text in a reasonably extensible way. You can define your own structures (the definition itself in SGML), and produce documents according to them. XML (extensible markup language), a lightweight version of SGML, is now used extensively for web-based applications. For bitmap storage the range of formats is seemingly unending. The stored image needs to record the size of the image, the number of bits per pixel, possibly a color map, as well as the bits of the image itself. In addition, an icon may have a 'hot-spot' for use as a cursor. If you think of all the ways of encoding these features, or leaving them implicit, and then consider all the combinations of these different encodings, you can see why there are problems. And all this before we have even considered the effects of compression! There is, in fact, a whole software industry producing packages that convert from one format to another. Given the range of storage standards (or rather lack of standards), there is no easy advice as to which is best, but if you are writing a new word processor and are about to decide how to store the document on disk, think, just for a moment, before defining yet another format.

1.Short-term memory: RAM

2.Long-term memory: magnetic and optical disks

3.capacity limitations related to document and video storage

4.Access methods as they limit or help the user.

1.7 Processing & Networks

The effects when systems run too slow or too fast, the myth of the infinitely fast machine

limitations on processing speed

Networks and their impact on system performance.

1.7.1 Effects of finite processor speed

Speed of processing can seriously affect the user interface. These effects must be taken into account when designing an interactive system. There are two sorts of faults due to processing speed: those when it is too slow, and those when it is too fast!

Example of the former above. This was a functional fault, in that the program did the wrong thing. The system is supposed to draw lines from where the mouse button is depressed to where it is released. However, the program gets it wrong – after realizing the button is down, it does not check the position of the mouse fast enough, and so the user may have moved the mouse before the start position is registered. This is a fault at the implementation

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

stage of the system rather than of the design. But to be fair, the programmer may not be given the right sort of information from lower levels of system software.

A second fault due to slow processing is where, in a sense, the program does the right thing, but the feedback is too slow, leading to strange effects at the interface. In order to avoid faults of the first kind, the system buffers the user input; that is, it remembers key presses and mouse buttons and movement. Unfortunately, this leads to problems of its own. One example of this sort of problem is cursor tracking, which happens in character-based text editors. The user is trying to move backwards on the same line to correct an error, and so presses the cursor-left key. The cursor moves and when it is over the correct position, the user releases the key. Unfortunately, the system is behind in responding to the user, and so has a few more cursor-left keys Moore's law.

Everyone knows that computers just get faster and faster. However, in 1965 Gordon Moore, co-founder of Intel, noticed a regularity. It seemed that the speed of processors, related closely to the number of transistors that could be squashed on a silicon wafer, was doubling every 18 months – exponential growth. One of the authors bought his first 'proper' computer in 1987; it was a blindingly fast 1.47 MHz IBM compatible (Macs were too expensive). By 2002 a system costing the same in real terms would have had a 1.5 GHz processor – 1000 times faster or 210 in 15 years, that is 10 to 18 months. There is a similar pattern for computer memory, except that the doubling time for magnetic storage seems to be closer to one year. For example, when the first edition of this book was written one of the authors had a 20 Mbyte hard disk; now, 11 years later, his disk is 30 Gbytes – around210 times more storage in just 10 years. The effects of this are dramatic. If you took a young baby today and started recording a full audio video diary of every moment, day and night, of that child's life, by the time she was an old lady her whole life experience would fit into memory the size of a small grain of dust.

The computer to process – the cursor then overshoots. The user tries to correct this by pressing the cursor-right key, and again overshoots. There is typically no way for the user to tell whether the buffer is empty or not, except by interacting very slowly with the system and observing that the cursor has moved after every keypress. A similar problem, icon wars, occurs on window systems. The user clicks the mouse on a menu or icon, and nothing happens; for some reason the machine is busy or slow. So the user clicks again, tries something else – then, suddenly, all the buffered mouse clicks are interpreted and the screen becomes a blur of flashing windows and menus. This time, it is not so much that the response is too slow – it is fast enough when it happens – but that the response is variable. The delays due to swapping programs in and out of main memory typically cause these problems. Furthermore, a style of interaction that is optimal on one machine may not be so on a slower

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

machine. In particular, mouse-based interfaces cannot tolerate delays between actions and feedback of more than a fraction of a second, otherwise the immediacy required for successful interaction is lost. If these responses cannot be met then a more old-fashioned, command-based interface may be required. Whereas it is immediately obvious that slow responses can cause problems for the user, it is not so obvious why one should not always aim for a system to be as fast as possible. However, there are exceptions to this – the user must be able to read and understand the output of the system. For example, one of the authors was once given a demonstration disk for a spreadsheet. Unfortunately, the machine the demo was written on was clearly slower than the author's machine, not much, at worst half the speed, but different enough. The demo passed in a blur over the screen with nothing remaining on the screen long enough to read. Many high-resolution monitors suffer from a similar problem when they display text. Whereas older character-based terminals scrolled new text from the bottom of the screen or redrew from the top, bitmap screens often 'flash' up the new page, giving no indication of direction of movement. A final example is the rate of cursor flashing: the rate is often at a fixed.

1.7.2 Limitations on Interactive performance

There are several factors that can limit the speed of an interactive system:

Computation bound

Storage channel bound

Graphics bound

Network capacity

1.7.3 Networked Computing

Computer systems in use today are much more powerful than they were a few years ago, which means that the standard computer on the desktop is quite capable of high-performance interaction without recourse to outside help. However, it is often the case that we use computers not in their standalone mode of operation, but linked together in networks. This brings added benefits in allowing communication between different parties, provided they are connected into the same network, as well as allowing the desktop computer to access resources remote from itself. Such networks are inherently much more powerful than the individual computers that make up the network: increased computing power and memory are only part of the story, since the effects of allowing people much more extensive, faster and easier access to information are highly significant to individuals, groups and institutions.

1.8 INTERACTION

Interaction involves at least two participants: the user and the system. The interface must therefore effectively translate between them to allow the interaction to be successful. This translation can fail at a number of points and for a number of reasons. The use of models

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

of interaction can help us to understand exactly what is going on in the interaction and identify the likely root of difficulties. They also provide us with a framework to compare different interaction styles and to consider interaction problems.

1.8.1 The terms of interaction

The purpose of an interactive system is to aid a user in accomplishing goals from some application domain. A domain defines an area of expertise and knowledge in some real-world activity. Some examples of domains are graphic design, authoring and process control in a factory.

A domain consists of concepts that highlight its important aspects. In a graphic design domain, some of the important concepts are geometric shapes, a drawing surface and a drawing utensil. Tasks are operations to manipulate the concepts of a domain. A goal is the desired output from a performed task. For example, one task within the graphic design domain is the construction of a specific geometric shape with particular attributes on the drawing surface. A related goal would be to produce a solid red triangle centered on the canvas. An intention is a specific action required to meet the goal.

1.8.2 The execution–evaluation cycle

The interactive cycle can be divided into two major phases: execution and evaluation. These can then be subdivided into further stages, seven in all. The stages in Norman‘s model of interaction are as follows:

1. Establishing the goal.
2. Forming the intention.
3. Specifying the action sequence.
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.

It is liable to be imprecise and therefore needs to be translated into the more specific intention, and the actual actions that will reach the goal, before it can be executed by the user. The user perceives the new state of the system, after execution of the action sequence, and interprets it in terms of his expectations. If the system state reflects the user‘s goal then the computer has done what he wanted and the interaction has been successful; otherwise the user must formulate a new goal and repeat the cycle.

Norman uses this model of interaction to demonstrate why some interfaces cause problems to their users. He describes these in terms of the gulfs of execution and the gulfs of evaluation. As we noted earlier, the user and the system do not use the same terms to describe the domain and goals – remember that we called the language of the system the core language

STUCOR APP
DOWNLOADED FROM STUCOR APP

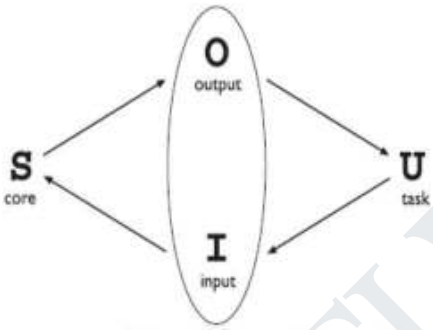
DOWNLOADED FROM STUCOR APP

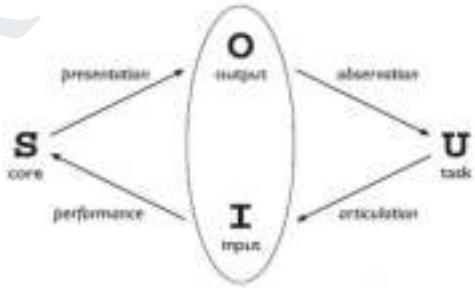
STUCOR APP

and the language of the user the task language. The gulf of execution is the difference between the user's formulation of the actions to reach the goal and the actions allowed by the system. If the actions allowed by the system correspond to those intended by the user, the interaction will be effective. The interface should therefore aim to reduce this gulf. The gulf of evaluation is the distance between the physical presentation of the system state and the expectation of the user. If the user can readily evaluate the presentation in terms of his goal, the gulf of evaluation is small. The more effort that is required on the part of the user to interpret the presentation, the less effective the interaction.

1.8.3 The interaction framework

The interaction framework attempts a more realistic description of interaction by including the system explicitly, and breaks it into four main components. The nodes represent the four major components in an interactive system – the System, the User, the Input and the Output. Each component has its own language. In addition to the User's task language and the System's core language, which we have already introduced, there are languages for both the Input and Output components. Input and Output together form the Interface.





The general interaction framework Translations between components

The System then transforms itself as described by the operations; the execution phase of the cycle is complete and the evaluation phase now begins. The System is in a new state, which must now be communicated to the User. The current values of system attributes are rendered as concepts or features of the Output. It is then up to the User to observe the Output and assess the results of the interaction relative to the original goal, ending the evaluation phase and, hence, the interactive cycle. There are four main translations involved in the interaction: articulation, performance, presentation and observation.

Assessing overall interaction

The interaction framework is presented as a means to judge the overall usability of an entire interactive system. This is not surprising since it is only in attempting to perform a particular task within some domain that we are able to determine if the tools we use are adequate. For a particular editing task, one can choose the text editor best suited for interaction relative to the task. The best editor, if we are forced to choose only one, is the one

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

that best suits the tasks most frequently performed. Therefore, it is not too disappointing that we cannot extend the interaction analysis beyond the scope of a particular task.

1.9 MODELS

FRAMEWORKS AND HCI

The field of ergonomics addresses issues on the user side of the interface, covering input and output, as well as the user's immediate context. Dialog design and interface styles can be placed particularly along the input branch of the framework, addressing both articulation and performance.

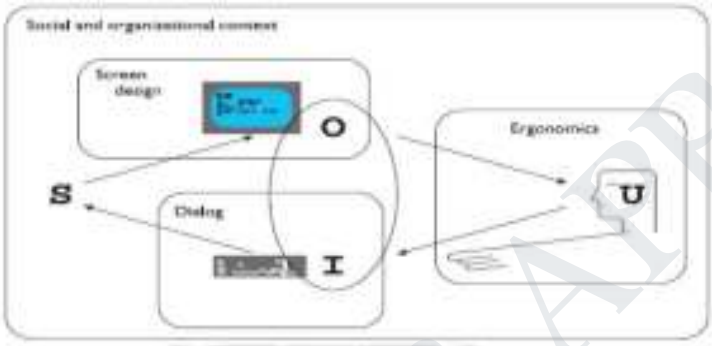


Figure: A framework for human–computer interaction.

Presentation and screen design relates to the output branch of the framework. The entire framework can be placed within a social and organizational context that also affects the interaction. Each of these areas has important implications for the design of interactive systems and the performance of the user.

1.10 ERGONOMICS

Ergonomics (or human factors) is traditionally the study of the physical characteristics of the interaction: how the controls are designed, the physical environment in which the interaction takes place, and the layout and physical qualities of the screen. A primary focus is on user performance and how the interface enhances or detracts from this. In seeking to evaluate these aspects of the interaction, ergonomics will certainly also touch upon human psychology and system constraints. It is a large and established field, which is closely related to but distinct from HCI, and full coverage would demand a book in its own right. Here we consider a few of the issues addressed by ergonomics as an introduction to the field. We will briefly look at the arrangement of controls and displays, the physical environment, health issues and the use of colour. These are by no means exhaustive and are intended only to give an indication of the types of issues and problems addressed by ergonomics.

1.10.1Arrangement of controls and displays

The exact organization that this will suggest will depend on the domain and the application, but possible organizations include the following:

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

- Functional controls and displays are organized so that those that are functionally related are placed together;
- Sequential controls and displays are organized to reflect the order of their use in a typical interaction (this may be especially appropriate in domains where a particular task sequence is enforced, such as aviation);
- Frequency controls and displays are organized according to how frequently they are used, with the most commonly used controls being the most easily accessible.

The physical environment of the interaction

Physical issues in the layout and arrangement of the machine interface, ergonomics is concerned with the design of the work environment itself. This will depend largely on the domain and will be more critical in specific control and operational settings than in general computer use. The physical environment in which the system is used may influence how well it is accepted and even the health and safety of its users. It should therefore be considered in all design. The first consideration here is the size of the users. Obviously this is going to vary considerably. All users should be comfortably able to see critical displays. For long periods of use, the user should be seated for comfort and stability. Seating should provide back support. If required to stand, the user should have room to move around in order to reach all the controls.

1.10.2 Health issues

There are a number of factors that may affect the use of more general computers. Again these are factors in the physical environment that directly affect the quality of the interaction and the user‘s

Performance:

Users should be able to reach all controls comfortably and see all displays. Users should not be expected to stand for long periods and, if sitting, should be provided with back support. If a particular position for a part of the body is to be adopted for long periods (for example, in typing) support should be provided to allow rest.

Temperature

Extremes of hot or cold will affect performance and, in excessive cases, health. Experimental studies show that performance deteriorates at high or low temperatures, with users being unable to concentrate efficiently.

Lighting The lighting level will again depend on the work environment. Adequate lighting should be provided to allow users to see the computer screen without discomfort or eyestrain. The light source should also be positioned to avoid glare affecting the display.

Noise Excessive noise can be harmful to health, causing the user pain, and in acute cases, loss of hearing. Noise levels should be maintained at a comfortable level in the work environment.

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

This does not necessarily mean no noise at all. Noise can be a stimulus to users and can provide needed confirmation of system activity.

Time The time users spend using the system should also be controlled. it has been suggested that excessive use of CRT displays can be harmful to users, particularly pregnant women.

The use of color

Colors used in the display 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 used should also correspond to common conventions and user expectations. Red, green and yellow are colors frequently associated with stop, go and standby respectively. Therefore, red may be used to indicate emergency and alarms; green, normal activity; and yellow, standby and auxiliary function. These conventions should not be violated without very good cause.

Ergonomics and HCI

Ergonomics is a huge area, which is distinct from HCI but sits alongside it. Its 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.

1.11 INTERACTION STYLES

Interaction can be seen as a dialog between the computer and the user. The choice of interface style can have a profound effect on the nature of this dialog. There are a number of common interface styles including

- command line interface
- menus
- natural language
- question/answer and query dialog
- form-fills and spreadsheets
- WIMP
- point and click
- Three-dimensional interfaces.

1.11.1 Command line interface

The command line interface was the first interactive dialog style to be commonly used and, in spite of the availability of menu-driven interfaces, it is still widely

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

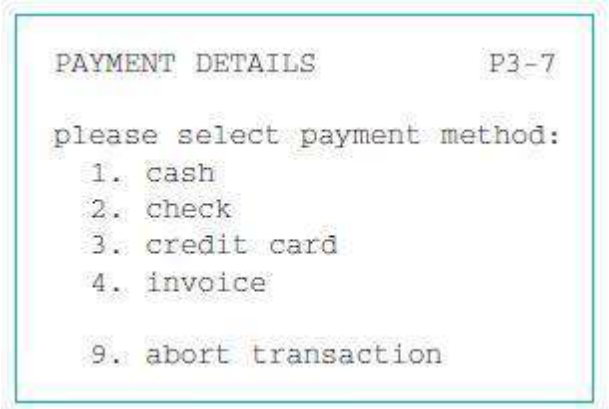
STUCOR APP

used. It provides a means of expressing instructions to the computer directly, using function keys, single characters, abbreviations or whole-word commands. In some systems the command line is the only way of communicating with the system, especially for remote access using telnet. Menu-based interfaces, providing accelerated access to the system‘s functionality for experienced users. Command line interfaces are powerful in that they offer direct access to system functionality and can be combined to apply a number of tools to the same data. They are also flexible: the command often has a number of options or parameters that will vary its behavior in some way, and it can be applied to many objects at once, making it useful for repetitive tasks. Flexibility and power brings with it difficulty in use and learning.

Commands must be remembered, as no cue is provided in the command line to indicate which command is needed. They are therefore better for expert users than for novices. This problem can be alleviated a little by using consistent and meaningful commands and abbreviations. The commands used should be terms within the vocabulary of the user rather than the technician. Unfortunately, commands are often obscure and vary across systems, causing confusion to the user and increasing the overhead of learning.

1.11.2 Menus

In a menu-driven interface, the set of options available to the user is displayed on the screen, and selected using the mouse, or numeric or alphabetic keys. Since the options are visible they are less demanding of the user, relying on recognition rather than recall. Menu options still need to be meaningful and logically grouped to aid recognition. Often menus are hierarchically ordered and the option required is not available at the top layer of the hierarchy. The grouping and naming of menu options then provides the only cue for the user to find the required option. Such systems either can be purely text based, with the menu options being presented as numbered choices, or may have a graphical component in which the menu appears within a rectangular box and choices are made, perhaps by typing the initial letter of the desired selection, or by entering the associated number, or by moving around the menu with the arrow keys.



STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

Figure Menu-driven interface

1.11.3 Natural language

Users, unable to remember a command or lost in a hierarchy of menus, may long for the computer that is able to understand instructions expressed in everyday words! Natural language understanding, both of speech and written input, is the subject of much interest and research. the ambiguity of natural language makes it very difficult for a machine to understand. Language is ambiguous at a number of levels. First, the syntax, or structure, of a phrase may not be clear. If we are given the sentence—The boy hit the dog with the stick‖

1.11.4 Question/answer and query dialog

Question and answer dialog is a simple mechanism for providing input to an application in a specific domain. The user is asked a series of questions (mainly with yes/no responses, multiple choice, or codes) and so is led through the interaction step by step These interfaces are easy to learn and use, but are limited in functionality and power. As such, they are appropriate for restricted domains (particularly information systems) and for novice or casual users.

Form-fills and spreadsheets

Form-filling interfaces are used primarily for data entry but can also be useful in data retrieval applications. The user is presented with a display resembling a paper form, with slots to fill in .Often the form display is based upon an actual form with which the user is familiar, which makes the interface easier to use. The user works through the form, filling in appropriate values. The data are then entered into the application in the correct place. Most form-filling interfaces allow easy movement around the form and allow some fields to be left blank. They also require correction facilities, as users may change their minds or make a mistake about the value that belongs in each field. The dialog style is useful primarily for data entry applications and, as it is easy to learn and use, for novice users.

Spreadsheets are a sophisticated variation of form filling. The spreadsheet comprises a grid of cells, each of which can contain a value or a formula. The formula can involve the values of other cells (for example, the total of all cellsin this column). 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. The user can therefore manipulate values to see the effects of changing different parameters. Spreadsheets are an attractive medium for interaction: the user is free to manipulate values at will and the distinction between input and output is blurred, making the interface more flexible and natural.

The WIMP interface

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

WIMP stands for windows, icons, menus and pointers (sometimes windows, icons, mice and pull-down menus), and is the default interface style for the majority of interactive computer systems in use today, especially in the PC and desktop workstation arena. Examples of WIMP interfaces include Microsoft Windows for IBM PC compatibles, MacOS for Apple Macintosh compatibles and various X Windows-based systems for UNIX.

Point-and-click interfaces

This point-and-click interface style is obviously closely related to the WIMP style. It clearly overlaps in the use of buttons, but may also include other WIMP elements. The philosophy is simpler and more closely tied to ideas of hypertext. In addition, the point-and-click style is not tied to mouse-based interfaces, and is also extensively used in touch screen information systems. In this case, it is often combined with a menu-driven interface. The point-and-click style has been popularized by World Wide Web pages, which incorporate all the above types of point-and-click navigation: highlighted words, maps and iconic buttons.

1.11.5 Three-dimensional interfaces

There is an increasing use of three-dimensional effects in user interfaces. The most obvious example is virtual reality, but VR is only part of a range of 3D techniques available to the interface designer. The simplest technique is where ordinary WIMP elements, buttons, scroll bars, etc., are given a 3D appearance using shading, giving the appearance of being sculpted out of stone. By unstated convention, such interfaces have a light source at their top right. Where used judiciously, the raised areas are easily identifiable and can be used to highlight active areas. Some interfaces make indiscriminate use of sculptural effects, on every text area, border and menu, so all sense of differentiation is lost.

1.12 ELEMENTS INTERACTIVITY

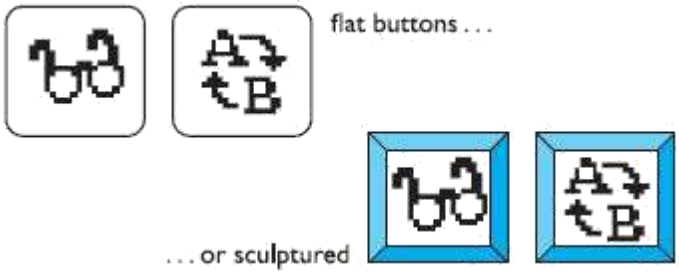
Dialog design is focused almost entirely on the choice and specification of appropriate sequences of actions and corresponding changes in the interface state. It is typically not used

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

at a fine level of detail and deliberately ignores the _semantic'level of an interface: for example, the validation of numeric information in a forms-based system.



It is worth remembering that interactivity is the defining feature of an interactive system. This can be seen in many areas of HCI. For example, the recognition rate for speech recognition is too low to allow transcription from tape, but in an airline reservation system, so long as the system can reliably recognize yes and no it can reflect back its understanding of what you said and seek confirmation. Speech-based input is difficult, speech-based interaction easier. Also, in the area of information visualization the most exciting developments are all where users can interact with visualization in real time, changing parameters and seeing the effect.

Interactivity is also crucial in determining the feel'of a WIMP environment. All WIMP systems appear to have virtually the same elements: windows, icons, menus, pointers, dialog boxes, buttons, etc. In fact, menus are a major difference between the MacOS and Microsoft Windows environments: in MacOS you have to keep the mouse depressed throughout menu selection; in Windows you can click on the menu bar and a pull-down menu appears and remains there until an item is selected or it is cancelled. Similarly the detailed behavior of buttons is quite complex.

1.13 PARADIGMS

1.13.1 Paradigms of Interaction

The paradigms of interaction are

Time sharing

Major contributions to come out of this new emphasis in research were the concept of time sharing, in which a single computer could support multiple users. The human (or more accurately, the programmer) was restricted to batch sessions, in which complete jobs were submitted on punched cards or paper tape to an operator who would then run them individually on the computer. Time-sharing systems of the 1960s made programming a truly interactive venture and brought about a subculture of programmers known as _hackers' – single-minded masters of detail who took pleasure in understanding complexity. Though the purpose of the first interactive time-sharing systems was simply to augment the programming capabilities of the early hackers, it marked a significant stage in computer applications for

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

human use. Rather than rely on a model of interaction as a pre-planned activity that resulted in a complete set of instructions being laid out for the computer to follow, truly interactive exchange between programmer and computer was possible. The computer could now project itself as a dedicated partner with each individual user and the increased throughput of information between user and computer allowed the human to become a more reactive and spontaneous collaborator.

Video display units

In mid-1950s researchers were experimenting with the possibility of presenting and manipulating information from a computer in the form of images on a video display unit (VDU). These display screens could provide a more suitable medium than a paper printout for presenting vast quantities of strategic information for rapid assimilation. The earliest applications of display screen images were developed in military applications, most notably the Semi-Automatic Ground Environment (SAGE) project of the US Air Force.

Programming toolkits

Douglas Engelbart‘s ambition since the early 1950s was to use computer technology as a means of complementing human problem-solving activity.

Personal computing

Programming toolkits provide a means for those with substantial computing skills to increase their productivity greatly. One of the first demonstrations that the powerful tools of the hacker could be made accessible to the computer novice was a graphics programming language for children called LOGO. A child could quite easily pretend they were inside‘the turtle and direct it to trace out simple geometric shapes, such as a square or a circle. By typing in English phrases, such as go forward or Turn left, the child/programmer could teach the turtle to draw more and more complicated figures. By adapting the graphical programming language to a model which children could understand and use, Paper demonstrated a valuable maxim for interactive system development – no matter how powerful a system may be, it will always be more powerful if it is easier to use.

Window systems and the WIMP interface

Humans are able to think about more than one thing at a time, and in accomplishing some piece of work, they frequently interrupt their current train of thought to pursue some other related piece of work. A personal computer system which forces the user to progress in order through all of the tasks needed to achieve some objective, from beginning to end without any diversions, does not correspond to that standard working pattern.

One presentation mechanism for achieving this dialog partitioning is to separate physically the presentation of the different logical threads of user–computer conversation on

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

the display device. The window is the common mechanism associated with these physically and logically separate display spaces.

The metaphor

Papert used the metaphor of a turtle dragging its tail in the dirt. Children could quickly identify with the real-world phenomenon and that instant familiarity gave them an understanding of how they could create pictures. The danger of a metaphor is usually realized after the initial honeymoon period. When word processors were first introduced, they relied heavily on the typewriter metaphor. The keyboard of a computer closely resembles that of a standard typewriter, so it seems like a good metaphor from which to start.

Hypertext

Hypertext is text which is not constrained to be linear. Hypertext is text which contains links to other texts. The term was coined by Ted Nelson around 1965. HyperMedia is a term used for hypertext which is not constrained to be text: it can include graphics, video and sound, for example. Apparently Ted Nelson was the first to use this term too. Hypertext and HyperMedia are concepts, not products.

Multi-modality

Genuine multi-modal systems rely to a greater extent on simultaneous use of multiple communication channels for both input and output. Humans quite naturally process information by simultaneous use of different channels. We point to someone and refer to them as you‘, and it is only by interpreting the simultaneous use of voice and touch that our directions are easily articulated and understood. Designers have wanted to mimic this flexibility in both articulation and observation by extending the input and output expressions an interactive system will support. So, for example, we can modify a gesture made with a pointing device by speaking, indicating what operation is to be performed on the selected object.

Computer-supported cooperative work

Personal computing provides individuals with enough computing power so that they were liberated from dumb terminals which operated on a time-sharing system. It is interesting to note that as computer networks became widespread, individuals retained their powerful workstations but now wanted to reconnect themselves to the rest of the workstations in their immediate working environment, and even throughout the world! One result of this reconnection was the emergence of collaboration between individuals via the computer – called computer-supported cooperative work, or CSCW.

The World Wide Web

WWW or "Web" is a global information medium which users can read and write via computers connected to the Internet. The term is often mistakenly used as a synonym for the

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

Internet itself, but the Web is a service that operates over the Internet, just as e-mail also does. The history of the Internet dates back significantly further than that of the World Wide Web.

The internet is simply a collection of computers, each linked by any sort of data connection, whether it be slow telephone line and modem or high-bandwidth optical connection. The computers of the internet all communicate using common data transmission protocols (TCP/IP) and addressing systems (IP addresses and domain names). This makes it possible for anyone to read anything from anywhere, in theory, if it conforms to the protocol. The web builds on this with its own layer of network protocol (http), a standard markup notation (such as HTML) for laying out pages of information and a global naming scheme (uniform resource locators or URLs). Web pages can contain text, color images, movies, sound and, most important, hypertext links to other web pages. Hypermedia documents can therefore be _published'by anyone who has access to a computer connected to the internet.

Ubiquitous computing

Ubiquitous computing is a paradigm in which the processing of information is linked with each activity or object as encountered. It involves connecting electronic devices, including embedding microprocessors to communicate information. Devices that use ubiquitous computing have constant availability and are completely connected. Ubiquitous computing focuses on learning by removing the complexity of computing and increases efficiency while using computing for different daily activities. Ubiquitous computing is also known as pervasive computing, every ware and ambient intelligence.

STUCOR APP
STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

UNIT II

DESIGN & SOFTWARE PROCESS

Interactive Design basics – process – scenarios – navigation – screen design – Iteration and prototyping. HCI in software process – software life cycle – usability engineering – Prototyping in practice – design rationale. Design rules – principles, standards, guidelines, rules. Evaluation Techniques – Universal Design.

2.1 INTERACTION DESIGN BASICS

Interaction design is about creating interventions in often complex situations using technology of many kinds including PC software, the web and physical devices.

- Design involves:

  1 Achieving goals within constraints and trade-off between these

  2Understanding the raw materials: computer and human

  3 Accepting limitations of humans and of design.

- The design process has several stages and is iterative and never complete.
- Interaction starts with getting to know the users and their context:

  1. Finding out who they are and what they are like . . .probably not like you!

  2 Talking to them, watching them.

- Scenarios are rich design stories, which can be used and reused throughout design:

  they help us see what users will want to do

  they give a step-by-step walkthrough of users‘ interactions: including what they see, do and are thinking.

- Users need to find their way around a system. This involves:

  1. Helping users know where they are, where they have been and what they can do next

  2. Creating overall structures that are easy to understand and fit the users‘ needs

  3 Designing comprehensible screens and control panels.

- Complexity of design means we don‘t get it right first time:

  1. So we need iteration and prototypes to try out and evaluate

  2. But iteration can get trapped in local maxima; designs that have no simple improvements, but are not good theory and models can help give good start points.

WHAT IS DESIGN?

A simple definition is: achieving goals within constraints

Goals: what is the purpose of the design we are intending to produce? Who is it for? Why do they want it? For example, if we are designing a wireless personal movie player, we may think about young affluent users wanting to watch the latest movies whilst on the move and download free copies, and perhaps wanting to share the experience with a few friends.

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

Constraints: What materials must we use? What standards must we adopt? How much can it cost? How much time do we have to develop it? Are there health and safety issues? In the case of the personal movie player: does it have to withstand rain? Must we use existing video standards to download movies? Do we need to build in copyright protection?

Trade-off Choosing which goals or constraints can be relaxed so that others can be met. For example, we might find that an eye-mounted video display, a bit like those used in virtual reality, would give the most stable image whilst walking along. However, this would not allow you to show friends, and might be dangerous if you were watching a gripping part of the movie as you crossed the road.

The golden rule of design

The designs we produce may be different, but often the raw materials are the same. This leads us to the golden rule of design: **understand your materials**

- understand computers
  - limitations, capacities, tools, platforms
- understand people
  - psychological, social aspects, human error.

2.2PROCESS

2.2.1 The Process of Design

A system has been designed and built, and only when it proves unusable do they think to ask how to do it right! In other companies usability is seen as equivalent to testing – checking whether people can use it and fixing problems, rather than making sure they can from the beginning. In the best companies, however, usability is designed in from the start.

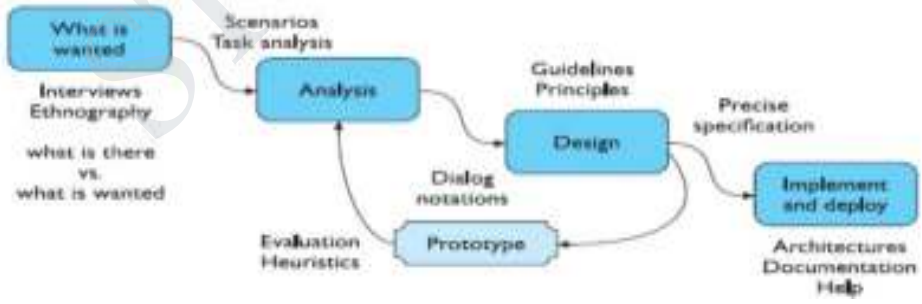


Figure: Interaction design process

Requirements – what is wanted The first stage is establishing what exactly is needed. As a precursor to this it is usually necessary to find out what is currently happening.

Analysis: The results of observation and interview need to be ordered in some way to bring out key issues and communicate with later stages of design.

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

Design: Well, this is all about design, but there is a central stage when you move from what you want, to how to do it. There are numerous rules, guidelines and design principles that can be used to help

Iteration and prototyping: Humans are complex and we cannot expect to get designs right first time. We therefore need to evaluate a design to see how well it is working and where there can be improvements.

Implementation and deployment Finally, when we are happy with our design, we need to create it and deploy it. This will involve writing code, perhaps making hardware, writing documentation and manuals – everything that goes into a real system that can be given to others.

2.3 Scenarios

Scenarios are stories for design: rich stories of interaction. They are perhaps the simplest design representation, but one of the most flexible and powerful. Some scenarios are quite short: ‗the user intends to press the ―save‖ button, but accidentally presses the ―quit‖ button so loses his work'. Others are focussed more on describing the situation or context.

Scenarios force you to think about the design in detail and notice potential problems before they happen. What is the system doing now?'. This can help to verify that the design would make sense to the user and also that proposed implementation architectures would work.

In addition scenarios can be used to:

Communicate with others – other designers, clients or users. It is easy to misunderstand each other whilst discussing abstract ideas. Concrete examples of use are far easier to share.

Validate other models: A detailed scenario can be ‗played' against various more formal representations such as task models or dialog and navigation models .

Express dynamics Individual screen shots and pictures give you a sense of what a system would look like, but not how it behaves.

2.4 Navigation Design

Navigation Design is the process or activity of accurately ascertaining one's position and planning and following a route. the process or activity of accurately ascertaining one's position and planning and following a route.

Widgets The appropriate choice of widgets and wording in menus and buttons will help you know how to use them for a particular selection or action.

Screens or windows You need to find things on the screen, understand the logical grouping of buttons.

Navigation within the application You need to be able to understand what will happen when a button is pressed, to understand where you are in the interaction.

STUCOR APP
DOWNLOADED FROM STUCOR APP

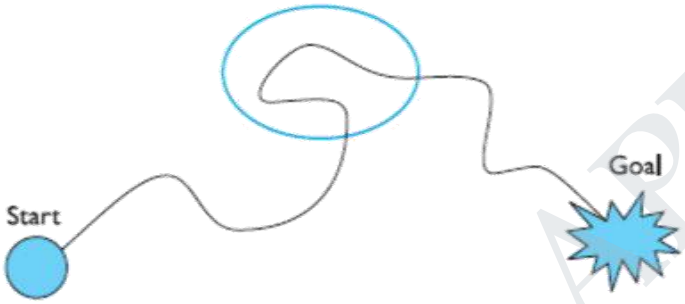
DOWNLOADED FROM STUCOR APP

STUCOR APP

Environment The word processor has to read documents from disk, perhaps some are on remote networks. You swap between applications, perhaps cut and paste.

2.4.1 Local structure

In an ideal world if users had perfect knowledge of what they wanted and how the system worked they could simply take the shortest path to what they want, pressing all the right buttons and links. The important thing is not so much that they take the most efficient route, but that at each point in the interaction they can make some assessment of whether they are getting closer to their (often partially formed) goal.



To do this goal seeking, each state of the system or each screen needs to give the user enough knowledge of what to do to get closer to their goal.

- knowing where you are
- knowing what you can do
- knowing where you are going – or what will happen
- knowing where you‘ve been – or what you‘ve done.

2.4.2 Global structure – hierarchical organization

The hierarchy links screens, pages or states in logical groupings. The Diagram gives a high-level breakdown of some sort of messaging system. This sort of hierarchy can be used purely to help during design, but can also be used to structure the actual system. For example, this may reflect the menu structure of a PC application or the site structure on the web. Any sort of information structuring is difficult, but there is evidence that people find hierarchies simpler than most. One of the difficulties with organizing information or system functionality is that different people have different internal structures for their knowledge, and may use different vocabulary.

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

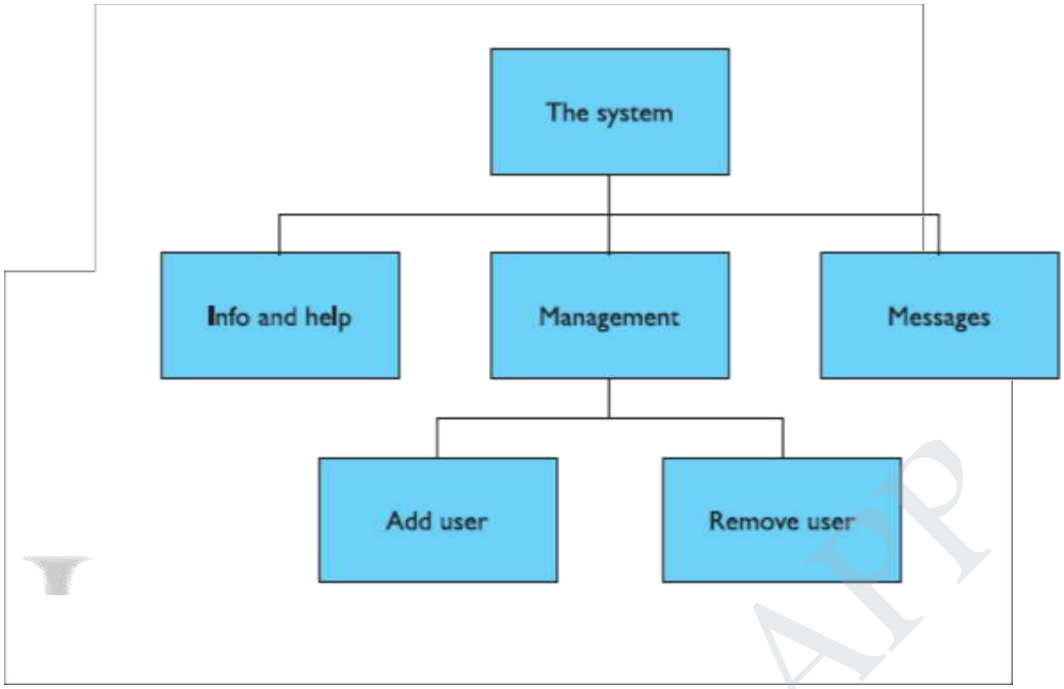


Figure: Application functional hierarchy

2.3 SCREEN DESIGN AND LAYOUT

2.3.1 Tools for layout

We have a number of visual tools available to help us suggest to the user appropriate ways to read and interact with a screen or device.

Billing details:
Name:
Address: ...
Credit card no:

Delivery details:
Name:
Address: ...
Delivery time:

Order details:

item	quantity	cost/item	cost
size 10 screws (boxes)	7	3.71	25.97
... ...	...	...	...

Figure: Grouping related items in an order screen

Grouping and structure

If things logically belong together, then we should normally physically group them together. This may involve multiple levels of structure. We can see a potential design for an ordering screen. Notice how the details for billing and delivery are grouped together spatially; also note how they are separated from the list of items actually ordered by a line as well as spatially. This reflects the following logical structure:

Order:

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

- Administrative information
  - Billing details
  - Delivery details
- Order information
  - Order line 1
  - Order line 2

Order of groups and items

In general we need to think: what is the natural order for the user? This should normally match the order on screen. For data entry forms or dialog boxes we should also set up the order in which the tab key moves between fields. Occasionally we may also want to force a particular order; for example we may want to be sure that we do not forget the credit card details

Decoration

Decorative features like font style, and text or background colors can be used to emphasize groupings.

Alignment

Alignment of lists is also very important. For users who read text from left to right, lists of text items should normally be aligned to the left. Numbers, however, should normally be aligned to the right (for integers) or at the decimal point. This is because the shape of the column then gives an indication of magnitude – a sort of mini histogram. Items like names are particularly difficult.

White space

Spacing or whitespace, white space is any section of a document that is unused or space around an object. White spaces help separate paragraphs of text, graphics, and other portions of a document, and help a document look less crowded. Using white space effectively in a document keeps the reader reading the document and helps the reader quickly find what they are interested in reading.

How to create white space

White space is created by pressing the return key, spacebar key, or the tab key and can also be created by setting the document's margins and inserting form feeds or tables.

2.3.2 User action and control

- Entering information

In each case the screen consists not only of information presented to the user, but also of places for the user to enter information or select options. Many of the same layout issues for data presentation also apply to fields for data entry. Alignment is still important. It is

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

especially common to see the text entry boxes aligned in a jagged fashion because the field names are of different lengths. This is an occasion where right-justified text for the field labels may be best or, alternatively, in a graphical interface a smaller font can be used for field labels and the labels placed just above and to the left of the field they refer to. For both presenting and entering information a clear logical layout is important.

The task analysis techniques can help in determining how to group screen items and also the order in which users are likely to want to read them or fill them in. Knowing also that users are likely to read from left to right and top to bottom (depending on their native language!) means that a screen can be designed so that users encounter items in an appropriate order for the task at hand.

- **Knowing what to do**

If everyone designs buttons to look the same and menus to look the same, then users will be able to recognize them when they see them. It is important that the labels and icons on menus are also clear. Standards can help for common actions such as save, delete or print. For more system-specific actions, one needs to follow broader principles. For example, a button says _bold‘: does this represent the current state of a system or the action that will be performed if the button is pressed?

- **Affordances**

These are especially difficult problems in multimedia applications where one may deliberately adopt a non-standard and avant-garde style. How are users supposed to know where to click? The psychological idea of affordance says that things may suggest by their shape and other attributes what you can do to them: a handle affords pulling or lifting; a button affords pushing. These affordances can be used when designing novel interaction elements. One can either mimic real-world objects directly, or try to emulate the critical aspects of those objects. What you must not do is depict a real-world object in a context where its normal affordances do not work!

2.3.3 Appropriate appearance

- **Presenting information**

The way of presenting information on screen depends on the kind of information: text, numbers, maps, tables; on the technology available to present it: character display, line drawing, graphics, and virtual reality; and, most important of all, on the purpose for which it is being used. The file listing is alphabetic, which is fine if we want to look up the details of a particular file, but makes it very difficult to find recently updated files. Of course, if the list were ordered by date then it would be difficult to find a particular file. Different purposes require different representations. For more complex numerical data, we may be considering scatter graphs, histograms or 3D surfaces; for hierarchical structures, we may consider

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

outlines or organization diagrams. But, no matter how complex the data, the principle of matching presentation to purpose remains. 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.

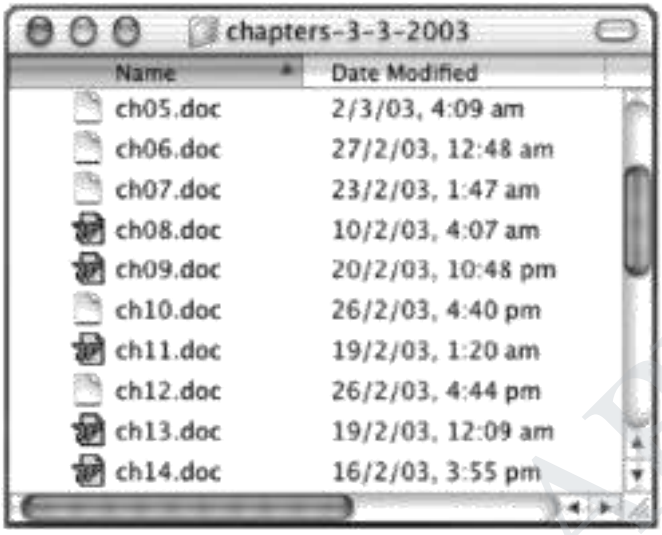


Figure : Alphabetic file listing. Screen shot reprinted by permission from Apple Computer, Inc.

Aesthetics and utility

The beauty and utility may sometimes be at odds. For example, an industrial control panel will often be built up of the individual controls of several subsystems, some designed by different teams, some bought in. The resulting inconsistency in appearance may look a mess and suggest tidying up. Certainly some of this inconsistency may cause problems.

The conflict between aesthetics and utility can also be seen in many _well designed' posters and multimedia systems. In particular, the backdrop behind text must have low contrast in order to leave the text readable; this is often not the case and graphic designers may include excessively complex and strong backgrounds because they look good. The results are impressive, perhaps even award winning, but completely unusable! In consumer devices these aesthetic considerations may often be the key differentiator betweenproducts, for example, the sleek curves of a car. This is not missed by designers of electronic goods: devices are designed to be good to touch and feel as well as look at and this is certainly one of the drivers for the futuristic shapes of the Apple iMac family.

Making a mess of it: colour and 3D

The increasing use of 3D effects in interfaces has posed a whole new set of problems for text and numerical information. Whilst excellent for presenting physical information and certain sorts of graphs, text presented in perspective can be very difficult to read and the all too common 3D pie chart is all but useless.

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

Localization / internationalization

If you are working in a different country, you might see a document being word processed where the text of the document and the file names are in the local language, but all the menus and instructions are still in English. The process of making software suitable for different languages and cultures is called localization or internationalization.

It is clear that words have to change and many interface construction toolkits make this easy by using resources. When the program uses names of menu items, error messages and other text, it does not use the text directly, but instead uses a resource identifier, usually simply a number. A simple database is constructed separately that binds these identifiers to particular words and phrases. A different resource database is constructed for each language, and so the program can be customized to use in a particular country by simply choosing the appropriate resource database.

2.4 ITERATION AND PROTOTYPING

All interaction design includes some form of iteration of ideas. This often starts early on with paper designs and storyboards being demonstrated to colleagues and potential users. Any of these prototypes, whether paper-based or running software, can then be evaluated to see whether they are acceptable and where there is room for improvement. This sort of evaluation, intended to improve designs, is called formative evaluation. This is in contrast to summative evaluation, which is performed at the end to verify whether the product is good enough. One approach is to get an expert to use a set of guidelines, for example the _knowing where you are‘ list above, and look screen by screen to see if there are any violations.

The other main approach is to involve real users either in a controlled experimental setting, or _in the wild‘ – a real-use environment. The result of evaluating the system will usually be a list of faults or problems and this is followed by a redesign exercise, which is then prototyped, evaluated The end point is when there are no more problems that can economically be fixed. So iteration and prototyping are the universally accepted _best practice ‘approach for interaction design.

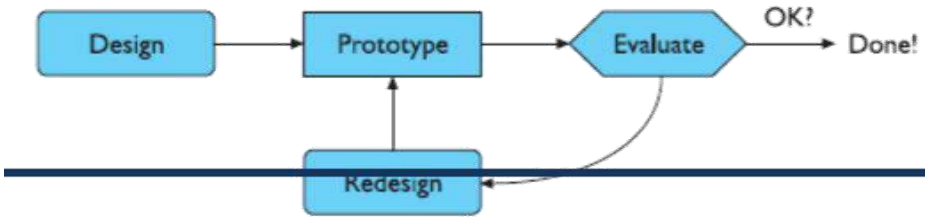


Figure :Role of prototyping

Prototyping is an example of what is known as a hill-climbing approach. Imagine you are standing somewhere in the open countryside. You walk uphill and keep going uphill as steeply as possible. Eventually you will find yourself at a hill top.is

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

exactly how iterative prototyping works: you start somewhere, evaluate it to see how to make it better, change it to make it better and then keep on doing this until it can‘t get any better.

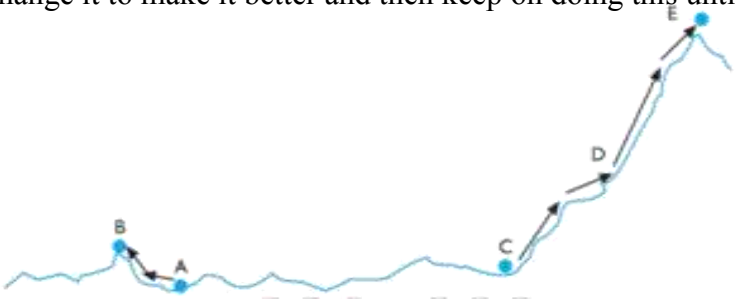


Figure: Moving little by little ..... but to where?

2.5 HCI IN THE SOFTWARE PROCESS

- Software engineering provides a means of understanding the structure of the design process, and that process can be assessed for its effectiveness in interactive system design.
- Usability engineering promotes the use of explicit criteria to judge the success of a product in terms of its usability.
- Iterative design practices work to incorporate crucial customer feedback early in the design process to inform critical decisions which affect usability.
- Design involves making many decisions among numerous alternatives. Design rationale provides an explicit means of recording those design decisions and the context in which the decisions were made.

2.6 Software Life cycle models

In the development of a software product, we consider two main parties: the customer who requires the use of the product and the designer who must provide the product. Typically, the customer and the designer are groups of people and some people can be both customer and designer. It is often important to distinguish between the customer who is the client of the designing company and the customer who is the eventual user of the system. These two roles of customer can be played by different people. The group of people who negotiate the features of the intended system with the designer may never be actual users of the system. This is often particularly true of web applications. In this chapter, we will use the term _customer‘ to refer to the group of people who interact with the design team and we will refer to those who will interact with the designed system as the user or end-user.

2.6.1 Activities

The graphical representation is reminiscent of a waterfall, in which each activity naturally leads into the next. The analogy of the waterfall is not completely faithful to the real relationship between these activities, but it provides a good starting point for discussing the logical flow of activity. We describe the activities of this waterfall model of the software life cycle

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

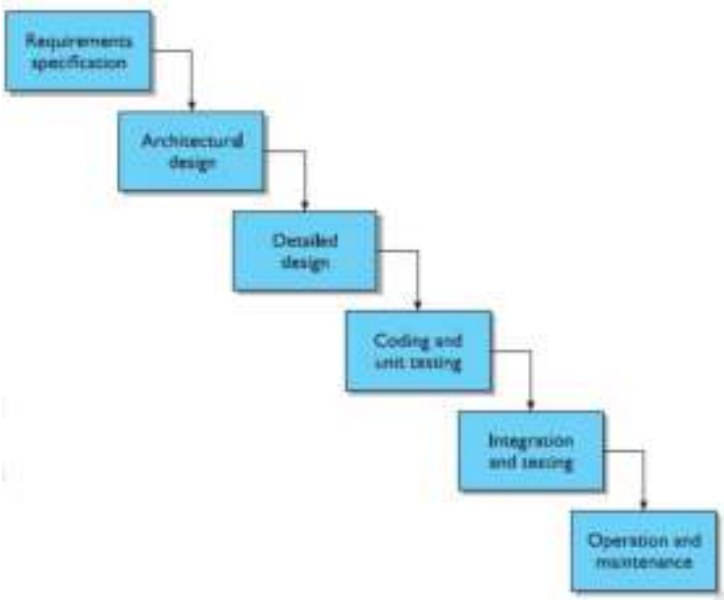


Figure The activities in the waterfall model of the software life cycle

Requirements specification

Requirements specification begins at the start of product development. Though the requirements are from the customer‘s perspective, if they are to be met by the software product they must be formulated in a language suitable for implementation. Requirements are usually initially expressed in the native language of the customer. The executable languages for software are less natural and are more closely related to a mathematical language in which each term in the language has a precise interpretation, or semantics. The transformation from the expressive but relatively ambiguous natural language of requirements to the more precise but less expressive executable languages is one key to successful development. Task analysis techniques, which are used to express work domain requirements in a form that is both expressive and precise.

Architectural design

The requirements specification concentrates on what the system is supposed to do. The next activities concentrate on how the system provides the services expected from it. The first activity is a high-level decomposition of the system into components that can either be brought in from existing software products or be developed from scratch independently. An architectural design performs this decomposition. It is not only concerned with the functional decomposition of the system, determining which components provide which services. It must also describe the interdependencies between separate components and the sharing of resources that will arise between components.

Detailed design

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

The architectural design provides a decomposition of the system description that allows for isolated development of separate components which will later be integrated. For those components that are not already available for immediate integration, the designer must provide a sufficiently detailed description so that they may be implemented in some programming language. The detailed design is a refinement of the component description provided by the architectural design. The behavior implied by the higher-level description must be preserved in the more detailed description.

There will be more than one possible refinement of the architectural component that will satisfy the behavioral constraints. Choosing the best refinement is often a matter of trying to satisfy as many of the non-functional requirements of the system as possible. Thus the language used for the detailed design must allow some analysis of the design in order to assess its properties.

Coding and unit testing

The detailed design for a component of the system should be in such a form that it is possible to implement it in some executable programming language. After coding, the component can be tested to verify that it performs correctly, according to some test criteria that were determined in earlier activities. Research on this activity within the life cycle has concentrated on two areas. There is plenty of research that is geared towards the automation of this coding activity directly from a low-level detailed design. Most of the work in formal methods operates under the hypothesis that, in theory, the transformation from the detailed design to the implementation is from one mathematical representation to another and so should be able to be entirely automated. Other, more practical work concentrates on the automatic generation of tests from output of earlier activities which can be performed on a piece of code to verify that it behaves correctly.

Integration and testing

Once enough components have been implemented and individually tested, they must be integrated as described in the architectural design. Further testing is done to ensure correct behavior and acceptable use of any shared resources. It is also possible at this time to perform some acceptance testing with the customers to ensure that the system meets their requirements. It is only after acceptance of the integrated system that the product is finally released to the customer.

Maintenance

After product release, all work on the system is considered under the category of maintenance, until such time as a new version of the product demands a total redesign or the product is phased out entirely. Consequently, the majority of the lifetime of a product is spent in the maintenance activity. Maintenance involves the correction of errors in the systems

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

which are discovered after release and the revision of the system services to satisfy requirements that were not realized during previous development.

2.6.2 Validation and verification

Throughout the life cycle, the design must be checked to ensure that it both satisfies the high-level requirements agreed with the customer and is also complete and internally consistent. These checks are referred to as validation and verification, respectively. Verification of a design will most often occur within a single life-cycle activity or between two adjacent activities. For example, in the detailed design of a component of a payroll accounting system, the designer will be concerned with the correctness of the algorithm to compute taxes deducted from an employee‘s gross income.

The architectural design will have provided a general specification of the information input to this component and the information it should output. The detailed description will introduce more information in refining the general specification. The detailed design may also have to change the representations for the information and will almost certainly break up a single high-level operation into several low-level operations that can eventually be implemented. In introducing these changes to information and operations, the designer must show that the refined description is a legal one within its language (internal consistency) and that it describes all of the specified behavior of the high-level description (completeness) in a provably correct way (relative consistency). Validation of a design demonstrates that within the various activities the customer‘s requirements are satisfied.

Validation is a much more subjective exercise than verification, mainly because the disparity between the language of the requirements and the language of the design forbids any objective form of proof. In interactive system design, the validation against HCI requirements is often referred to as evaluation and can be performed by the designer in isolation or in cooperation with the customer.

2.6.3 Management and contractual issues

The life cycle described above concentrated on the more technical features of software development. In a technical discussion, managerial issues of design, such as time constraints and economic forces, are not as important. The different activities of the life cycle are logically related to each other. We can see that requirements for a system precede the high-level architectural design which precedes the detailed design, and so on. In reality, it is quite possible that some detailed design is attempted before all of the architectural design. In management, a much wider perspective must be adopted which takes into account the marketability of a system, its training needs, the availability of skilled personnel or possible subcontractors, and other topics outside the activities for the development of the isolated system.

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

2.6.4 Interactive systems and the software life cycle

The life cycle for development we described above presents the process of design in a somewhat pipeline order. In reality, even for batch-processing systems, the actual design process is iterative, work in one design activity affecting work in any other activity either before or after it in the life cycle.

A final point about the traditional software life cycle is that it does not promote the use of notations and techniques that support the user‘s perspective of the interactive system. We discussed earlier the purpose of validation and the formality gap. It is very difficult for an expert on human cognition to predict the cognitive demands that an abstract design would require of the intended user if the notation for the design does not reflect the kind of information the user must recall in order to interact. The same holds for assessing the timing behavior of an abstract design that does not explicitly mention the timing characteristics of the operations to be invoked or their relative ordering. Though no structured development process will entirely eliminate the formality gap, the particular notations used can go a long way towards making validation of non-functional requirements feasible with expert assistance. In the remaining sections of this chapter, we will describe various approaches to augment the design process to suit better the design of interactive systems. These approaches are categorized under the banner of user-centered design.

2.7 USABILITY ENGINEERING

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 requirements specification that concentrates on features of the user–system interaction which contribute to the usability of the product. Various attributes of the system are suggested as gauges for testing the usability. For each attribute, six items are defined to form the usability specification of that attribute.

Table : Sample usability specification for undo with a VCR

Attribute:	Backward recoverability
Measuring concept:	Undo an erroneous programming sequence
Measuring method:	Number of explicit user actions to undo current program
Now level:	No current product allows such an undo
Worst case:	As many actions as it takes to program in mistake
Planned level:	A maximum of two explicit user actions
Best case:	One explicit cancel action

Recoverability refers to the ability to reach a desired goal after recognition of some error in previous interaction. The recovery procedure can be in either a backward or forward sense.

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

Current VCR design has resulted in interactive systems that are notoriously difficult to use; the redesign of a VCR provides a good case study for usability engineering. In designing a new VCR control panel, the designer wants to take into account how a user might recover from a mistake he discovers while trying to program the VCR to record some television program in his absence. One approach that the designer decides to follow is to allow the user the ability to undo the programming sequence, reverting the state of the VCR to what it was before the programming task began. The backward recoverability attribute is defined in terms of a measuring concept, which makes the abstract attribute more concrete by describing it in terms of the actual product. So in this case, we realize backward recoverability as the ability to undo an erroneous programming sequence. The measuring method states how the attribute will be measured, in this case by the number of explicit user actions required to perform the undo, regardless of where the user is in the programming sequence. The remaining four entries in the usability specification then provide the agreed criteria for judging the success of the product based on the measuring method. The now level indicates the value for the measurement with the existing system, whether it is computer based or not. The worst case value is the lowest acceptable measurement for the task, providing a clear distinction between what will be acceptable and what will be unacceptable in the final product. The planned level is the target for the design and the best case is the level which is agreed to be the best possible measurement given the current state of development tools and technology. In the example, the designers can look at their previous VCR products and those of their competitors to determine a suitable now level. In this case, it is determined that no current model allows an undo which returns the state of the VCR to what it was before the programming task.

Table: Criteria by which measuring method can be determined

1. Time to complete a task
2. Per cent of task completed
3. Per cent of task completed per unit time
4. Ratio of successes to failures
5. Time spent in errors
6. Per cent or number of errors
7. Per cent or number of competitors better than it
8. Number of commands used
9. Frequency of help and documentation use
10. Per cent of favorable/unfavorable user comments
11. Number of repetitions of failed commands
12. Number of runs of successes and of failures
13. Number of times interface misleads the user

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

14. Number of good and bad features recalled by users
15. Number of available commands not invoked
16. Number of regressive behaviors
17. Number of users preferring your system
18. Number of times users need to work around a problem
19. Number of times the user is disrupted from a work task
20. Number of times user loses control of the system
21. Number of times user expresses frustration or satisfaction

2.7.1 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. There is a very solid argument which points out that it is only through empirical approaches such as the use of usability metrics that we can reliably build more usable systems. Although the ultimate yardstick for determining usability may be by observing and measuring user performance, that does not mean that these measurements are the best way to produce a predictive design process for usability.

The problem with usability metrics is that they rely on measurements of very specific user actions in very specific situations. When the designer knows what the actions and situation will be, then she can set goals for measured observations. However, at early stages of design, designers do not have this information. Take our example usability specification for the VCR. In setting the acceptable and unacceptable levels for backward recovery, there is an assumption that a button will be available to invoke the undo. In fact, the designer was already making an implicit assumption that the user would be making errors in the programming of the VCR. We should recognize another inherent limitation for usability engineering, which provides a means of satisfying usability specifications and not necessarily usability. The designer is still forced to understand why a particular usability metric enhances usability for real people.

2.8 ITERATIVE DESIGN AND PROTOTYPING

The design can then be modified to correct any false assumptions that were revealed in the testing. This is the essence of iterative design, a purposeful design process which tries to overcome the inherent problems of incomplete requirements specification by cycling through several designs, incrementally improving upon the final product with each pass.

The problems with the design process, which lead to an iterative design philosophy, are not unique to the usability features of the intended system. The problem holds for requirements specification in general, and so it is a general software engineering problem,

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

together with technical and managerial issues. On the technical side, iterative design is described by the use of prototypes, artifacts that simulate or animate some but not all features of the intended system. There are three main approaches to prototyping:

Throw-away :The prototype is built and tested. The design knowledge gained from this exercise is used to build the final product, but the actual prototype is discarded. Figure depicts the procedure in using throw-away prototypes to arrive at a final requirements specification in order for the rest of the design process to proceed.

Incremental The final product is built as separate components, one at a time. There is one overall design for the final system, but it is partitioned into independent and smaller components. The final product is then released as a series of products, each subsequent release including one more component.

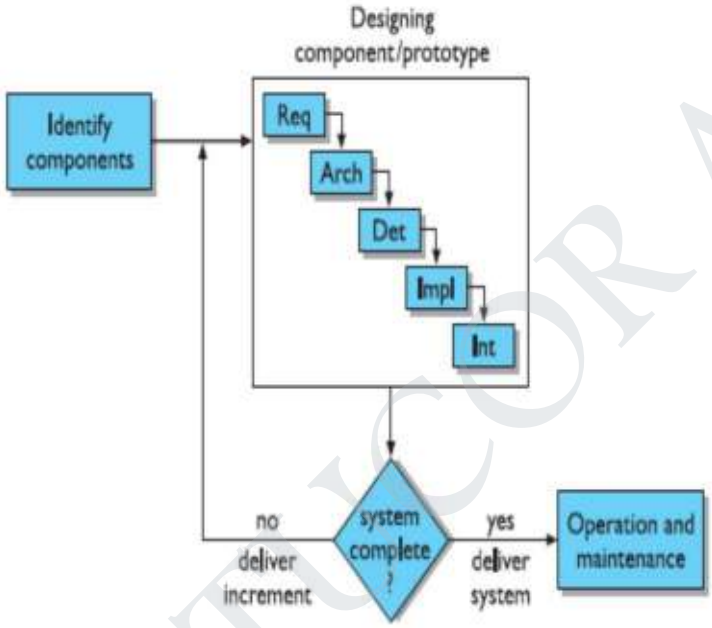


Figure: Incremental prototyping within the life cycle

Evolutionary Here the prototype is not discarded and serves as the basis for the next iteration of design. In this case, the actual system is seen as evolving from a very limited initial version to its final release,

Evolutionary prototyping also fits in well with the modifications which must be made to the system that arise during the operation and maintenance activity in the life cycle.

Prototypes differ according to the amount of functionality and performance they provide relative to the final product. An animation of requirements can involve no real functionality, or limited functionality to simulate only a small aspect of the interactive behavior for evaluative purposes. At the other extreme, full functionality can be provided at the expense of other performance characteristics, such as speed or error tolerance. Regardless of the level of functionality, the importance of a prototype lies in its projected realism. The prototype of an interactive system is used to test requirements by evaluating

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

their impact with real users. An honest appraisal of the requirements of the final system can only be trusted if the evaluation conditions are similar to those anticipated for the actual operation. But providing realism is costly, so there must be support.

Time Building prototypes takes time and, if it is a throw-away prototype, it can be seen as precious time taken away from the real design task. So the value of prototyping is only appreciated if it is fast, hence the use of the term rapid prototyping. Rapid development and manipulation of a prototype should not be mistaken for rushed evaluation which might lead to erroneous results and invalidate the only advantage of using a prototype in the first place. Planning Most project managers do not have the experience necessary for adequately planning and costing a design process which involves prototyping.

Non-functional features Often the most important features of a system will be non-functional ones, such as safety and reliability, and these are precisely the kinds of features which are sacrificed in developing a prototype. For evaluating usability features of a prototype, response time – yet another feature often compromised in a prototype – could be critical to product acceptance. This problem is similar to the technical issue of prototype realism.

Contracts The design process is often governed by contractual agreements between customer and designer which are affected by many of these managerial and technical issues. Prototypes and other implementations cannot form the basis for a legal contract, and so an iterative design process will still require documentation which serves as the binding agreement. There must be an effective way of translating the results derived from prototyping into adequate documentation. A rapid prototyping process might be amenable to quick changes, but that does not also apply to the design process.

2.8.1 Techniques for prototyping

Probably the simplest notion of a prototype is the storyboard, which is a graphical depiction of the outward appearance of the intended system, without any accompanying system functionality. Storyboards do not require much in terms of computing power to construct; in fact, they can be mocked up without the aid of any computing resource. The origins of storyboards are in the film industry, where a series of panels roughly depicts snapshots from an intended film sequence in order to get the idea across about the eventual scene. Similarly, for interactive system design, the storyboards provide snapshots of the interface at particular points in the interaction. Evaluating customer or user impressions of the storyboards can determine relatively quickly if the design is heading in the right direction.

Modern graphical drawing packages now make it possible to create storyboards with the aid of a computer instead of by hand. Though the graphic design achievable on screen may not be as sophisticated as that possible by a professional graphic designer, it is more

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

realistic because the final system will have to be displayed on a screen. Also, it is possible to provide crude but effective animation by automated sequencing through a series of snapshots. Animation illustrates the dynamic aspects of the intended user–system interaction, which may not be possible with traditional paper-based storyboards. If not animated, storyboards usually include annotations and scripts indicating how the interaction will occur.

Limited functionality simulations

Storyboards and animation techniques are not sufficient for this purpose, as they cannot portray adequately the interactive aspects of the system. To do this, some portion of the functionality must be simulated by the design team. Programming support for simulations means a designer can rapidly build graphical and textual interaction objects and attach some behavior to those objects, which mimics the system‘s functionality. Once this simulation is built, it can be evaluated and changed rapidly to reflect the results of the evaluation study with various users.

High-level programming support

HyperTalk and many similar languages allow the programmer to attach functional behavior to the specific interactions that the user will be able to do, such as position and click on the mouse over a button on the screen. Previously, the difficulty of interactive programming was that it was so implementation dependent that the programmer would have to know quite a bit of intimate detail of the hardware system in order to control even the simplest of interactive behavior. These high-level programming languages allow the programmer to abstract away from the hardware specifics and think in terms that are closer to the way the input and output devices are perceived as interaction devices. The frequent conceptual model put forth for interactive system design is to separate the application functionality from its presentation. It is then possible to program the underlying functionality of the system and to program the behavior of the user interface separately. The job of a UIMS, then, is to allow the programmer to connect the behavior at the interface with the underlying functionality.

2.8.2 Warning about iterative design

The ideal model of iterative design, in which a rapid prototype is designed, evaluated and modified until the best possible design is achieved in the given project time, is appealing. But there are two problems.

First, it is often the case that design decisions made at the very beginning of the prototyping process are wrong and, in practice, design inertia can be so great as never to overcome an initial bad decision. So, whereas iterative design is, in theory, amenable to great changes through iterations, it can be the case that the initial prototype has bad features that will not be amended.

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

The second problem is slightly more subtle, and serious. If, in the process of evaluation, a potential usability problem is diagnosed, it is important to understand the reason for the problem and not just detect the symptom.

2.9 DESIGN RATIONALE

Design rationale is the information that explains why a computer system is the way it is, including its structural or architectural description and its functional or behavioral description. In this sense, design rationale does not fit squarely into the software life cycle described in this chapter as just another phase or box. Rather, design rationale relates to an activity of both reflection (doing design rationale) and documentation (creating a design rationale) that occurs throughout the entire life cycle.

In an explicit form, a design rationale provides a communication mechanism among the members of a design team so that during later stages of design and/or maintenance it is possible to understand what critical decisions were made, what alternatives were investigated (and, possibly, in what order) and the reason why one alternative was chosen over the others. This can help avoid incorrect assumptions later.

- Accumulated knowledge in the form of design rationales for a set of products can be reused to transfer what has worked in one situation to another situation which has similar needs. The design rationale 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.
- The effort required to produce a design rationale forces the designer to deliberate more carefully about design decisions. The process of deliberation can be assisted by the design rationale technique by suggesting how arguments justifying or discarding a particular design option are formed.

In the area of HCI, design rationale has been particularly important, again for several reasons:

- There is usually no single best design alternative. More often, the designer is faced with a set of trade-offs between different alternatives. For example, a graphical interface may involve a set of actions that the user can invoke by use of the mouse and the designer must decide whether to present each action as a _button‘on the screen, which is always visible, or hide all of the actions in a menu which must be explicitly invoked before an action can be chosen. The former option maximizes the operation visibility but the latter option takes up less screen space. It would be up to the designer to determine which criterion for evaluating the options was more important and then communicating that information in a design rationale.
- Even if an optimal solution did exist for a given design decision, the space of alternatives is so vast that it is unlikely a designer would discover it. In this case, it is

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

important that the designer indicates all alternatives that have been investigated. Then later on it can be determined if she has not considered the best solution or had thought about it and discarded it for some reason. In project management, this kind of accountability for design is good.

- The usability of an interactive system is very dependent on the context of its use. The flashiest graphical interface is of no use if the end-user does not have access to a high-quality graphics display or a pointing device. Capturing the context in which a design decision is made will help later when new products are designed.

If the context remains the same, then the old rationale can be adopted without revision. If the context has changed somehow, the old rationale can be re-examined to see if any rejected alternatives are now more favorable or if any new alternatives are now possible.

2.9.1 Process-oriented design rationale

Rationale is based on Rittel's issue-based information system, or IBIS, a style for representing design and planning dialog developed in the 1970s. In IBIS (pronounced _ibbiss'), a hierarchical structure to a design rationale is created. A root issue is identified which represents the main problem or question that the argument is addressing. Various positions are put forth as potential resolutions for the root issue, and these are depicted as descendants in the IBIS hierarchy directly connected to the root issue. Each position is then supported or refuted by arguments, which modify the relationship between issue and position. The hierarchy grows as secondary issues are raised which modify the root issue in some way. Each of these secondary issues is in turn expanded by positions and arguments, further sub-issues, and so on.

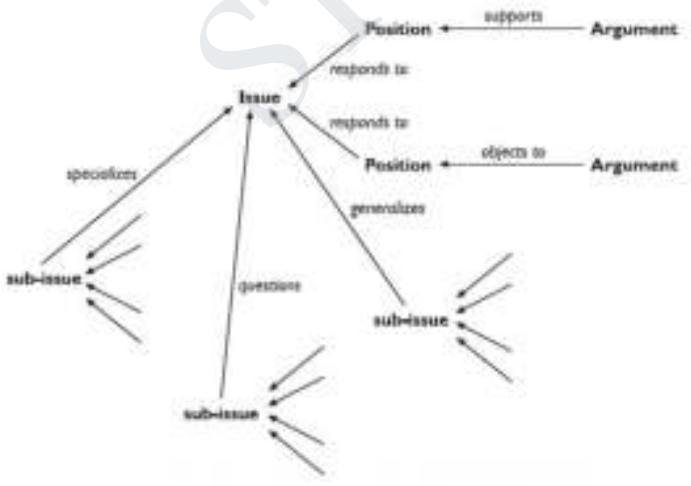


Figure: The structure of a gIBIS design rationale

A graphical version of IBIS has been defined by Conklin and Yakemovic called gIBIS (pronounced _gibbiss'), which makes the structure of the design rationale more apparent

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

visually in the form of a directed graph which can be directly edited by the creator of the design rationale. Above figure gives a representation of the gIBIS vocabulary. Issues, positions and arguments are nodes in the graph and the connections between them are labeled to clarify the relationship between adjacent nodes. So, for example, an issue can suggest further sub-issues, or a position can respond to an issue or an argument can support a position. The gIBIS structure can be supported by a hypertext tool to allow a designer to create and browse various parts of the design rationale.

2.9.2 Design space analysis

MacLean and colleagues have proposed a more deliberative approach to design rationale which emphasizes a post hoc structuring of the space of design alternatives that have been considered in a design project. Their approach, embodied in the Questions, Options and Criteria (QOC) notation, is characterized as design space analysis issues raised based on reflection and understanding of the actual design activities. Questions in a design space analysis are therefore similar to issues in IBIS except in the way they are captured. Options provide alternative solutions to the question. They are assessed according to some criteria in order to determine the most favorable option. In Figure an option which is favorably assessed in terms of a criterion is linked with a solid line, whereas negative links have a dashed line.

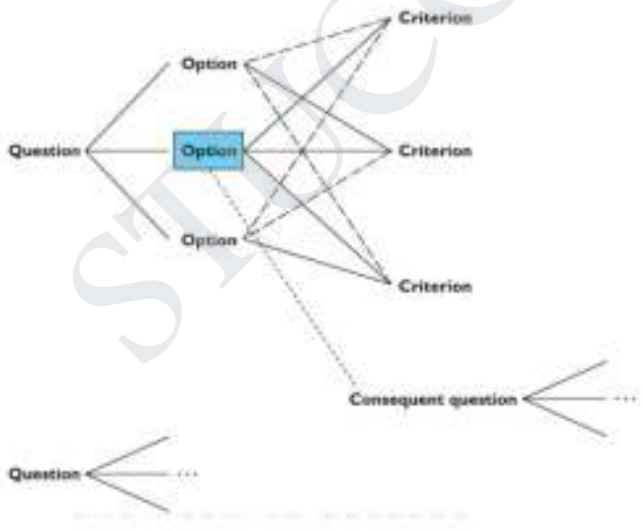


Figure: The QOC notation

The key to an effective design space analysis using the QOC notation is deciding the right questions to use to structure the space and the correct criteria to judge the options. The initial questions raised must be sufficiently general that they cover a large enough portion of the possible design space, but specific enough that a range of options can be clearly identified. It can be difficult to decide the right set of criteria with which to assess the options.

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

Structure-oriented technique, called Decision Representation Language (DRL), developed by Lee and Lai, structures the design space in a similar fashion to QOC, though its language is somewhat larger and it has a formal semantics. The questions, options and criteria in DRL are given the names: decision problem, alternatives and goals. QOC assessments are represented in DRL by a more complex language for relating goals to alternatives. The sparse language in QOC used to assess an option relative to a criterion (positive or negative assessment only) is probably insufficient, but there is a trade-off involved in adopting a more complex vocabulary which may prove too difficult to use in practice. The advantage of the formal semantics of DRL is that the design rationale can be used as a computational mechanism to help manage the large volume of information. For example, DRL can track the dependencies between different decision problems, so that subsequent changes to the design rationale for one decision problem can be automatically propagated to other dependent problems. Design space analysis directly addresses the claim that no design activity can hope to uncover all design possibilities, so the best we can hope to achieve is to document the small part of the design space that has been investigated. An advantage of the post hoc technique is that it can abstract away from the particulars of a design meeting and therefore represent the design knowledge in such a way that it can be of use in the design of other products. The major disadvantage is the increased overhead such an analysis warrants. More time must be taken away from the design activity to do this separate documentation task. When time is scarce, these kinds of overhead costs are the first to be trimmed.

2.9.3 Psychological design rationale

The final category of design rationale tries to make explicit the psychological claims of usability inherent in any interactive system in order better to suit a product for the tasks users have. This psychological design rationale has been introduced by Carroll and Rosson, and before we describe the application of the technique it is important to understand some of its theoretical background.

When designing a new interactive system, the designers take into account the tasks that users currently perform and any new ones that they may want to perform. This task identification serves as part of the requirements for the new system, and can be done through empirical observation of how people perform their work currently and presented through informal language or a more formal task analysis language. When the new system is implemented, or becomes an artifact, further observation reveals that in addition to the required tasks it was built to support, it also supports users in tasks that the designer never intended. Once designers understand these new tasks, and the associated problems that arise between them and the previously known tasks, the new task definitions can serve as requirements for future artifacts.

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

Carroll refers to this real-life phenomenon as the task–artifact cycle. He provides a good example of this cycle through the evolution of the electronic spreadsheet. When the first electronic spreadsheet, VisiCalc, was marketed in the late 1970s, it was presented simply as an automated means of supporting tabular calculation, a task commonly used in the accounting world. Within little over a decade of its introduction, the application of spreadsheets had far outstripped its original intent within accounting. Spreadsheets were being used for all kinds of financial analysis, _what-if 'simulations, report formatting and even as a general programming language paradigm! As the set of tasks expands, new spreadsheet products have flooded the marketplace trying to satisfy the growing customer base. Another good example of the task–artifact cycle in action is with word processing, which was originally introduced to provide more automated support for tasks previously achieved with a typewriter and now provides users with the ability to carry out various authoring tasks that they never dreamed possible with a conventional typewriter. And today, the tasks for the spreadsheet and the word processor are intermingled in the same artifact.

The purpose of psychological design rationale is to support this natural task– artifact cycle of design activity. The main emphasis is not to capture the designer's intention in building the artifact. Rather, psychological design rationale aims to make explicit the consequences of a design for the user, given an understanding of what tasks he intends to perform. Previously, these psychological consequences were left implicit in the design, though designers would make informal claims about their systems

The first step in the psychological design rationale is to identify the tasks that the proposed system will address and to characterize those tasks by questions that the user tries to answer in accomplishing them. For instance, Carroll gives an example of designing a system to help programmers learn the Smalltalk object-oriented programming language environment. The main task the system is to support is learning how Smalltalk works. In learning about the programming environment, the programme will perform tasks that help her answer the questions:

- What can I do: that is, what are the possible operations or functions that this programming environment allows?
- How does it work: that is, what do the various functions do?
- How can I do this: that is, once I know a particular operation I want to perform,
- How do I go about programming it?

2.10 DESIGN RULES

- Designing for maximum usability is the goal of interactive systems design.
- Abstract principles offer a way of understanding usability in a more general sense, especially if we can express them within some coherent catalog.

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

- Design rules in the form of standards and guidelines provide direction for design, in both general and more concrete terms, in order to enhance the interactive properties of the system.
- The essential characteristics of good design are often summarized through _golden rules‘ or heuristics.
- Design patterns provide a potentially generative approach to capturing and reusing design knowledge.

2.10.1 PRINCIPLES TO SUPPORT USABILITY

The principles we present are first divided into three main categories:

Learnability – the ease with which new users can begin effective interaction and achieve maximal performance.

Flexibility – the multiplicity of ways in which the user and system exchange information.

Robustness – the level of support provided to the user in determining successful achievement and assessment of goals.

Table: Summary of principles affecting learnability

Principle	Definition	Related principles
Predictability	Support for the user to determine the effect of future action based on past interaction history	Operation visibility
Synthesizability	Support for the user to assess the effect of past operations on the current state	Immediate/eventual honesty
Familiarity	The extent to which a user's knowledge and experience in other real-world or computer-based domains can be applied when interacting with a new system	Guessability, affordance
Generalizability	Support for the user to extend knowledge of specific interaction within and across applications to other similar situations	–
Consistency	Likeness in input–output behavior arising from similar situations or similar task objectives	–

Predictability

Predictability of an interactive system is distinguished from deterministic behavior of the computer system alone. Most computer systems are ultimately deterministic machines, so that given the state at any one point in time and the operation which is to be performed at that time, there is only one possible state that can result. Predictability is a user-centered concept; it is deterministic behavior from the perspective of the user. It is not enough for the behavior of the computer system to be determined completely from its state, as the user must be able to take advantage of the determinism.

Synthesizability

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

When an operation changes some aspect of the internal state, it is important that the change is seen by the user. The principle of honesty relates to the ability of the user interface to provide an observable and informative account of such change. In the best of circumstances, this notification can come immediately, requiring no further interaction initiated by the user. At the very least, the notification should appear eventually, after explicit user directives to make the change observable. A good example of the distinction between immediacy and eventuality can be seen in the comparison between command language interfaces and visual desktop interfaces for a file management system. You have moved a file from one directory to another. The principle of honesty implies that after moving the file to its new location in the file system you are then able to determine its new whereabouts. In a command language system, you would typically have to remember the destination directory and then ask to see the contents of that directory in order to verify that the file has been moved (in fact, you would also have to check that the file is no longer in its original directory to determine that it has been moved and not copied). In a visual desktop interface, a visual representation (or icon) of the file is dragged from its original directory and placed in its destination directory where it remains visible (assuming the destination folder is selected to reveal its contents). In this case, the user need not expend any more effort to assess the result of the move operation. The visual desktop is immediately honest.

Familiarity

New users of a system bring with them a wealth of experience across a wide number of application domains. This experience is obtained both through interaction in thereal world and through interaction with other computer systems. For a new user, the familiarity of an interactive system measures the correlation between the user‘s existing knowledge and the knowledge required for effective interaction. For example, when word processors were originally introduced the analogy between the word processor and a typewriter was intended to make the new technology more immediately accessible to those who had little experience with the former but a lot of experience with the latter. Familiarity has to do with a user‘s first impression of the system. In this case, we are interested in how the system is first perceived and whether the user can determine how to initiate any interaction.

Generalizability

The generalizability of an interactive system supports this activity, leading to a more complete predictive model of the system for the user. We can apply generalization to situations in which the user wants to apply knowledge that helps achieve one particular goal to another situation where the goal is in some way similar. Generalizability can be seen as a form of consistency. Generalization can occur within a single application or across a variety of applications. For example, in a graphical drawing package that draws a circle as a

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

constrained form of ellipse, we would want the user to generalize that a square can be drawn as a constrained rectangle. A good example of generalizability across a variety of applications can be seen in multi-windowing systems that attempt to provide cut/paste/copy operations to all applications in the same way (with varying degrees of success). Generalizability within an application can be maximized by any conscientious designer.

Consistency

Consistency relates to the likeness in behavior arising from similar situations or similar task objectives. Consistency is probably the most widely mentioned principle in the literature on user interface design. _Be consistent!' we are constantly urged. The user relies on a consistent interface. However, the difficulty of dealing with consistency is that it can take many forms. Consistency is not a single property of an interactive system that is either satisfied or not satisfied. Instead, consistency must be applied relative to something. Thus we have consistency in command naming, or consistency in command/argument invocation.

Consistency can be expressed in terms of the form of input expressions or output responses with respect to the meaning of actions in some conceptual model of the system. For example, before the introduction of explicit arrow keys, some word processors used the relative position of keys on the keyboard to indicate directionality for operations (for example, to move one character to the left, right, up or down).The conceptual model for display-based editing is a two-dimensional plane, so the user would think of certain classes of operations in terms of movements up, down, left or right in the plane of the display. Operations that required directional information, such as moving within the text or deleting some unit of text, could be articulated by using some set of keys on the keyboard that form a pattern consistent with up, down, left and right (for example, the keys e, x, s and d, respectively). For output responses, a good example of consistency can be found in a warnings system for an aircraft. Warnings to the pilot are classified into three categories, depending on whether the situation with the aircraft requires immediate recovery action, eventual but not immediate action, or no action at all (advisory) on the part of the crew.

Table: Summary of principles affecting flexibility

Principle	Definition	Related principles
Dialog initiative	Allowing the user freedom from artificial constraints on the input dialog imposed by the system	System/user pre-emptiveness
Multi-threading	Ability of the system to support user interaction pertaining to more than one task at a time	Concurrent vs. interleaving, multi-modality
Task migratability	The ability to pass control for the execution of a given task so that it becomes either internalized by the user or the system or shared between them	–
Substitutivity	Allowing equivalent values of input and output to be arbitrarily substituted for each other	Representation multiplicity, equal opportunity
Customizability	Modifiability of the user interface by the user or the system	Adaptivity, adaptability

Dialog initiative

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

The system can initiate all dialog, in which case the user simply responds to requests for information. We call this type of dialog system pre-emptive. For example, a modal dialog box prohibits the user from interacting with the system in any way that does not direct input to the box. Alternatively, the user may be entirely free to initiate any action towards the system, in which case the dialog is user pre-emptive. The system may control the dialog to the extent that it prohibits the user from initiating any other desired communication concerning the current task or some other task the user would like to perform. From the user‘s perspective, a system-driven interaction hinders flexibility whereas a user-driven interaction favours it. In general, we want to maximize the user‘s ability to pre-empt the system and minimize the system‘s ability to pre-empt the user. Although a system pre-emptive dialog is not desirable in general, some situations may require it. In a cooperative editor (in which two people edit a document at the same time) it would be impolite for you to erase a paragraph of text that your partner is currently editing. For safety reasons, it may be necessary to prohibit the user from the _freedom‘ to do potentially serious damage. A pilot about to land an aircraft in which the flaps have asymmetrically failed in their extended position2 should not be allowed to abort the landing, as this failure will almost certainly result in a catastrophic accident.

Multi-threading

A thread of a dialog is a coherent subset of that dialog. In the user–system dialog, we can consider a thread to be that part of the dialog that relates to a given user task. Multi-threading of the user–system dialog allows for interaction to support more than one task at a time. Concurrent multi-threading allows simultaneous communication of information pertaining to separate tasks. Interleaved multi-threading permits a temporal overlap between separate tasks, but stipulates that at any given instant the dialog is restricted to a single task.

Task migratability

Task migratability concerns the transfer of control for execution of tasks between system and user. It should be possible for the user or system to pass the control of a task over to the other or promote the task from a completely internalized one to a shared and cooperative venture. Hence, a task that is internal to one can become internal to the other or shared between the two partners.

Substitutivity

Substitutivity requires that equivalent values can be substituted for each other. For example, in considering the form of an input expression to determine the margin for a letter, you may want to enter the value in either inches or centimeters. You may also want to input the value explicitly (say 1.5 inches) or you may want to enter a calculation which produces the right input value (you know the width of the text is 6.5 inches and the width of the paper

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

is 8.5 inches and you want the left margin to be twice as large as the right margin, so you enter 2/3(8.5 − 6.5) inches). This input substitutivity contributes towards flexibility by allowing the user to choose whichever form best suits the needs of the moment. By avoiding unnecessary calculations in the user‘s head, substitutivity can minimize user errors and cognitive effort.

Robustness

A user is engaged with a computer in order to achieve some set of goals. The robustness of that interaction covers features that support the successful achievement and assessment of the goals.

Observability

Observability allows the user to evaluate the internal state of the system by means of its perceivable representation at the interface. Observability can be discussed through five other principles: browsability, defaults, reachability, persistence and operation visibility.

Browsability

Allows the user to explore the current internal state of the system via the limited view provided at the interface. Usually the complexity of the domain does not allow the interface to show all of the relevant domain concepts at once. Indeed, this is one reason why the notion of task is used, in order to constrain the domain information needed at one time to a subset connected with the user‘s current activity. While you may not be able to view an entire document‘s contents, you may be able to see all of an outline view of the document, if you are only interested in its overall structure. Even with a restriction of concepts relevant to the current task, it is probable that all of the information a user needs to continue work on that task is not immediately perceivable. Or perhaps the user is engaged in a multi-threaded dialog covering several tasks. There needs to be a way for the user to investigate, or browse, the internal state. This browsing itself should not have any side-effects on that state; that is, the browsing commands should be passive with respect to the domain specific parts of the internal state.

The availability of **defaults** can assist the user by passive recall .It also reduces the number of physical actions necessary to input a value. Thus, providing default values is a kind of error prevention mechanism. There are two kinds of default values: static and dynamic. Static defaults do not evolve with the session. They are either defined within the system or acquired at initialization. On the other hand, dynamic defaults evolve during the session. They are computed by the system from previous user inputs; the system is then adapting default values.

Reachability refers to the possibility of navigation through the observable system states. There are various levels of reachability that can be given precise mathematical definitions,

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

but the main notion is whether the user can navigate from any given state to any other state. Reachability in an interactive system affects the recoverability of the system, as we will discuss later. In addition, different levels of reachability can reflect the amount of flexibility in the system as well, though we did not make that explicit in the discussion on flexibility.

Persistence deals with the duration of the effect of a communication act and the ability of the user to make use of that effect. The effect of vocal communication does not persist except in the memory of the receiver. Visual communication, on the other hand, can remain as an object which the user can subsequently manipulate long after the act of presentation. If you are informed of a new email message by a beep at your terminal, you may know at that moment and for a short while later that you have received a new message. If you do not attend to that message immediately, you may forget about it. If, however, some persistent visual information informs you of the incoming message , then that will serve as a reminder that an unread message remains long after its initial receipt.

Recoverability

Recoverability is the ability to reach a desired goal after recognition of some error in a previous interaction. There are two directions in which recovery can occur, forward or backward. Forward error recovery involves the acceptance of the current state and negotiation from that state towards the desired state. Forward error recovery may be the only possibility for recovery if the effects of interaction are not revocable (for example, in building a house of cards, you might sneeze whilst placing a card on the seventh level, but you cannot undo the effect of your misfortune except by rebuilding). Backward error recovery is an attempt to undo the effects of previous interaction in order to return to a prior state before proceeding. In a text editor, a mistyped keystroke might wipe out a large section of text which you would want to retrieve by an equally simple undo button. Recovery can be initiated by the system or by the user. When performed by the system, recoverability is connected to the notions of fault tolerance, safety, reliability and dependability, all topics covered in software engineering. However, in software engineering this recoverability is considered only with respect to system functionality; it is not tied to user intent. When recovery is initiated by the user, it is important that it determines the intent of the user‘s recovery actions; that is, whether he desires forward (negotiation) or backward (using undo/redo actions) corrective action.

Responsiveness

Responsiveness measures the rate of communication between the system and the user. Response time is generally defined as the duration of time needed by the system to express state changes to the user. In general, short durations and instantaneous response times are desirable. Instantaneous means that the user perceives system reactions as immediate. But even in situations in which an instantaneous response cannot be obtained, there must be some

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

indication to the user that the system has received the request for action and is working on a response. As significant as absolute response time is response time stability. Response time stability covers the invariance of the duration for identical or similar computational resources. For example, pull-down menus are expected to pop up instantaneously as soon as a mouse button is pressed. Variations in response time will impede anticipation exploited by motor skill.

Task conformance

Since the purpose of an interactive system is to allow a user to perform various tasks in achieving certain goals within a specific application domain, we can ask whether the system supports all of the tasks of interest and whether it supports these as the user wants.

Task completeness addresses the coverage issue and task adequacy addresses the user‘s understanding of the tasks. It is not sufficient that the computer system fully implements some set of computational services that were identified at early specification stages. It is essential that the system allows the user to achieve any of the desired tasks in a particular work domain as identified by a task analysis that precedes system specification

Task completeness refers to the level to which the system services can be mapped onto all of the user tasks. However, it is quite possible that the provision of a new computer based tool will suggest to a user some tasks that were not even conceivable before the tool. Therefore, it is also desirable that the system services be suitably general so that the user can define new tasks.

2.10.2 STANDARDS

Standards for interactive system design are usually set by national or international bodies to ensure compliance with a set of design rules by a large community. Standards can apply specifically to either the hardware or the software used to build the interactive system. Smith points out the differing characteristics between hardware and software, which affect the utility of design standards applied to them:

Underlying theory Standards for hardware are based on an understanding of physiology or ergonomics/human factors, the results of which are relatively well known, fixed and readily adaptable to design of the hardware. On the other hand, software standards are based on theories from psychology or cognitive science, which are less well formed, still evolving and not very easy to interpret in the language of software design. Consequently, standards for hardware can directly relate to a hardware specification and still reflect the underlying theory, whereas software standards would have to be more vaguely worded.

Change Hardware is more difficult and expensive to change than software, which is usually designed to be very flexible. Consequently, requirements changes for hardware do not occur

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

as frequently as for software. Since standards are also relatively stable, they are more suitable for hardware than software.

A given standards institution, such as the British Standards Institution (BSI) or the International Organization for Standardization (ISO) or a national military agency, has had standards for hardware in place before any for software. For example, the UK Ministry of Defence has published an Interim Defence Standard 00–25 on Human Factors for Designers of Equipment, produced in 12 parts:

- Part 1 Introduction
- Part 2 Body Size
- Part 3 Body Strength and Stamina
- Part 4 Workplace Design
- Part 5 Stresses and Hazards
- Part 6 Vision and Lighting
- Part 7 Visual Displays
- Part 8 Auditory Information
- Part 9 Voice Communication
- Part 10 Controls
- Part 11 Design for Maintainability
- Part 12 Systems

One component of the ISO standard 9241, pertaining to usability specification, applies equally to both hardware and software design. In the beginning of that document, the following definition of usability is given:

Usability The effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments.

Effectiveness The accuracy and completeness with which specified users can achieve specified goals in particular environments.

Efficiency The resources expended in relation to the accuracy and completeness of goals achieved.

Satisfaction The comfort and acceptability of the work system to its users and other people affected by its use.

2.10.3 GUIDELINES

A major concern for all of the general guidelines is the subject of dialog styles, which in the context of these guidelines pertains to the means by which the user communicates input to the system, including how the system presents the communication device. Smith and Mosier identify eight different dialog styles and Mayhew identifies seven . The only real difference is

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

the absence of query languages in Mayhew's list, but we can consider a query language as a special case of a command language.Most guidelines are applicable for the implementation of any one of these dialog styles in isolation. It is also important to consider the possibility of mixing dialog styles in one application. In contrasting the action and language paradigms , we concluded that it is not always the case that one paradigm wins over the other for all tasks in an application and, therefore, an application may want to mix the two paradigms. This equates to a mixing of dialog styles – a direct manipulation dialog being suitable for the action paradigm and a command language being suitable for the language paradigm. Mayhew provides guidelines and a technique for deciding how to mix dialog styles.

Table: Comparison of dialog styles mentioned in guidelines

Smith and Mosier [325]	Mayhew [230]
Question and answer	Question and answer
Form filling	Fill-in forms
Menu selection	Menus
Function keys	Function keys
Command language	Command language
Query language	–
Natural language	Natural language
Graphic selection	Direct manipulation

2.10.4 GOLDEN RULES AND HEURISTICS

Shneiderman's Eight Golden Rules of Interface Design

They are intended to be used during design but can also be applied, like Nielsen's heuristics, to the evaluation of systems. Strive for consistency in action sequences, layout, terminology, command use and so on.

1. Enable frequent users to use shortcuts, such as abbreviations, special key sequences and macros, to perform regular, familiar actions more quickly.
2. Offer informative feedback for every user action, at a level appropriate to the magnitude of the action.
3. Design dialogs to yield closure so that the user knows when they have completed a task.
4. Offer error prevention and simple error handling so that, ideally, users are prevented from making mistakes and, if they do, they are offered clear and informative instructions to enable them to recover.
5. Permit easy reversal of actions in order to relieve anxiety and encourage exploration, since the user knows that he can always return to the previous state.
6. Support internal locus of control so that the user is in control of the system, which responds to his actions.

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

7. Reduce short-term memory load by keeping displays simple, consolidating multiple page displays and providing time for learning action sequences.

Norman's Seven Principles for Transforming Difficult Tasks into Simple Ones

1.Use both knowledge in the world and knowledge in the head. People work better when the knowledge they need to do a task is available externally – either explicitly or through the constraints imposed by the environment. But experts also need to be able to internalize regular tasks to increase their efficiency. So systems should provide the necessary knowledge within the environment and their operation should be transparent to support the user in building an appropriate mental model of what is going on.

2. Simplify the structure of tasks. Tasks need to be simple in order to avoid complex problem solving and excessive memory load. There are a number of ways to simplify the structure of tasks. One is to provide mental aids to help the user keep track of stages in a more complex task. Another is to use technology to provide the user with more information about the task and better feedback. A third approach is to automate the task or part of it, as long as this does not detract from the user's experience. The final approach to simplification is to change the nature of the task so that it becomes something simpler. In all of this, it is important not to take control away from the user.

3. Make things visible: bridge the gulfs of execution and evaluation. The interface should make clear what the system can do and how this is achieved, and should enable the user to see clearly the effect of their actions on the system.

4. Get the mappings right. User intentions should map clearly onto system controls. User actions should map clearly onto system events. So it should be clear what does what and by how much. Controls, sliders and dials should reflect the task – so a small movement has a small effect and a large movement a large effect.

5. Exploit the power of constraints, both natural and artificial. Constraints are things in the world that make it impossible to do anything but the correct action in the correct way. A simple example is a jigsaw puzzle, where the pieces only fit together in one way. Here the physical constraints of the design guide the user to complete the task.

6. Design for error. To err is human, so anticipate the errors the user could make and design recovery into the system.

7. When all else fails, standardize. If there are no natural mappings then arbitrary mappings should be standardized so that users only have to learn them once. It is this standardization principle that enables drivers to get into a new car and drive it with very little difficulty – key controls are standardized. Occasionally one might switch on the indicator lights instead of the windscreen wipers, but the critical controls (accelerator, brake, clutch, steering) are always the same.

STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

EVALUATION TECHNIQUES

• Evaluation

– tests usability and functionality of system

– occurs in laboratory, field and/or in collaboration with users

– evaluates both design and implementation

– should be considered at all stages in the design life cycle

2.11.1 Goals of Evaluation

Evaluation has three main goals:

- to assess the extent and accessibility of the system's functionality
- to assess users' experience of the interaction
- to identify any specific problems with the system

The system's functionality is important in that it must accord with the user's requirements. In other words, the design of the system should enable users to perform their intended tasks more easily. This includes not only making the appropriate functionality available within the system, but making it clearly reachable by the user in terms of the actions that the user needs to take to perform the task. It also involves matching the use of the system to the user's expectations of the task. For example, if a filing clerk is used to retrieving a customer's file by the postal address, the same capability (at least) should be provided in the computerized file system. Evaluation at this level may also include measuring the user's performance with the system, to assess the effectiveness of the system in supporting the task.

In addition to evaluating the system design in terms of its functional capabilities, it is important to assess the user's experience of the interaction and its impact upon him. This includes considering aspects such as how easy the system is to learn, its usability and the user's satisfaction with it. It may also include his enjoyment and emotional response, particularly in the case of systems that are aimed at leisure or entertainment. It is important to identify areas of the design that overload the user in some way, perhaps by requiring an excessive amount of information to be remembered, for example. Much evaluation is aimed at measuring features such as these.

The final goal of evaluation is to identify specific problems with the design. These may be aspects of the design which, when used in their intended context, cause unexpected results, or confusion amongst users. This is, of course, related to both the functionality and usability of the design (depending on the cause of the problem). However, it is specifically concerned with identifying trouble-spots which can then be rectified.

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

2.11.2 Evaluation through expert analysis

There are four approaches to expert analysis:

- Cognitive walkthrough,
- Heuristic evaluation,
- The use of models
- Use of previous work

Cognitive walkthrough,

1. A specification or prototype of the system. It doesn't have to be complete, but it should be fairly detailed. Details such as the location and wording for a menu can make a big difference.

2. A description of the task the user is to perform on the system. This should be a representative task that most users will want to do.

3. A complete, written list of the actions needed to complete the task with the proposed system.

4. An indication of who the users are and what kind of experience and knowledge the valuators can assume about them.

Given this information, the evaluators step through the action sequence (identified in item 3 above) to critique the system and tell a believable story about its usability. To do this, for each action, the evaluators try to answer the following four questions for each step in the action sequence.

1. **Is the effect of the action the same as the user's goal at that point?** Each user action will have a specific effect within the system. Is this effect the same as what the user is trying to achieve at this point? For example, if the effect of the action is to save a document, is 'saving a document' what the user wants to do?

2. **Will users see that the action is available?**

Will users see the button or menu item, for example, that is used to produce the action? This is not asking whether they will recognize that the button is the one they want. This is merely asking whether it is visible to them at the time when they will need to use it. Instances where the answer to this question might be 'no' are, for example, where a VCR remote control has a covered panel of buttons or where a menu item is hidden away in a submenu.

3. **Once users have found the correct action, will they know it is the one they need?**

Complements the previous question. It is one thing for a button or menu item to be visible, but will the user recognize that it is the one he is looking for to complete his task? Where the previous question was about the visibility of the action, this one is about whether its meaning and effect is clear.

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

4. **After the action is taken, will users understand the feedback they get?**

If you now assume that the user did manage to achieve the correct action, will he know that the has done so? Will the feedback given be sufficient confirmation of what has actually happened? This is the completion of the execution–evaluation interaction cycle In order to determine if they have accomplished their goal, users need appropriate feedback. It is vital to document the cognitive walkthrough to keep a record of what is good

Heuristic evaluation

Nielsen's ten heuristics are:

1. **Visibility of system status** Always keep users informed about what is going on, through appropriate feedback within reasonable time. For example, if a system operation will take some time, give an indication of how long and how much is complete.

2. **Match between system and the real world** The system should speak the user's language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in natural and logical order.

3. **User control and freedom** Users often choose system functions by mistake and need a clearly marked 'emergency exit' to leave the unwanted state without having to go through an extended dialog. Support undo and redo.

4. **Consistency and standards** Users should not have to wonder whether words, situations or actions mean the same thing in different contexts. Follow platform conventions and accepted standards.

5. **Error prevention** Make it difficult to make errors. Even better than good error messages is a careful design that prevents a problem from occurring in the first place.

6. **Recognition rather than recall** Make objects, actions and options visible. The user should not have to remember information from one part of the dialog to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

7. **Flexibility and efficiency of use** Allow users to tailor frequent actions. Accelerators – unseen by the novice user – may often speed up the interaction for the expert user to such an extent that the system can cater to both in experienced and experienced users.

The use of models

A third expert-based approach is the use of models. Certain cognitive and design models provide a means of combining design specification and evaluation into the same framework. For example, the GOMS (goals, operators, methods and selection) model predicts user performance with a particular interface and can be used to filter particular design options. Similarly, lower-level modeling techniques such as the keystroke-level model provide predictions of the time users will take to perform low-level physical tasks. Design

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

methodologies, such as design rationale also have a role to play in evaluation at the design stage. Design rationale provides a framework in which design options can be evaluated. By examining the criteria that are associated with each option in the design, and the evidence that is provided to support these criteria, informed judgments can be made in the design. Dialog models can also be used to evaluate dialog sequences for problems, such as unreachable states, circular dialogs and complexity.

Using previous studies in evaluation

A final approach to expert evaluation exploits this inheritance, using previous results as evidence to support (or refute) aspects of the design. It is expensive to repeat experiments continually and an expert review of relevant literature can avoid the need to do so. It should be noted that experimental results cannot be expected to hold arbitrarily across contexts. The reviewer must therefore select evidence carefully, noting the experimental design chosen, the population of participants used, the analyses performed and the assumptions made.

2.11.3 User Participation

The different approaches to evaluation through user participation. These include empirical or experimental methods, observational methods, query techniques, and methods that use physiological monitoring, such as eye tracking and measures of heart rate and skin conductance.

2.11.4 Empirical methods

One of the most powerful methods of evaluating a design or an aspect of a design is to use a controlled experiment. This provides empirical evidence to support a particular claim or hypothesis. It can be used to study a wide range of different issues at different levels of detail. Any experiment has the same basic form. 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. Within this basic form there are a number of factors that are important to the overall reliability of the experiment, which must be considered carefully in experimental design. These include the participants chosen, the variables tested and manipulated, and the hypothesis tested.

2.12 Universal design

2.12.1 Design principles

These principles give us a framework in which to develop universal designs.

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 identical use is not possible, equivalent use

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

should be supported. Where appropriate, 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 choice of methods of use and adaptivity to the user's pace, precision and custom.

3.The system be simple and intuitive to use, regardless of the knowledge, experience, language or level of concentration of the user. The design needs to support the user's expectations and accommodate different language and literacy skills. It should not be unnecessarily complex and should be organized to facilitate access to the most important areas. It should provide prompting and feedback as far as possible.

4. Perceptible information: the design should provide effective communication of information regardless of the environmental conditions or the user's abilities. Redundancy of presentation is important: information should be represented in different forms or modes (e.g. graphic, verbal, text, touch). Essential information should be emphasized and differentiated clearly from the peripheral content. Presentation should support the range of devices and techniques used to

access information by people with different sensory abilities.

5. Tolerance for error: minimizing the impact and damage caused by mistakes or unintended behavior. Potentially dangerous situations should be removed or made hard to reach. Potential hazards should be shielded by warnings. Systems should fail safe from the user's perspective and users should be supported in tasks that require concentration.

6. Low physical effort: systems should be designed to be comfortable to use, minimizing physical effort and fatigue. The physical design of the system should allow the user to maintain a natural posture with reasonable operating effort. Repetitive or sustained actions should be avoided.

7.Requires size and 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. Important elements should be on the line of sight for both seated and standing users. All physical components should be comfortably reachable by seated or standing users. Systems should allow for variation in hand size and provide enough room for assistive devices to be used. These seven principles give us a good starting point in considering universal design. They are not all equally applicable to all situations, of course. For example, principles six and seven would be vital in designing an information booth but less important in designing word-processing software. But they provide a useful checklist of considerations for designers, together with guidelines on how each principle can be achieved.

2.12.2 Multimodal Interaction

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

Providing access to information through more than one mode of interaction is an important principle of universal design.Such design relies on multi-modal interaction. there are five senses: sight, sound, touch, taste and smell. Sight is the predominant sense for the majority of people, and most interactive systems consequently use the visual channel as their primary means of presentation, through graphics, text, video and animation. However, sound is also an important channel, keeping us aware of our surroundings, monitoring people and events around us, reacting to sudden noises, providing clues and cues that switch our attention from one thing to another. It can also have an emotional effect on us, particularly in the case of music. Music is almost completely an auditory experience, yet is able to alter moods, conjure up visual images,

Evoke atmospheres or scenes in the mind of the listener. Touch, too, provides important information: tactile feedback forms an intrinsic part of the operation of many common tools – cars, musical instruments, pens, anything that requires holding or moving. It can form a sensuous bond between individuals, communicating a wealth of non-verbal information. Taste and smell are often less appreciated (until they are absent) but they also provide useful information in daily life: checking if food is bad, detecting early signs of fire, noticing that manure has been spread in a field, pleasure.

2.12.3 Designing for diversity

1. Designing for users with disability

Visual Impairment

Hearing Impairment

Physical Impairment

Speech impairment

2. Designing for different age groups

Older people

Children

3. Designing for cultural differences

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

UNIT III

MODELS AND THEORIES

Cognitive models –Socio-Organizational issues and stake holder requirements – Communication and collaboration models-Hypertext, Multimedia and WWW

3.1 COGNITIVE MODELS

Cognitive models represent users of interactive systems.

- Hierarchical models represent a user‘s task and goal structure.
- Linguistic models represent the user–system grammar.
- Physical and device models represent human motor skills.
- Cognitive architectures underlie all of these cognitive models.

3.1.1 GOAL AND TASK HIERARCHIES

To achieve this goal we divide it into several subgoals, say gathering the data together, producing the tables and histograms, and writing the descriptive material. Concentrating on the data gathering, we decide to split this into further subgoals: find the names of all introductory HCI textbooks and then search the book sales database for these books. Similarly, each of the other subgoals is divided up into further subgoals, until some level of detail is found at which we decide to stop. We thus end up with a hierarchy of goals and subgoals.

We can go on decomposing tasks until we get down to the individual hand and eye movements of the user, or we can stop at a more abstract level. Where do we start? In a similar way, we can start our analyses at different points in the hierarchy of goals. At the extreme we could extend our analysis to larger and larger goals: _light hob‘ is a subgoal of _boil peas‘ and so on to goals such as _have my dinner‘, _feed‘ and _stay alive‘.

These two questions are issues of granularity, and both of the methods described below leave this to some extent in the hands of the designer. Different design issues demand different

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

levels of analysis. However, both methods operate at a relatively low level; neither would attempt to start with such an abstract goal as _produce a report‘ which will involve real creativity and difficult problem solving. Instead they confine themselves to more routine learned behavior. This most abstract task is referred to as the unit task. The unit task does not require any problem-solving skills on the part of the user, though it frequently demands quite sophisticated problem-solving skills on the part of the designer to determine them. What do we do when there are several ways of solving a problem, or if the solutions to two subgoals interact? Users will often have more than one way to achieve a goal and there must be some way of representing how they select between competing solutions.

GOMS

The GOMS model of Card, Moran and Newell is an acronym for Goals, Operators, Methods and Selection

Goals These are the user‘s goals, describing what the user wants to achieve. GOMS the goals are taken to represent a _memory point‘ for the user, from which he can evaluate what should be done and to which he may return should any errors occur.

Operators These are the lowest level of analysis. They are the basic actions that the user must perform in order to use the system. They may affect the system (for example, press the _X‘ key) or only the user‘s mental state (for example, read the dialog box). There is still a degree of flexibility about the granularity of operators; we may take the command level _issue the SELECT command‘ or be more primitive: _move mouse to menu bar, press center mouse button .

Methods There are typically several ways in which a goal can be split into subgoals. For instance, in a certain window manager a currently selected window can be closed to an icon either by selecting the _CLOSE‘ option from a pop-up menu, or by hitting the _L7‘ function key. In GOMS these two goal decompositions are referred to as methods, so we have the CLOSE-METHOD and the L7-METHOD:

GOAL: ICONIZE-WINDOW
. [select GOAL: USE-CLOSE-METHOD
. . MOVE-MOUSE-TO-WINDOW-HEADER
. . POP-UP-MENU
. . CLICK-OVER-CLOSE-OPTION
GOAL: USE-L7-METHOD
. . PRESS-L7-KEY]

The dots are used to indicate the hierarchical level of goals. Selection From the above snippet we see the use of the word select where the choice of methods arises. GOMS does not leave

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

this as a random choice, but attempts to predict which methods will be used. This typically depends both on the particular user and on the state of the system and details about the goals.

Rule 1: Use the CLOSE-METHOD unless another rule applies.

Rule 2: If the application is _blocks' use the L7-METHOD.

The goal hierarchies described in a GOMS analysis are almost wholly below the level of the unit task defined earlier. A typical GOMS analysis would therefore consist of a single high-level goal, which is then decomposed into a sequence of unit tasks, all of which can be further decomposed down to the level of basic operators:

GOAL: EDIT-MANUSCRIPT

GOAL: EDIT-UNIT-TASK repeat until no more unit tasks The goal decomposition between the overall task and the unit tasks would involve detailed understanding of the user's problem-solving strategies and of the application domain.

Cognitive complexity theory

Cognitive complexity refer to the number of mental structures an individual uses, how abstract they are and how they interact to shape his discernment or an individual difference variable linked with a wide range of communication skills and associated abilities.

Individuals with high cognitive complexity have the capacity to analyze a situation to discern various constituent elements and explore connections and possible relationships among the elements. These individuals think in a multidimensional way. The assumption of the complexity theory is that the more an event can be differentiated and parts considered in novel relationships, the more sophisticated the response and successful the solution. Whereas less complex individuals can be trained to understand a complicated set of detailed differentiations for a specific context, highly complex individuals are highly flexible in creating distinctions in new situations.

Individuals with high cognitive complexity are open to new information, attracted to other individuals of high complexity, highly flexibility, socially influential, problem solvers, strategic planners, highly creative, effective communicators and generally good leaders.

Problems and extensions of goal hierarchies

The formation of a goal hierarchy is largely a post hoc technique and runs a very real risk of being defined by the computer dialog rather than the user. One way to rectify this is to produce a goal structure based on pre-existing manual procedures and thus obtain a natural hierarchy . GOMS defines its domain to be that of expert use, and thus the goal structures that are important are those which users develop out of their use of the system. The conceptual framework of goal hierarchies and user goal stacks can be used to express interface issues, not directly addressed by the notations above. For instance, we can use this to examine in more detail the closure problem

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

with early automated teller machines (ATMs) mentioned in the Design Focus box These early ATMs gave the customers the money before returning their cards. Unfortunately, this led to many customers leaving their cards behind. This was despite on-screen messages telling them to wait. This is referred to as a problem of closure. The user‘s principal goal is to get money; when that goal is satisfied, the user does not complete or close the various subtasks which still remain open:

GOAL: GET-MONEY

. GOAL: USE-ATM

. . INSERT-CARD

. . ENTER-PIN

. . ENTER-AMOUNT

. . COLLECT-MONEY

<< outer goal now satisfied goal stack popped >>

. . COLLECT-CARD – subgoal operators missed

3.1.2 LINGUISTIC MODELS

The user‘s interaction with a computer is often viewed in terms of a language, so it is not surprising that several modeling formalisms have developed centered around this concept. BNF grammars are frequently used to specify dialogs.

The models here, although similar in form to dialog design notations, have been proposed with the intention of understanding the user‘s behavior and analyzing the cognitive difficulty of the interface.

BNF

Representative of the linguistic approach is Reisner‘s use of Backus–Naur Form (BNF) rules to describe the dialog grammar . This views the dialog at a purely syntactic level, ignoring the semantics of the language. BNF has been used widely to specify the syntax of computer programming languages, and many system dialogs can be described easily using BNF rules. For example, imagine a graphics system that has a line-drawing function. To select the function the user must select the _line‘ menu option. The line-drawing function allows the user to draw a polyline, that is a sequence of line arcs between points. The user selects the points by clicking the mouse button in the drawing area. The user double clicks to indicate the last point of the polyline.

```
draw-line      ::= select-line + choose-points
                   + last-point
select-line    ::= position-mouse + CLICK-MOUSE
choose-points  ::= choose-one
                   | choose-one + choose-points
choose-one     ::= position-mouse + CLICK-MOUSE
last-point     ::= position-mouse + DOUBLE-CLICK-MOUSE
position-mouse ::= empty | MOVE-MOUSE + position-mouse
```

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

The aims in the description are of two types: non-terminals, shown in lower case, and terminals, shown in upper case. Terminals represent the lowest level of user behavior, such as pressing a key, clicking a mouse button or moving the mouse. Non-terminals are higher-level abstractions. The non-terminals are defined in terms of other non-terminals and terminals by a definition of the form name ::= expression The _::=‘ symbol is read as _is defined as‘. Only non-terminals may appear on the left of a definition. The right-hand side is built up using two operators _+‘ (sequence) and _|‘ (choice). For example, the first rule says that the non-terminal draw-line is defined to be select-line followed by choose-points followed by lastpoint. All of these are non-terminals, that is they do not tell us what the basic user actions are. The second rule says that select-line is defined to be position mouse (intended to be over the _line‘ menu entry) followed by CLICK-MOUSE. This is our first terminal and represents the actual clicking of a mouse.

Position-mouse is, we look at the last rule. This tells us that there are two possibilities for position-mouse (separated by the _|‘ symbol). One option is that position-mouse is empty – a special symbol representing no action. That is, one option is not to move the mouse at all. The other option is to doa MOVE-MOUSE action followed by position-mouse. This rule is recursive, and this second position-mouse may itself either be empty or be a MOVE-MOUSE action followed by position-mouse, and so on. That is, position-mouse may be any number of MOVE-MOUSE actions whatsoever. Similarly, choose-points is defined recursively, but this time it does not have the option of being empty. It may be one or more of the non-terminal choose-one which is itself defined to be (like select-line) position-mouse followed by CLICK-MOUSE.

The BNF description of an interface can be analyzed in various ways. One measure is to count the number of rules. The more rules an interface requires to use it, the more complicated it is. This measure is rather sensitive to the exact way the interface is described. For example, we could have replaced the rules for choose points and choose-one with the single definition choose-points ::= position-mouse + CLICK-MOUSE | position-mouse + CLICK-MOUSE + choose-points.

Task–action grammar

Measures based upon BNF have been criticized as not _cognitive‘ enough. They ignore the advantages of consistency both in the language‘s structure and in its use of command names and letters. Task–action grammar (TAG)

3.1.3 THE CHALLENGE OF DISPLAY-BASED SYSTEMS

hierarchical and grammar-based techniques were initially developed when most interactive systems were command line, or at most, keyboard and cursor based. There are significant

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

worries, therefore, about how well these approaches can generalize to deal with more modern windowed and mouse-driven interfaces. Pressing a cursor key is a reasonable lexeme, but moving a mouse one pixel is less sensible. In addition, pointer-based dialogs are more display oriented. Clicking a cursor at a particular point on the screen has a meaning dependent on the current screen contents. This problem can be partially resolved by regarding operations such as _select region of text‘ or _click on quit button‘ as the terminals of the grammar. If this approach is taken, the detailed mouse movements and parsing of mouse events in the context of display information (menus, etc.) are abstracted away. Goal hierarchy methods have different problems, as more display-oriented systems encourage less structured methods for goal achievement. Instead of having well-defined plans, the user is seen as performing a more exploratory task, recognizing fruitful directions and backing out of others. Typically, even when this exploratory style is used at one level,

WRITE_LETTER

. FIND_SIMILAR_LETTER

. COPY_IT

. EDIT_COPY

3.1.4 PHYSICAL AND DEVICE MODELS

Keystroke-level model

The human motor system is well understood. KLM (Keystroke-Level Model) uses this understanding as a basis for detailed predictions about user performance. It is aimed at unit tasks within interaction – the execution of simple command sequences, typically taking no more than 20 seconds. Examples of this would be using a search and replace feature, or changing the font of a word. It does not extend to complex actions such as producing a diagram. The assumption is that these more complex tasks would be split into subtasks (as in GOMS) before the user attempts to map them into physical actions.

The task is split into two phases:

Acquisition of the task, when the user builds a mental representation of the task;

Execution of the task using the system‘s facilities.

During the acquisition phase, the user will have decided how to accomplish the task using the primitives of the system, and thus, during the execution phase, there is no high-level mental activity – the user is effectively expert. KLM is related to the GOMS model, and can be thought of as a very low-level GOMS model where the method is given.

The model decomposes the execution phase into five different physical motor operators, a mental operator and a system response operator:

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

K Key stroking, actually striking keys, including shifts and other modifier keys. B Pressing a mouse button.

P Pointing, moving the mouse (or similar device) at a target.

H Homing, switching the hand between mouse and keyboard.

D Drawing lines using the mouse.

M Mentally preparing for a physical action.

R System response which may be ignored if the user does not have to wait for it, as in copy typing.

The execution of a task will involve interleaved occurrences of the various operators. For instance, imagine we are using a mouse-based editor. If we notice a single character error we will point at the error, delete the character and retype it, and then return to our previous typing point. This is decomposed as follows:

1. Move hand to mouse H[mouse]
2. Position mouse after bad character PB[LEFT]
3. Return to keyboard H[keyboard]
4. Delete character MK[DELETE]
5. Type correction K[char]
6. Reposition insertion point H[mouse]MPB[LEFT]

3.1.5 COGNITIVE ARCHITECTURES

The concept of taking a problem and solving it by divide and conquer using subgoals is central to GOMS. CCT assumes the distinction between long- and short-term memory, with production rules being stored in long-term memory and _matched‘ against the contents of short-term (or working) memory to determine which _fire‘. The values for various motor and mental operators in KLM were based on the Model Human Processor (MHP) architecture of Card, Moran and Newell. Another common assumption, itis the distinction between linguistic levels – semantic, syntactic and lexical – as an architectural model of the user‘s understanding.

The problem space model Rational behavior is characterized as behavior that is intended to achieve a specific goal. This element of rationality is often used to distinguish between intelligent and machine-like behavior. In the field of artificial intelligence (AI), a system exhibiting rational behavior is referred to as a knowledge-level system. A knowledge-level system contains an agent behaving in an environment. The agent has knowledge about itself and its environment, including its own goals. It can perform certain actions and sense information about its changing environment. As the agent behaves in its environment, it changes the environment and its own knowledge. We can view the overall behavior of the

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

knowledge-level system as a sequence of environment and agent states as they progress in time. The goal of the agent is characterized as a preference over all possible sequences of agent/environment states. The search proceeds by moving from one state to another possible state by means of operations or actions, the ultimate goal of which is to arrive at one of the desired states. This very general model of computation is used in the ordinary task of the programmer. Once she has identified a problem and a means of arriving at the solution to the problem (the algorithm), the programmer then represents the problem and algorithm in a programming language, which can be executed on a machine to reach the desired state. The architecture of the machine only allows the definition of the search or problem space and the actions that can occur to traverse that space. Termination is also assumed to happen once the desired state is reached.

The new computational model is the problem space model, based on the problem-solving work of Newell and Simon at Carnegie–Mellon University. A problem space consists of a set of states and a set of operations that can be performed on the states. Behavior in a problem space is a two-step process. First, the current operator is chosen based on the current state and then it is applied to the current state to achieve the new state. The problem space must represent rational behavior, and so it must characterize the goal of the agent. A problem space represents a goal by defining the desired states as a subset of all possible states. Once the initial state is set, the task within the problem space is to find a sequence of operations that form a path within the state space from the initial state to one of the desired states, whereupon successful termination occurs.

We can highlight four different activities that occur within a problem space: goal formulation, operation selection, operation application and goal completion. The relationship between these problem space processes and knowledge-level activity is key. Perception that occurs at the knowledge level is performed by the goal formulation process, which creates the initial state based on observations of the external environment. Actions at the knowledge level are operations in the problem space which are selected and applied. The real knowledge about the agent and its environment and goals is derived from the state/operator information in the problem space. Because of the goal formulation process, the set of desired states indicates the knowledge-level goal within the problem space. The operation selection process selects the appropriate operation at a given point in time because it is deemed the most likely to transform the state in the problem space to one of the desired states; hence rational behavior is implied.

Interacting cognitive subsystems (ICS) provides a model of perception, cognition and action, but unlike other cognitive architectures, it is not intended to produce a description of

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

the user in terms of sequences of actions that he performs. ICS provides a more holistic view of the user as an information-processing machine. The emphasis is on determining how easy particular procedures of action sequences become as they are made more automatic within the user.

ICS attempts to incorporate two separate psychological traditions within one cognitive architecture. On the one hand is the architectural and general-purpose information-processing approach of short-term memory research. On the other hand is the computational and representational approach characteristic of psycholinguistic research and AI problem-solving literature.

The architecture of ICS is built up by the coordinated activity of nine smaller subsystems: five peripheral subsystems are in contact with the physical world and four are central, dealing with mental processes. Each subsystem has the same generic structure. A subsystem is described in terms of its typed inputs and outputs along with a memory store for holding typed information. It has transformation functions for processing the input and producing the output and permanently stored information. Each of the nine subsystems is specialized for handling some aspect of external or internal processing. For example, one peripheral subsystem is the visual system for describing what is seen in the world.

3.2 SOCIO-ORGANIZATIONAL ISSUES AND STAKEHOLDER REQUIREMENTS

There are several organizational issues that affect the acceptance of technology by users and that must therefore be considered in system design:

- systems may not take into account conflict and power relationships
- those who benefit may not do the work
- not everyone may use systems.

In addition to generic issues, designers must identify specific stakeholder requirements within their organizational context.

- Socio-technical models capture both human and technical requirements.
- Soft systems methodology takes a broader view of human and organizational issues.
- Participatory design includes the user directly in the design process.
- Ethnographic methods study users in context, attempting to take an unbiased perspective.

3.2.1 ORGANIZATIONAL ISSUES

Cooperation or conflict?

The term _computer-supported cooperative work‘ (CSCW) seems to assume that groups will be acting in a cooperative manner. This is obviously true to some extent;even opposing football teams cooperate to the extent that they keep (largely) within the rules of the

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

game, but their cooperation only goes so far. People in organizations and groups have conflicting goals, and systems that ignore this are likely to fail spectacularly.

Imagine that an organization is already highly computerized, the different departments all have their own systems and the board decides that an integrated information system is needed. The production manager can now look directly at stocks when planning the week‘s work, and the marketing department can consult the sales department‘s contact list to send out marketing questionnaires.

The storekeeper always used to understate stock levels slightly in order to keep an emergency supply, or sometimes inflate the quoted levels when a delivery was due from a reliable supplier. Also, requests for stock information allowed the storekeeper to keep track of future demands and hence plan future orders. The storekeeper has now lost a sense of control and important sources of information. Members of the sales department are also unhappy: their contacts are their livelihood. The last thing they want is someone from marketing blundering in and spoiling a relationship with a customer built up over many years.

Some of these people may resort to subverting the system, keeping _sanitized‘ information online, but the real information in personal files.

Changing power structures

The identification of stakeholders will uncover information transfer and power relationships that cut across the organizational structure. Indeed, all organizations have these informal networks that support both social and functional contacts. The official lines of authority and information tend to flow up and down through line management.

The physical layout of an organization often reflects the formal hierarchy: each department is on a different floor, with sections working in the same area of an office. If someone from sales wants to talk to someone from marketing then one of them must walk to the other‘s office. Their respective supervisors can monitor the contact.

In face-to-face conversation, the manager can easily exert influence over a subordinate: both know their relative positions and this is reflected in the patterns of conversation and in other non-verbal cues. Email messages lose much of this sense of presence and it is more difficult for a manager to exercise authority. The _levelling‘ effect even makes it possible for subordinates to direct messages _diagonally‘ across the hierarchy, to their manager‘s peers, or, even worse, to their manager‘s manager!

The invisible worker

The ability to work and collaborate at a distance can allow functional groups to be distributed over different sites. This can take the form of cross-functional neighbourhood centers, where workers from different departments do their jobs in electronic contact with

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

their functional colleagues. If the approach in an organization is _management by presence‘, that is you know someone is working because they are in the office, then there is no way a remote worker is going to be trusted. If, on the other hand, the style is _management by objectives‘, that is you know your subordinates are working because they are doing their jobs and producing results, then remote working is not so problematical.

Who benefits?

In these systems the sender has to do work in putting information into fields appropriately, but it is the recipient who benefits. Another example is shared calendars. The beneficiary of the system is a manager who uses the system to arrange meeting times, but whose personal secretary does the work of keeping the calendar up to date. Subordinates are less likely to have secretarial support, yet must keep up the calendar with little perceived benefit. Of course, chaos results when a meeting is automatically arranged and the subordinates may have to rearrange commitments that have not been recorded on the system. The manager may force use by edict or the system may simply fall into disuse. Many such groupware systems are introduced on a _see if it works‘ basis,

Free rider problem

A system may still not function symmetrically, which may be a problem, particularly with shared communication systems. One issue is the free rider problem. Take an electronic conferencing system. If there is plenty of discussion of relevant topics then there are obvious advantages to subscribing and reading the contributions. When considering writing a contribution, the effort of doing so may outweigh any benefits.

The total benefit of the system for each user outweighs the costs, but for any particular decision the balance is overturned. A few free riders in a conference system are often not a problem, as the danger is more likely from too much activity. In addition, in electronic conferences the patterns of activity and silence may reflect other factors such as expertise. It is easy for the number of free riders gradually to increase and the system slide into disuse. It is hard to enforce equal use, except by restrictive schemes such as round-robin contributions (everyone contributes something however short). In the real world, such problems are often solved by social pressure, and the free rider reacts to the collective censure of the group. Increasing the visibility of participants‘ contributions might also help these social mechanisms.

Critical mass

When telephones were only in public places, their use as a form of pervasive interpersonal communication was limited. However, once a large number of people have telephones in their homes it becomes worthwhile paying to have a telephone installed. In

STUCOR APP

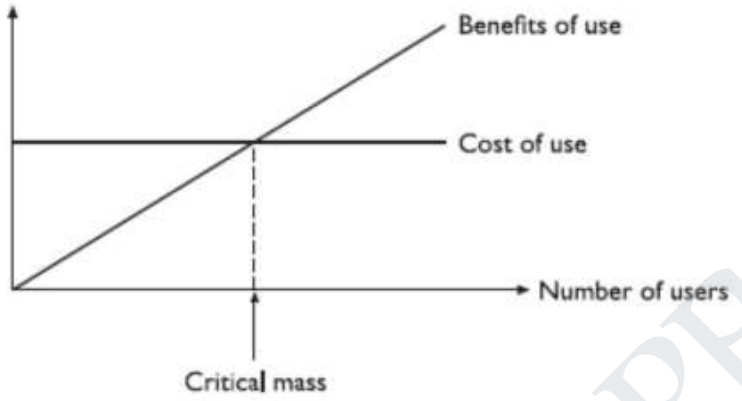
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

cost/benefit terms, the early subscribers probably have a smaller benefit than the cost. Only when the number of subscribers increases beyond the critical mass does the benefit for all dominate the cost .

Figure: Cost/benefit of system use



The telephone was useful for subgroups before it became beneficial for all. Even when only a small proportion of the population had personal telephones, they still formed a significant proportion of their social group, so these cliques of use could grow gradually over time.

Automating processes – workflow and BPR

Organizations have many such processes, and workflow systems aim to automate much of the process using electronic forms, which are forwarded to the relevant person based on pre-coded rules. Some workflow systems are built using special purpose groupware, often based on a notation for describing the desired workflow.

The rigid form of a typical workflow system is an example of global structuring. The danger with any form of global structuring is that it may conflict with or inhibit more informal and less structured patterns of activity which also contribute to the organization‘s free running.

A more radical approach to organizational processes is found in business process re-engineering (BPR). Traditionally, organizations have been structured around functions: sales, accounts, stores, manufacturing. However, the purpose of an organization can be seen in terms of key business processes. The ordering/delivery process described above is a typical and important example. In BPR these processes are recorded and analyzed. Problems in the current process are noted and the whole process may be redesigned in order to make the path of the process more efficient. For example, instead of sending an order to the accounts department to approve, a list of customer credit limits could be given to the sales executives.

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

They could then check the credit rating of the customer whilst on the phone and only forward the order to accounts if there are any unusual problems.

Evaluating the benefits

The benefits from cooperative systems, especially organization-wide systems such as email or electronic conferencing, are in terms of job satisfaction or more fluid information flow. Some, such as the video wall , are expected primarily to help social contact within the organization. It may be possible to measure contentment and job satisfaction using attitude questionnaires.

3.2.2 CAPTURING REQUIREMENTS

Who are the stakeholders?

A stakeholder can be defined as anyone who is affected by the success or failure of the system. It can be useful to distinguish different categories of stakeholder, and the following categorization from the CUSTOM approach is helpful for this:

Primary stakeholders are people who actually use the system – the end-users.

Secondary stakeholders are people who do not directly use the system, but receive output from it or provide input to it (for example, someone who receives a report produced by the system).

Tertiary stakeholders are people who do not fall into either of the first two categories but who are directly affected by the success or failure of the system .Facilitating stakeholders are people who are involved with the design, development and maintenance of the system.

The aim of the design team is to meet the needs of as many stakeholders as possible. The reality is that usually stakeholder needs are in conflict with each other. Sometimes this does not matter: a company is unlikely to be too concerned that its competitors‘ requirement to maintain advantage over it is under threat by the new system.

Socio-technical models

The socio-technical systems view came about to counter this technology-centric position, by stressing that work systems were composed of both human and machine elements and that it was the interrelationship between these that should be central.

Socio-technical models for interactive systems are therefore concerned with technical, social, organizational and human aspects of design. They recognize the fact that technology is not developed in isolation but as part of a wider organizational environment. It is important to consider social and technical issues side by side so that human issues are not overruled by technical considerations.

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

The key focus of the socio-technical approach is to describe and document the impact of the introduction of a specific technology into an organization. Methods vary but most attempt to capture certain common elements:

- The problem being addressed: there is a need to understand why the technology is being proposed and what problem it is intended to solve.
- The stakeholders affected, including primary, secondary, tertiary and facilitating, together with their objectives, goals and tasks.
- The workgroups within the organization, both formal and informal.
- The changes or transformations that will be supported.
- The proposed technology and how it will work within the organization.
- External constraints and influences and performance measures.

CUSTOM methodology

CUSTOM is a socio-technical methodology designed to be practical to use in small organizations. It is based on the User Skills and Task Match (USTM) approach, developed to allow design teams to understand and fully document user requirements. CUSTOM focusses on establishing stakeholder requirements: all stakeholders are considered, not just the end-users.

It is applied at the initial stage of design when a product opportunity has been identified, so the emphasis is on capturing requirements. It is a forms-based methodology, providing a set of questions to apply at each of its stages.

There are six key stages to carry out in a CUSTOM analysis:

1. Describe the organizational context, including its primary goals, physical characteristics, political and economic background.

2. Identify and describe stakeholders. All stakeholders are named, categorized (as primary, secondary, tertiary or facilitating) and described with regard to personal issues, their role in the organization and their job. For example, CUSTOM addresses issues such as stakeholder motivation, disincentives, knowledge, skills, power and influence within the organization, daily tasks and so on.

3. Identify and describe work-groups. A work-group is any group of people who work together on a task, whether formally constituted or not. Again, work-groups are described in terms of their role within the organization and their characteristics.

4. Identify and describe task–object pairs. These are the tasks that must be performed, coupled with the objects that are used to perform them or to which they are applied.

5. Identify stakeholder needs. Stages 2–4 are described in terms of both the current system and the proposed system. Stakeholder needs are identified by considering the differences

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

between the two. For example, if a stakeholder is identified as currently lacking a particular skill that is required in the proposed system then a need for training is identified.

6. Consolidate and check stakeholder requirements. Here the stakeholder needs list is checked against the criteria determined at earlier stages.

Open System Task Analysis (OSTA)

OSTA attempts to describe what happens when a technical system is introduced into an organizational work environment. Like CUSTOM, OSTA specifies both social and technical aspects of the system. However, whereas in CUSTOM these aspects are framed in terms of stakeholder perspectives, in OSTA they are captured through a focus on tasks.

OSTA has eight main stages:

1. The primary task which the technology must support is identified in terms of users‘ goals.

2. Task inputs to the system are identified. These may have different sources and forms that may constrain the design.

3. The external environment into which the system will be introduced is described, including physical, economic and political aspects.

4. The transformation processes within the system are described in terms of actions performed on or with objects.

5. The social system is analyzed, considering existing work-groups and relationships within and external to the organization.

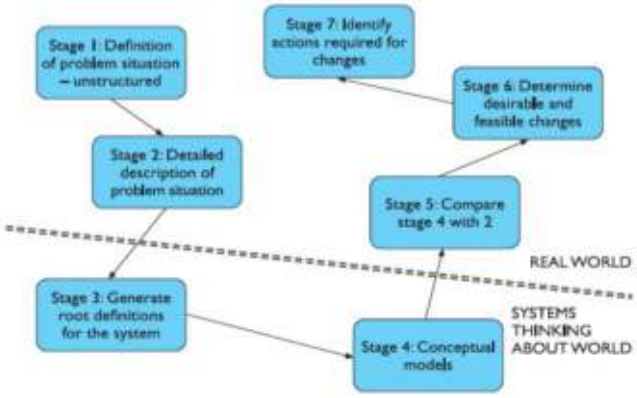
6. The technical system is described in terms of its configuration and integration with other systems.

7. Performance satisfaction criteria are established, indicating the social and technical requirements of the system.

8. The new technical system is specified.

Soft systems methodology

Soft systems methodology (SSM) arises from the same tradition but takes a view of the organization as a system of which technology and people are components. There is no assumption of a particular solution: the emphasis is rather on understanding the situation fully.



STUCOR AP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

Participatory design

Participatory design is a philosophy that encompasses the whole design cycle. It is design in the workplace, where the user is involved not only as an experimental subject or as someone to be consulted when necessary but as a member of the design team. Users are therefore active collaborators in the design process, rather than passive participants whose involvement is entirely governed by the designer. The argument is that users are experts in the work context and a design can only be effective within that context if these experts are allowed to contribute actively to the design process. In addition, introduction of a new system is liable to change the work context and organizational processes, and will only be accepted if these changes are acceptable to the user. Participatory design therefore aims to refine system requirements iteratively through a design process in which the user is actively involved.

Participatory design has three specific characteristics. It aims to improve the work environment and task by the introduction of the design. This makes design and evaluation context or work oriented rather than system oriented. Secondly, it is characterized by collaboration: the user is included in the design team and can contribute to every stage of the design. Finally, the approach is iterative: the design is subject to evaluation and revision at each stage.

The participatory design process utilizes a range of methods to help convey information between the user and designer. They include

Brainstorming This involves all participants in the design pooling ideas. This is informal and relatively unstructured although the process tends to involve _on the- structuring of the ideas as they materialize.

Storyboarding : Storyboards can be used as a means of describing the user‘s day-to-day activities as well as the potential designs and the impact they will have.

Workshops These can be used to fill in the missing knowledge of both user and designer and provide a more focussed view of the design. They may involve mutual enquiry in which both parties attempt to understand the context of the design from each other‘s point of view. The designer questions the user about the work environment in which the design is to be used, and the user can query the designer on the technology and capabilities that may be available. This establishes common ground between the user and designer and sets the foundation for the design that is to be produced. The use of role play can also allow both user and designer to step briefly into one another‘s shoes.

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

Pencil and paper exercises These allow designs to be talked through and evaluated with very little commitment in terms of resources. Users can _walk through' typical tasks using paper mock-ups of the system design. This is intended to show up discrepancies between the user's requirements and the actual design as proposed. Such exercises provide a simple and cheap technique for early assessment of models.

Effective Technical and Human Implementation of Computer-based Systems (ETHICS)

ETHICS methodology, stakeholders are included as participants in the decision making process. ETHICS considers the process of system development as one of managing change: conflicts will occur and must be negotiated to ensure acceptance and satisfaction with the system. If any party is excluded from the decision-making process then their knowledge and contribution is not utilized and they are more likely to be dissatisfied. However, participation is not always complete. Mumford recognizes three levels of participation:

Consultative – the weakest form of participation where participants are asked for their opinions but are not decision makers.

Representative – a representative of the participant group is involved in the decision making process.

Consensus – all stakeholders are included in the decision-making process.

The usual practice is that design groups are set up to include representatives from each stakeholder group and these groups make the design decisions, overseen by a steering committee of management and employee representatives.

1. **Make the case for change.** Change for its own sake is inappropriate. If a case cannot be made for changing the current situation then the process ends and the system remains as it is.

2. **Identify system boundaries**. This focusses on the context of the current system and its interactions with other systems, in terms of business, existing technology, and internal and external organizational elements. How will the change impact upon each of these?

3. **Describe the existing system**, including a full analysis of inputs and outputs and the various other activities supported, such as operations, control and coordination.

4. **Define key objectives,** identifying the purpose and function of each area of the organization.

5. **Define key tasks:** what tasks need to be performed to meet these objectives?

6. **Define key information needs,** including those identified by analysis of the existing system and those highlighted by definition of key tasks.

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

7. **Diagnose efficiency needs,** those elements in the system that cause it to underperform or perform incorrectly. If these are internal they can be redesigned out of the new system; if they are external then the new system must be designed to cope with them.

8. **Diagnose job satisfaction needs,** with a view to increasing job satisfaction where it is low.

9. **Analyze likely future changes,** whether in technology, external constraints (such as legal requirements), economic climate or stakeholder attitudes. This is necessary to ensure that the system is flexible enough to cope with change.

10. **Specify and prioritize objectives based on efficiency,** job satisfaction and future needs. All stakeholders should be able to contribute here as it is a critical stage and conflicting priorities need to be negotiated. Objectives are grouped as either primary (must be met) or secondary

Ethnographic methods

Ethnography is based on very detailed recording of the interactions between people and between people and their environment. It has a special focus on social relationships and how they affect the nature of work. The ethnographer does not enter actively into the situation, and does not see things from a particular person‘s viewpoint. However, an aim is to be uncultured, to understand the situation from within its own cultural framework. Culture here means that of the particular workgroup or organization, rather than that of society as a whole. Ethnographers try to take an unbiased and open-ended view of the situation. They report and do not like to speculate, so it is often unclear how well their approach can contribute to the design of new systems.

Contextual inquiry

Contextual inquiry has much in common with the ethnographic tradition: it studies the user in context, trying to capture the reality of his work culture and practice. However, it is also an approach rooted in practice and it differs in a number of significant ways from pure ethnographic study: the intention is to understand and to interpret the data gathered, and rather than attempting to take an open-ended view, the investigator acknowledges and challenges her particular focus. In addition, the explicit aim is to design a new system, whereas in a pure ethnographic study, it would be open ended.

The model of contextual inquiry is of the investigator being apprenticed to the user to learn about his work. Interviews take place in the workplace so that the objects, artifacts and relationships of the work can be better understood. Examples of work are collected and both verbal and non-verbal communication is studied. The idea is to be as comprehensive in the data gathering as possible and to be concrete. Another central notion of contextual inquiry is

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

that of partnership: the user is the expert in the workplace and is therefore encouraged to lead the investigation. the investigator is not a passive observer. Her objective is to gain a shared understanding of how the work happens and, to do so, she questions meaning and offers interpretations of what she observes. The aim is to draw out the implications of comments and actions and understand (rather than assume) what they really mean. In order to do this honestly and effectively the investigator must know her focus – her pre-existing beliefs and assumptions about the situation – and be prepared to challenge and adjust them in the face of new information.

A number of models of the work are developed to capture what is important in the user's work situation:

The sequence model elaborates the steps required to complete a specific task, as well as the triggers that initiate that sequence of steps.

- The physical model maps the physical work environment and how it impacts upon work practice, for example, an office plan showing where different work activities happen.
- The flow model shows the lines of coordination and communication between the user and other participants within and outside the workplace.
- The cultural model reflects the influences of work culture and policy and shows the scope of these influences. This may include official or unofficial codes of behavior, common expectations (which may or may not be explicit) and value systems.
- The artifact model describes the structure and use of a particular artifact within the work process.

3.3 COMMUNICATION AND COLLABORATION MODELS

All computer systems, single-user or multi-user, interact with the work-groups and organizations in which they are used.

- We need to understand normal human–human communication:
  – face-to-face communication involves eyes, face and body
  – conversation can be analyzed to establish its detailed structure.
- This can then be applied to text-based conversation, which has:
  – reduced feedback for confirmation
  – less context to disambiguate utterances
  – slower pace of interaction but is more easily reviewed.
- Group working is more complex than that of a single person:

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

– it is influenced by the physical environment

– experiments are more difficult to control and record

– field studies must take into account the social situation.

3.3.1 FACE-TO-FACE COMMUNICATION

Face-to-face contact is the most primitive form of communication – primitive, that is, in terms of technology.

Transfer effects and personal space

When we come to use computer-mediated forms of communication, we carry forward all our expectations and social norms from face-to-face communication. People are very adaptable and can learn new norms to go with new media. However, success with new media is often dependent on whether the participants can use their existing norms. The rules of face-to-face conversation are not conscious, so, when they are broken, we do not always recognize the true problem. We may just have

feeling of unease, or we may feel that our colleague has been rude.

Eye contact and gaze

Normal conversation uses eye contact extensively, if not as intently. Our eyes tell us whether our colleague is listening or not; they can convey interest, confusion or boredom. Sporadic direct eye contact (both looking at one another‘s eyes) is important in establishing a sense of engagement and social presence. People who look away when you look at them may seem shifty and appear to be hiding something. Furthermore, relative frequency of eye contact and who _gives way‘ from direct eye contact is closely linked to authority and power.

Gestures and body language

When the participants are in the same room, the existence of electronic equipment can interfere with the body language used in normal face-to-face communication. The fact that attention is focused on keyboard and screen can reduce the opportunities for eye contact.

Also, large monitors may block participants‘ views of one another‘s bodies, reducing their ability to interpret gestures and body position. Most computer-supported meeting rooms recess monitors into the desks to reduce these problems.

Back channels, confirmation and interruption

It is easy to think of conversation as a sequence of utterances: A says something, then B says something, then back to A. This process is called turn-taking and is one of the fundamental structures of conversation. However, each utterance is itself the result of intricate negotiation and interaction. Consider the following transcript:

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

Alison: Do you fancy that film . . . er . . . _The Green‘ um . . . it starts at eight.

Brian: Grea

The nods, grimaces, shrugs of the shoulder and small noises are called back channels. They feed information back from the listener to the speaker at a level below the turn-taking of the conversation. The existence of back channels means that the speaker can afford to be slightly vague, adding details until it is obvious that the listener understands. Imagine making no response as someone talks to you, no little _yes‘es, no nods or raised eyebrows. You could answer questions and speak in turn, but not use back channels. It is likely that your colleague would soon become very uncomfortable, possibly rambling on with ever more detailed explanations, looking for some sign of understanding:

Do you fancy that film . . . er . . . ‘The Green’ um . . . the one with Charles Dermot in . . . you know with that song, er and the black cat on the poster . . . uhh

Turn-taking

Starting to speak in the middle of someone‘s utterance can be rude, but one can say something like _well uh‘ accompanied by a slight raising of the hands and a general tensing of the body and screwing of the eyes. This tells the speaker that you would like to interrupt, allowing a graceful transition. In this case, the listener requested the floor. Turn-taking is the process by which the roles of speaker and listener are exchanged. Back channels are often a crucial part of this process.

The role of _um‘s and _ah‘s is very important. They can be used by either participant during the gap to claim the turn. So, if Brian wanted to respond in the middle of the utterance, but had not yet framed his utterance, he might begin _um the one . . .‘. As it was, Brian did not respond, so Alison starts _er‘ which says to Brian _I‘m going to continue, but I‘m thinking‘. Alternatively, Alison could have started to _er‘ as soon as she had said the word _film‘. This would have told Brian not to interrupt. These turn-offering gaps are just the places where the speaker expects some back channel response even if no turn exchange takes place. A total lack of response will be taken, depending on the circumstances, as assent to the speaker, or perhaps as lack of understanding.

3.3.2 CONVERSATION

Conversational analyses are sociological and psychological understandings of conversation.

Basic conversational structure

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

Imagine we have a transcript of a conversation, recalling from that the production of such a transcript is not a simple task. For example, a slightly different version of Alison and Brian‘s conversation may look like this:

Alison: Do you fancy that film?

Brian: The uh (500 ms) with the black cat – _The Green whatsit‘?

Alison: Yeah, go at uh . . . (looks at watch – 1.2 s) . . . 20 to?

This transcript is quite heavily annotated with the lengths of pauses and even Alison‘s action of looking at her watch. it certainly lacks the wealth of gesture and back channel activity that were present during the actual conversation.

Transcripts may be less well documented, perhaps dropping the pause timings, or more detailed, adding more actions, where people were looking and some back channelling. Whilst thinking about the structure of conversation, the transcript above is sufficient.

The most basic conversational structure is turntaking. On the whole we have an alternating pattern: Alison says something, then Brian, then Alison again. The speech within each turn is called an utterance. There can be exceptions to this turn-taking structure even within two-party conversation.

Context

Take a single utterance from a conversation, and it will usually be highly ambiguous if not meaningless: _the uh with the black cat – —The Green whatsit‖‘. Each utterance and each fragment of conversation is heavily dependent on context, which must be used to disambiguate the utterance. We can identify two types of context within conversation: **internal context** – dependence on earlier utterances. For example, when Brian says _masses‘ in the last transcript, this is meaningful in the light of Alison‘s question _and lots of chocolate?‘. This in turn is interpreted in the context of Brian‘s original offer of gateau. **external context –** dependence on the environment. For example, if Brian had said simply _do you want one?‘, this could have meant a slice of gateau, or, if he had been holding a bottle, a glass of wine, or, if accompanied by a clenched fist, a punch on the nose.

Topics, focus and forms of utterance

Alison began the conversation with the topic of roses. Brian shifts to the related, but distinct, topic of greenfly. However, for some reason Alison has missed this shift in focus, so when she makes her second utterance, her focus and Brian‘s differ, leading to the breakdown in communication. The last two utterances are a recovery which re-establishes a shared dialog focus.

Breakdown and repair

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

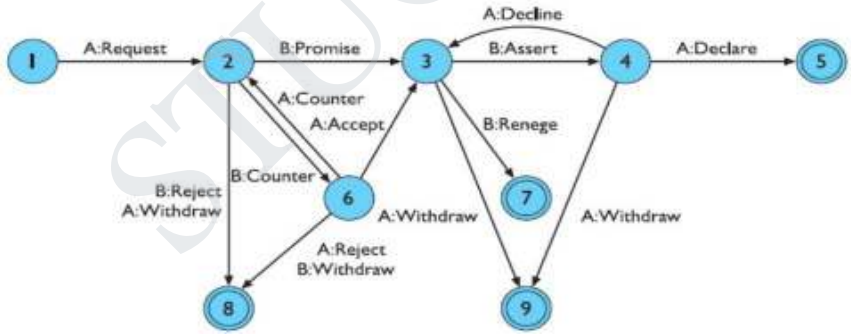
When Alison and Brian were talking about Brian‘s roses, they failed to maintain a shared focus. Brian

tried to interpret Alison‘s utterance in terms of his focus and failed, or rather the meaning in that focus was unusual – greenfly are the symbol of the English summer? He then questioned Alison and the confusion was cleared. This correction after breakdown is called repair.

Speech act theory

Speech act theory, has been both influential and controversial in CSCW. Not only is it an analytic technique, but it has been used as the guiding force behind the design of a commercial system, The basic premise of speech act theory is that utterances can be characterized by what they do. If you say _I‘m hungry‘, this has a certain propositional meaning – that you are feeling hungry. However, depending on who is talking and to whom, this may also carry the meaning _get me some food‘ – the intent of the statement is to evoke an action on the part of the hearer. Speech act theory concerns itself with the way utterances interact with the actions of the participants. The act of saying the words changes the state of the couple. Other acts include promises by the speaker to do something and requests that the hearer do something. These basic acts are called illocutionary points.

Individual speech acts can contribute to a conversation. The basic structure of conversations can then be seen as instances of generic conversations. One example of such a generic structure is a conversation for action (CfA).



3.3.3 TEXT BASED COMMUNICATION

Text-based communication is familiar to most people, in that they will have written and received letters. However, the style of letter writing and that of face-to face communication are very different. The text-based communication in groupware systems is acting as a speech substitute, and, thus, there are some problems adapting between the two media.

There are four types of textual communication in current groupware:

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

discrete – directed message as in email. There is no explicit connection between different messages, except in so far as the text of the message refers to a previous one.

linear – participants‘ messages are added in (usually temporal) order to the end of a single transcript.

non-linear – when messages are linked to one another in a hypertext fashion.

spatial – where messages are arranged on a two-dimensional surface.

Back channels and affective state

One of the most profound differences between face-to-face and text-based communication is the lack of fine-grained channels. Much of the coordination of face-to-face conversation depends on back channels and interpretation of the listener‘s expressions. Text-based communication loses these back channels completely. speaker would pause to seek back channel confirmation or to offer the floor, the text _speaker‘ must either continue regardless, or finish the message, effectively passing the turn.

These normally convey the affective state of the speaker (happy, sad, angry, humorous) and the illocutionary force of the message (an important and urgent demand or a deferential request). Email users have developed explicit tokens of their affective state by the use of _flaming‘ and _smilies‘, using punctuation and acronyms; for example:

:-) – smiling face, happy

:-(– sad face, upset or angry

;-) – winking face, humorous

LOL – laughing out loud.

Grounding constraints

This grounding process is linked strongly with the types of channels through which the conversants communicate. Clark and Brennan describe the properties of these channels in terms of grounding constraints. These include:

Co-temporality – an utterance is heard as soon as it is said (or typed); **simultaneity** – the participants can send and receive at the same time; **sequence** – the utterances are ordered.

These are all constraints which are weaker in text-based compared with face-to-face interaction.

In a text-based system, different participants can compose simultaneously, but they lack cotemporality. As we saw, even if the messages appear as they are produced, they will not be read in real time. In addition, the messages may only be delivered when complete and even then may be delayed by slow communications networks.

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

Turn-taking

In a pair of participants, turn-taking is simple; first one person says something, then the other. The only problem is deciding exactly when the exchange should happen. With three or more participants, turn-taking is more complex. They must decide who should have the next turn. This is resolved by face-to-face groups in a number of ways. First, the conversation may, for a period, be focused on two of the parties, in which case normal two-party turn-taking holds. Secondly, the speaker may specifically address another participant as the utterance is finished, either implicitly by body position, or explicitly: _what do you think Alison?‘ Finally, the next speaker may be left open, but the cotemporality of the audio channel allows the other participants to negotiate the turn. Basically, whoever speaks first, or most strongly, gets in. These mechanisms are aided by back channels, as one of the listeners may make it clear that she wants to speak. In this case, either the speaker will explicitly pass the turn (the second option above), or at least the other listeners are expecting her to speak. In addition, the movement between effective two-party conversation (the first option) and open discussion will be mediated by back channel messages from the other participants.

In an unstructured text-based conversation the third option is not available, nor, of course, are the back channels. Paired conversation is quite common and the second option, explicitly naming the next speaker, is possible. This naming is not particularly natural unless a direct question is being asked. In both options, the absence of back channels makes it difficult for another listener to interrupt the conversation. Some systems use more structured mechanisms to get round these problems, perhaps having a round-robin protocol (each participant _speaks‘ in turn) or having a queue of turn-requests. Whether the strictures of such mechanisms are worse than the problems of occasional breakdown depends very much on the context and is a matter of opinion.

Context and deixis

Utterances are highly ambiguous and are only meaningful with respect to external context, the state of the world, and internal context, the state of the conversation. Both of these are problems in text-based communication.

The very fact that the participants are not co-present makes it more difficult to use external context to disambiguate utterances. This is why many groupware systems strive so hard to make the participants‘ views the same; that is, to maintain WYSIWIS (_what you see is what I see‘).

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

Whatever the means of direct communication, remote participants have difficulty in using deictic reference. They cannot simply say _that one‘, but must usually describe the referrant: _the big circle in the corner‘. If their displays are not WYSIWIS then they must also ensure that their colleague‘s display includes the object referred to and that the description is unambiguous. Asynchronous participants have even more problems with deixis as there is no opportunity for their colleagues to clarify a reference (without extremely lengthy exchanges). The objects referred to by a message may have changed by the time someone comes to read it! Similarly, group pointers are not really an option, but one can use methods of linking the conversation to its context, either by embedding it within the objects as annotations or by having hypertext links between the conversation and the object. The trouble does not end with external context; there are also problems with deictic reference to internal context. In speech, the context is intimately connected to linear sequence and adjacency. As we have seen, even in linear text transcripts, overlap breaks the strict sequentiality of the conversation, and thus causes problems with indexicals and with context in general.

1. Alison: Brian‘s got some lovely roses.
2. Brian: I‘m afraid they‘re covered in greenfly.
3. Clarise: I‘ve seen them, they‘re beautiful.

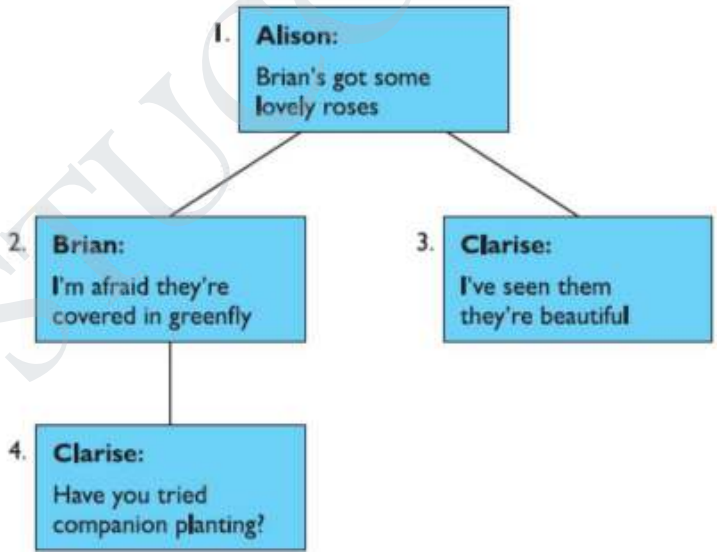


Fig: Hypertext conversation structure

Pace and granularity

The term pace is being used in a precise sense above. Imagine a message being composed and sent, the recipient reading (or hearing) the message and then composing and sending a reply. The pace of the conversation is the rate of such a sequence of connected

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

messages and replies. Clearly, as the pace of a conversation reduces, there is a tendency for the granularity to increase. To get the same information across, you must send more per message. However, it is not as easy as that. We have seen the importance of feedback from listener to speaker in clarifying meaning and negotiating common ground. Even most monologs are interactive in the sense that the speaker is constantly looking for cues of comprehension in the listener. Reducing the pace of a conversation reduces its interactivity.

In a hypertext-based system one can expand several branches of a conversation tree, but in speech or in a linear text transcript the conversation follows one branch. To overcome these limitations, people adopt several coping strategies. The simplest strategy is just to avoid conversation. This can be done by delegating parts of a task to the different participants. Each participant can then perform much of the task without communication. They must still communicate for large-scale strategic decisions, but have significantly reduced the normal communications. Of course, this approach reduces communication by reducing collaboration. More interesting in a cooperative work setting are two coping strategies which increase the chunk size of messages in order to reduce the number of interactions required to complete a task. These strategies are frequently seen in both text-based conferences and in letter writing.

The first of these coping strategies is multiplexing. Basically, the conversants hold several conversations in parallel, each message referring to several topics. In terms of the conversation tree, this corresponds to going down several branches at once.

Linear text vs. Hypertext

Multiplexed messages can be represented as updates to several parts of the hypertext, thus reducing the likelihood of breakdown and lost topics. In addition, if the messages themselves can be mini-hypertexts, then eager messages listing several possible courses of action can be explicitly represented by the message.

Even static hypertexts, which have been carefully crafted by their authors, can be difficult to navigate. A hypertext that is created _on the fly‘ is unlikely to be comprehensible to any but those involved in its creation. Conklin and Begeman, themselves associated with the hypertext based argumentation tool gIBIS, conclude that _traditional linear text provides a continuous, unwinding thread of context.

For the asynchronous reader trying to catch up with a conversation, a linear transcript is clearly easier, but it is precisely in more asynchronous settings where overlap in linear text is most likely to cause confusion.

3.3.4 GROUP WORKING

Group behavior is more complex still as we have to take into account the dynamic social relationships during group working. We will begin by looking at several factors which affect group working, and then discuss the problems of studying group working .

Group dynamics

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

organizational relationships such as supervisor/supervisee are relatively stable, the roles and relationships within a group may change dramatically within the lifetime of a task and even within a single work session. For example, studies of joint authoring have found that roles such as author, co-author and commentator change throughout the lifetime of a document. This means that systems, such as co-authoring systems, which use a formal concept of role, must allow these roles to change together with the socially defined roles.

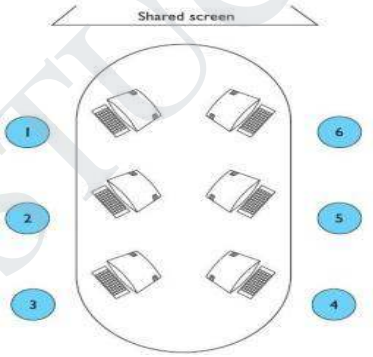
A person may be an author of a book or paper, but never write the words in it, acting instead as a source of ideas and comments. A particular case of this is the biographical story where the individual

concerned and a professional writer co-author the book, but only the professional author writes. A co-authoring system such as Quilt would call the non-writing author a

_commentator‘ or a _reviewer‘, but not an _author‘. One can imagine some of the social friction such naming will cause.

Physical layout

The designers of Capture Lab, an eight-person meeting room, considered all these features and many other subtle effects. However, the users still had some difficulty in adapting to the power positions in the electronic meeting room. At first sight, the electronic meeting room is not unlike a normal conference room. If the shared screen is a whiteboard or an overhead projector, then the most powerful position is toward the front of the room . Managers would normally take this seat as they can then easily move to the whiteboard or overhead projector to point out some item and draw the group‘s attention.



Distributed cognition

Traditional views talk about the movement of information between working memory and long-term memory: it is not so difficult then to regard bits of paper, books and computer systems as extensions to these internal memory systems. Similarly, many models of human cognition regard the mind as a set of interacting subsystems. The step to regarding several people as involved in joint thinking is not difficult.

3.4 HYPERTEXT, MULTIMEDIA AND THE WORLD WIDE WEB

- Hypertext allows documents to be linked in a nonlinear fashion.
- Multimedia incorporates different media: sound, images, and video.

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

- The World Wide Web is a global hypermedia system.
- Animation and video can show information that is difficult to convey statically.
- Applications of hypermedia include online help, education and e-commerce.
- Design for the World Wide Web illustrates general hypermedia design, but also has its own special problems.
- Dynamic web content can be used for simple online demonstration

3.4.1 Hypertext.

•The term hypertext means certain extra capabilities imparted to normal or standard text.

•Technical documentation consists often of a collection of independent information units.

•It consists of cross references which lead to multiple searches at different places for the reader.

•Hypertext is text which is not constrained to be linear and it contains links to other texts which is known as hyperlinks.

•Hypertext is mostly used on World Wide Web for linking and navigating through different web pages.

•A hypertext consists of two different parts: Anchor and link

•An anchor or node is an entry point to another document. In some cases instead of a text an image a video or some other non-textual element.

•A link or pointer provide connection to other information unit known as target documents.

3.4.2 Multimedia refers to using computers to integrate text, graphics, animation, audio, and video into one

application. Most multimedia applications are interactive, so that users may choose the material to view, define the order in which it is presented, and obtain feedback on their actions.

Interactivity also makes multimedia very suitable for video games, electronic newspapers and magazines, electronic books and references, simulations, virtual reality, and computer-based training. Multimedia applications can be created by using a multimedia authoring software. Many multimedia applications are also deliverable via the World Wide Web.

Graphics

A graphic is a digital representation of information such as a drawing, a chart, or a photograph.Graphics were the first media used to enhance the originally text-based Internet. Two of the more common graphical formats on the Web are JPEG and GIF. Other graphical formats such as BMP and TIFF have larger file sizes, and may require special viewer software to display on the Web. To reduce download times for graphics, some Web sites use thumbnails, which is a smaller version of a larger graphical image that a user may click to display the fullsized image.

Audio

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

Audio can be music, speech, or any other sound. Common audio formats include WAV, MID, and MP3. Some Web sites use streaming audio, which allows a user to listen to the sound as it downloads to the computer. Two accepted standards for streaming audio on the Web are Windows Media Player and RealAudio.

Video

Video consists of full-motion images that are played back at various speed. Most video is also accompanied with audio. MPEG is a popular video compression standard defined by the Moving Picture Experts Group (MPEG). Streaming video allows a user to view longer or live video images as they download to the computer from the Web. Two popular streaming video formats are Windows Media Player and RealVideo.

Animation is the appearance of motion that is created by displaying a series of still images in rapid sequence. Animated GIF is a popular type of animation format, which combines several images into a single GIF file.

Multimedia Authoring Software

Multimedia authoring software combines text, graphics, animation, audio, and video into an application. Multimedia is widely used in video games, electronic newspapers and magazines, electronic books and references, simulations, virtual reality, and computer-based training. Popular multimedia authoring software includes Macromedia AuthorWare, Macromedia Director, and Macromedia Flash. Multimedia computers have facilities for handling sound and video as well as text and graphics. Most computers are now sold with a multimedia capacity.

3.4.3 Web - World Wide Web

The Web, or World Wide Web, is basically a system of Internet servers that support specially formatted documents. The documents are formatted in a markup language called HTML (HyperText Markup Language) that supports links to other documents, as well as graphics, audio, and video files.

This means you can jump from one document to another simply by clicking on hot spots. Not all Internet servers are part of the World Wide Web.

The Internet is a worldwide collection of networks that links millions of businesses, government offices, educational institutions, and individuals. Data is transferred over the Internet using servers, which are computers that manage network resources and provide centralized storage areas, and clients, which are computers that can access the contents of the storage areas. The data travels over communications lines. Each computer or device on a communications line has a numeric address called an IP (Internet protocol) address, the text version of which is called a domain name. Every time you specify a domain name, a DNS (domain name system) server translates the domain name into its associated IP address, so data can route to the correct computer.

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

An Internet service provider (ISP) provides temporary Internet connections to individuals and companies. An online service provider (OSP) also supplies Internet access, in addition to a variety of special content and services. A wireless service provider (WSP) provides wireless Internet access to users with wireless modems or Web-enabled handheld computers or devices.

Employees and students often connect to the Internet through a business or school network that connects to a service provider. For home or small business users, dial-up access provides an easy and inexpensive way to connect to the Internet. With dial-up access, you use a computer, a modem, and a regular telephone line to dial into an ISP or OSP. Some home and small business users opt for newer, high-speed technologies. DSL (digital subscriber line) provides high-speed connections over a regular copper telephone line. A cable modem provides high-speed Internet connections through a cable television network.

The World Wide Web (WWW or Web) consists of a worldwide collection of electronic documents called Web pages. A browser is a software program used to access and view Web pages. Each Web page has a unique address, called a URL (Uniform Resource Locator), that tells a browser where to locate the Web page. A URL consists of a protocol, domain name, and sometimes the path to a specific Web page or location on a Web page. Most URLs begin with http://, which stands for hypertext transfer protocol, the communications standard that enables pages to transfer on the Web.

A search engine is a software program you can use to find Web sites, Web pages, and Internet files. To find a Web page or pages, you enter a relevant word or phrase, called search text or keywords, in the search engine's text box. Many search engines then use a program called a spider to read pages on Web sites and create a list of pages that contain the keywords. Any Web page that is listed as the result of the search is called a hit. Each hit is a link that can be clicked to display the associated Web site or Web page.

There are six basic types of Web pages. An advocacy Web page contains content that describes a cause, opinion, or idea. A business/marketing Web page contains content that promotes or sells products or services. An informational Web page contains factual information. A news Web page contains newsworthy material including stories and articles relating to current events, life, money, sports, and the weather. A portal Web page offers a variety of Internet services from a single, convenient location. A personal Web page is maintained by a private individual who normally is not associated with any organization.

Many exciting Web pages use multimedia. Multimedia refers to any application that integrates text with one of the following elements: graphics, sound, video, virtual reality, or other media elements.

A graphic is a digital representation of information such as a drawing, chart, or photograph. Two common file formats for graphical images on the Web are JPEG (Joint

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

Photographic Experts Group) and GIF (Graphics Interchange Format), which use compression techniques to reduce the size of graphics files and thus speed downloading.

Animation is the appearance of motion created by displaying a series of still images in rapid sequence. One popular type of animation, called an animated GIF, uses computer animation and graphics software to combine several images into a single GIF file.

Audio is music, speech, or any other sound. A common format for audio files on the Web is MP3, a popular technology that compresses audio. More advanced Web audio applications use streaming audio, which transfers audio data in a continuous and even flow, allowing users to listen to the sound as it downloads. Video consists of full-motion images that are played back at various speeds. Video files often are quite large in size. The Moving Pictures Experts Group (MPEG) defines a popular video compression standard. Streaming video allows you to view longer or live video images as they are downloaded.

Virtual reality (VR) is the use of computers to simulate a real or imagined environment that appears as a three-dimensional (3-D) space. A VR world is an entire 3-D site that contains infinite space and depth.

A variety of services are used widely on the Internet, including e-mail, FTP, newsgroups and message boards, mailing lists, chat rooms, and instant messaging. E-mail (electronic mail) is the transmission of messages and files via a computer network. You use an e-mail program to create, send, receive, forward, store, print, and delete messages. To receive messages, you need an e-mail address, which is a combination of a username and a domain name that identifies a user.

FTP (File Transfer Protocol) is an Internet standard that allows you to upload and download files with other computers on the Internet. An FTP server is a computer that allows you to use FTP to upload files to, and download files from, an FTP site. With anonymous FTP, anyone can transfer some, if not all, available files. A newsgroup is an online area in which users conduct written discussions about a particular subject. The computer that stores and distributes newsgroup messages is called a news server. You use a program called a newsreader to access a newsgroup, read previously entered messages (called articles), and add (post) messages of your own.

A thread consists of the original article and all subsequent related replies. In a moderated newsgroup, a moderator reviews articles and posts them, if appropriate. A message board is a popular Web-based type of discussion group that does not require a newsreader and typically is easier to use than a newsgroup. A mailing list is a group of e-mail names and addresses given a single name. To add your e-mail name and address to a mailing list you subscribe to it; to remove your name, you unsubscribe.

A chat is real-time (meaning everyone involved in the chat is online at the same time) typed conversation that takes place on a computer. A location on an Internet server that

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

permits users to chat is called a chat room. Some chat rooms support voice chats and video chats, where you can hear or see others and they can hear or see you as you chat. A chat client is a program on your computer that allows you to connect to a chat server and start a chat session. Instant messaging (IM) is a real-time Internet communications service that notifies you when one or more people are online and then allows you to exchange messages or join a private chat room.

STUCOR APP
STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

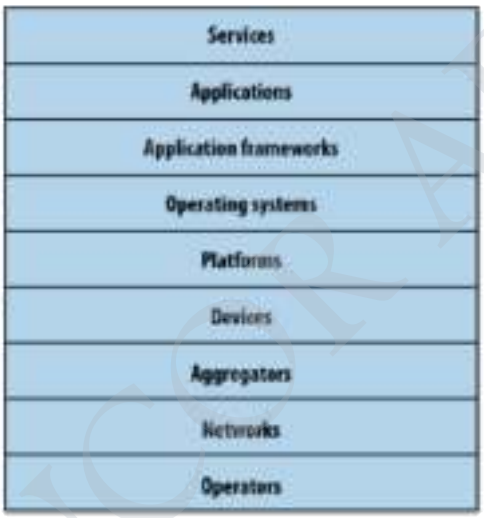
UNIT IV

MOBILE HCI

Mobile Ecosystem: Platforms, Application frameworks- Types of Mobile Applications: Widgets, Applications, Games- Mobile Information Architecture, Mobile 2.0, Mobile Design: Elements of Mobile Design, Tools.

4.1 The Mobile Ecosystem

Mobile is an entirely unique ecosystem and, like the Internet, it is made up of many different parts that must all work seamlessly together. With mobile technology, the parts are different, and because you can use mobile devices to access the Internet, that means that not only do you need to understand the facets of the Internet, but you also need to understand the mobile ecosystem.



4.1.1 Operators

The base layer in the mobile ecosystem is the operator. Operators go by many names, depending on what part of the world you happen to be in or who you are talking to. Operators can be referred to as Mobile Network Operators (MNOs); mobile service providers, wireless carriers, or simply carriers; mobile phone operators; or cellular companies. In the mobile community, we officially refer to them as operators, though in the United States, there is a tendency to call them Carriers. Operators are what essentially make the entire mobile ecosystem work. They are the gatekeepers to the kingdom. They install cellular towers, operate the cellular network, make services (such as the Internet) available for mobile subscribers, and they often maintain relationships with the subscribers, handling billing and support, and offering subsidized device sales and a network of retail stores. The operator's role in the ecosystem is to create and maintain a specific set of wireless services over a reliable cellular network. That's it. However, to grow the mobile market over the past decade,

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

the operator has been required to take a greater role in the mobile ecosystem, doing much more than just managing the network. For example, they have had to establish trust with subscribers to handle the billing relationship and to offer.

4.1.2 Networks

Operators operate wireless networks. Remember that cellular technology is just a radio that receives a signal from an antenna. The type of radio and antenna determines the capability of the network and the services you can enable on it. You'll notice that the vast majority of networks around the world use the GSM standard, using GPRS or GPRS EDGE for 2G data and UMTS or HSDPA for 3G. We also have CDMA (Code Division Multiple Access) and its 2.5G hybrid CDMA2000, which offers greater coverage than its more widely adopted rival. So in places like the United States or China, where people are more spread out, CDMA is a great technology. It uses fewer towers, giving subscribers fewer options as they roam networks.

4.1.3 Devices

What you call phones, the mobile industry calls handsets or terminals. These are terms that I think are becoming outdated with the emergence of wireless devices that rely on operator networks, but do not make phone calls. The number of these "other" devices is a small piece of the overall pie right now, but it's growing rapidly. Let's focus on the biggest slice of the device pie—mobile phones. As of 2008, there are about 3.6 billion mobile phones currently in use around the world; just more than half the planet's population has a mobile phone Most of these devices are feature phones, making up the majority of the marketplace. Smartphones make up a small sliver of worldwide market share and maintain a healthy percentage in the United States and the European Union; smartphone market share is growing with the introduction of the iPhone and devices based on the Android platform. As next-generation devices become a reality, the distinction between feature phones and smartphones will go away. In the next few years, feature phones will largely be located in emerging and developing markets.

4.2 PLATFORMS

A mobile platform's primary duty is to provide access to the devices. To run software and services on each of these devices, you need a platform, or a core programming language in which all of your software is written. Like all software platforms, these are split into three categories: licensed, proprietary, and open source.

4.2.1 Licensed

Licensed platforms are sold to device makers for nonexclusive distribution on devices. The goal is to create a common platform of development Application Programming Interfaces (APIs) that work similarly across multiple devices with the least possible effort

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

required to adapt for device differences, although this is hardly reality. Following are the licensed platforms:

Java Micro Edition (Java ME)

Java ME is by far the most predominant software platform of any kind in the mobile ecosystem. It is a licensed subset of the Java platform and provides a collection of Java APIs for the development of software for resource constrained devices such as phones.

Binary Runtime Environment for Wireless (BREW)

BREW is a licensed platform created by Qualcomm for mobile devices, mostly for the U.S. market. It is an interface-independent platform that runs a variety of application frameworks, such as C/C++, Java, and Flash Lite.

Windows Mobile

Windows Mobile is a licensable and compact version of the Windows operating system, combined with a suite of basic applications for mobile devices that is based on the Microsoft Win32 API.

LiMo

LiMo is a Linux-based mobile platform created by the LiMo Foundation. Although Linux is open source, LiMo is a licensed mobile platform used for mobile devices. LiMo includes SDKs for creating Java, native, or mobile web applications using the WebKit browser framework.

4.2.2 Proprietary

Proprietary platforms are designed and developed by device makers for use on their devices. They are not available for use by competing device makers. These include: Palm

Palm uses three different proprietary platforms. Their first and most recognizable is the Palm OS platform based on the C/C++ programming language; this was initially developed for their Palm Pilot line, but is now used in low-end smartphones such as the Centro line. As Palm moved into higher-end smartphones, they started using the Windows Mobile-based platform for devices like the Treo line. The most recent platform is called webOS, is based on the WebKit browser framework, and is used in the Prē line.

BlackBerry

Research in Motion maintains their own proprietary Java-based platform, used exclusively by their BlackBerry devices.

iPhone

Apple uses a proprietary version of Mac OS X as a platform for their iPhone and iPod touch line of devices, which is based on Unix.

4.2.3 Open Source

Open source platforms are mobile platforms that are freely available for users to download, alter, and edit. Open source mobile platforms are newer and slightly controversial,

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

but they are increasingly gaining traction with device makers and developers. Android is one of these platforms. It is developed by the Open Handset Alliance, which is spearheaded by Google. The Alliance seeks to develop an open source mobile platform based on the Java programming language.

4.3 APPLICATION FRAMEWORKS

Application frameworks often run on top of operating systems, sharing core services such as communications, messaging, graphics, location, security, authentication, and many others.

Java

Applications written in the Java ME framework can often be deployed across the majority of Java-based devices, but given the diversity of device screen size and processor power, cross-device deployment can be a challenge.

S60

The S60 platform, formerly known as Series 60, is the application platform for devices

that run the Symbian OS. S60 is often associated with Nokia devices—Nokia owns the platform—but it also runs on several non-Nokia devices. S60 is an open source framework. S60 applications can be created in Java, the Symbian C++ framework, or even Flash Lite.

BREW

Applications written in the BREW application framework can be deployed across the majority of BREW-based devices, with slightly less cross-device adaption than other frameworks.

Flash Lite

Adobe Flash Lite is an application framework that uses the Flash Lite and ActionScript frameworks to create vector-based applications. Flash Lite applications can be run within the Flash Lite Player, which is available in a handful of devices around the world.

Flash Lite is a promising and powerful platform, but there has been some difficulty getting it on devices. A distribution service for applications written in Flash Lite is long overdue.

Windows Mobile

Applications written using the Win32 API can be deployed across the majority of Windows Mobile-based devices. Like Java, Windows Mobile applications can be downloaded and installed over the air or loaded via a cable-connected computer.

Cocoa Touch

Cocoa Touch is the API used to create native applications for the iPhone and iPod touch. Cocoa Touch applications must be submitted and certified by Apple before being

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

included in the App Store. Once in the App Store, applications can be purchased, downloaded, and installed over the air or via a cable-connected computer.

Android SDK

The Android SDK allows developers to create native applications for any device that runs the Android platform. By using the Android SDK, developers can write applications in C/C++ or use a Java virtual machine included in the OS that allows the creation of applications with Java, which is more common in the mobile ecosystem.

Web Runtimes (WRTs)

Nokia, Opera, and Yahoo! provide various Web Runtimes, or WRTs. These are meant to be miniframeworks, based on web standards, to create mobile widgets. Both Opera‘s and

Nokia‘s WRTs meet the W3C-recommended specifications for mobile widgets.

WebKit

WebKit is a browser technology, so applications can be created simply by using web technologies such as HTML, CSS, and JavaScript. WebKit also supports a number of recommended standards not yet implemented in many desktop browsers. Applications can be run and tested in any WebKit browser, desktop, or mobile device.

The Web

The Web is the only application framework that works across virtually all devices and all platforms. Although innovation and usage of the Web as an application framework in mobile has been lacking for many years, increased demand to offer products and services outside of operator control, together with a desire to support more devices in shorter development cycles, has made the Web one of the most rapidly growing mobile application platforms to date.

4.4 Types of Mobile Applications

4.4.1 Mobile Application Medium Types

The mobile medium type is the type of application framework or mobile technology that presents content or information to the user. It is a technical approach regarding which type of medium to use; this decision is determined by the impact it will have on the user experience.

SMS

The most basic mobile application you can create is an SMS application. Although it might seem odd to consider text messages applications, they are nonetheless a designed experience. Given the ubiquity of devices that support SMS, these applications can be useful tools when integrated with other mobile application types. Typically, the user sends a single keyword to a five-digit short code in order to return information or a link to premium content. For example, sending the keyword “freebie” to a hypothetical short code “12345” might

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

return a text message with a coupon code that could be redeemed at a retail location, or it could include a link to a free ringtone. SMS applications can be both "free," meaning that there is no additional charge beyond the text message fees an operator charges, and "premium," meaning that you are charged an additional fee in exchange for access to premium content. The most common uses of SMS applications are mobile content, such ringtones and images, and to interact with actual goods and services. Some vending machines can dispense beverages when you send them an SMS; SMS messages can also be used to purchase time at a parking meter or pay lot. A great example of how SMS adds incredible value would be Twitter, where users can receive SMS alerts from their friends and post to their timeline from any mobile device.

4.4.2 Mobile websites

Mobile website is a website designed specifically for mobile devices, not to be confused with viewing a site made for desktop browsers on a mobile browser. Mobile websites are characterized by their simple "drill-down" architecture, or the simple presentation of navigation links that take you to a page a level deeper.

Mobile websites often have a simple design and are typically informational in nature, offering few—if any—of the interactive elements you might expect from a desktop site. Mobile websites have made up the majority of what we consider the mobile web for the past decade, starting with the early WML-based sites (not much more than a list of links) and moving to today's websites, with a richer experience that more closely resembles the visual aesthetic users have come to expect with web content. Though mobile websites are fairly easy to create, they fail to display consistently across multiple mobile browsers—a trait common to all mobile web mediums. The mobile web has been gradually increasing in usage over the years in most major markets, but the limited experience offered little incentive to the user. Many compare the mobile web to a 10-year-old version of the Web: slow, expensive to use, and not much to look at. As better mobile browsers started being introduced to device platforms like the iPhone and Android, the quality of mobile websites began to improve dramatically, and with it, usage improved. For example, in just one year, the U.S. market

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

went from being just barely in the top five consumers of the mobile web to number one, largely due to the impact of the iPhone alone.

Pros

• They are easy to create, maintain, and publish.

• They can use all the same tools and techniques you might already use for desktop sites.

• Nearly all mobile devices can view mobile websites.

Cons

The cons of mobile websites are:

• They can be difficult to support across multiple devices.

• They offer users a limited experience.

• Most mobile websites are simply desktop content reformatted for mobile devices.

4.4.3 Mobile Web Widgets

Largely in response to the poor experience provided by the mobile web over the years, there has been a growing movement to establish mobile widget frameworks and platforms. For years the mobile web user experience was severely underutilized and failed to gain traction in the market, so several operators, device makers, and publishers began creating widget platforms (Figure) to counter the mobile web's weaknesses.

Figure: An example mobile web widget

I initially saw mobile web widgets as another attempt by the mobile industry to hype a technology that no one wants. I liked to quiz mobile web widget advocates about what makes mobile web widgets different than what we can do with the mobile web.

A component of a user interface that operates in a particular

way. The ever-trusty Wikipedia defines a web widget this way:

A portable chunk of code that can be installed and executed within any separate HTML based web page by an end user without requiring additional compilation.

Between these two definitions is a better answer:

A mobile web widget is a standalone chunk of HTML-based code that is executed by the end user in a particular way. Mobile web widgets are small web applications that can't run by themselves; they need to be executed on top of something else. I think one reason for

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

all the confusion around what is a mobile web widget is that this definition can also encompass any web application that runs in a browser. Opera Widgets, Nokia Web Run Time (WRT), Yahoo! Blueprint, and Adobe Flash Lite are all examples of widget platforms that work on a number of mobile handsets. Using a basic knowledge of HTML (or vector graphics in the case of Flash), you can create compelling user experiences that tap into device features and, in many cases, can run while the device is offline.

Pros

The pros of mobile web widgets are:

• They are easy to create, using basic HTML, CSS, and JavaScript knowledge.

• They can be simple to deploy across multiple handsets.

• They offer an improved user experience and a richer design, tapping into device features and offline use.

Cons

The cons of mobile web widgets are:

• They typically require a compatible widget platform to be installed on the device.

• They cannot run in any mobile web browser.

• They require learning additional proprietary, non-web-standard techniques.

4.4.4 Mobile Web Applications

Mobile web applications are mobile applications that do not need to be installed or compiled on the target device. Using XHTML, CSS, and JavaScript, they are able to provide an application-like experience to the end user while running in any mobile web browser. By Application -like experience, I mean that they do not use the drill-down or page metaphors in which a click equals a refresh of the content in view. Web applications allow users to interact with content in real time, where a click or touch performs an action within the current view.

The history of how mobile web applications came to be so commonplace is interesting, and is one that I think can give us an understanding of how future mobile trends can be assessed and understood. Shortly after the explosion of Web 2.0, web applications like Facebook, Flickr, and Google Reader hit desktop browsers, and there was discussion of how to bring those same web applications to mobile devices. The Web 2.0 movement brought user-centered design principles to the desktop web, and those same principles were sorely needed in the mobile web space as well.

The challenge, as always, was device fragmentation. The mobile browsers were years behind the desktop browsers, making it nearly impossible for a mobile device to render a comparable experience. While XHTML support had become fairly commonplace across devices, the rendering of CSS2 was wildly inconsistent, and support for Java- Script, necessary or simple DHTML, and Ajax was completely nonexistent. To make matters worse, the perceived market demand for mobile web applications was not seen as a priority with

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

many operators and device makers. It was the classic chickenor- the-egg scenario. What had to come first, market demand to drive browser innovation or optimized content to drive the market? With the introduction of the first iPhone, we saw a cataclysmic change across the board.

Using WebKit, the iPhone could render web applications not optimized for mobile devices as perfectly usable, including DHTML- and Ajax-powered content. Developers quickly got on board, creating mobile web applications optimized mostly for the iPhone (Figure). The combination of a high-profile device with an incredibly powerful mobile web browser and a quickly increasing catalog of nicely optimized experiences created the perfect storm the community had been waiting for.

Usage of the mobile web exploded with not just users of the iPhone, but users of other handsets, too. Because Web applications being created for the iPhone were based on web standards, they actually worked reasonably well on other devices. Operators and device makers saw that consumers wanted not just the mobile web on their handsets, but the regular Web, too.

Pros:

The pros of mobile web applications are:

• They are easy to create, using basic HTML, CSS, and JavaScript knowledge.

• They are simple to deploy across multiple handsets.

• They offer a better user experience and a rich design, tapping into device features & offline use.

• Content is accessible on any mobile web browser.

Cons:

The cons of mobile web applications are:

• The optimal experience might not be available on all handsets.

• They can be challenging (but not impossible) to support across multiple devices.

• They don‘t always support native application features, like offline mode, location lookup, file system access, camera, and so on.

4.4.5 Games

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

The most popular of all media available to mobile devices. Technically games are really just native applications that use the similar platform SDKs to create immersive experiences (Figure). But I treat them differently from native applications for two reasons: they cannot be easily duplicated with web technologies, and porting them to multiple mobile platforms is a bit easier than typical platform-based applications.

Seeing as how we have yet to see these types of gaming experiences appear on the desktop using standard web technologies, I believe we are still a few years out from seeing them on mobile devices. Adobe‘s Flash and the SVG (scalable vector graphics) standard are the only way to do it on the Web now, and will likely be how it is done on mobile devices in the future, the primary obstacle being the performance of the device in dealing with vector graphics. The reason games are relatively easy to port (—relatively‖ being the key word), is that the bulk of the gaming experience is in the graphics and actually uses very little of the device APIs. The game mechanics are the only thing that needs to adapted to the various platforms. Like in console gaming, there are a great number of mobile game porting shops that can quickly take a game written in one language and port it to another.

These differences, in my mind, are what make mobile games stand apart from all other application genres—their capability to be unique and difficult to duplicate in another application type, though the game itself is relatively easy to port. Looking at this model for other application areas—namely, the mobile web—could provide helpful insight into how we create the future of mobile web applications.

Pros: The pros of game applications are:

• They provide a simple and easy way to create an immersive experience.

• They can be ported to multiple devices relatively easily.

Cons: The cons of game applications are:

• They can be costly to develop as an original game title.

• They cannot easily be ported to the mobile web.

4.5 MOBILE INFORMATION ARCHITECTURE

What Is Information Architecture?

The structural design of shared information environments

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

- The combination of organizations, labelling, search, and navigation systems within websites and intranets
- The art and science of shaping information products and experiences to support usability and find ability
- An emerging discipline and community of practice focused on bringing principles of design and architecture to the digital landscape

Information architecture

The organization of data within an informational space. In other words, how the user will get to information or perform tasks within a website or application.

Interaction design

The design of how the user can participate with the information present, either in a direct or indirect way, meaning how the user will interact with the website of application to create a more meaningful experience and accomplish her goals.

Information design

The visual layout of information or how the user will assess meaning and direction given the information presented to him.

Navigation design

The words used to describe information spaces; the labels or triggers used to tell the users what something is and to establish the expectation of what they will find.

Interface design

The design of the visual paradigms used to create action or understanding.

The role of information architecture is played by a variety of people, from product managers to designers and even developers. To make things more confusing, information architecture can be called many different things throughout the design and development process. Words like intuitive, simple, findable, usable, or the executive favourite easy to-use—all describe the role that information architects play in creating digital experiences.

The visual design of your product, what frameworks you use, and how it is developed are integral to the success of any product, but the information architecture stands apart as being the most crucial element of your product. It is the first line of scrimmage—the user‘s first impression of your product. Even if you have the best design, the best code, and the best backend service, if the user cannot figure out how to use it, she will fail and so will your product.

Mobile Information Architecture

Information architecture has become a common discipline in the web industry; unfortunately, the mobile industry like software has only a handful of specialized mobile information architects. Although mobile information architecture is hardly a discipline in its own right, it certainly ought to be. This is not because it is so dissimilar from its desktop cousin, but

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

because of context, added technical constraints, and needing to display on a smaller screen as much information as we would on a desktop.

The role of a mobile information architect would be to interpret this content to the mobile context. Do you use the same structure, or sections? Do you present the same information above the fold? If so, how should that be prioritized? How does the user navigate to other areas? Do you use the same visual and interaction paradigms, or invent new ones? And if you do start to invent new paradigms, will you lose the visual characteristics of what users expect?

Keeping It Simple

When thinking about your mobile information architecture, you want to keep it as simple as possible.

Support your defined goals

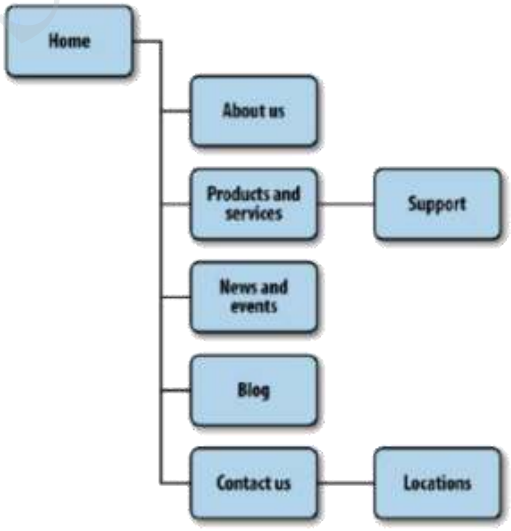
If something doesn‘t support the defined goals, lose it. Go back to your user goals and needs, and identify the tasks that map to them. Find those needs and fill them.

Clear, simple labels

Good trigger labels, the words we use to describe each link or action, are crucial in Mobile. Words like products or services aren‘t good trigger labels. Users have a much higher threshold of pain when clicking about on a desktop site or application, hunting and pecking for tasty morsels. Mobile performs short, to-the-point, get-it-quick, and get-out types of tasks. What is convenient on the desktop might be a deal breaker on mobile.

4.5.1 Site Maps

Relationship of content to other content and provide a map for how the user will travel through the informational space Limit opportunities for mistakes In Figure, you can see a poorly designed mobile information architecture that too closely mimics its desktop cousin; it was not designed with the mobile user in mind.



In the mobile context, tasks are short and users have limited time to perform them. And with mobile websites, we can‘t assume that the users have access to a reliable broadband

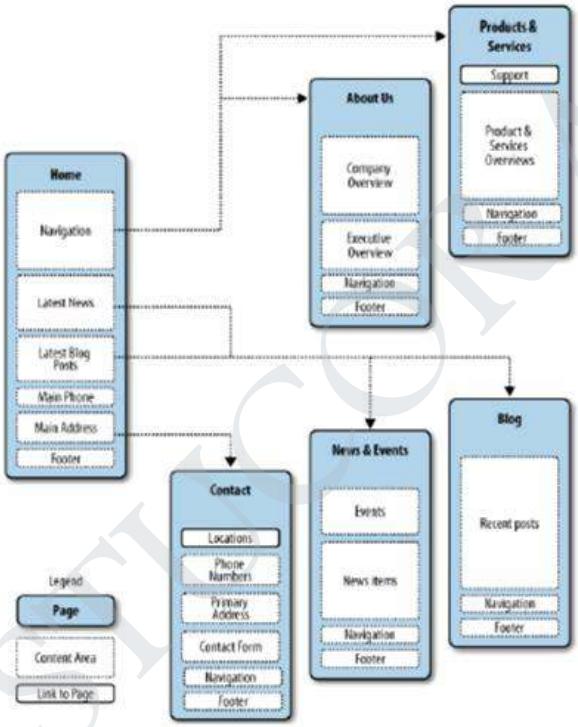
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

connection that allows them to quickly go back to the previous page. In addition, the users more often than not have to pay for each page view in data charges. So not only do they pay cash for viewing the wrong page by mistake, they pay to again download the page they started from: we can't assume that pages will be cached properly.

Confirm the path by teasing content

Information-heavy sites and applications often employ nested or drill-down architectures, forcing the user to select category after category to get to their target. To reduce risking the user's time and money, we want to make sure we present enough information for the user to wade through our information architecture successfully. On the Web, we take these risks very lightly, but with mobile, we must give our users a helping hand. We do this by teasing content within each category— that is, providing at least one content item per category.



4.5.2 Clickstreams

Clickstream is a term used for showing the behavior on websites, displaying the order in which users travel through a site's information architecture, usually based on data gathered from server logs. Clickstreams are usually historical, used to see the flaws in your information architecture, typically using heat-mapping or simple percentages to show where your users are going. I've always found them to be a useful tool for researching large websites. The maps the ideal path the user will take to perform common tasks. Being able to visually lay out the path users will take gives you a holistic or bird's-eye view of your mobile information architecture, just as a road map does. When you can see all the paths next to each other and take a step back, you start to see shortcuts and how you can get users to their goal

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

faster or easier, as shown in Figure. Just create user or process flows, such as the esoteric diagram shown in Figure, which is made up of boxes and diamonds that look more like circuit board diagrams than information architecture. If that is what your team prefers, then by all means, flow away. Personally, I like to present all of my information architecture deliverables from the perspective of the user, using the same metaphors she will use to make her way through my information architecture in this case, either a screen or page metaphor.

4.5.3 Wireframes

The next information architecture tool at our disposal is wireframes. Wireframes are a way to lay out information on the page, also referred to as information design. Site maps show how our content is organized in our informational space; wireframes show how the user will directly interact with it. Wireframes are like the peanut butter to the site map jelly in our information architecture sandwich. It‘s the stuff that sticks. Wireframes like the one in Figure serve to make our information space tangible and useful. Wireframes to be one of the most valuable information deliverables to communicate my vision for how a site or app will work, the challenge is that a diagram on a piece of paper doesn‘t go a long way toward describing how the interactions will work. Most common are what I call ―in-place interactions or areas where the user can interact with an element without leaving the page. This can be done with Ajax or a little show/hide JavaScript. These interactions can include copious amounts of annotation, describing each content area in as much length as you can fit in the margins of the page.

4.5.4 Prototyping

Prototypes might sound like a scary (or costly) step in the process. Some view them as redundant or too time-consuming, preferring to jump in and start coding things. But as with wireframes, I‘ve found that each product we‘ve built out some sort of prototype has saved both time and money.

Paper prototypes

The most basic level we have is paper prototyping: taking our printed-out wireframes or even drawings of our interface, like the one shown in Figure, and putting them in front of people.

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

Context prototype

STUCOR APP

The next step is creating a context prototype (Figure). Take a higher-end device that enables you to load full-screen images on it. Take your wireframes or sketches and load them onto the device, sized to fill the device screen. Leave the office. Go for a walk down to your nearest café. Or get on a bus or a train. Pay particular attention to what you are thinking and your physical behavior while you are using your interface and then write it down. If you are brave and don‘t have strict nondisclosure issues, ask the people around you to use it, too. I wouldn‘t bother with timing interactions or sessions, but try to keep an eye on a clock to determine how long the average session is.

4.6 Mobile 2.0

The Web as a platform for the mobile context, this means ―write once, deploy everywhere,‖ moving away from the costly native applications deployed over multiple frameworks and networks.

Harnessing collective intelligence this isn‘t something the mobile community has done much of, but projects like WURFL—an open source repository of device profiles provided by the community—is exactly what mobile needs more of.

Data is the next Intel inside Mobile takes this principle several steps further. It can include the data we seek, the data we create, and the data about or around our physical locations.

End of the software release cycle Long development and testing cycles heavily weigh on mobile projects, decreasing all hopes of profitability. Shorter agile cycles are needed to make mobile development work as a business. Releasing for one device, iterating, improving, and then releasing for another is a great way to ensure profitability in mobile.

Lightweight programming models Because mobile technology is practically built on enterprise Java, the notion of using lightweight models is often viewed with some skepticism. But decreasing the programming overhead required means more innovation occurs faster.

Software above the level of a single device This effectively means that software isn‘t just about computers anymore. We need to approach new software as though the user will demand it work in multiple contexts, from mobile phones to portable gaming consoles and e-book readers.

Rich user experiences a great and rich user experience helps people spend less time with the software and more time living their lives. Mobile design is about enabling users to live their lives better.

4.6.1 The Convergence of the Web and Mobile

It is obvious that in the minds of many, Mobile 2.0 is the Web. At this point, the mobile web has always been viewed as a second-class citizen within the mobile ecosystem, for many reasons, as discussed later. Mobile is already a medium, but the consensus is that by leveraging the power of the Web, integrating web services into the mobile medium is the

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

future of mobile development. When the iPhone exploded onto the scene, it increased the usage of the mobile web by its users to levels never seen before. The spur of new mobile web apps created just for the iPhone doubled the number of mobile websites available in under a year. If Web 2.0 taught us that the Web is the platform, then Mobile 2.0 tells us that mobile will be the primary context in which we leverage the Web in the future.

4.6.2 The Mobile Web Browser

If the future of mobile is the Web, then it only makes sense that the mobile web browser is the next killer app of mobile. Again, this is something we saw confirmed with WebKit in the iPhone and later in Android. However, of particular concern is how device fragmentation factors into mobile browsers. For example, how can we expect developers to support more than 30 different mobile browsers? A fellow panelist from the Mozilla Minimo project offered a potential solution in consolidation—that we will see only a few mobile browsers in the future; specifically, browsers built on Mozilla, Opera, Internet Explorer, and WebKit technologies. At the time, I thought that prediction was too focused on smart phones, but in the years since, the line between smart phone and feature phone seems to be going away, so this prediction is fairly accurate. But the single biggest challenge in mobile remains device fragmentation. The mobile browser enables us to penetrate the problem by not having our content locked so specifically to the device abilities, screen size, and form factor, but device fragmentation still causes old, outdated browsers to remain in the market long after they should be put out to pasture. What appears to be solving browser fragmentation is actually the iPhone. The Mobile Safari browser included with the iPhone provided such an excellent web experience on a mobile device that it drove use of the mobile web to huge levels, which means big profits for the operators. This also means that the mobile web is no longer a secondclass citizen. In the post-iPhone market, all new devices are judged by the quality of their mobile web browser. Operators know it and therefore are demanding better

4.6.3 JavaScript

If you are going to provide mobile web applications, you have to have a mobile web browser that supports Ajax, or, as it is technically known, XML Http Request. It makes a lot of those cool interactions in your web browser work via the capability to load content asynchronously into your browser view. But it isn't just Ajax; it is JavaScript, a web technology that has largely been ignored with most mobile web browsers. Ajax is great, but just being able to do a little show hide or change a style after you click or touch it goes a long way toward improving the user experience. This is probably where mobile web browsers fall behind desktop browsers the most. Because they both support XHTML and CSS relatively well, JavaScript has been a nogo in mobile for years. In order for mobile web apps to rival native applications, you have to support some JavaScript. Modern mobile browsers have made much progress over the last few years, but there is still plenty of work to be done. For

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

example, accessing the device capabilities like the phone book or file system with JavaScript doesn't work in a consistent way. These problems still need to be solved in order to truly reap the benefits of the Web.

4.6.4 Mobile User Experience

Traditionally, the user experience available in the mobile web has been like using a website from 1995: mostly text-based, difficult to use, and ugly as sin. This isn't to say that the user experience of mobile applications has been much better, but it used to be that if you wanted a good experience, you built a native app. Descriptions within the industry range from the honest "the mobile user experience is utterly horrid," to the sales pitch of "look at these cool things you can do," to the optimistic "the mobile user experience is the future!" These polar attitudes toward the mobile user experience are somewhat ironic, given that the mobile user experience was largely ignored for close to a decade. People in mobile treat the user experience like a chicken-and-egg scenario: bad input/output of the user experience prevents adoption, but designing a shiny user experience with bells and whistles will bring them in droves. Device APIs usually force you to use their models of user experience, meaning that you have to work in the constraints of the API. This is good for creating consistent experiences for that platform, but these experiences don't translate to others. For example, would you take an iPhone app design and put it on an Android device? The user experience for these devices is similar but still remains different. The beauty of the Web, literally, is that you can design whatever experience you want, for better or worse. You are in control of the presentation and can establish your own visual metaphors. The problem has been that traditionally complex (which often equates to good) user experiences haven't been possible on mobile devices. Modern mobile web browsers, as they come closer to their desktop counterparts, remove this distinction, giving us the same canvas on mobile devices that we have for the desktop. This means that creating mobile experiences just got a whole lot easier. It also means we can have a consistent user experience across multiple mediums.

4.7 MOBILE DESIGN

When building mobile experiences, it is impossible to create a great experience without three ingredients: context, information architecture, and visual design. The visual design of your experience is the direct representation of everything underneath; it is the first impression the user will have. A great design gives the user high expectations of your site or application; a poor design leads to lower expectations. Users' expectations translate to value and trust. In the mobile space, where content is often "free" (they still need to pay for data charges), users often have low expectations, due to the limitations of the canvas. Confusing site structures and slow download speeds reinforce those initial low expectations. In the downloadable application space, where application design can be much more robust, users must purchase applications almost sight unseen. For example, they may see just a small

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

screenshot of your application or game. But if the application doesn't meet the higher expectations of the design, your application downloads will drop like a stone. The design, that first impression, determines right from the start if the user will spend five seconds, five minutes, or five hours with your product. This leads us to the most significant challenge in mobile design: creativity. You can't always be as creative as you want to be. Many devices just can't support complex designs for every channel; for example, on many lower-end devices, the mobile web experience may just be a list of links. But every device has the capability to create a bestin- device experience; it just depends on how you take advantage of the application medium and context that you plan to use.

4.7.1 Interpreting Design

Days were spent creating print designs and advertisements, defining precisely what each design would be, down to the picas and points. The method of design is meant for creating a vision for how to communicate information or ideas in his head, and then duplicating that on the printed page. Precise designs might look better, but they can be brutal to implement. More flexible designs might not be much to look at, but they work for the most users, or the lowest common denominator. But more than that, our backgrounds and our training can actually get in the way of creating the best design for the medium. We like to apply the same rules to whatever the problem in front of us might be. In mobile design, you interpret what you know about good design and translate it to this new medium that is both technologically precise and at times incredibly unforgiving, and you provide the design with the flexibility to present information the way you envision on a number of different devices. In the end, the graphic designer and I scrapped the work, and he provided me his pica perfect designs as giant images, which I turned into a series of massive image maps.

4.7.2 The Mobile Design Tent-Pole

In Hollywood, executives like to use the term "tent-pole" to describe their movies and television shows. The term has dual meanings: one is business, and the other creative. The business goal of a tent-pole production is to support or prop up the losses from other productions. However, to create a tent-pole production, the creators involved must make an artistic work that they know will appeal to the largest possible audience, providing something for everyone. You probably know tent-pole movies as "blockbusters"; in television, they are known as "anchor" shows. Trying to reach the widest possible audience poses a problem. Hollywood is learning with great pains that with so many entertainment options and with today's audience being so hard to reach through traditional advertising channels, tent-pole productions are failing. As the number of social niches increases, it becomes difficult to satisfy the specific tastes of each social group. What one group finds hysterically funny, several other groups might find offensive. Today, tent-pole productions often come off as bland and half-hearted, failing to appease anyone.

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

One of the most interesting examples of how the tide turned in entertainment is with the animated films of Disney versus those of Pixar. For years, Disney produced tentpole family fare quite successfully. But as competition increased, notably from Pixar, Disney films would spend millions to create stale and dated films, losing audiences and revenue. Meanwhile, Pixar found that their movies could be successful by avoiding the traditional storytelling formats of animated film, which Disney essentially defined. Instead, Pixar based their stories around specific emotional themes and was able to connect with audiences of all ages, in multiple cultures and across multiple niches. In 2006, Disney acquired Pixar, making its top executives the new leaders of all Disney creative projects. Although Disney technically acquired Pixar, I've always looked at it as the other way around. Disney realized that it needed to be more Pixar and less Disney in order to grow and adapt to today's changing audiences and niches. This is something that Pixar was doing correctly. Back in the world of mobile design, the de facto strategy is to create tent-pole products. Like the old days of Disney, the strategy is to sink millions into creating tent-pole products, or products that support the largest number of devices that no one will ever use. They are creatively stale, they lack inspiration, and they simply exist with no meaningful purpose to the user. They make the same mistake Disney made, thinking that it could simply put something on the market that might not be the best quality, but because it carried the Disney name, people would buy it. To have a successful mobile design, you have to think like Pixar. Find that emotional connection, that fundamental need that serves many audiences, many cultures, and many niches and design experiences. Too often, designers simply echo the visual trends of the day, mimicking the inspiration of others—I'm certainly guilty of it. But with mobile design, once you find that essential thing, that chewy nougat we call "context" that lives at the center of your product, then you will find ample inspiration of your own to start creating designs that translate not only across multiple devices, but across multiple media. Sure, there are countless examples of poorly designed mobile products that are considered a success. You only need to look as far as the nearest mobile app store to find them. This is because of the sight unseen nature of mobile commerce. Traditionally, you couldn't demo—or in some cases even see—screenshots of a game or mobile application before you bought it. You simply had to purchase it and find out if it was any good. Apple's App Store quickly changed that. You can clearly see that the best-selling games and applications for the iPhone are the ones with the best designs.

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

THE ELEMENTS OF MOBILE DESIGN

Good design requires three abilities: the first is a natural gift for being able to see visually how something should look that produces a desired emotion with the target audience. The second is the ability to manifest that vision into something for others to see, use, or participate in. The third knows how to utilize the medium to achieve your design goals.

Six elements of mobile design that you need to consider, starting with the context and layering in visual elements or laying out content to achieve the design goal. Then, you need to understand how to use the specific tools to create mobile design, and finally, you need to understand the specific design considerations of the mobile medium.

4.8.1 Context

I won‘t belabor the point except to say that context is core to the mobile experience. As the designer, it is your job to make sure that the user can figure out how to address context using your app. Make sure you do your homework to answer the following questions:

• Who are the users? What do you know about them? What type of behavior can you assume or predict about the users?

• What is happening? What are the circumstances in which the users will best absorb the content you intend to present?

• When will they interact? Are they at home and have large amounts of time? Are they at work where they have short periods of time? Will they have idle periods of time while waiting for a train, for example?

• Where are the users? Are they in a public space or a private space? Are they inside or outside? Is it day or is it night?

• Why will they use your app? What value will they gain from your content or services in their present situation?

• How are they using their mobile device? Is it held in their hand or in their pocket?

•How are they holding it? Open or closed? Portrait or landscape?

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

The answers to these questions will greatly affect the course of your design. Treat these questions as a checklist to your design from start to finish.

STUCOR APP

4.8.2 Message

Message is the overall mental impression you create explicitly through visual design. I like to think of it as the holistic or at times instinctual reaction someone will have to your design. If you take a step back, and look at a design from a distance, what is your impression? Or conversely, look at a design for 30 seconds, and then put it down. What words would you use to describe the experience?

Branding shouldn‘t be confused with messaging. Branding is the impression your company name and logo gives—essentially, your reputation. Branding serves to reinforce the message with authority, not deliver it. In mobile, the opportunities for branding are limited, but the need for messaging is great. With such limited real estate, the users don‘t care about your brand, but they will care about the messaging, asking themselves questions like, ―What can this do for me? or ―Why is this important to me?

Yahoo!

Yahoo! sort of delivers a message. This app provides a clean interface, putting a focus on search and location, using color to separate it from the news content. But I‘m not exactly sure what it is saying. Words you might use to describe the message are crisp, clean, and sharp.

ESPN

The ESPN site clearly is missing a message. It is heavily text-based, trying to put a lot of content above the fold, but doesn‘t exactly deliver a message of any kind. If you took out the ESPN logo, you likely would have indifferent expectations of this site; it could be about anything, as the design doesn‘t help set expectations for the user in any way. Words you might use to describe the message: bold, cluttered, and content-heavy.

Disney

Disney creates a message with its design. It gives you a lot to look at—probably too much—but it clearly tries to say that the company is about characters for a younger audience. Words you might use to describe the message: bold, busy, and disorienting.

Wikipedia

The Wikipedia design clearly establishes a message. With a prominent search and text-heavy layout featuring an article, you know what you are getting with this design. Words you might use to describe the message: clean, minimal, and text-heavy.

Amazon

Amazon sort of creates a message. Although there are some wasted opportunities above the fold with the odd ad placement, you can see that it is mostly about products (which are improved even more if you scroll down). Words you might use to describe the message: minimal but messy, product-heavy, and disorienting.

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

4.8.3 Look and Feel

STUCOR APP

Look and feel is used to describe appearance, as in I want a clean look and feel or I want a usable look and feel. The problem is: as a mobile designer, what does it mean? And how is that different than messaging?

I think of look and feel in a literal sense, as something real and tactile that the users can look at, and then feel something they can touch or interact with. Look and feel is used to evoke action how the user will use an interface. Messaging is holistic, as the expectation the users will have about how you will address their context. It is easy to confuse the two, because feel can be interpreted to mean our emotional reaction to design and the role of messaging.

I often find myself explaining the look and feel with the word because, with a cause-and-effect rationale for design decisions, as in The user will press this button because...‖ or The user will go to this screen because… followed by a reason why a button or control is designed a certain way. Establishing a look and feel usually comes from wherever design inspiration comes from. However, your personal inspiration can be a hard thing to justify. Therefore we have design patterns, or documented solutions to design problems, sometimes referred to as style guides. On large mobile projects or in companies with multiple designers, a style guide or pattern library is crucial, maintaining consistency in the look and feel and reducing the need for each design decision to be justified.

4.8.4 Layout

Layout is an important design element, because it is how the user will visually process the page, but the structural and visual components of layout often get merged together, creating confusion and making your design more difficult to produce. The first time layout should rear its head is during information architecture. In fact, I prefer to make about 90 percent of my layout decisions during the information architecture period. I ask myself

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

questions like: where should the navigation go on the page or screen? What kind of navigation type should I use? Should I use tabs or a list? What about a sidebar for larger screens? All of these should be answered when defining the information architecture and before you begin to design. Design is just too subjective of an issue. If you are creating a design for anyone but yourself, chances are good that there will be multiple loosely-based-on-experience opinions that will be offered and debated.

Where the design opinions of the CEO or Chief Marketing Officer (CMO) might influence a design direction more than, say, the Creative Director or Design Director. By defining design elements like layout prior to actually applying the look and feel, you can separate the discussion. As a self-taught designer, I started out in this business making designs for my own projects. I could just put pen to paper and tweak it to my heart's content. If I wanted to radically change the layout, I could. When I started my mobile design career with my first mobile company more than a decade ago, I realized that this approach didn't work. The majority of comments that reviewers would make were about the layout. They focused on the headers, the navigation, the footer, or how content blocks are laid out, and so on. But their feedback got muddied with the look and feel, the colors, and other design elements.

Reviewers do make remarks like I like the navigation list, but can you make it look more raised? Most designers don't hear that; they hear the navigation isn't right, do it again. But, with this kind of feedback, there are two important pieces of information about different types of design. First, there is confirmation that the navigation and layout are correct. Second, there is a question about the look and feel. Because designers hear Do it again, they typically redo the layout, even though it was actually fine.

Creating mobile designs in an environment with multiple reviewers is all about getting the right feedback at the right time. Your job is to create a manifestation of a shared vision. Layout is one of the elements you can present early on and discuss independently. People confuse the quality and fidelity of your deliverables as design. By keeping it basic, you don't risk having reviewers confuse professionalism with design. The irony is that as I become more adept at defining layouts, I make them of increasingly lower fidelity. For example, when I show my mobile design layouts as wireframes during the information architecture phase, I intentionally present them on blueprint paper, using handwriting fonts for my annotations. It also helps to say that this is not a design; it is a layout, so please give me feedback on the layout.

4.8.5 Color

The fifth design element, color, is hard to talk about in a black-and-white book. Maybe it is fitting, because it wasn't that long ago that mobile screens were available only

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

inblack and white well, technically, it was black on a green screen). These days, we have nearly the entire spectrum of colors to choose from for mobile designs.

The most common obstacle you encounter when dealing with color is mobile screens, which come in a number of different color or bit depths, meaning the number of bits (binary digits) used to represent the color of a single pixel in a bitmapped image. When complex designs are displayed on different mobile devices, the limited color depth on one device can cause banding, or unwanted posterization in the image.

For an example of posterization, the technical term for when the gradation of tone is replaced with regions of fewer tones, see in Figure 8-10 how dramatically the color depth can affect the quality of a photo or gradient, producing banding in several parts in the image.

Color palettes

Defining color palettes can be useful for maintaining a consistent use of color in your mobile design. Color palettes typically consist of a predefined number of colors to use throughout the design. Selecting what colors to use varies from designer to designer, each having different techniques and strategies for deciding on the colors. I‘ve found that I use three basic ways to define a color palette:

Sequential

In this case, there are primary, secondary, and tertiary colors. Often the primary color is reserved as the ―brand color or the color that most closely resembles the brand‘s using a color wheel.

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

Color	Represents
White	Light, reverence, purity, truth, snow, peace, innocence, cleanliness, simplicity, security, humility, sterility, winter, coldness, surrender, fearfulness, lack of imagination, air, death (in Eastern cultures), life, marriage (in Western cultures), hope, bland
Black	Absence, modernity, power, sophistication, formality, elegance, wealth, mystery, style, evil, death (in Western cultures), fear, seriousness, conventionality, rebellion, anarchism, unity, sorrow, professionalism
Gray	Elegance, humility, respect, reverence, stability, subtlety, wisdom, old age, pessimism, boredom, decay, decrepitude, dullness, pollution, urban sprawl, strong emotions, balance, neutrality, mourning, formality
Yellow	Sunlight, joy, happiness, earth, optimism, intelligence, idealism, wealth (gold), summer, hope, air, liberalism, cowardice, illness (quarantine), fear, hazards, dishonesty, avarice, weakness, greed, decay or aging, femininity, gladness, sociability, friendship
Green	Intelligence, nature, spring, fertility, youth, environment, wealth, money (U.S.), good luck, vigor, generosity, go, grass, aggression, coldness, jealousy, disgrace (China), illness, greed, drug culture, corruption (North Africa), life eternal, air, earth (classical element), sincerity, renewal, natural abundance, growth
Blue	Seas, men, productiveness, interiors, skies, peace, unity, harmony, tranquility, calmness, trust, coolness, confidence, conservatism, water, ice, loyalty, dependability, cleanliness, technology, winter, depression, coldness, idealism, air, wisdom, royalty, nobility, Earth (planet), strength, steadfastness, light, friendliness, peace, truthfulness, love, liberalism (U.S. politics), and conservatism (UK, Canadian, and European politics)
Violet	Nobility, envy, sensuality, spirituality, creativity, wealth, royalty, ceremony, mystery, wisdom, enlightenment, arrogance, flamboyance, gaudiness, mourning, exaggeration, profanity, bisexuality, confusion, pride

Color	Represents
Red	Passion, strength, energy, fire, sex, love, romance, excitement, speed, heat, arrogance, ambition, leadership, masculinity, power, danger, gaudiness, blood, war, anger, revolution, radicalism, aggression, respect, martyrs, conservatism (U.S. politics), Liberalism (Canadian politics), wealth (China), and marriage (India)
Orange	Energy, enthusiasm, balance, happiness, heat, fire, flamboyance, playfulness, aggression, arrogance, gaudiness, overemotion, warning, danger, autumn, desire
Pink	Spring, gratitude, appreciation, admiration, sympathy, socialism, femininity, health, love, romance, marriage, joy, flirtatiousness, innocence and child-like qualities
Brown	Calm, boldness, depth, nature, richness, rustic things, stability, tradition, anachronism, boorishness, dirt, dullness, heaviness, poverty, roughness, earth

Adaptive

An adaptive palette is one in which you leverage the most common colors present in a supporting graphic or image. When creating a design that is meant to look native on the device, I use an adaptive palette to make sure that my colors are consistent with the target mobile platform.

Inspired

This is a design that is created from the great pieces of design you might see online, as shown in Figure below, or offline, in which a picture of the design might inspire you. This could be anything from an old poster in an alley, a business card, or some packaging. When I sit down with a new design, I thumb through some of materials to create an inspired palette. Like with the adaptive palette, you actually extract the colors from the source image, though you should never ever use the source material in a design.

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

4.9 MOBILE DESIGN TOOLS

STUCOR APP

Mobile design requires understanding the design elements and specific tools. The closest thing to a common design tool is Adobe Photoshop, though each framework has a different method of implementing the design into the application. Some frameworks provide a complete interface toolkit, allowing designers or developers to simply piece together the interface, while others leave it to the designer to define from scratch.

Mobile framework	Design tool	Interface toolkits
Java ME	Photoshop, NetBeans	JavaFX, Capuchin
BREW	Photoshop, Flash	BREW UI Toolkit, uiOne, Flash
Flash Lite	Flash	Flash Lite
iPhone	Photoshop, Interface Builder	iPhone SDK

Mobile framework	Design tool	Interface toolkits
Android	Photoshop, XML-based themes	Android SDK
Palm webOS	Photoshop, HTML, CSS, and JavaScript	Mojo SDK
Mobile web	Photoshop, HTML, CSS, and JavaScript	W3C Mobile Web Best Practices
Mobile widgets	Photoshop, HTML, CSS, and JavaScript	Opera Widget SDK, Nokia Web Runtime
Mobile web apps	Photoshop, HTML, CSS, and JavaScript	iUI, jQTouch, W3C Mobile Web App Best Practices

Designing for the Right Device

The truly skilled designer doesn‘t create just one product—she translates ideas into experiences. The spirit of your design should be able to be adapted to multiple devices. What device suits this design best? What market niche would appreciate it most? What devices are the most popular within that niche? The days of tent-poles are gone. Focus instead on getting your best possible experience to the market that will appreciate it most. It might not be the largest or best long-term market, but what you will learn from the best possible scenario will tell you volumes about your mobile product‘s potential for success or failure.

This knowledge will help you develop your porting and/or adaptation strategy, the most expensive and riskiest part of the mobile equation. iPhone users consume more mobile content and products than the average mobile user. This platform has an easy-to-learn framework and excellent documentation, for both web and native products, and an excellent display and performance means. Although iPhone users might not be the majority of your market, the ability to create the best possible design and get it in front of those users presents the least expensive product to produce with the lowest risk.

With a successful single device launch, you can start to adapt designs from the best possible experience to the second best possible experience, then the third, and fourth, and so

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

on. The best possible experience is how it should be, so it serves as a reference point for how we will adapt the experience to suit more devices.

STUCOR APP

Designing for Different Screen Sizes

Mobile devices come in all shapes and sizes. Choice is great for consumers, but bad for design. It can be incredibly difficult to create that best possible experience for a plethora of different screen sizes. For example, your typical feature phone might only be 140 pixels wide, whereas your higher-end smartphone might be three to four times wider.

Landscape or portrait? Fixed width or fluid? Do you use one column or two? These are common questions that come up when thinking about your design on multiple screen sizes. The bad news is that there is no simple answer. How you design each screen of content depends on the scope of devices you look to support, your content, and what type of experience you are looking to provide. The good news is that the vast majority of mobile device screens share the same vertical or portrait orientation, even though they vary greatly in dimension.

UNIT V WEB INTERFACE DESIGN

Designing Web Interfaces – Drag & Drop, Direct Selection, Contextual Tools, Overlays, Inlays and Virtual Pages, Process Flow. Case Studies.

User Interface Design Basics

User Interface (UI) Design focuses on anticipating what users might need to do and ensuring that the interface has elements that are easy to access, understand, and use to facilitate those actions. UI brings together concepts from interaction design, visual design, and information architecture.

Choosing Interface Elements

Users have become familiar with interface elements acting in a certain way, so try to be consistent and predictable in your choices and their layout. Doing so will help with task completion, efficiency, and satisfaction.

Interface elements include but are not limited to:

Input Controls: buttons, text fields, checkboxes, radio buttons, dropdown lists, list boxes, toggles, date field Navigational Components: breadcrumb, slider, search field, pagination, slider, tags, icons Informational Components: tooltips, icons, progress bar, notifications, message boxes, modal windows Containers: accordion

There are times when multiple elements might be appropriate for displaying content. When this happens, it‘s important to consider the trade-offs. For example, sometimes elements that can help save you space, put more of a burden on the user mentally by forcing them to guess what is within the dropdown or what the element might be.

STUCOR APP

Best Practices for Designing an Interface

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

Everything stems from knowing your users, including understanding their goals, skills, preferences, and tendencies. Once you know about your user, make sure to consider the following when designing your interface:

Keep the interface simple. The best interfaces are almost invisible to the user. They avoid unnecessary elements and are clear in the language they use on labels and in messaging. Create consistency and use common UI elements. By using common elements in your UI, users feel more comfortable and are able to get things done more quickly. It is also important to create patterns in language, layout and design throughout the site to help facilitate efficiency. Once a user learns how to do something, they should be able to transfer that skill to other parts of the site.

Be purposeful in page layout. Consider the spatial relationships between items on the page and structure the page based on importance. Careful placement of items can help draw attention to the most important pieces of information and can aid scanning and readability. Strategically use color and texture. You can direct attention toward or redirect attention away from items using color, light, contrast, and texture to your advantage. Use typography to create hierarchy and clarity. Carefully consider how you use typeface. Different sizes, fonts, and arrangement of the text to help increase scalability, legibility and readability. Make sure that the system communicates what's happening. Always inform your users of location, actions, changes in state, or errors. The use of various UI elements to communicate status and, if necessary, next steps can reduce frustration for your user. Think about the defaults. By carefully thinking about and anticipating the goals people bring to your site, you can create defaults that reduce the burden on the user. This becomes particularly important when it comes to form design where you might have an opportunity to have some fields pre-chosen or filled out.

5.1 DRAG AND DROP

5.1.1 Interesting Moments

At first blush, drag and drop seems simple. Just grab an object and drop it somewhere. But, as always, the devil is in the details. There are a number of individual states at which interaction is possible. We call these microstates interesting moments:

- How will users know what is draggable?
- What does it mean to drag and drop an object?
- Where can you drop an object, and where is it not valid to drop an object?
- What visual affordance will be used to indicate draggability?
- ill valid and invalid drop targets be signified?
- Do you drag the actual object?
- Or do you drag just a ghost of the object?
- Or is it a thumbnail representation that gets dragged?

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

• What visual feedback should be used during the drag and drop interaction?

STUCOR APP

What makes it challenging is that there are a lot of events during drag and drop that can be used as opportunities for feedback to the user. Additionally, there are a number of elements on the page that can participate as actors in this feedback loop.

The Events: There are at least 15 events available for cueing the user during a drag and drop interaction:

Page Load: Before any interaction occurs, you can pre-signify the availability of drag and drop. For example, you could display a tip on the page to indicate draggability.

Mouse Hover: The mouse pointer hovers over an object that is draggable.

Mouse Down: The user holds down the mouse button on the draggable object.

Drag Initiated: After the mouse drag starts (usually some threshold—3 pixels).

Drag Leaves Original Location: After the drag object is pulled from its location or object that contains it.

Drag Re-Enters Original Location: When the object re-enters the original location.

Drag Enters Valid Target: Dragging over a valid drop target.

Drag Exits Valid Target: Dragging back out of a valid drop target.

Drag Enters Specific Invalid Target: Dragging over an invalid drop target.

Drag Is Over No Specific Target: Dragging over neither a valid or invalid target. Do you treat all areas outside of valid targets as invalid?

Drag Hovers Over Valid Target : User pauses over the valid target without dropping the object. This is usually when a spring loaded drop target can open up. For example, drag over a folder and pause, the folder opens revealing a new area to drag into.

Drag Hovers Over Invalid Target: User pauses over an invalid target without dropping the object. Do you care? Will you want additional feedback as to why it is not a valid target?

Drop Accepted : Drop occurs over a valid target and drop has been accepted.

Drop Rejected: Drop occurs over an invalid target and drop has been rejected. Do you zoom back the dropped object?

Drop on Parent Container: Is the place where the object was dragged from special? Usually this is not the case, but it may carry special meaning in some contexts.

The Actors : During each event you can visually manipulate a number of actors. The page elements available include:

• Page (e.g., static messaging on the page)

• Cursor

• Tool Tip

• Drag Object (or some portion of the drag object, e.g., title area of a module)

• Drag Object‘s Parent Container

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

• Drop Target

STUCOR APP

Interesting Moments Grid

That's 15 events times 6 actors. That means there are 90 possible interesting moment search requiring a decision involving an almost unlimited number of style and timing choices. You can pull all this together into a simple interesting moment's grid for Drag and Drop.

	Page Generation	Mouse Hover	Drag Initiated	Drag over Valid	Drag over Invalid	Drag over Original	Drop Accepted	Drop Rejected	Drop on Original
Page Content	Hint	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Cursor	Normal	Move Cursor	Move Cursor	Move Cursor	Move Cursor	Move Cursor	Normal	Normal	Normal
Drag Object	Normal	Normal	Reduced Opacity & Tracking	Reduced Opacity & Tracking	Reduced Opacity & Tracking + Invalid Badge	Reduced Opacity & Tracking	2. Modules animates into the area just below insertion bar 3. Module comes to rest in new area 4. Modules slide up in a self-healing transition to close hole	Normal Opacity + Zoom Back to Original	Normal Opacity + Zoom Back to Original
Orig Location	Normal	Normal	Hole Opens	Hole Remains	Hole Remains	Hole Remains	Hole Remains	Hole refilled with drag object	Hole refilled with drag object
Drop Target	Normal	Normal	Normal	Insertion Bar	N/A	N/A	1. Insertion Bar Removed	N/A	N/A

5.1.3 Drag and Drop Module

One of the most useful purposes of drag and drop is to allow the user to directly place objects where she wants them on the page. A typical pattern is Drag and Drop Modules on a page. Netvibes provides a good example of this interaction pattern

Considerations

Netvibes allows its modules to be rearranged with drag and drop. A number of interesting moments decide the specific interaction style for this site. Figure shows the interesting moment's grid for Netvibes. While dragging, it is important to make it clear what will happen when the user drops the dragged object. There are two common approaches to targeting a drop:

• Placeholder target

• Insertion target

Placeholder target

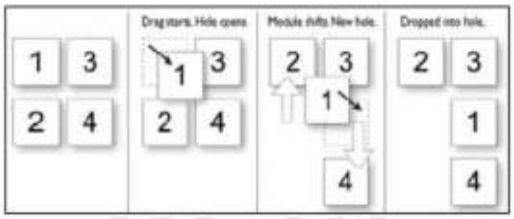
Netvibes uses a placeholder (hole with dashed outline) as the drop target. The idea (illustrated in Figure) is to always position a hole in the spot where the drop would occur. When module 1 starts dragging, it gets ―rippedǁ out of the spot. In its place is the

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

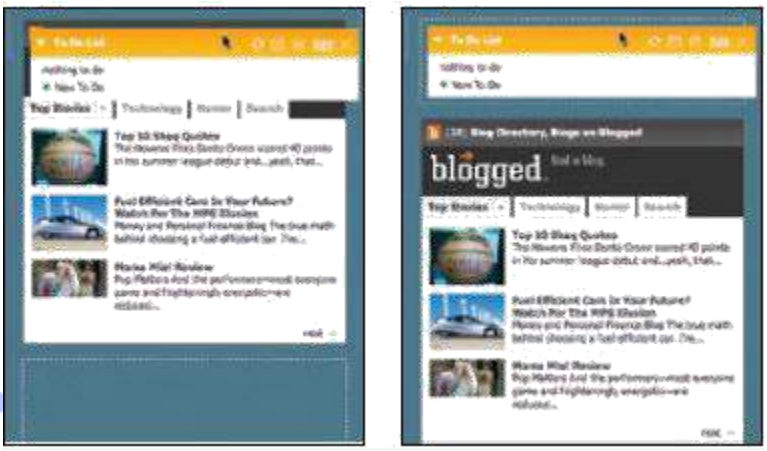
placeholder target (dashed outline). As 1 gets dragged to the spot between 3 and 4, the placeholder target jumps to fill in this spot as 4 moves out of the way.



The hole serves as a placeholder and always marks the spot that the dragged module will land when dropped. It also previews what the page will look like (in relation to the other modules) if the drop occurs there. For module drag and drop, the other modules only slide up or down within a vertical column to make room for the dragged module.

One complaint with using placeholder targets is that the page content jumps around a lot during the drag. This makes the interaction noisier and can make it harder to understand what is actually happening. This issue is compounded when modules look similar. The user starts dragging the modules around and quickly gets confused about what just got moved. One way to resolve this is to provide a quick animated transition as the modules move. It is important, however, that any animated transitions not get in the way of the normal interaction.

Boundary-based placement. Since most sites that use placeholder targeting drag the module in its original size, targeting is determined by the boundaries of the dragged object and the boundaries of the dragged-over object. The mouse position is usually ignored because modules are only draggable in the title (a small region). Both Netvibes and iGoogle take the boundary-based approach. But, interestingly, they calculate the position of their placeholders differently. In Netvibes, the placeholder changes position only after the dragged module‘s title bar has moved beyond the dragged-over module‘s title bar. In practice, this means if you are moving a small module to be positioned above a large module, you have to move it to the very top of the large module. In Figure you have to drag the small ―To Do List module all the way to the top of the ―Blog Directory module before the placeholder changes position.



STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

Insertion target

Placeholder positioning is a common approach, but it is not the only way to indicate drop targeting. An alternate approach is to keep the page as stable as possible and only move around an insertion target (usually an insertion bar). A previous version of My Yahoo! Used the insertion bar approach as the dragged module was moved around.

Drag distance

Dragging the thumbnail around does have other issues. Since the object being dragged is small, it does not intersect a large area. It requires moving the small thumbnail directly to the place it will be dropped. With iGoogle, the complete module is dragged. Since the module will always be larger than the thumbnail, it intersects a drop target with much less movement. The result is a shorter drag distance to accomplish a move

Drag rendering

How should the dragged object be represented? Should it be rendered with a slight transparency (ghost)? Or should it be shown fully opaque? Should a thumbnail representation be used instead?

5.1.4 Drag and Drop List

The Drag and Drop List pattern defines interactions for rearranging items in a list. 37 Signal‘s Backpackit allows to-do items to be rearranged with Drag and Drop List **Considerations**

Backpackit takes a real-time approach to dragging items. Since the list is constrained, this is a natural approach to moving objects around in a list. You immediately see the result of the drag.

Placeholder target

This is essentially the same placeholder target approach we discussed earlier for dragging and dropping modules. The difference is that when moving an item in a list, we are constrained to a single dimension. Less feedback is needed. Instead of a ―ripped-out‖ area (represented earlier with a dotted rectangle), a simple hole can be exposed where the object will be placed when dropped.

Insertion target

Drag and Drop Modules, placeholder targeting is not the only game in town. You can also use an insertion bar within a list to indicate where a dropped item will land. Netflix uses an insertion target when movies are dragged to a new location in a user‘s movie queue

The upside to this approach is that the list doesn‘t have to shuffle around during drag. The resulting experience is smoother than the Backpack it approach. The downside is that it is not as obvious where the movie is being positioned. The insertion bar appears under the ghosted item. The addition of the brackets on the left and right of the insertion bar is an attempt to make the targeting clearer.

Non–drag and drop alternative

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

Besides drag and drop, the Netflix queue actually supports two other ways to move objects around:

Edit the row number and then press the ―Update DVD Queue button.

Click the ―Move to Top‖ icon to pop a movie to the top.

Modifying the row number is straightforward. It‘s a way to rearrange items without drag and drop. The ―Move to Top button is a little more direct and fairly straightforward (if the user really understands that this icon means ―move to top). Drag and drop is the least discoverable of the three, but it is the most direct, visual way to rearrange the list. Since rearranging the queue is central to the Netflix customer‘s satisfaction, it is appropriate to allow multiple ways to do so.

Hinting at drag and drop

When the user clicks the ―Move to Top button, Netflix animates the movie as it moves up. But first, the movie is jerked downward slightly and then spring-loaded to the top The combination of the downward jerk and then the quick animation to the top gives a subtle clue that the object is draggable. This is also an interesting moment to advertise drag and drop. After the move to top completes, a simple tip could appear to invite users to drag and drop. The tip should probably be shown only once, or there should be a way to turn it off. Providing an invitation within a familiar idiom is a good way to lead users to the new idiom.

Drag and Drop Object

Another common use for drag and drop is to change relationships between objects. This is appropriate when the relationships can be represented visually. Drag and drop as a means of visually manipulating relationships is a powerful tool.Cogmap is a wiki for organizational charts. Drag and Drop Object is used to rearrange.

Considerations

When object relationships can be clearly represented visually, drag and drop is a natural choice to make these type of changes. Cogmap uses the target insertion approach. This allows the dragging to be nondistracting, since the chart does not have to be disturbed during targeting.

5.1.6 Drag and Drop Action

Drag and drop is also useful for invoking an action or actions on a dropped object. The Drag and Drop Action is a common pattern. Its most familiar example is dropping an item in the trash to perform the delete action.

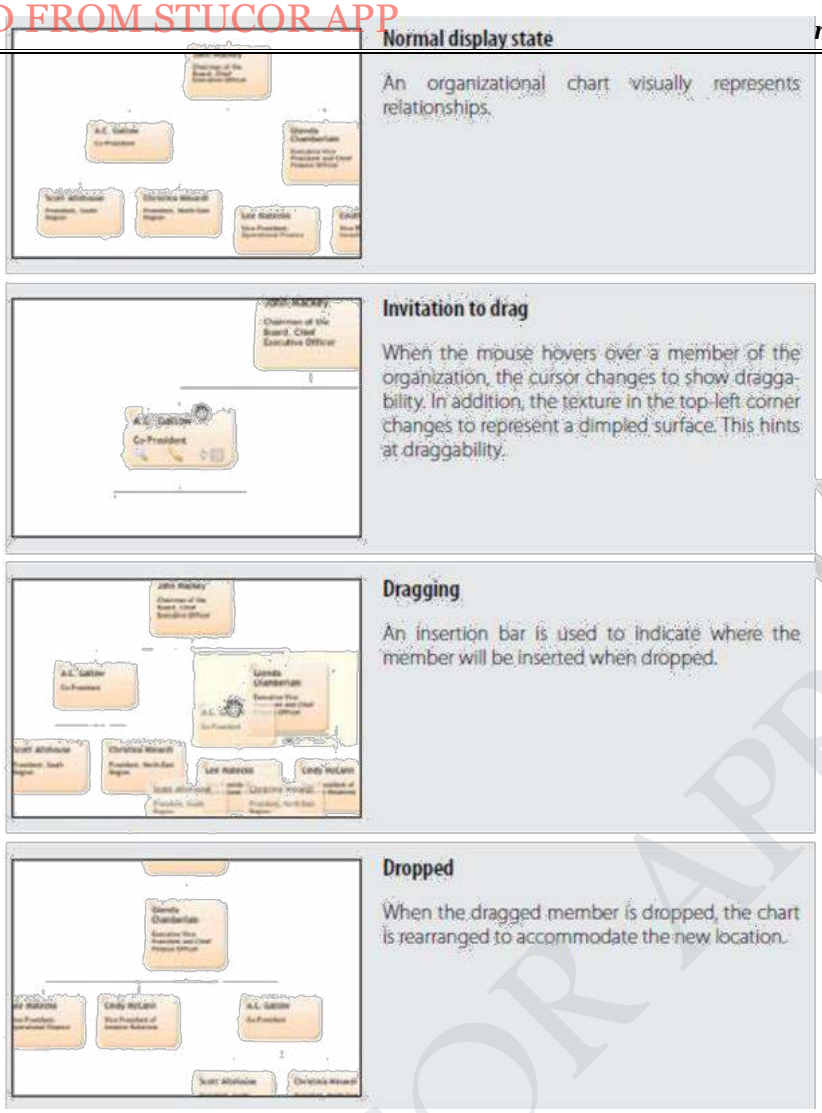
Normally uploading files to a web application includes pressing the upload button and browsing for a photo. This process is repeated for each photo.

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP



Considerations

This is not a trivial implementation. But it does clearly illustrate the benefit of drag and drop for operating on a set of files. The traditional model requires each photo to be selected individually for upload. Drag and drop frees you to use whatever browsing method is available on your system and then drop those photos for upload.

Anti-pattern: Artificial Visual Construct

Unfortunately, drag and drop can sometimes drive the design of an interface instead of being an extension of a natural interface. These interactions are almost always doomed, as they are the tail wagging the proverbial dog. Rating movies, books, and music is a common feature found on many sites. But what happens if you try to use drag and drop to rate movies?

5.1.7 Drag and Drop Collection

A variation on dragging objects is collecting objects for purchase, bookmarking, or saving into a temporary area. This type of interaction is called Drag and Drop Collection. Drag and drop is a nice way to grab items of interest and save them to a list. Drag and drop is a natural way collect items for purchase. It mimics the shopping experience in the real world. Grab an item. Drop it in your basket. This is fast and convenient once you know about the feature. However, as a general rule, you should never rely solely.

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

The Challenges of Drag and Drop

STUCOR APP

As you can see from the discussion in this chapter, Drag and Drop is complex. There are four broad areas where Drag and Drop may be employed: Drag and Drop Module, Drag and Drop List, Drag and Drop Object, and Drag and Drop Action. And in each area, there are a large number of interesting moments that may be handled in numerous ways. Being consistent in visual and interaction styles across all of these moments for all of these types of interactions is a challenge in itself. And keeping the user informed throughout the process with just the right amount of hints requires design finesse.

5.2 DIRECT SELECTION

- **Toggle Selection:** Checkbox or control-based selection.
- **Collected Selection:** Selection that spans multiple pages.
- **Object Selection:** Direct object selection.
- **Hybrid Selection:** Combination of Toggle Selection and Object Selection

Toggle Selection

The most common form of selection on the Web is Toggle Selection. Checkboxes and toggle buttons are the familiar interface for selecting elements on most web pages.

The way to select an individual mail message is through the row's checkbox. Clicking on the row itself does not select the message. We call this pattern of selection Toggle Selection since toggle-style controls are typically used for selecting items. Once items have been check-selected, actions can be performed on them. Usually these actions are performed on the selection by clicking on a separate button (e.g., the Delete button). Gmail is a good example of actions in concert with Toggle Selection.

Toggle Selection with checkboxes has some nice attributes:

- Clear targeting, with no ambiguity about how to select the item or deselect it.
- Straightforward discontinuous selection, and no need to know about Shift or Control key ways to extend a selection. Just click the checkboxes in any order, either in a continuous or discontinuous manner.
- Clear indication of what has been selected.

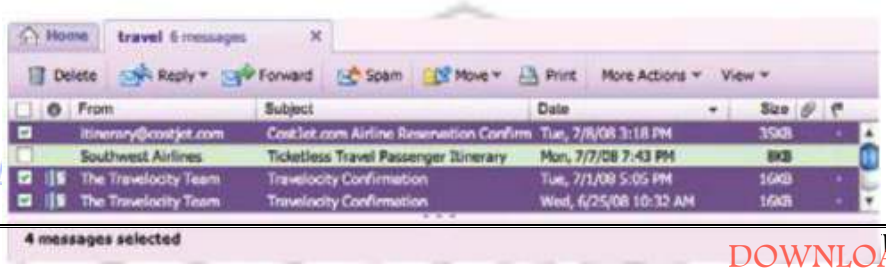
STUCOR APP
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

Scrolling versus paging

The previous examples were with paged lists. But what about a scrolled list? Yahoo! Mail uses a scrolled list to show all of its mail messages (Figure). While not all messages are visible at a time, the user knows that scrolling through the list retains the currently selected items. Since the user understands that all the messages not visible are still on the same continuous pane, there is no confusion about what an action will operate on—it will affect all selected items in the list. Sometimes the need for clarity of selection will drive the choice between scrolling and paging.



DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

Making selection explicit

With Yahoo! Bookmarks you can manage your bookmarks by selecting bookmarked pages and then acting on them. The selection model is visually explicit.

5.2.2 Collected Selection

Toggle Selection is great for showing a list of items on a single page. But what happens if you want to collect selected items across multiple pages? Collected Selection is a pattern for keeping track of selection as it spans multiple pages. In Gmail, you can select items as you move from page to page. The selections are remembered for each page. If you select two items on page one, then move to page two and select three items, there are only three items selected. This is because actions only operate on a single page. This makes sense, as users do not normally expect selected items to be remembered across different pages.

Considerations

Gmail does provide a way to select all items across different pages. When selecting all items on a individual page (with the ―All‖ link), a prompt appears inviting the user to ―Select all 2785 conversations in Spam‖. Clicking that will select all items across all pages (Figure). The ―Delete Forever‖ action will operate on all 2785 conversations, not just the 25 selected on the page.

Keeping the selection visible

The real challenge for multi-page selection is finding a way to show selections gathered across multiple pages. You need a way to collect and show the selection as it is being created. Here is one way that Collected Selection comes into play. LinkedIn uses Collected Selection to add potential contacts to an invite list.

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

The list of potential invitees is shown in a paginated list on the lefthand side. Clicking the checkbox adds them to the invite list. The invite list becomes the place where selected contacts across multiple pages are remembered.

Collected Selection and actions

In the menu system it was hard to discern whether the user meant to operate on the selection (photos on the page could be selected through an Object Selection model) or on the collected items in the tray. To resolve this ambiguity, the drop-down menus contained two identical sets of commands. The first group of commands in the menu operated on the collected items in the tray. The second set of commands operated on the selected objects. Needless to say, this was confusing since it required the user to be fully aware of these two selection models when initiating a command.

One way to remove this ambiguity would have been to have a single set of commands that operated on either the tray or the photos—depending on which had the focus. This would require a way to select the tray and a way to deselect it (by clicking outside the tray). A possible approach would be to slightly dim the photo gallery when the tray is selected (causing it to clearly have the focus), and do the opposite when the tray is not the focus.

Object Selection

Object Selection, is when selection is made directly on objects within the interface. Sometimes using a checkbox does not fit in with the style of interaction desired. Laszlo's WebTop mail allows the user to select messages by clicking anywhere in the row

Considerations

Desktop applications tend to use Object Selection. It is also natural that web-based mail applications that mimic desktop interactions employ this same style of selection. Instead of showing a control (like a checkbox), the object itself can be selected and acted on directly. Object Selection can be extended by holding down the Shift key while clicking on a different item. The Command key (Macintosh) or Control key (Windows) can be used to individually add items in a discontinuous manner. The downside to this approach is that it is not obvious

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

to use the modifier keys for extending the selection. Toggle Selection's use of toggle buttons makes the selection extension model completely obvious.

Desktop-style selection

For now Object Selection is not as common on the Web. Given that most sites have been content-oriented, there have been few objects to select. Also, with the Web's simple event model, Object Selection was not easy to implement. In typical web pages, keyboard events have rarely made sense since they are also shared with the browser.

Object Selection interactions include ways to use the mouse to drag-select objects. Yahoo! Photos introduced this same type of object selection to its photo gallery (Figure below). Individually clicking on a photo selects it. Using the Shift key and clicking also extends the selection. In addition, using the Control key and clicking discontinuously selects photos. And like most desktop applications, you can drag a selection box around a group of items to add them to the selected set (in this case, photos).

Hybrid Selection

Mixing Toggle Selection and Object Selection in the same interface can lead to a confusing interface. Referring back to Yahoo! Bookmarks, you'll see an odd situation arise during drag and drop

Figure: In Yahoo! Bookmarks, one item is selected, but two items can be dragged by dragging on the unselected item.

Considerations

There are a few important issues to consider when using Hybrid Selection.

Confusing two models

One bookmark element is selected (notice the checkbox Toggle Selection). The second bookmark element (—Dr. Dobb's) is unselected (the checkbox is clear). In the right panel of clicking and dragging on the unselected bookmark element initiates a drag. The drag includes both the selected element and the unselected element. Since only one is shown as selected, this creates a confusing situation.

This occurs because three things are happening in the same space:

- Toggle Selection is used for selecting bookmarks for editing, deleting, etc.
- Object Selection is used for initiating a drag drop.
- Mouse click is used to open the bookmark on a separate page.

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

The problem is that more than one interaction idiom is applied to the same place on the same page. In this case, if you happen to try to drag, but instead click, you will be taken to a new page. And if you drag an unselected item, you now have two items selected for drag but only one shown as selected for other operations. This is definitely confusing. Simply selecting the item (automatically checking the box) when the drag starts would keep the selection model consistent in the interface. However, it might lead the user to expect a single click to also do the same (which it cannot since it opens the bookmark). So, mixing the two selection models together can be problematic. However, there is a way to integrate the Toggle Selection and Object Selection and have them coexist peacefully as well as create an improved user experience.

CONTEXTUAL TOOLS

Interaction in Context

Desktop applications separate functionality from data. Menu bars, toolbars, and palettes form islands of application functionality. Either the user chooses a tool to use on the data or makes a selection and then applies the tool. They were completely content-oriented. Rich tool sets were not needed for simply viewing and linking to content pages. Even in e-commerce sites like Amazon or eBay, the most functionality needed was the hyperlink and Submit button.

Touch-based interfaces were the stuff of research labs and, more recently, interesting You- Tube videos. But now they‘re as close as our phones. Most notably, the Apple iPhone brought touch to the masses. Gesture-based interfaces seemed even further out. Yet these became reality with the Nintendo Wii.

5.3.2 Fitts’s Law

Fitts‘s Law is an ergonomic principle that ties the size of a target and its contextual proximity to ease of use. Bruce Tognazzini restates it simply as:

The time to acquire a target is a function of the distance to and size of the target. In other words, if a tool is close at hand and large enough to target, then we can improve the user‘s interaction. Putting tools in context makes for lightweight interaction.

5.3.3 Contextual Tools

Contextual Tools are the Web‘s version of the desktop‘s right-click menus. Instead of having to right-click to reveal a menu, we can reveal tools in context with the content. We can do this in a number of ways:

- **Always-Visible Tools:** Place Contextual Tools directly in the content.
- **Hover-Reveal Tools:** Show Contextual Tools on mouse hover.
- **Toggle-Reveal Tools:** A master switch to toggle on/off Contextual Tools for the page.
- **Multi-Level Tools:** Progressively reveal actions based on user interaction.

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

- **Secondary Menus:** Show a secondary menu (usually by right-clicking on an object).

STUCOR APP

5.3.4 Always-Visible Tools

The simplest version of Contextual Tools is to use Always-Visible Tools. Digg is an example of making Contextual Tools always visible

The —digg it button and Digg scorecard provide Always-Visible Tools next to each story. Clear call to action Why not hide the tools and only reveal them when the mouse is over the story? Since digging stories is central to the business of Digg, always showing the tool provides a clear call to action. There are other actions associated with news stories (comments, share, bury, etc.) but they are represented less prominently. In the case of Digg, the designers chose to show these at all times.

Relative importance

The —digg it action is represented as a button and placed prominently in the context of the story. The —bury it action is represented as a hyperlink along with other —minor actions just below the story. The contrast of a button and a hyperlink as well as its placement gives a strong indication as to the relative importance of each action.

Discoverability

Discoverability is a primary reason to choose Always-Visible Tools. On the flip side, it can lead to more visual clutter. In the case of Digg and Netflix, there is a good deal of visual space given to each item (story, movie). But what happens when the items you want to act on are in a list or table?

Generally Contextual Tools in a list work well when the number of actions is kept to a minimum. Gmail provides a single Always-Visible Tool in its list of messages—the star rating—for flagging emails

Simply clicking the star flags the message as important. The unstarred state is rendered in a visually light manner, which minimizes the visual noise in the list.

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

5.3.5 Hover-Reveal Tools

STUCOR APP

One way to do this is to reveal the tools when the user pauses the mouse over an object. The Hover-Reveal Tools pattern is most clearly illustrated by 37 Signal's Backpackit (Figure below). To-do items may be deleted or edited directly in the interface. The tools to accomplish this are revealed on mouse hover.

Discoverability

A serious design consideration for Hover-Reveal Tools is just how discoverable the additional functionality will be. While the Contextual Tools are revealed on hover, the checkbox is always visible for each to-do item. To check off an item, users have to move the mouse over it. When they do, they will discover the additional functionality.

Contextual Tools in an overlay

There are several actions available for a focused object. Instead of placing tools beside the object being acted on, the revealed tools can be placed in an overlay. However, there can be issues with showing contextual tools in an overlay:

1. Providing an overlay feels heavier. An overlay creates a slight contextual switch for the user's attention.

2. The overlay will usually cover other information—information that often provides context for the tools being offered.

3. Most implementations shift the content slightly between the normal view and the overlay view, causing the users to take a moment to adjust to the change.

4. The overlay may get in the way of navigation. Because an overlay hides at least part of the next item, it becomes harder to move the mouse through the content without stepping into a —landmine.

5.3.6 Toggle-Reveal Tools

A variation on the two previous approaches is to not show any Contextual Tools until a special mode is set on the page. A good example of Toggle-Reveal Tools is in Basecamp's category editing

STUCOR

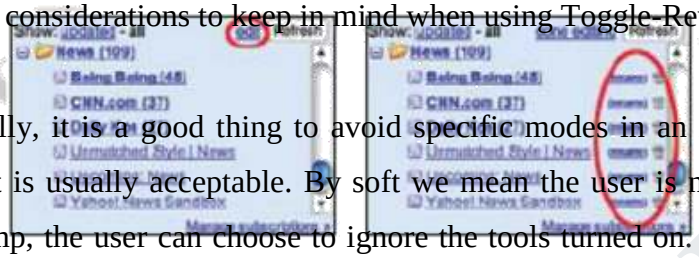
DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

Considerations

Here are a few considerations to keep in mind when using Toggle-Reveal Tools.



Soft mode

Generally, it is a good thing to avoid specific modes in an interface. However, if a mode is soft it is usually acceptable. By soft we mean the user is not trapped in the mode. With Base camp, the user can choose to ignore the tools turned on. It just adds visual noise and does not restrict the user from doing other actions. This is a nice way to keep the interaction lightweight.

When would you use this technique? When the actions are not the main thing and you want to reduce visual noise. This fits the category example perfectly. Items are renamed or deleted occasionally. It is common, however, to want to click through and see the contents of a category (the category is always hyperlinked). Hence, make it readable and easily navigable in the normal case but still give the user a way to manage the items in context.

Google Reader could potentially be improved in this manner. In the current interface, clicking ―Manage Subscriptions takes the user to another page to edit subscriptions. One possible change is the addition of an edit button that toggles in a set of context tools for each subscription. This would allow the user to rename and unsubscribe without leaving the context of the reading pane.

5.3.7 Multi-Level Tools

Contextual Tools can be revealed progressively with Multi-Level Tools. Songza* provides a set of tools that get revealed after a user clicks on a song. Additional tools are revealed when hovering over the newly visible tools

Radial menus

Radial menus* such as in Songza have been shown to have some advantages over more traditional menus. First, experienced users can rely on muscle memory rather than having to look directly at the menu items. Second, the proximity and targeting size make the

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

menu easy to navigate since the revealed menu items are all equally close at hand (recall Fitts's Law).

The one potential downside to this approach is that rating a song requires several steps: an initial click on the song, moving the mouse over the rate menu item, then clicking either the thumbs up or thumbs down option. If rating songs was an important activity, the extra effort might prevent some users from doing so. An alternate approach would be to replace —rate directly with the thumbs up and the thumbs down options.

Activation

Another interesting decision Songza made was to not activate the radial menu on hover. Instead, the user must click on a song to reveal the menu. Activating on click makes the user intent more explicit. Making activation more explicit avoids the issues described earlier in the Hover and Cover anti-pattern. The user has chosen to interact with the song. Conversely, with a mouse hover, it's never quite clear if the user meant to activate the menu or just happened to pause over a song title.

Default action

Playing a song requires moving to the top leaf. One possible solution would be to place the play option in the middle of the menu (at the stem) instead of in one of the leaves. Clicking once would activate the menu. Clicking a second time (without moving the mouse) would start playing the song. This interaction is very similar to one commonly used in desktop application: allowing a double-click to activate the first item (default action) in a right-click menu.

5.4 OVERLAYS

Overlays are really just lightweight pop ups. We use the term lightweight to make a clear distinction between it and the normal idea of a browser pop up. Browser pop ups are created as a new browser window Lightweight overlays are shown within the browser page as an overlay. Older style browser pop ups are undesirable because: Browser pop ups display a new browser window. As a result • these windows often take time and a sizeable chunk of system resources to create. Browser pop ups often display browser interface controls (e.g., a URL bar). Due to security concerns, in Internet Explorer 7 the URL bar is a permanent fixture on any browser pop-up window.

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

STUCOR APP

By using either Flash or Ajax-style techniques (Dynamic HTML), a web application can present a pop up in a lightweight overlay within the page itself. This has distinct advantages: Lightweight overlays are just a lightweight in-page object. They are inexpensive to create and fast to display. The interface for lightweight overlays is controlled by the •web application and not the browser. There is complete control over the visual style for the overlay. This allows the overlay to be more visually integrated into the application's interface

We will look at three specific types of overlays: Dialog Overlays, Detail Overlays, and Input Overlays.

5.4.1 Dialog Overlay

Dialog Overlays replace the old style browser pop ups. Netflix provides a clear example of a very simple Dialog Overlay. In the "previously viewed movies for sale" section, a user can click on a "Buy" button to purchase a DVD. Since the customer purchasing the DVD is a member of Netflix, all the pertinent shipping and purchasing information is already on record. The complete checkout experience can be provided in a single overlay

5.4.2 Detail Overlay

The second type of overlay is somewhat new to web applications. The Detail Overlay allows an overlay to present additional information when the user clicks or hovers over a link or section of content. Toolkits now make it easier to create overlays across different browsers and to request additional information from the server without refreshing the page. Taking

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

another example from Netflix, information about a specific movie is displayed as the user hovers over the movie's box shot.

STUCOR APP

5.4.3 Input Overlay

Input Overlay is a lightweight overlay that brings additional input information for each field tabbed into. American Express uses this technique in its registration for premium card such as its gold card.

5.5 INLAYS

Information, or dialog with the user needs to be an overlay. Another approach is to inlay the information directly within the page itself. To distinguish from the pop-up overlay, we call these in-page panels Inlays.

5.5.1 Dialog Inlay

A simple technique is to expand a part of the page, revealing a dialog area within the page. The BBC recently began experimenting with using a Dialog Inlay as a way to reveal customization controls for its home page.

5.5.2 List Inlay

Lists are a great place to use Inlays. Instead of requiring the user to navigate to a new page for an item's detail or popping up the information in an Overlay, the information can be shown with a List Inlay in context. The List Inlay works as an effective way to hide detail until needed—while at the same time preserving space on the page for high-level overview information. Google Reader provides an expanded view and a list view for unread blog articles. In the list view, an individual article can be expanded in place as a List Inlay

5.5.3 Detail Inlay

A common idiom is to provide additional detail about items shown on a page. We saw this with the example of the Netflix movie detail pop up . Hovering over a movie revealed a Detail Overlay calling out the back-of-the-box information. Details can be shown inline as well. Roost allows house photos to be viewed in-context for a real estate listing with a Detail Inlay

Inlay Versus Overlay?

Use an overlay when there may be more than one place a • dialog can be activated from (the exception may be showing details for items in a list).

• Use an overlay to interrupt the process.

• Use an overlay if there is a multi-step process.

• Use an inlay when you are trying to avoid covering information on the page needed in the dialog.

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

• Use an inlay for contextual information or details about one of many items (as in a list): a typical example is expanding list items to show detail.

5.6 Virtual pages

Overlays allow you to bring additional interactions or content in a layer above the current page. Inlays allow you to do this within the page itself. However, another powerful approach to keeping users engaged on the current page is to create a virtual page. That is to say, we create the illusion of a larger virtual page.

Patterns that support virtual pages include:

• Virtual Scrolling

• Inline Paging

• Scrolled Paging

• Panning

• Zoomable User Interface

5.6.1 Virtual Scrolling

The traditional Web is defined by the "page." In practically every implementation of websites (for about the first 10 years of the Web's existence) pagination was the key way to get to additional content. Of course, websites could preload data and allow the user to scroll through it. However, this process led to long delays in loading the page. So most sites kept it simple: go fetch 10 items and display them as a page and let the user request the next page of content. Each fetch resulted in a page refresh. The classic example of this is Google Search. Each page shows 10 results. Moving through the content uses the now-famous Google pagination control

Virtual Scrolling demonstrate three different ways to manage the virtual space:

• Yahoo! Mail creates the illusion that all data has been loaded up-front by having the scrollbar reflect the total virtual space.

• Microsoft Live Search creates the virtual illusion as the user moves down through the search results.

- And PicLens does the same with the caveat that it • shows a virtual window in the larger virtual space (by only providing a scroller control for where the user is and some before and after context).

5.6.2 Inline Paging

What if instead of scrolling through content we just wanted to make pagination feel less like a page switch? By only switching the content in and leaving the rest of the page stable, we can create an Inline Paging experience. This is what Amazon's Endless.com site does with its search results.

5.6.3 Scrolled Paging

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

Besides Virtual Scrolling and Virtual Paging, there is another option. You can combine both scrolling and paging into Scrolled Paging. Paging is performed as normal. But instead the content is "scrolled" into view.

The Carousel pattern takes this approach. A Carousel provides a way to page-in more data by scrolling it into view. On one hand it is a variation on the Virtual Scrolling pattern. In other ways it is like Virtual Paging since most carousels have paging controls. The additional effect is to animate the scrolled content into view. Yahoo! Underground uses a Carousel to provide a way to page/scroll through articles.

5.6.4 Virtual Panning

One way to create a virtual canvas is to allow users the freedom to roam in two-dimensional space. A great place for Virtual Panning is on a map. Google Maps allows you to pan in any direction by clicking the mouse down and dragging the map around

5.6.5 Zoomable user interface

A Zoomable User Interface (ZUI) is another way to create a virtual canvas. Unlike panning or flicking through a flat, two-dimensional space, a ZUI allows the user to also zoom in to elements on the page. This freedom of motion in both 2D and 3D supports the concept of an infinite interface.Practically speaking, ZUIs have rarely been available in everyday software applications, much less on the Web. But with more advanced features added to Flash and the advent of Silverlight, this type of interface is starting to emerge and may be commonplace in the not-too-distant future.

5.6.6 Paging Vs Scrolling

Leading web designers and companies have taken different approaches to solving the same problems. Yahoo! Mail chose Virtual Scrolling. Gmail chose Inline Paging. How do you choose between paging and scrolling? While there are no hard and fast rules, here are some things to consider when making the decision:

• When the data feels "more owned" by the user—in other words, the data is not transient but something users want to interact with in various ways. If they want to sort it, filter it, and so on, consider Virtual Scrolling (as in Yahoo! Mail).

• When the data is more transient (as in search results) and will get less and less relevant the further users go in the data, Inline Paging works well (as with the iPhone).

• For transient data, if you don't care about jumping around in the data to specific sections, Consider using Virtual Scrolling (as in Live Image Search).

• If you are concerned about scalability and performance, paging is usually the best choice. Originally Microsoft's Live Web Search also provided a scrollbar. However, the scrollbar increased server-load considerably since users are more likely to scroll than page.

• If the content is really continuous, scrolling is more natural than paging.

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

• If you get your revenue by page impressions, scrolling may not be an option for your business model.

• If paging causes actions for the content to become cumbersome, move to a scrolling model. This is an issue in Gmail. The user can only operate on the current page. Changing items across page boundaries is unexpected. Changing items in a continuous scrolled list is intuitive.

5.7 Process flow

It has long been common practice on the Web to turn each step into a separate page. While this may be the simplest way break down the problem, it may not lead to the best solution. For some Process Flows it makes sense to keep the user on the same page throughout the process.

5.7.1 Google blogger

The popular site Google Blogger generally makes it easy to create and publish blogs. One thing it does not make easy, though, is deleting comments that others may leave on your blog. This is especially difficult when you are the victim of hundreds of spam comments left by nefarious companies hoping to increase their search ranking. Blogger forces you to delete these comments through a three-step process. Each step is an individual page, all punctuated with a page refresh.

My (Bill's) blog site was recently spammed. It turns out that my 100 or so articles all had 4 or more spam comments. That means that I had to delete more than 400 spam comments. Given the way Google Blogger implemented comment deleting, I had to follow these steps for each comment on each blog article:

1. Scroll to find the offending comment.
2. Click the trash icon to delete the comment.
3. After page refreshes, click the "Remove Forever" checkbox.
4. Click the "Delete Comment" button.
5. After the page refreshes, click the link to return to my blog article.
6. Repeat steps 1–5 for each article with spam comments.

It took 1,600 clicks, 1,200 page refreshes, 400 scroll operations, and several hours to finally rid myself of all of the spam comments. If the delete action could have been completed in the same page as the comments, that would have eliminated hundreds of clicks and well over a thousand page refreshes, and scrolling would have been all but eliminated. I would not have wasted all the mental energy to reorient myself after each page transition. And I would have been a much happier man. This is a common interaction flow on the Web. It turns out to be

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

simpler to design and implement a process as a series of pages rather than a single interactive space.

5.7.2 The magic principle

Alan Cooper discusses a wonderful technique for getting away from a technology-driven approach and discovering the underlying mental model of the user. He calls it the "magic principle." Ask the question, "What if when trying to complete a task the user could invoke some magic?" For example, let's look at the problem of taking and sharing photos.

The process for this task breaks down like this:

• Take pictures with a digital camera.

• Sometime later, upload the photos to a photo site like Flickr. This involves:

— Finding the cable.

— Starting iTunes.

— Importing all photos.

Using a second program, such as Flickr Uploadr, to upload the photos to Flickr.

— Copying the link for a Flickr set (which involves first locating the page for the uploaded set).

Send the link in email to appropriate friends. If some magic were invoked, here is how it might happen:

• The camera would be event-aware. It would know that is your daughter's eighth birthday.

• When finished taking pictures of the event, the camera would upload the pictures to Flickr.

• Flickr would notify family and friends that the pictures of the birthday party are available.

5.7.3 Interactive single page processor

Consumer products come in a variety of shapes, sizes, textures, colors, etc. Online shoppers will not only have to decide that they want shoes, but do they want blue suede shoes? And what size and width do they want them in? In the end the selection is constrained by the available inventory. As the user makes decisions, the set of choices gets more and more limited. This type of product selection is typically handled with a multi-page workflow. On one page, the user selects a shirt and its color and size. After submitting the choice, a new page is displayed. Only when the user arrives at this second page does he find out that the "true navy" shirt is not available in the medium size.

5.7.4 Inline assistant process

Another common place where multiple pages are used to complete a process is when adding items to a shopping cart. As mentioned earlier, Amazon provides the typical experience So what magic can we apply to move this from a multi-page experience to a single-page experience? Instead of thinking about the cart as a process, we can think about it as a real-world object. Given this mindset, the cart can be realized in the interface as an

STUCOR APP

DOWNLOADED FROM STUCOR APP

DOWNLOADED FROM STUCOR APP

object and be made available on the page. The Gap employed an Inline Assistant Process pattern for its shopping cart when it re-launched its site a few years back.

STUCOR APP

5.7.5 Dialog overlay process

As mentioned before, any page switch is an interruption to the user's mental flow. In addition, any context switch is a chance for a user to leave the site. We seek an experience that has as little mental friction as possible. But sometimes the step-by-step flow is necessary. The Netflix approach just described uses a Dialog Overlay Process to encapsulate a multi-step flow inside a Dialog Overlay.. Overlays allow us to keep the context of the page yet present a virtual space to conduct a conversation with the user. Discover.com recently expanded its account section with a more detailed profile. The profile captures things like your payment date, mobile fraud alerts, paperless statements, and general contact information .The overlay pops up when you first enter your account.

5.7.6 Configuration process

Sometimes a Process Flow is meant to invoke delight. In these cases, it is the engagement factor that becomes most important. This is true with various Configurator Process interfaces on the Web. We can see this especially at play with car configurators. Porsche provides a configurator that allows users to build their own Porsche Being able to tweak the colors, wheels, interior, and options for a car and see the results in real time is an engaging experience. Live Previews allow the user to see the effect of his changes on a simulated version of the real thing.

5.7.7 Static single page process

The Apple example illustrates another way to get rid of multiple pages in a Process Flow. Just put the complete flow on one page in a Static Single-Page Process. The user sees all the tasks needed to complete the full process. This can be both good and bad. Seeing just one step to complete the process can encourage users to finish the task. But if the single step seems too long or too confusing, the user will most likely bail out of the process early. In other words, if placing all the tasks on a single page is enough to cause the user to bail out; it is not a good idea. In the case of the Apple store, each item is optionally set, and it's just a single click to include or exclude an item from the purchase. EBay provides two ways to sell an item. An introductory panel gathers the description of the item for sale. The "Customize your listings…" option takes the user through a traditional multi-page process.

STUCOR APP

DOWNLOADED FROM STUCOR APP