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.

PART-A

1 **What is Input/Ouput channel?**

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. Input in the human occurs mainly through the senses and output through the motor control of the effectors.

2 **What are the types of memory or memory function?**

(i) Sensory buffers ,(ii) Short-term memory or working memory, (iii) Long term memory

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

4 **What is iconic memory?**

We can demonstrate the existence of iconic memory by moving a finger in front of the eye. Can you see it in more than one place at once? This indicates a persistence of the image after the stimulus has been removed. A similar effect is noticed most vividly at firework displays where moving sparklers leave a persistent image. Information remains in iconic memory very briefly, in the order of 0.5 seconds.

5 **Write brief on existence of echoic memory.**

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. However, since this information is received at different times, we must store the stimulus in the meantime. Echoic memory allows brief 'play-back' of information.

6 **Write short notes on short term memory or working 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. For example, calculate the multiplication 35 × 6 in your head. Short-term memory can be accessed rapidly, in the order of 70 ms. However, 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.

7 **What are the two types of long term memory?**

- Episodic memory
- Semantic memory

8 **State Reasoning. What are the types of 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:

1. deductive ,2.inductive ,3.abductive.

9 What is problem solving?

Reasoning is a means of inferring new information from what is already known, problem solving is the process of finding a solution to an unfamiliar task, using the knowledge we have. Human problem solving is characterized by the ability to adapt the information we have to deal with new situations. However, often solutions seem to be original and creative.

10 State Gestalt theory.

Psychology concept is used in training. It proposes that what is 'seen' is what appears to the seer and not what may 'actually be there,' and that the nature of a unified whole is not understood by analyzing its parts. It views learning as a reorganizing of a whole situation in contrast to the behavioral psychology view that learning consists of associations between stimuli and responses. Gestalt experiments show that the brain does not act like a sponge but actively filters, structures, and matches all incoming information against known patterns to make sense of it.

11 What are the basic levels of skill in Anderson's ACT* model?

1. The learner uses general-purpose rules which interpret facts about a problem. This is slow and demanding on memory access.
2. The learner develops rules specific to the task.
3. The rules are tuned to speed up performance.

12 List out all text entry devices.

1.The alphanumeric keyboard, 2.Chord keyboards, 3.Phone pad and T9 entry, 4.Handwriting recognition, 5.Speech recognition.

13 What are touch screens?

Touch screens are another method of allowing the user to point and select objects on the screen as they detect the presence of the user's finger, or a stylus, on the screen itself. They work in one of a number of different ways: by the finger (or stylus) interrupting a matrix of light beams, or by capacitance changes on a grid overlaying the screen, or by ultrasonic reflections. The touch screen is very fast, and requires no specialized pointing device. Because the screen acts as an input device as well as an output device, there is no separate hardware to become damaged or destroyed by dirt; this makes touch screens suitable for use in hostile environments.

14 What is Eyegaze?

Eyegaze systems allow you to control the computer by simply looking at it. Some systems require you to wear special glasses or a small head-mounted box. A low-power laser is shone into the eye and is reflected off the retina. The reflection changes as the angle of the eye alters, and by tracking the reflected beam the eyegaze system can determine the direction in which the eye is looking. Eyegaze is a very fast and accurate device, but the more accurate versions can be expensive.

15 What is icon wars?

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 and 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.

16 **What are the limitations on interactive performance?**
1.Computational bound, 2.Storage channel bound, 3.Graphics bound, 4.Network capacity

17 **What are the stages in Norman's model of interaction?**
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.

18 **State 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.

19 **What are the common interface styles?**
Common interface styles includes,
1. command line interface
2. natural language menus
3.
4. question/answer and query dialog
5. form-fills and spreadsheets
6. WIMP
7. point and click
8. three-dimensional interfaces.

20 **Write notes on WIMP interface.**
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.

1 Explain the concept of (i) Input-output channel (ii) The Human Memory (iii) Computer Memory. (**April/May 2019)**

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 g 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.

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

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 focussed 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.

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

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.

2 Explain about devices used for positioning, pointing and drawing in detail

Types of Pointing Devices

Different types of pointing devices are as following:

- **Mouse**
- **Trackball**
- **Touchpad/ Trackpad**
- **Pointing Stick**
- **Graphics Tablet**
- **Touchscreen**
- **Light Pen**
- **Joystick**
- **Stylus**

pointing device is an input interface (specifically a human interface device) that allows a user to input spatial (i.e., continuous and multi-dimensional) data to a computer. CAD systems and graphical user interfaces (GUI) allow the user to control and provide data to the computer using physical gestures - point, click, and drag - for example, by moving a hand-held mouse across the surface of the physical desktop and activating switches on the mouse. Movements of the pointing device are echoed on the screen by movements of the pointer (or cursor) and other visual changes.

3 . Write short notes on Display devices.
i)Bitmap displays ii)Cathode Ray Tube iii)Liquid crystal Display iv)Digital paper

4

out physical controls, sensors and special devices with examples.

6	State and explain Gestalt theory and problem space theory.
7	Write short notes on (**April/May 2019)** (i)Psychology and design of interactive systems (ii)Text entry devices (iii)Models of interaction
8	Explain in detail about Ergonomics.

Human factors and ergonomics (commonly referred to as **human factors**) is the application of psychological and physiological principles to the engineering and design of products, processes, and systems. The goal of human factors is to reduce human error, increase productivity, and enhance safety and comfort with a specific focus on the interaction between the human and the thing of interest.[1]

The field is a combination of numerous disciplines, such as psychology, sociology, engineering, biomechanics, industrial design, physiology, anthropometry, interaction design, visual design, user experience, and user interface design. In research, human factors employs the scientific method to study human behavior so that the resultant data may be applied to the four primary goals. In essence, it is the study of designing equipment, devices and processes that fit the human body and its cognitive abilities. The two terms "human factors" and "ergonomics" are essentially synonymous.[2][3][4]

The International Ergonomics Association defines ergonomics or human factors as follows:[5]

Ergonomics (or human factors) is the scientific discipline concerned with the understanding of interactions among humans and other elements of a system, and the profession that applies theory, principles, data and methods to design to optimize human well-being and overall system performance.

Part of a series on

Psychology

Ψ

- **Outline**
- **History**
- **Subfields**

Basic types[show]

Applied psychology[hide]

- Applied behavior analysis
- Clinical
- Community
- Consumer
- Counseling
- Critical
- Educational

- Environmental
- Ergonomics
- Forensic
- Health
- Humanistic
- Industrial and organizational
- Legal
- Medical
- Military
- Music
- Occupational health
- Political
- Religion
- School
- Sport
- Traffic

Lists[show]

- Ψ **Psychology portal**

- v
- t
- e

Human factors is employed to fulfill the goals of occupational health and safety and productivity. It is relevant in the design of such things as safe furniture and easy-to-use interfaces to machines and equipment.

Proper ergonomic design is necessary to prevent repetitive strain injuries and other musculoskeletal disorders, which can develop over time and can lead to long-term disability.

Human factors and ergonomics is concerned with the "fit" between the user, equipment, and environment or "fitting a job to a person".[6] It accounts for the user's capabilities and limitations in seeking to ensure that tasks, functions, information, and the environment suit that user.

To assess the fit between a person and the used technology, human factors specialists or ergonomists consider the job (activity) being done and the demands on the user; the equipment used (its size, shape, and how appropriate it is for the task), and the information used (how it is presented, accessed, and changed). Ergonomics draws on many disciplines in its study of humans and their environments, including anthropometry, biomechanics, mechanical engineering, industrial engineering, industrial design, information design, kinesiology, physiology, cognitive psychology, industrial and organizational psychology, and space psychology.

9 Write in detail about Elements of the WIMP interface. .APRIL/MAY 2018

In human–computer interaction, **WIMP** stands for "windows, icons, menus, pointer", denoting a style of interaction using these elements of the user interface. It was coined by Merzouga Wilberts in 1980.[4] Other expansions are sometimes used, such as substituting "mouse" and "mice" for menus, or "pull-down menu" and "pointing" for pointer.

Although the term has fallen into disuse, some use it as an approximate synonym for *graphical user interface (GUI)*. Any interface that uses graphics can be called a GUI, and WIMP systems

ms. However, while all WIMP systems use graphics as a key elements (the icon and pointer elements), and therefore are GUIs, the reverse is not true. Some GUIs are not based in windows, icons, menus, and pointers. For example, most mobile phones represent actions as icons, and some might have menus, but very few include a pointer or run programs in a window.[citation needed]

WIMP interaction was developed at Xerox PARC (see Xerox Alto, developed in 1973) and popularized with Apple's introduction of the Macintosh in 1984, which added the concepts of the "menu bar" and extended window management.[8]

In a WIMP system:[citation needed]

- A window runs a self-contained program, isolated from other programs that (if in a multi-program operating system) run at the same time in other windows.
- An icon acts as a shortcut to an action the computer performs (e.g., execute a program or task).
- A menu is a text or icon-based selection system that selects and executes programs or tasks.
- The pointer is an onscreen symbol that represents movement of a physical device that the user controls to select icons, data elements, etc.

10 Explain the concept of paradigms for interaction in detail.

A paradigm is a frame of reference or theory that affects how we see and experience a situation. By definition, it represents a "group of ideas about how something should be done... or thought about."

Although there is an appropriate time to use each paradigm, Win/Win is the preferable approach for those who must work together to achieve maximum results.

The Six Paradigms

Win/Win

Win/Win is a mindset or approach to a situation that seeks to ensure the interest of all parties are considered. It is not an "either/or" proposition.

Covey describes Win/Win as "not your way or my way; it's a better way, a higher way." And it leads to what Covey calls the "Third Alternative."

The person who practices this paradigm is: principle-centered, places emphasis on what is right/best for **ALL** parties involved, and is not focused on getting the "upper-hand" or best deal for themselves.

Win/Lose

Win/Lose is where one party asserts their interests over the interests of another. They subscribe to the "scarcity mentality" and believe "you must lose so that I can win".

This interaction often "feels" highly competitive in nature and destroys teamwork.

Lose/Win

Lose/Win means I cooperate with the other person to the degree that I lose so they can win. In this approach, the "loser" often walks away very resentful.

More often than not, this person chooses to lose in order to avoid conflict.

Lose/Lose

Lose/Lose is when no one wins; both parties lose and both end up with very hard feelings.

This approach does not serve to advance positive team dynamics and is very unhealthy for all parties involved.

Win

The person operating with this paradigm says "I want to win. I don't want you to lose, but getting what you want is your business, not mine."

They do not actively seek to ensure that the interests of the other party are considered.

Win/Win or No Deal

The Win/Win or No Deal mindset actively seeks the interests of all parties involved, including theirs.

It requires a high character ethic, investment of time and lots of two-way communication.

In the event an agreement suitable to all parties cannot be reached, the only Win/Win is "No Deal."

Even if no deal is the best outcome, the intent and engagement of this interaction sets the stage for possible Win/Wins in the future.

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.

PART-A

1 **What is Design?**

Design is defined as achieving Goals within constraints and encompasses work tasks data design, architectural design, interface design and component-level design and odel or design specification.

2 **What are the steps for Interaction Design process?**

- Requirements
- Analysis and Design
- Iteration and prototyping
- Implementation and Deployment.

3 **What are the classification of evaluation techniques**?

Cognitive walkthrough, Heuristic evaluation ,Review based ,Model based

4 **What are the advantages and disadvantages of Prototyping Model?**

Advantages:

- It produces the products quickly and thus saves the time and solves the waiting problem in waterfall model.
- It minimizes the cost and product failure.
- It is possible for the developers and client to check the function of preliminary implementations of system models before committing to a final system.
- It obtains feedback from clients and changes in system concept.

Disadvantages:

- It ignores quality, reliability maintainability and safety requirements. Customer satisfaction is not achieved.

5 **What are the Levels of Interaction?**

Widgets ,Screen design, Navigation design, Other apps and operating system

6 **What are the two things you need in order for prototyping methods to work?**

1. To understand what is wrong and how to improve.

2. A good start point.

7 **What are the activities in the waterfall model of the software life cycle?**

1.Requirements specification , 2.Design , 3.Construction ,4.Integration and Testing , 5.Installation and Maintenance.

8 **What are the Emphasis for usability engineering**

The emphasis for usability engineering is in knowing exactly what criteria will be used to judge a product for its usability. The ultimate test of a product's usability is based on measurements of users' experience with it. Therefore, since a user's direct experience with an interactive system is at the physical interface, focus on the actual user interface
is understandable.

9 **What are the 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

10 **What are the possible ways to set measurement levels in a usability specification?**

1.Existing system or previous version

2. competitive systems

3. carrying out the task without use of a computer system

4. an absolute scale

5. your own prototype

ier performance

	7. each component of a system separately **8. a successive split of the difference between best and worst values observed in user tests**
11	**What are the three main goals of Evaluation?** **1. To assess the extent and accessibility of the system's functionality.** **2.To assess users' experience of the interaction.3.To identify any specific problems with** **the system.**
12	**Define 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.**
13	**What is the beneficial to have access to the design rationale?** **1.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 .** **2. 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.**
14	**What is Design space Analysis?** The design space is initially structured by a set of questions representing the major issues of the design. Since design space analysis is structure oriented, it is not so important that the questions recorded are the actual questions asked during design meetings.
15	**What is key to an effective design space analysis?** The key to an effective design space analysis using the QOC(Questions,Options and Criteria) notation is deciding the right questions to use to structure the space and the correct criteria to judge the options.
16	**What are the principles to support Usability?** 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.

17	**Define Usability and Effectiveness.** 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.
18	**Define 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.
19	**Define Efficiency and Satisfaction.** 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
20	**Define HCI Patterns.** A pattern is an invariant solution to a recurrent problem within a specific context. Patterns address the problems that designers face by providing a 'solution statement'. Patterns are an approach to capturing and reusing this knowledge – of abstracting the essential details of successful design so that these can be applied again and again in new situations.
21	**Define Universal Design principles.** It is the process of designing products so that they can be used by as many people as possible in as many situations as possible.

PART-B

1

Explain in detail about process of Design and golden rule of Design. (**April/May 2019**)

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. For example, how do people currently watch movies? What sort of personal appliances do they currently use? There are a number of techniques used for this in HCI: interviewing people, videotaping them, looking at the documents and objects that they work with, observing them directly.

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 models, which are a means to capture how people carry out the various tasks that are part of their work and life. we will look at scenarios, rich stories of interaction, which can be used in conjunction with a method like task analysis or on their own to record and make vivid actual interaction. These techniques can be used both to represent the situation as it is and also the desired situation.

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. We need to record our design choices in some way and there are various notations and methods to do this, including those used to record the existing situation

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.. Some forms of evaluation can be done using the design on paper, but it is hard to get real feedback without trying it out. Most user interface design therefore involves some form of prototyping, producing early versions of systems to try out with real users.

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 5.4 User focus 197 system that can be given to others

Golden rule of Design.

Part of the understanding we need is about the circumstances and context of the particular design problem. However, there are also more generic concepts to understand. The designs we produce may be different, but often the raw materials are the same. This leads us to the golden rule of design Interaction design basics In the case of a physical design this is obvious. Look at a chair with a steel frame and one with a wooden frame. They are very different: often the steel frames are tubular or thin L or H section steel. In contrast wooden chairs have thicker solid legs. If you made a wooden chair using the design for a metal one it would break; if you made the metal one in the design for the wooden one it would be too heavy to move. For Human–Computer Interaction the obvious materials are the human and the computer. That is we must: n understand computers – limitations, capacities, tools, platforms n understand people – psychological, social aspects, human error. e must understand the fundamental materials of human–computer interaction in order to design areas. For example, the way you fit seats and windows into an airplane's hull affects the safety and strength of the aircraft as a whole.

2	**Explain in detail about Scenarios and usage of 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 example of a scenario for the personal movie player. Like the persona it is perhaps more detailed than appears necessary, but the detail helps make the events seem real. 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. In the next section we will discuss ways of describing the patterns of interaction with a system. These are more complex and involve networks or hierarchies. In contrast scenarios are linear – they represent a single path amongst all the potential interactions. This linearity has both positive and negative points: **Time is linear**- Our lives are linear as we live in time and so we find it easier to understand simple linear narratives. We are natural storytellers and story listeners. **But no alternatives** -Real interactions have choices, some made by people, some by systems. A simple scenario does not show these alternative paths. In particular, it is easy to miss the unintended things a person may do. Scenarios are a resource that can be used and reused throughout the design process: helping us see what is wanted, suggesting how users will deal with the potential design, checking that proposed implementations will work, and generating test cases for final evaluation.

3 **Explain in detail about Navigation design through Levels of Interaction and Screen design. (April/May 2019 , (April/May 2018)**

Imagine yourself using a word processor. You will be doing this in some particular social and physical setting, for a purpose. But now we are focusing on the computer system itself. You interact at several levels:

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.

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

The structure of an application is to think about actual use:

- who is going to use the application?
- how do they think about it?
- what will they do with it?

This can then drive the second task – thinking about structure. Individual screens or the layout of devices will have their own structure, but this is for the next section. Here we will consider two main kinds of issue:

local structure – looking from one screen or page out

global structure – structure of site, movement between screens.

local structure:

Much of interaction involves goal-seeking behavior. Users have some idea of what they are after and a partial model of the system. 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. However, in a world of partial knowledge users meander through the system. The Levels of interaction PC application Widgets Screen design Navigation design Other apps and operating system Physical device Buttons, dials, lights, displays Physical layout Main modes of device The real world! Website Form elements, tags and links Page design Site structure The web, browser, external links 5.6 Navigation design 205 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.

To get you started, here are four things to look for when looking at a single web page, screen or state of a device.

- 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.

You then need to know where you are going when you click a button or what will happen. Of course you can try clicking the button to see. In the case of a website or information system this may mean you then have to use some sort of 'back' mechanism to return, but that is all; however, in an application or device the action of clicking the button may already have caused some effect. If the system has an easy means to undo or reverse actions this is not so bad, but it is better if users do not have to use this 'try it and see' interaction. Where response times are slow this is particularly annoying.

Global structure:

Hierarchical organization of overall structure of an application. This is the way the various screens, pages or device states link to one another. One way to organize a system is in some form of hierarchy. This is typically organized along functional boundaries (that is, different kinds of things), but may be organized by roles, user type, or some more esoteric breakdown such as modules in an educational system. The hierarchy links screens, pages or states in logical groupings. 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.

In a pure information system or static website it may be sufficient to have a fully hierarchical structure, perhaps with next/previous links between items in the same group. However, for any system that involves doing things, constantly drilling down from one part of the hierarchy to another is very frustrating. Usually there are ways of getting more quickly from place to place. For example, in a stock control system there may be a way of going from a stock item to all orders outstanding on that item and then from an order to the purchase record for the customer who placed the order. These would each be in a very different part of a hierarchical view of the application, yet directly accessible from one another.

A simple way is to use a network diagram showing the principal states or screens linked together with arrows. This can

- show what leads to what
- show what happens when
- include branches and loops
- be more task oriented than a hierarchy

SCREEN DESIGN AND LAYOUT

The basic principles at the screen level reflect those in other areas of interaction design: Ask What is the user doing? Think What information is required? What comparisons may the user need to make? In what order are things likely to be needed? Design Form follows function: let the required interactions drive the layout.

4 **Explain in detail about Waterfall model of Software development Life cycle**

(April/May 2019)

A fundamental feature of software engineering, therefore, is that it provides the structure for applying techniques to develop software systems. The software life cycle is an attempt to identify the activities that occur in software development. These activities must then be ordered in time in any development project and appropriate techniques must be adopted to carry them through. 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. 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

Requirements specification

In requirements specification, the designer and customer try to capture a description of what the eventual system will be expected to provide. This is in contrast to determining how the system will provide the expected services, which is the concern of later activities. Requirements specification involves eliciting information from the customer about the work environment, or domain, in which the final product will function. Aspects of the work domain include not only the particular functions that the software product must perform but also details about the environment in which it must operate, such as the people whom it will potentially affect and the new product’s relationship to any other products which it is updating or replacing. 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 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

Architectural design

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. There are many structured techniques that are used to assist a designer in deriving an architectural description from information in the requirements specification (such as CORE, MASCOT and HOOD). Details of these techniques are outside the scope of this book, but can be found in any good software engineering textbook. What we will mention here is that the majority of these techniques are adequate for capturing the functional requirements of the system – the services the system must provide in the work domain – but do not provide an immediate way to capture other non-functional requirements – features of the system that are not directly related to the actual services provided but relate to the manner in which those services must be provided. Some classic examples of non-functional requirements are the efficiency, reliability, timing and safety features of the system. Interactive features of the system,

Detailed design

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. Typically, 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.

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. It may also be necessary to certify the final system according to requirements imposed by some outside authority, such as an aircraft certification board.

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 system which are discovered after release and the revision of the system services to satisfy requirements that were not realized during previous development. Therefore, maintenance provides feedback to all of the other activities in the life cycle.

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

Proofs that are for verification of a design can frequently occur within one language or between two languages which both have a precise mathematical semantics. Time constraints for a design project and the perceived economic implications of the separate components usually dictate which proofs are carried out in full formality and which are done only rigorously (if at all). As research in this area matures and automated tools provide assistance for the mechanical aspects of proof, the cost of proof should decrease.

Validation proofs are much trickier, as they almost always involve a transformation between

	languages. Furthermore, the origin of customer requirements arises in the inherent ambiguity of the real world and not the mathematical world. This precludes the possibility of objective proof, rigorous or formal. Instead, there will always be a leap from the informal situations of the real world to any formal and structured development process
5	**Explain in detail about Prototyping Model** 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. **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. **Evolutionary**: 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 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 for a designer/programmer to create a realistic prototype quickly and efficiently. **Techniques for prototyping:** **Storyboards**: 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 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: More functionality must be built into the prototype to demonstrate the work that the application will accomplish. 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: Hyper Talk was an example of a special-purpose high-level programming language which makes it easy for the designer to program certain features of an interactive system at the expense of other system features like speed of response or space efficiency. Hyper Talk 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.

6 **Explain in detail about the usability engineering and principles to support Usability**

One approach to user-centered design has been the introduction of explicit usability engineering goals into the design process, as suggested by Whiteside and colleagues at IBM and Digital Equipment Corporation and by Nielsen at Bellcore.Engineering depends on interpretation against a shared background of meaning, agreed goals and an understanding of how satisfactory completion will be judged. The emphasis for usability engineering is in knowing exactly what criteria will be used to judge a product for its usability. The ultimate test of a product's usability is based on measurements of users' experience with it. Therefore, since a user's direct experience with an interactive system is at the physical interface, focus on the actual user interface is understandable. The danger with this limited focus is that much of the work that is accomplished in interaction involves more than just the surface features of the systems used to perform that work. In reality, the whole functional architecture of the system and the cognitive capacity of the users should be observed in order to arrive at meaningful measures. But it is not at all simple to derive measurements of activity beyond the physical actions in the world, and so usability engineering is limited in its application. 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.

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

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. Measurements such as those promoted by usability engineering are also called usability metrics.

Possible ways to set measurement levels in a usability specification Set levels with respect to information on:

1. an existing system or previous version
2. competitive systems
3. carrying out the task without use of a computer system
4. an absolute scale
5. your own prototype
6. user's own earlier performance
7. each component of a system separately
8. a successive split of the difference between best and worst values observed in user tests

The major feature of usability engineering is the assertion of explicit usability metrics early on in the design process which can be used to judge a system once it is delivered. The problem with usability metrics 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.

7 **Explain in detail about the Standards ,guidelines and golden rules for Interactive system design**

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

The strength of a standard lies in its ability to force large communities to abide – the so-called authority we have referred to earlier. It should be noted that such authority does not necessarily follow from the publication of a standard by a national or international body. In fact, many standards applying to software design are put forth as suggestive measures, rather than obligatory. The authority of a standard (or a guideline, for that matter) can only be determined from its use in practice. Some software products become de facto standards long before any formal standards document is published

Guidelines

We have observed that the incompleteness of theories underlying the design of interactive software makes it difficult to produce authoritative and specific standards. As a result, the majority of design rules for interactive systems are suggestive and more general guidelines. Our concern in examining the wealth of available guidelines is in determining their applicability to the various stages of design. The more specific the guideline, the more suited it is to detailed design. The guidelines can also be automated to some extent, providing a direct means for translating detailed design specifications into actual implementation. There are a vast amount of published guidelines for Design rules interactive system design (they are frequently referred to as guidelines for user interface design). We will present only a few examples here to demonstrate the content of guidelines in that vast literature. Several books and technical reports contain huge catalogs of guidelines.

The basic categories of the Smith and Mosier guidelines are:

1. Data Entry
2. Data Display
3. Sequence Control
4. User Guidance
5. Data Transmission
6. Data Protection

Each of these categories is further broken down into more specific subcategories which contain the particular guidelines.. A striking feature of this compendium of guidelines is the extensive cross-referencing within the catalog, and citation to published work that supports each guideline.

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.

In moving from abstract guidelines to more specific and automated ones, it is necessary to introduce assumptions about the computer platform on which the interactive system is designed. So, for example, in Apple's Human Interface Guidelines: the Apple Desktop Interface, there is a clear distinction between the abstract guidelines (or principles), independent of the specific Macintosh hardware and software, and the concrete guidelines, which assume them. The abstract guidelines provide the so-called philosophy of programming that Apple would like designers to adopt in programming applications for the Macintosh. The more concrete guidelines are then seen as more concrete manifestations of that philosophy.

Design Shneiderman's eight golden rules provide a convenient and succinct summary of the key principles of interface design.

1. Strive for consistency in action sequences, layout, terminology, command use and so on.

2. Enable frequent users to use shortcuts, such as abbreviations, special key sequences and macros, to perform regular, familiar actions more quickly.

3. Offer informative feedback for every user action, at a level appropriate to the magnitude of the action.

4. Design dialogs to yield closure so that the user knows when they have completed a task.

5. 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.

6. 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.

7. Support internal locus of control so that the user is in control of the system, which responds to his actions.

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

These rules provide a useful shorthand for the more detailed sets of principles described earlier. Like those principles, they are not applicable to every eventuality and need to be interpreted for each new situation. However, they are broadly useful and their application will only help most design projects.

8 **Explain in detail about the Universal Design principles**

universal design as 'the process of designing products so that they can be used by as many people as possible in as many situations as possible'. But what does that mean in practice? Is it possible to design anything so that anyone can use it – and if we could, how practical would it be? Wouldn't the cost be prohibitive? In reality, we may not be able to design everything to be accessible to everyone, and we certainly cannot ensure that everyone has the same experience of using a product, but we can work toward the aim of universal design and try to provide an equivalent experience. Although it may seem like a huge task, universal design does not have to be complex or costly. In fact, if you are observant, you will see many examples of design that attempt to take account of user diversity. Next time you cross the road, look at the pavement. The curb may be lowered, to enable people who use wheelchairs to cross more easily. The paving near the curb may be of a different texture – with raised bumps or ridges – to enable people who cannot see to find the crossing point. Notice how many modern buildings have automatic doors that open on approach. Or lifts that offer both visual and auditory notification of the floor reached. And, whilst these designs make the crossing, the building and the lift more accessible to people who have disabilities, notice too how they also help other users. The parent with a child in a buggy, or the traveller with wheeled luggage, can cross the road more easily; the shopper with heavy bags, or the small child, can enter the building; and people are less likely to miss their floor because they weren't paying attention. Universal design is primarily about trying to ensure that you do not exclude anyone through the design choices you make but, by giving thought to these issues, you will invariably make your design better for everyone.

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

Principle one is 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 should be supported. Where appropriate, security, privacy and safety provision should be available to all.

Principle two is 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. **Principle three** is that 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.

Principle four is 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.

Principle five is 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.

Principle six is 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.

Principle seven 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. It is interesting to note that these principles are closely related to the context of general user-centered design rules, indicating again that universal design is fundamentally good design for all.

9 **Explain in detail about Evaluation Techniques**.

Evaluation should not be thought of as a single phase in the design process (still less as an activity tacked on the end of the process if time permits). Ideally, evaluation should occur throughout the design life cycle, with the results of the evaluation feeding back into modifications to the design. Clearly, it is not usually possible to perform extensive experimental testing continuously throughout the design, but analytic and informal techniques can and should be used.

Three main goals of Evaluation:

1. To assess the extent and accessibility of the system's functionality
2. To assess users' experience of the interaction.
3. To identify any specific problems with the system.

Classification of Evaluation Techniques:

- Cognitive walkthrough
- Heuristic evaluation
- Review based and Model based

Cognitive walkthrough

The origin of the cognitive walkthrough approach to evaluation is the code walkthrough familiar in software engineering. Walkthroughs require a detailed review of a sequence of actions. In the code walkthrough, the sequence represents a segment of the program code that is stepped through by the reviewers to check certain characteristics (for example, that coding style is adhered to, conventions for spelling variables versus procedure calls, and to check that system-wide invariants are not violated). In the cognitive walkthrough, the sequence of actions refers to the steps that an interface will require a user to perform in order to accomplish some known task. The evaluators then 'step through' that action sequence to check it for potential usability problems. Usually, the main focus of the cognitive walkthrough is to establish how easy a system is to learn. More specifically, the focus is on learning through exploration. Experience shows that many users prefer to learn how to use a system by exploring its functionality hands on, and not after sufficient training or examination of a user's manual. So the checks that are made during the walkthrough ask questions that address this exploratory learning.

To do a walkthrough (the term walkthrough from now on refers to the cognitive walkthrough, and not to any other kind of walkthrough), you need four things:

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 evaluators 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.

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

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 he 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.

Heuristic evaluation

A heuristic is a guideline or general principle or rule of thumb that can guide a design decision or be used to critique a decision that has already been made. Heuristic evaluation, developed by Jakob Nielsen and Rolf Molich, is a method for structuring the critique of a system using a set of relatively simple and general heuristics. Heuristic evaluation can be performed on a design specification so it is useful for evaluating early design. But it can also be used on prototypes, storyboards and fully functioning systems. It is therefore a flexible, relatively cheap approach. Hence it is often considered a discount usability technique. The general idea behind heuristic evaluation is that several evaluators independently critique a system to come up with potential usability problems. It is important that there be several of these evaluators and that the evaluations be done independently.

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 inexperienced and experienced users.

8. Aesthetic and minimalist design Dialogs should not contain information that is irrelevant or rarely needed. Every extra unit of information in a dialog competes with the relevant units of information and diminishes their relative visibility.

9. Help users recognize, diagnose and recover from errors Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

10. Help and documentation Few systems can be used with no instructions so it may be necessary to provide help and documentation. Any such information should be easy to search, focussed on the user's task, list concrete steps to be carried out, and not be too large.

Once each evaluator has completed their separate assessment, all of the problems are collected and the mean severity ratings calculated. The design team will then determine the ones that are the most important and will receive attention first.

Review based

Experimental psychology and human–computer interaction between them possess a wealth of experimental results and empirical evidence. Some of this is specific to a particular domain, but much deals with more generic issues and applies in a variety of situations. Examples of such issues are the usability of different menu types, the recall of command names, and the choice of icons. 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 9.4 Evaluation through user participation 327 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. For example, an experiment testing the usability of a particular style of help system using novice participants may not provide accurate evaluation of a help system designed for expert users. The review should therefore take account of both the similarities and the differences between the experimental context and the design under consideration.

Model-based evaluation

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 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. Models such as state transition networks are useful for evaluating dialog designs prior to implementation.

10 **Explain in detail about Design rationale.**

It 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.

It is beneficial to have access to the design rationale for several reasons:

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 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 reexamined to see if any rejected alternatives are now more favorable or if any new alternatives are now possible.

There are 3 types:

Process-oriented design rationale

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.

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 visually in the form of a directed graph which can be directly edited by the creator of the design rationale. 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

Design space analysis

The design space is initially structured by a set of questions representing the major issues of the design. Since design space analysis is structure oriented, it is not so important that the questions recorded are the actual questions asked during design meetings. Rather, these questions represent an agreed characterization of the 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.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.

Another 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

Psychological design rationale

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. People use computers to accomplish some tasks in their particular work domain, as we have seen before. 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 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 (for example, that it is more ‘natural’ for the user, or easier to learn). 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.

> The main task the system is to support is learning how Smalltalk works. In learning about the programming environment, the programmer 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?

UNIT –III MODELS AND THEORIES

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

PART -A

1. **What is a Cognitive model?**
A Cognitive model is the designer's intended mental model for the user of the system: a set of ideas about how it is organized and operates.

2. **What is a models and theories?**
"analyze and design user interfaces and new user-interface technologies", "created software tools and development environment to facilitate the construction of graphical user interfaces", "pioneered the user of voice and video in user interfaces, hypertext links, interactive tutorials and context-sensitive help systems."

3. **Define Cognition psychology.**
Cognitive psychology is the study of mental processes such as "attention, language use, memory, perception, problem solving, creativity, and thinking

4. **Define user modeling?**
User modeling is the subdivision of human–computer interaction which describes the process of building up and modifying a conceptual understanding of the user. The main goal of user modeling is customization and adaptation of systems to the user's specific needs. The system needs to "say the 'right' thing at the 'right' time in the 'right' way".

5. **What do we do when there are several ways of solving a problem, or if the solutions to two sub goals 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.

6. **What are issues for goal hierarchies**
1.Granularity,
2.Routine learned behavior, not problem solving,
3.Conflict,
4.Error

7. **What is GOMS.**
GOMS is a specialized human information processor model for human-computer interaction observation that describes a user's cognitive structure on four components. a set of **Goals**, a set of **Operators**, a set of **Methods** for achieving the goals, and a set of **Selections** rules for choosing among competing methods for goals.

8. **Define Goals and Operators.**

Goals are symbolic structures that define a state of affairs to be achieved and determinate a set of possible methods by which it may be accomplished

Operators are elementary perceptual, motor or cognitive acts, whose execution is necessary to change any aspect of the user's mental state or to affect the task environment

9. **Define Methods and Selections.**

Methods describe a procedure for accomplishing a goal

Control Structure: Selection Rules are needed when a goal is attempted, there may be more than one method available to the user to accomplish it.

10. **Give an example for GOMS**

```
GOAL: CLOSE-WINDOW
 . [select GOAL: USE-MENU-METHOD
. MOVE-MOUSE-TO-FILE-MENU
 . PULL-DOWN-FILE-MENU
 . CLICK-OVER-CLOSE-OPTION
GOAL: USE GOAL: USE-CTRL-W-METHOD
 . PRESS-CONTROL-W-KEYS]
```

For a particular user:

Rule 1: Select USE-MENU-METHOD unless another rule applies

Rule 2: If the application is GAME, select CTRL-W-METHOD

11. **Describe Cognitive complexity theory**

Cognitive complexity theory, begins with the basic premises of goal decomposition from GOMS and enriches the model to provide more predictive power. CCT has two parallel descriptions: one of the user's goals and the other of the computer system (called the device in CCT).

12. **Describe various problem with CCT.**

There are various problems with CCT. As with many 'rich' description methods, the size of description for even a part of an interface can be enormous. Furthermore, there may be several ways of representing the same user behavior and interface behavior, yielding different measures of dissonance.

13. **How to Representative of the linguistic approach?**

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.

14. **What is Task Action Grammar?**

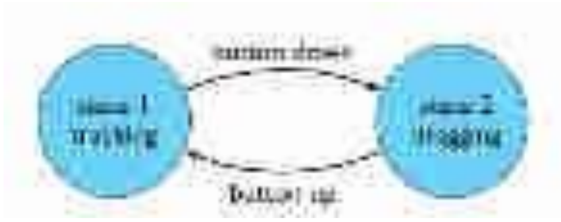
Measures based upon BNF have been criticized as not 'cognitive' enough. They ignore the advantages of consistency both in the language's structure and in its use of command names and letters. Task–action grammar (TAG) [284] attempts to deal with some of these problems by including elements such as parametrized grammar rules to emphasize consistency and encoding the user's world knowledge

15. **Define Keystroke Level Model (KLM).**

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.

16. **Define three-state model.**

The three-state model, which captures some of these crucial distinctions. He begins by looking at a mouse. If you move it with no buttons pushed, it normally moves the mouse cursor about. This tracking behavior is termed state 1. Depressing a button over an icon and then moving the mouse will often result in an object being dragged about. This he calls
state 2.



17. **Define term computer-supported cooperative work' (CSCW).**

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 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.

18. **What is use of storekeeper ?**

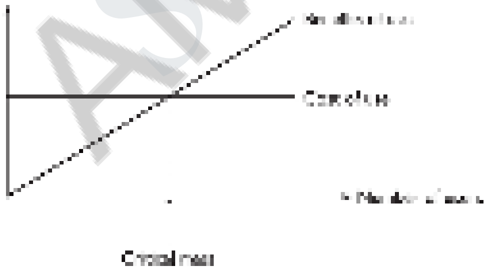
The storekeeper always used to understate stock levels slightly in order to keep an emergency supply, or sometimes inflate the quoted levels when a delivery was due from a reliable supplier. Also, requests for stock information allowed the storekeeper to keep track of future demands and hence plan future orders.

19. **What is Free rider problem ?**

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. However, it is easy for the number of free riders gradually to increase and the system slide into disuse.

20. **What is 'Critical Mass' ?**

Critical mass is the point at which a growing company becomes self-sustaining, and no longer needs additional investment to remain economically viable.



21. **Who are the stakeholders?**

Understanding stakeholders is key to many of the approaches to requirements capture, since in an organizational setting it is not simply the end-user who is affected by the introduction of new technology

22. **Define 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
t the end-users.

23. **What are the CATWOE approach?**

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 (for example, a director whose profits increase or decrease depending on the success of the system).
Facilitating stakeholders are people who are involved with the design, development and maintenance of the system.

24. **Define Open System Task Analysis (OSTA)**

OSTA is an alternative socio-technical approach, which 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.

25. **Define Soft systems methodology(SSM).**

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.

26. **Define ETHICS methodology.**

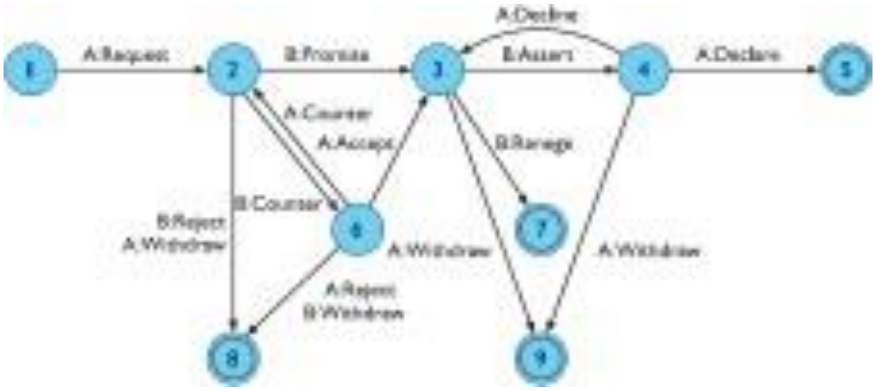
Effective Technical and Human Implementation of Computer-based Systems (ETHICS) ETHICS considers the process of system development as one of managing change: conflicts will occur and must be negotiated to ensure acceptance and satisfaction with the system. If any party is excluded from the decision-making process then their knowledge and contribution is not utilized and they are more likely to be dissatisfied. However, participation is not always complete.

27. **What is FACE-TO-FACE communication?**

Face-to-face contact is the most primitive form of communication – primitive, that is, in terms of technology. If, on the other hand, we consider the style of communication, the interplay between different channels and productivity, we instead find that face-to-face is the most sophisticated communication mechanism available. The first thing to note is that face-to-face communication involves not just speech and hearing, but also the subtle use of body language and eyegaze.

28. **What is Speech act theory?**

A particular form of conversational analysis, 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, Coordinator.



29. **What is the use Text-based communication?**

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

30. **What are four types of textual communication in current groupware?**

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.

31. **Draw Hypertext conversation structure.**

32. **What is use of distributed cognition**

A school of thinking has recently developed which regards thinking as happening not just within the head, but in the external relationships with things in the world and with other people. This viewpoint is called distributed cognition

33. **What is hypertext, multimedia and the world-wide web.**

- Hypertext allows documents to be linked in a non-linear fashion.
- Multimedia incorporates different media: sound, images, video.
- The world wide web is a global hypermedia system.

34. **What is the advantage of animation?**

1. Communication Skills
2. Building Bridges
3. Self-expression
4. Technical Skills
5. Presentation Skills

35. **Define web technology and issues.**

The web consists of a set of protocols built on top of the internet that, in theory, allow multimedia documents to be created and read from any connected computer in the world. The web supports hypertext, graphics, sound and movies, and, to structure and describe the information, uses a language called HTML (hypertext markup language) or in some cases, XML (extensible markup language).

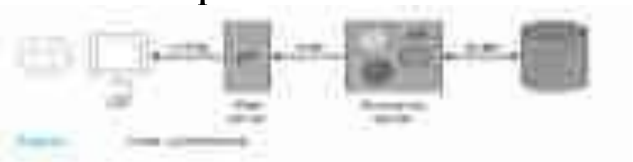
36. **Define web servers and clients**

A Web server is a program that uses HTTP (Hypertext Transfer Protocol) to serve the files that form Web pages to users, in response to their requests, which are forwarded by their computers' HTTP clients. Dedicated computers and appliances may be referred to as Web servers as well.

37. **What is the difference between static and dynamic web pages?**

In simplest terms, static Web pages are those with content that cannot change without a developer editing its source code, while dynamic Web pages can display different content from the same source code. When it comes to using static or dynamic pages for parts of your company's website, having the most advanced code on each of your pages is not important.

38. **Give an example for n-tier architecture.**



39. **What are the Network issues**

- bandwidth.
- latency.
- Jitter

PART B

1. **Create a goms description of the task of photocopying an article from a journal. Discuss the Issue of closure in terms of your goms description.**

One possible GOMS description of the goal hierarchy for this task is given below. Answers will vary depending on assumptions about the photocopier used as the model for the exercise. In this example, we will assume that the article is to be copied one page at a time and that a cover over the imaging surface of the copier has to be in place before the actual copy can be made.

```
GOAL: PHOTOCOPY-PAPER
. GOAL: LOCATE-ARTICLE
. GOAL: PHOTOCOPY-PAGE repeat until no more pages
. . GOAL: ORIENT-PAGE
. . . OPEN-COVER
. . . SELECT-PAGE
. . . POSITION-PAGE
. . . CLOSE-COVER
. . GOAL: PRESS-COPY-BUTTON
. . GOAL: VERIFY-COPY
. . . LOCATE-OUT-TRAY
. . . EXAMINE-COPY
. GOAL: COLLECT-COPY
. . LOCATE-OUT-TRAY
. . REMOVE-COPY (outer goal satisfied!)
. GOAL: RETRIEVE-JOURNAL
. . OPEN-COVER
. . REMOVE-JOURNAL
. . CLOSE-COVER
```

The closure problem which appears in this example occurs when the copy of the article **is removed from the photocopier out tray, satisfying the overall goal for the task. In the above description, however, the original journal article is still on the imaging surface of the photocopier, and the cover is closed. The user could easily forget to remove the journal. How could the photocopying procedure be revised to eliminate this problem? One answer is to force the goal RETRIEVE-JOURNAL to be satisfied before**

.

2. **Do a keystroke-level analysis for opening up an application in a visual desktop interface using A mouse as the pointing device, comparing at least two different methods for performing the Task. Repeat the exercise using a trackball. Consider how the analysis would differ for various Positions of the trackball relative to the keyboard and for other pointing devices.**

We provide a keystroke-level analysis for three different methods for launching an application on a visual desktop. These methods are analyzed for a conventional onebutton mouse, a trackball mounted away from the keyboard and one mounted close to the keyboard. The main distinction between the two trackballs is that the second one does not require an explicit repositioning of the hands, that is there is no time required for homing the hands between the pointing device and the keyboard.

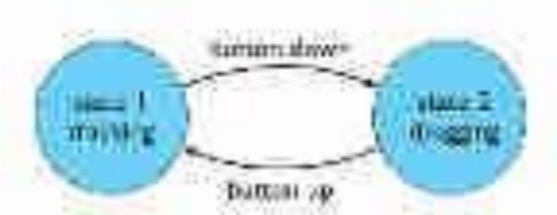
Method 1 Double clicking on application icon
StepsOperator Mouse Trackball1 Trackball2 1.Move hand to mouse
H[mouse] 0.400 0.400 0.000
2.Mouse to icon P[to icon] 0.664 1.113 1.113
3.Double click 2B[click] 0.400 0.400 0.400
4.Return to keyboard H[kbd] 0.400 0.400 0.000
Total times 1.864 2.313 1.513
Method 2 Using an accelerator key
StepsOperator Mouse Trackball1 Trackball2
1. Move hand to mouse H[mouse] 0.400 0.400 0.000
2. Mouse to icon P[to icon] 0.664 1.113 1.113
3. Click to select B[click] 0.200 0.200 0.200
4. Pause M 1.350 1.350 1.350
5. Return to keyboard H[kbd] 0.400 0.400 0.000
6. Press accelerator K 0.200 0.200 0.200
Total times 3.214 3.663 2.763
Method 3 Using a menu
StepsOperator Mouse Trackball1 Trackball2
1. Move hand to mouse H[mouse] 0.400 0.400 0.000
2. Mouse to icon P[to icon] 0.664 1.113 1.113
3. Click to select B[click] 0.200 0.200 0.200
4. Pause M 1.350 1.350 1.350
5. Mouse to file menu P 0.664 1.113 1.113
6. Pop-up menu B[down] 0.100 0.100 0.100
7. Drag to open P[drag] 0.713 1.248 1.248
8. Release mouse B[up] 0.100 0.100 0.100
9. Return to keyboard H[kbd] 0.400 0.400 0.000
Total times 4.591 6.024 5.224

3. **Explain in details of three-state model.**

- Buxton has developed a simple model of input devices , the three-state model, which captures some of these crucial distinctions. He begins by looking at a mouse. If you move it with no buttons pushed, it normally moves the mouse cursor about.
- This tracking behavior is termed state 1. Depressing a button over an icon and then moving the mouse will often result in an object being dragged about. This he calls state 2

(see Figure 1).

- If instead we consider a light pen with a button, it behaves just like a mouse when it is touching the screen. When its button is not depressed, it is in state 1, and when its button is down, state 2.
- However, the light pen has a third state, when the light pen is not touching the screen. In this state the system cannot track the light pen's position. This is called state 0 (see Figure2).
- A touchscreen is like the light pen with no button. While the user is not touching the screen, the system cannot track the finger – that is, state 0 again. When the user touches the screen, the system can begin to track – state 1. So a touchscreen is a state 0–1 device whereas a mouse is a state 1–2 device. As there is no difference between a state 0–2 and a state 0–1 device, there are only the three possibilities we have seen.



- The only additional complexity is if the device has several buttons, in which case we would have one state for each button: 2left, 2middle, 2right.
- One use of this classification is to look at different pointing tasks, such as icon selection or line drawing, and see what state 0–1–2 behavior they require.
- At first, the model appears to characterize the states of the device by the inputs available to the system. So, from this perspective, state 0 is clearly different from states 1 and 2. However, if we look at the state 1–2 transaction, we see that it is symmetric with respect to the two states.
- In principle, there is no reason why a program should not decide to do simple mouse tracking whilst in state 2 and drag things about in state 1. That is, there is no reason until you want to type something!
- The way we can tell state 1 from state 2 is by the activity of the user. State 2 requires a button to be pressed, whereas state 1 is one of relative relaxation (whilst still requiring hand–eye coordination for mouse movement).
- There is a similar difference in tension between state 0 and state 1.
- It is well known that Fitts' law has different timing constants for different devices.
- Recall that Fitts' law says that the time taken to move to a target of size S at a distance.
  D is:

  a +b log2(D/S +1)
- The constants a and b depend on the particular pointing device used and the skill of the user with that device. However, the insight given by the three-state model is that these constants also depend on the device state. In addition to the timing, the final accuracy may be affected.
- Experiments to calculate Fitts' law constants in states 1 and 2 have shown that these differences do exist.

4. **Explain in details of organizational issues.**

The organizational issues that affect the acceptance and relevance of information and communication systems. These factors often sit 'outside' the system as such, and may involve individuals who never use it.

- **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 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.

- **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. However, the official lines of authority and information tend to flow up and down through line management. New communications media may challenge and disrupt these formal managerial structures.

- The physical layout of an organization often reflects the formal hierarchy:
- An email system has no such barriers; it is as easy to 'chat' to someone in another department as in your own.
- Face-to-face conversation, the manager can easily exert influence over a subordinate:

Technology can be an important vector of social change, but if violent reaction is to be avoided, the impact of the technology must be assessed before it is introduced. In the short term, solutions must be carefully matched to the existing social and organizational structures.

- **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 their functional colleagues. Alternatively, distributed groupware can allow the true home-based teleworker to operate on similar terms to an office-based equivalent. The ecological and economic advantages of such working practices are now becoming well established, and it seems that communications and CSCW technology can overcome many of the traditional barriers.

- **Free rider problem.**

In economics, the free rider problem occurs when those who benefit from resources, goods, or ervices do not pay for them, which results in an under-provision of those goods or services. The free rider problem is the question of how to limit free riding and its negative effects in these situations. The free rider problem may occur when property rights are not clearly defined and imposed.

The free rider problem is common among public goods. These are goods that have two characteristics: non-excludability—non-paying consumers cannot be prevented from using it—and non-rivalry—when you consume the good, it does not reduce the amount available to others. The potential for free riding exists when people are asked to voluntarily pay for a public good.

- **Critical mass**

A critical mass is the smallest amount of fissile material needed for a sustained nuclear chain reaction. The critical mass of a fissionable material depends upon ies (specifically, the nuclear fission cross-section), its density, its shape, its enrichment, its purity, its temperature, and its surroundings. The concept is important in nuclear weapon design.

- 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.

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.

- Evaluating the benefits

We have seen several problems that can arise from the mismatch between information systems and organizational and social factors.

- ✓ 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.
- ✓ The benefits are difficult to quantify, but, over time, it has become clear that the competitive edge of information technology is necessary for survival in the modern world.

5. Explain in details custom methodology and open system task analysis (OSTA).

CUSTOM methodology

CUSTOM is a socio-technical methodology designed to be practical to use in small organizations [200]. It is based on the User Skills and Task Match (USTM)
approach, developed to allow design teams to understand and fully document user requirements [219]. 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 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 [116] is an alternative socio-technical approach, which 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.

OSTA uses notations familiar to designers, such as data flow diagrams and textual descriptions

6. **Explain in details soft systems methodology.**

The socio-technical models we have looked at focus on identifying requirements from both human and technical perspectives, but they assume a technological solution is being proposed. 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. SSM was developed by Checkland.

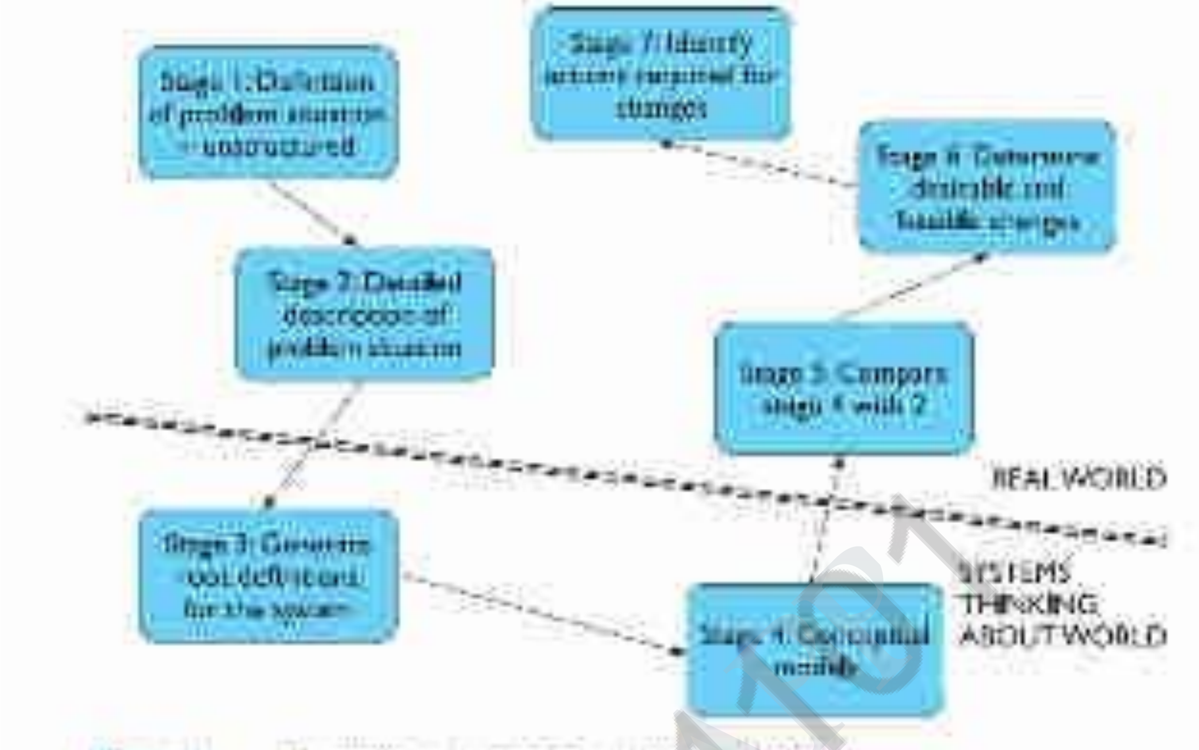

Stage 1: Definition of problem situation – unstructured
Stage 2: Detailed description of problem situation
Stage 3: Generate root definitions for the system
Stage 4: Conceptual models
Stage 5: Compare stage 4 with 2
Stage 6: Determine desirable and feasible changes
Stage 7: Identify actions required for changes
REAL WORLD
SYSTEMS THINKING ABOUT WORLD

Figure 1 The seven stages of soft systems methodology.

to help designers reach an understanding of the context of technological developments and the influences and concerns that exist within the system under consideration. SSM has seven stages (see Figure 1). A distinction is made between the 'real-world' stages (1–2, 5–7) and the systems stages (3–4).

We will outline the stages here but will focus on those that help capture requirements. The first stage of SSM is the recognition of the problem and initiation of analysis. This is followed by a detailed description of the problem situation: developing a rich picture. This will include all the stakeholders, the tasks they carry out and the groups they work in, the organizational structure and its processes and the issues raised by each stakeholder. Any knowledge elicitation techniques can be used to gather the information to build the rich picture, including observation (and video and audio recording), structured and unstructured interviews and questionnaires, and workshops incorporating such activities as role play, simulations and critical incident analysis. In general, less structured approaches are used initially to avoid artificially constraining the description. The rich picture can be in any style – there are no right or wrong answers – but it should be clear and informative to the designer. Certain conventions are widely accepted, however. Speech balloons are used to represent stakeholder issues; crossed swords represent conflicts within the system; and the eye represents external influences or observers. Figure 2 shows an example of a rich picture with all of these elements.

Rich pictures are in themselves useful tools to aid understanding of a situation.

The rich picture is informal and relatively intuitive. It captures succinctly the potentially conflicting interests of the various stakeholders and the other influences on a design situation. It provides an understandable summary of the designer's understanding that can be easily checked with stakeholders, and it can even be developed collaboratively with stakeholders as part of the consultation process – allowing all parties to contribute to the rich picture sketch. These benefits have led to a number of researchers in HCI proposing their use (outside the full SSM methodology) to inform the design process . At the next stage in SSM we move from the real world to the systems world and attempt to

tions for the system, which define the essence of what the system is

	about. There may be several root definitions of a system, representing different stakeholder perspectives, for example. Root definitions are described in terms of specific elements, summarized using the acronym, CATWOE: Clients – those who receive output or benefit from the system. Actors – those who perform activities within the system. Transformations – the changes that are effected by the system. This is a critical part of the root definition as it leads to the activities that need to be included in the next stage. These ‘transform’ the inputs of the system into the required outputs. Weltanschauung – (from the German) meaning world view. This is how the system is perceived in a particular root definition. Owner – those to whom the system belongs, to whom it is answerable and who can authorize changes to it. Environment – the world in which the system operates and by which it is influenced.

7 Explain in details effective technical and human implementation of computer-based systems (ethics).

ETHICS is a method developed by Enid Mumford within the socio-technical tradition, but it is distinct in its view of the role of stakeholders in the process. In the 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. The design groups then address the following issues and activities:

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.

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 (desirable to meet).

The final stages of the ETHICS approach focus on the actual design and evaluation of the system. Necessary organizational changes are designed alongside the technical system. These are then specified in detail, implemented and evaluated.

The ETHICS approach attempts to reach a solution that meets both user and task requirements by having specialist teams negotiate objectives and rank potential solutions.

The emphasis is on reaching a solution that ranks highly on job satisfaction to ensure that the solution is acceptable.

It ensures participation at every stage through representative design teams, but can be expensive and time consuming to carry out. However, the benefits in terms of job

er productivity may balance out the initial investment

8. Explain in details face-to-face communication.

Face-to-face contact is the most primitive form of communication – primitive, that is, in terms of technology. If, on the other hand, we consider the style of communication, the interplay between different channels and productivity, we instead find that face-to- face is the most sophisticated communication mechanism available.

- Transfer effects and personal space
  - ✓ We expect social norms from face We expect social norms from face-to-face communication when we come to use computermedicated forms of communication
  - ✓ People are very adaptable and can learn new norms to go with new media (ex. Walkie- talkie).
  - ✓ However, success with new media is often dependent on whether the participants can use their existing norms.
  - ✓ If conversation is broken, what is the true problem?
  - ✓ Direction is also important. We can accept people closer to us if they are at our sides or behind than if we are facing them. => Passenger on tube train forced to be close, will incline their faces at an angle to one another whilst talking.
  - ✓ Personal space also differ across cultures. Similar problem can occur in a video conference, ex. Wide focus, high level of zoom, camera position, different size of monitors. Even 'glass wall' makes precise distance less important, which could have a positive effect during cross-cultural meeting.
- Eye contact and gaze
  - ✓ Eye tell us that our colleague is listening or not; they can convey interest, confusion or boredom.
  - ✓ People who look away when you look at them may seem shifty and appear to be hiding something.
  - ✓ Relative frequency of eye contact and who 'give way' from direct eye contact is closely linked

to authority and power.

  - ✓ But if the camera is strapped to the top of monitor, both participants will look as their eyes are slightly dropped.
- Gestures and body language
  - ✓ We use our hand to indicate items of interest.
  - ✓ Video connection may not sufficient to read movement. Ex. Conversation as 'let's move this one there'. The 'this' and 'there' are indicated by gestures (or eyegaze). This is called deictic reference.
  - ✓ Even participants are in the same room, electronic equipment can interfere with the body language used in normal face-to-face communication. Ex. Large monitor, using keyboard and screen. => most computer-supported meeting rooms recess monitors into the desk to reduce these problems.
- Back channels, confirmation and interruption
  - ✓ The nods, grimaces, shrugs of the shoulder and small noises (ex. uh…er…) are called back channels.
  - ✓ Back channels means that the speaker can afford to be slightly vague, adding details until it obvious that the listener understand.
  - ✓ Even video communications are used, but we lost some body movement and gesture.
  - ✓ Audio-only links (ex. Telephone) have rely on purely verbal back channel responses.
  - ✓ Text-based communication has no back channels.
- Turn-taking
  - ✓ 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.
  - ✓ What is a problem if we talk during long-distance, satellite-based communications due to the

time lags.

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.

PART- A

1 **List out the layers of the mobile ecosystem.**

1.Services, 2.Applications
3.Application Frameworks
4.Operating Systems
5.Platforms
6.Devices
7.Aggregators
8.Networks
9.Operators.

2 **What are the services?**

Services include tasks such as accessing the Internet, sending a text message, or being able to get a location—basically, anything the user is trying to do.

3 **What do you mean by 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.

4 **What is the use of application layer?**

Application frameworks are used to create applications, such as a game, a web browser, a camera, or media player. Although the frameworks are well standardized, the devices are not. The largest challenge of deploying applications is knowing the specific device attributes and capabilities.

5 **What is the need of Application Framework layer?**

The first layer that you have any control over is the choice of application framework. Application frameworks often run on top of operating systems, sharing core services such as communications, messaging, graphics, location, security, authentication, and many others.

6 **What is Mobile platform?**

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.

7 **What are all the types of Mobile Platforms?**

Licensed:- Licensed platforms are sold to device makers for nonexclusive distribution on devices. Eg.*Java Micro Edition (Java ME),Binary Runtime Environment for Wireless (BREW),Windows Mobile,LiMo*

Proprietary:- Proprietary platforms are designed and developed by device makers for use on their devices.Eg.*Palm,BlackBerry,iPhone*

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, but they are increasingly gaining traction with device makers and developers. Android is one of these platforms.

8 **What is the use of Mobile Application medium type?**

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. The technical capabilities and capacity of the publisher also factor into which approach to take.

9 **What is Web Widget?**

A mobile web widget is a standalone chunk of HTML-based code that is executed by the end user in a particular way.

10 **Write about pros and cons of the Mobile web applications.**

Pros:-

- **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 and offline use.**
- **Content is accessible on any mobile web browser. Cons:-**
- **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, filesystem access, camera, and so on.**

11 **Give short notes on Immersive Full Screen Applications.**

The immersive full-screen applications is like a game, a media player, or possibly even a single-screen utility. These applications are meant to consume the user's focus, often doing so by filling the entire screen, and leaving no trace of the device user interface to distract the user. Again, the majority of mobile engagement occurs when the user has idle periods of time; the immersive context is typical in most entertainment applications, one of the most popular mobile content areas.

12 **What is the use of Productivity Application Context?**

The productivity application context is used for content and services that are heavily task-based and meant to increase the users' sense of efficiency. With these types of applications, we can assume that the users are more committed to accomplishing a particular goal, like managing content such as messages, contacts, or media, but we should still assume that they are doing so during idle periods.

13 **List down the disciplines of mobile Information architecture.**

- **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.

14 **What is the use of 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.

15 **Why Wireframes required?**

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. they also serve to separate layout from visual design, defining how the user will interact with the experience.

16 **Write about different types of Mobile Prototyping.**

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

Context prototype-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.

HTML prototypes-This is a prototype that you can actually load onto a device and produce the nearest experience to the final product, but with static dummy content and data. It takes a little extra time, but it is worth the effort.

17 **Define Subpixels.**

A subpixel is the division of each pixel into a red, green, and blue (or RGB) unit at a microscopic level, enabling a greater level of antialiasing for each font character or glyph. The addition of these RGB subpixels enables the eye to see greater variations of gray,
creating sharper antialiasing and crisp text.

18 **What is Pixel Density?**

The pixel density is determined by dividing the width of the display area in pixels by the width of the display area in inches. As this applies to mobile devices, the higher the density of pixels, the sharper the screen appears to the naked eye. This guideline especially applies to type, meaning that as text is antialiased on a screen with a high density of tiny pixels, the glyph appears sharper to the eye.

19 **What are all the ways of defining 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 meaning. The secondary and tertiary colors are often complementary colors.

Adaptive:- An adaptive palette is one in which you leverage the most common colors present in a supporting graphic or image.

Inspired:- This is a design that is created from the great pieces of design you might see online 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. 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.

20 **List out the rules to be followed for Readability in mobile design?**

- **Use a high-contrast typeface**
- **Use the right typeface**
- **Provide decent leading (rhymes with "heading") or line spacing**
- **Leave space on the right and left of each line; don't crowd the screen**
- **Generously utilize headings**
- **Use short paragraphs**

21 **What is an Iconography?**

The most common form of graphics used in mobile design is icons. Iconography is useful to communicate ideas and actions to users in a constrained visual space. The challenge is
making sure that the meaning of the icon is clear to the user.

22 **List out some Design tool and interface toolkits for different mobile frameworks?**

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

PART- B

1 Write detailed notes on Platforms in Mobile Ecosystem? (**April/May 2019)**

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.

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

Java Micro Edition (Java ME)

Formerly known as J2ME, 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 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.

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.

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,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.

2 **Write in detail about Application Frameworks in Mobile Ecosystem? (April/May 2019)**

The first layer that you have any control over is the choice of application framework. 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.Most Java applications are purchased and distributed through the operator, but they can also be downloaded and installed via cable or over the air.

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.However BREW applications must go through a costly and timely certification process and can be distributed only through an operator.

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 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.Although WRTs are very interesting and provide access to some device functions using mobile web principles, I've found them to be more complex than just creating a simple mobile web app, as they force the developer to code within an SDK rather than just code a simple web app. And based on the number of mobile web apps written for the iPhone versus the number written for other, more full-featured WRTs, I don't think I'm alone in thinking this.

WebKit

With Palm's introduction of webOS, a mobile platform based on WebKit, and given its predominance as a mobile browser included in mobile platforms like the iPhone,Android, and S60, and that the vast majority of mobile web apps are written specifically for WebKit, I believe we can t as a mobile framework in its own right.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.

3 **Write short notes on SMS and Mobile Websites with their pros and cons? (April/May 2019)**

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 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.

Pros

The pros of SMS applications include:

- They work on any mobile device nearly instantaneously.
- They're useful for sending timely alerts to the user.
- They can be incorporated into any web or mobile application.
- They can be simple to set up and manage.

Cons

The cons of SMS applications include:

- They're limited to 160 characters.
- They provide a limited text-based experience.
- They can be very expensive.

Mobile Websites

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

Pros

The pros of mobile websites are:

- 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.
- They can load pages slowly, due to network latency.

4 **Discuss in detail about Mobile Web Widgets and Native Applications with their pros and cons?**

Mobile Web Widgets

Definitions

- A component of a user interface that operates in a particular way.
- A portable chunk of code that can be installed and executed within any separate HTMLbased web page by an end user without requiring additional compilation.
- A mobile web widget is a standalone chunk of HTML-based code that is executed by the end user in a particular way.

Basically, mobile web widgets are small web applications that can’t run by themselves,they need to be executed on top of something else. Opera Widgets, Nokia Web RunTime (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. Widgets, however, are not to be confused with the utility application context, a user experience designed around short, task-based operations.

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.

Native Applications

Native applications, which is actually a misnomer because a mobile web app or mobile web widget can target the native features of the device as well. These applications actually should be called “platform applications,” as they have to be developed and compiled for each mobile platform.

The most common of all platforms is Java ME (formerly J2ME). In theory, a device written as a Java ME MIDlet should work on the vast majority of feature phones sold around the world. The reality is that even an application written as a Java ME MIDlet still requires some adaptation and testing for each device it is deployed on.

In the smartphone space, the platform SDKs get much more specific. Although many smartphones are also powered by Java, an operating system layer and APIs added to allow developers to more easily offload complex tasks to the API instead of writing methods from scratch. In addition to Java, other smartphone programming languages include versions of C, C++, and Objective-C . Because platform applications sit on top of the platform layer, they can tap into the majority of the device features, working online or offline, accessing the location and the filesystem—and if there’s camera on the device, then you can probably do something with it as well. Hence the need for certification before the application is distributed,to ensure that no one distributes an application that steals a user’s personal data or maliciously uses the device to spread viruses.

However, if you exclude games, the majority of native applications in use today could be created with a little bit of XHTML, CSS, and JavaScript—in other words, a mobile web application, with little or no feature loss to the user. The advantage is that a mobile application can be developed more devices, require less testing, and be updated more transparently than a native application, which requires third-party certification and publishing in order to get on users' devices. All of these aspects are highly desired in the platform application space. The downside is that it requires a fast and capable mobile web browser that supports offline data and access to device features like location.

Pros

The pros of native applications include:

- They offer a best-in-class user experience, offering a rich design and tapping into device features and offline use.
- They are relatively simple to develop for a single platform.
- You can charge for applications.

Cons

The cons of native applications include:

- They cannot be easily ported to other mobile platforms.
- Developing, testing, and supporting multiple device platforms is incredibly costly.
- They require certification and distribution from a third party that you have no control over.
- They require you to share revenue with the one or more third parties.

5 **Explain Mobile web applications and Games in mobile application medium types? (April/May 2019) (April/May 2017)**

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. Web applications allow users to interact with content in real time, where a click or touch performs an action within the current view.

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.

In less than a year, we saw a strong unilateral move by all operators and devices makers to put better mobile web browsers in their phones that could leverage this new application medium. We have not seen such rapid innovation in mobile devices since the inclusion of cameras. The downside, of course, like all things mobile-web-related, is that not all devices support the capability to render mobile web applications consistently. However, we do see a prevalent trend that the majority of usage of the mobile web is coming from the devices with better browsers, in some markets by a factor of 7:1. So although creating a mobile web application might not reach all devices, it will reach the devices that create the majority of traffic.

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 and 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, filesystem access, camera, and so on.

Games

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. It is also different 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.

The reason games are relatively easy to port 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 ty 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.

6 **Write short notes on**

a. Application Context

b. Utility Context

c. Locale Context

d. Informative Applications

a. Application Context

Once the application medium is decided upon, it is time to look at the application context, or the appropriate type of application to present to the user in order for the user to process and understand the information presented and complete her goals. Where the application medium refers mostly to the technical approach of creating an application, the application context deals with the user experience. Actually context is the surroundings in which information is processed, and the application user experience is no different.

Applications can be presented in a variety of ways, ranging from a simple task-based utility to an experience meant to consume the user's focus and attention. There of course is no right or wrong direction—only what is best for your user. In fact, nothing says that you can't use multiple application contexts within the same application. it is best to present only one application context so as to avoid confusing the user. If you think it best for your app to mix contexts, then give the user the option to switch between them; for example, some smartphones allow for an orientation change, so if the device is rotated to landscape mode, your app switches from an informative view to a utility view, or maybe from a locale view to an immersive view.

b. Utility Context

The most basic application context is the utility, or a simple user experience metaphor that is meant to address short, task-based scenarios. Information is meant to be presented in a minimal fashion, often using the least amount of user input as possible. An example of a utility might be a calculator, weather forecast, unit conversion, stocks, world clock, and so on. In each of these cases, the user enters a small amount of information into the utility, like a simple equation, a city, or a stock symbol, and either by performing a small background task or fetching information online, the utility is able to present data to the user in the desired context. The goal of the utility is to give users at-a-glance information, therefore offering users a minimal design aesthetic, focusing the design around the content in view, and often using larger type and a sparse layout.

It would be easy to mistake utilities for widgets, given that widgets are a "component of a user interface that operates in a particular way." But utilities can be much more than widgets; they are not merely an extension of the user experience, but are applications in their own right that can establish their own look and feel separate from the established user experience.

Use utilities for short, simple tasks, at-a-glance information, when there is limited content to display, and when combined with an immersive context to create dual-mode applications.

c. Locale Context

The locale context is a newer application type that is still being defined by the mobile community, but we are certainly seeing some clear patterns of how to create locale applications. As more location information is being published online, and more devices add GPS to pinpoint the user's location, locale is becoming an excellent data point to pivot information around. For example, I can use location to display the Cafes nearest to my current location.

Locale applications almost always have at least one thing in common: a map on which they plot the requested data points visually. At the very least, they list items in order of distance, with the nearest item first and the farthest last. The user's goal is to find information relative to his present location, and content should always be designed with this in mind. When creating locale apps, it is important to ensure that the user's present location is always clearly identified, as well as a means of adding data to it. This could be another location, in the case of finding point-to-point directions, or it could be a keyword query to find people, places, or things nearby.

Use locale applications for location-based applications, applications that might contain a dynamic map, and listing multiple location-based points of interest.

d. Informative Applications

The informative application is an application context in which the one and only goal is to provide information, like a news site, an online directory, a marketing site, or even a mobile commerce site, where the key task of the user is to read and understand and it is not necessary to interact.

For example, remember that most mobile tasks are short and are often undertaken during brief idle periods. The user doesn't have much extra time and the task can be interrupted at any moment. In the case of a mobile news site, provide the user with the option to mark a page or story to be read later. With an online directory, allow the user to flag favorite entries. With a marketing site, allow users to enter the shortest possible contact information, like their phone number or email. And with a mobile commerce site, allow users to save items to a wishlist to review and purchase later.

The theme here is that although reading information is a simple task, it usually creates a complex chain of events that can be anticipated. With mobile applications, we want to avoid forcing the users to input too much information with their mobile devices, which is more difficult and takes more time than it would on another medium such as a desktop or laptop computers. Instead, we want to look for ways we can interconnect experiences, having users use the informative context to filter to the most desirable information when they have a moment, and allowing them to interact with it later, when they have more time, from the medium of their choice.

Use informative applications when users need information to gain an understanding or perform a physical task, for applications that can be used in multimedia contexts such as desktop and mobile, for information-heavy applications, and for marketing or promotional applications.

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

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.

eed does my application fill? What are people trying to do here? What is their primary goal? Once you understand that, it is a simple process of reverse engineering the path from where they want to be to where they are starting. Cut out everything else—your site or application doesn't need it. For example, to get some news and information on a mobile device, you need to first ask what the goal is. What is the need you are trying to fill? Then you need to apply context. Where are your users? What are they doing? Are they waiting for the bus? Do they have only a minute to spare? Or, do they have five minutes to spare? With these answers, you get your informationarchitecture.

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. They don't tell us anything about that content or what we can expect. Now, I would argue that good trigger labels are crucial in the Web as well, that we've become lazy and we assume so much about the user that we ignore the use of 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.Keep all your labels short and descriptive, and never try to be clever with the words you use to evoke action. The worst sin is to introduce branding or marketing into your information architecture; this will just serve to confuse and distract your users.

Site Maps

The first deliverable we use to define mobile information architecture is the site map.Site maps are a classic information architecture deliverable. They visually represent the relationship of content to other content and provide a map for how the user will travel through the informational space.

Mobile site maps aren't that dissimilar from site maps we might use on the Web. But there are a few tips specific to mobile that we want to consider.

Limit opportunities for mistakes

Imagine a road with a fork in it. We can go either left or right. The risk that we will make the wrong choice is only 50 percent, meaning that we have a better than good chance that we will get to where we want to go. But imagine three roads. Now our chances have dropped to 33 percent. Four roads drops your chances to 25 percent, and five roads takes you down to 20 percent. Now a 20 percent chance isn't great, but it isn't too bad, either.

Now think of your own website. How many primary navigation areas do you have?Seven? Eight? Ten? Fifteen? What risk is there to the users for making a wrong choice? If they go down the wrong path, they can immediately click back to where they started and go down another path, eliminating the wrong choices to find the right ones. The risks for making the wrong choice are minor.

Confirm the path by teasing content

After the users have selected a path, it isn't always clear whether they are getting to where they need to be. 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.

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.

A good architect's job is to create a map of user goals, not map out every technical contingency or edge case. Too often, process flows go down a slippery slope of adding every project requirement, bogging down the user experience with unnecessary distractions,rather than focusing on streamlining the experience.

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 serve to make our information space tangible and useful. But the purpose of wireframes is not just to provide a visual for our site map; they also serve to separate layout from visual design, defining how the user will interact with the experience. How do we lay out our navigation? What visual or interaction metaphors will we use to evoke action? What are the best ways to communicate and show information in the assumed context of the user? These questions and many more are answered with wireframes.

Prototyping

As mentioned before, wireframes lack the capability to communicate more complex,often in-place, interactions of mobile experiences. This is where prototypes come in. 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. The following sections discuss some ways to do some simple and fast mobile prototyping.

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.

Fig: *A paper prototype, where the interaction is nothing more than drawings on note cards*

Context prototype

The next step is creating a context prototype . 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. As you are traveling about, pull out your device and start looking your interface in the various contexts you find yourself currently in.

HTML prototypes

The third step is creating a lightweight, semifunctional static prototype using XHTML,CSS, and JavaScript, if available. This is a prototype that you can actually load onto a device and produce the nearest experience to the final product, but with static dummy content and data . It takes a little extra time, but it is worth the effort. With a static XHTML prototype, you use all the device metaphors of navigation, you see how much content will really be displayed on screen (it is always less than you expect), and you have to deal with slow load times and network latency. In short, you will feel the same pains your user will go through

8 **Write brief notes on the following mobile design elements. (April/May 2019)**
(April/May 2017)

Context

The 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?

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. They can provide not only great inspiration for design challenges, but justification for your design decisions later.

Message

The important design element is your message, or what you are trying to say about your site or application visually. One might also call it the "branding," although I see branding and messaging as two different things. Your message is the overall mental impression you create explicitly through visual design.

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?"

Your approach to the design will define that message and create expectations. A sparse, minimalist design with lots of whitespace will tell the user to expect a focus on content.A "heavy" design with use of dark colors and lots of graphics will tell the user to expect something more immersive.

Examples:

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.

Look and Feel

The concept of "look and feel" is an odd one, being subjective and hard to define. Typically, 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."

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.

Establishing a look and feel usually comes from wherever design inspiration comes from. 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.

Although a lot of elements go into making Apple's App Store successful, the most important design element is how it looks and feels. Apple includes a robust user interface tool that enables developers to use prebuilt components, supported with detailed Human Interface Guidelines (or HIG) of how to use them, similar to a pattern library.This means that a developer can just sit down and create an iPhone application that looks like it came from Apple in a matter of minutes. During the App Store submission process, Apple then ensures that the developer uses these tools correctly according to the HIG.

The look and feel can either be consistent with the stock user interface elements that Apple provides; they can be customized, often retaining the "spirit" of Apple's original design; or an entirely new look and feel can be defined—this approach is often used for immersive experiences.

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.Users focused on the headers, the navigation, the footer, or howcontent blocks are laid out, and so on. But their feedback got muddied with the "look and feel, the colors, and other design elements."

Different layouts for different devices

The second part of layout design is how to visually represent content. In mobile design,the primary content element you deal with the is navigation. Whether you are designing a site or app, you need to provide users with methods of performing tasks, navigating to other pages, or reading and interacting with content. This can vary, depending on the devices you support. There are two distinct types of navigation layouts for mobile devices: touch and scroll.

With touch, you literally point to where you want to go; therefore, navigation can be anywhere on the screen. But we tend to see most of the primary actions or navigation areas living at the bottom of the screen and secondary actions living at the top of the screen, with the area in between serving as the content area.

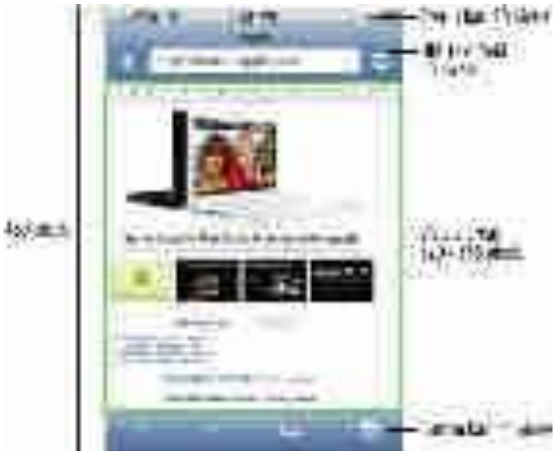


Fig: iPhone HIG, showing the layout dimensions of Safari on the iPhone

When designing for this type of device, the primary and often the secondary actions should live at the top of the screen. This is so the user doesn't have to press down dozens of times to get to the important stuff. When dealing with scroll navigation, you also have to make the choice of whether to display navigation horizontally or vertically. Visually, horizontally makes a bit more sense, but when you consider that it forces the user to awkwardly move left and right,it can quickly become a bit cumbersome for the user to deal with. There is no right or wrong way to do it, but my advice is just to try and keep it as simple as possible.

Fixed versus fluid

Another layout consideration is how your design will scale as the device orientation changes, for example if the device is rotated from portrait mode to landscape and vice versa. This is typically described as either being fixed (a set number of pixels wide), or fluid (having the ability to scale to the full width of the screen regardless of the device orientation).

Example layout of a scroll-based application, where the user had to press the D-pad past each link to scroll the page

Orientation switching has become commonplace in mobile devices, and your design should always provide the user with a means to scale the interface to take full advantage of screen real estate.

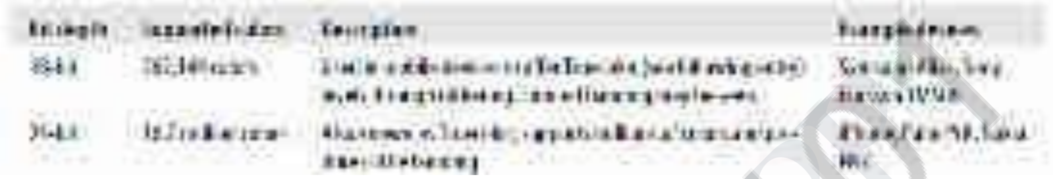
9 **Explain the concepts of Color, Typography and Graphics in mobile design elements.**

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 in black 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.

Different devices have different color depths.



[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]

The psychology of color

People respond to different colors differently. It is fairly well known that different colors produce different emotions in people, but surprisingly few talk about it outside of art school. Thinking about the emotions that colors evoke in people is an important aspect of mobile design, which is such a personal medium that tends to be used in personal ways. Using the right colors can be useful for delivering the right message and setting expectations.

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 meaning. The secondary and tertiary colors are often complementary colors that I select using a color wheel.

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 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.

Typography

The sixth element of mobile design is typography, which in the past would bring to mind the famous statement by Henry Ford:

Any customer can have a car painted any color that he wants so long as it is black.

As devices improved, so did their fonts. Higher-resolution screens allowed for a more robust catalog of fonts than just the device font. First, let's understand how mobile screens work.

Subpixels and pixel density

There seem to be two basic approaches to how type is rendered on mobile screens:using subpixel-based screens or having a greater pixel density or pixels per inch (PPI).A subpixel is the division of each pixel into a red, green, and blue (or RGB) unit at a microscopic level, enabling a greater level of antialiasing for each font character or glyph. The addition of these RGB subpixels enables the eye to see greater variations of gray, creating sharper antialiasing and crisp text.

Type options

Fortunately, today's mobile devices have a few more options than a single typeface, but the options are still fairly limited. Coming from web design, where we have a dozen or so type options, the limited choices available in mobile design won't come as a big surprise. when creating mobile designs for either web or native experiences, my advice is to stick with either the default device font, or web-safe fonts—your basic serif variants like Times New Roman and Georgia or sans-serif typefaces like Helvetica, Arial, or Verdana.

Font replacement

The ability to use typefaces that are not already loaded on the device varies from model to model and your chosen platform. Some device APIs will allow you to load a typeface into your native application. Some mobile web browsers support various forms of font replacement; the two most common are sIFR and Cufon. sIFR uses Flash to replace HTML text with a Flash representation of the text, but the device of course has to support Flash. Cufon uses JavaScript and the canvas element draws the glyphs in the browser, but the device of course needs to support both JavaScript and the canvas element. In addition, the @font-face CSS rule allows for a typeface file to be referenced and loaded into the browser, but a license for web use is usually not granted by type foundries.

Readability

The most important role of typography in mobile design is to provide the user with excellent readability, or the ability to clearly follow lines of text with the eye and not lose one's place or become disoriented.. This can be done by following these six simple rules:

Use a high-contrast typeface

Remember that mobile devices are usually used outside. Having a high-contrast typeface with regard to the background will increase visibility and readability.

Use the right typeface

The type of typeface you use tells the user what to expect. For example, a sans-serif font is common in navigation or compact areas, whereas serif typefaces come in handy for lengthy or dense content areas.

Provide decent leading (rhymes with "heading") or line spacing

Mobile screens are often held 10–12" away from the eye, which can make tracking each line difficult. Increase the leading to avoid having the users lose their place.

Leave space on the right and left of each line; don't crowd the screen

Most mobile frameworks give you full access to the screen, meaning that you normally need to provide some spacing between the right and left side of the screen's edge and your text—not much, typically about three to four character widths.

Generously utilize headings

Break the content up in the screen, using text-based headings to indicate to the user what is to come. Using different typefaces, color, and emphasis in headings can also help create a readable page.

Use short paragraphs

Like on the Web, keep paragraphs short, using no more than two to three sentences per paragraph.

Graphics

The final design element is graphics, or the images that are used to establish or aid a visual experience. Graphics can be used to supplement the look and feel, or as content displayed inline with the text.For example, in figure, you can see Ribot's Little Spender application for the iPhone and the S60 platform. The use of graphical icons in the iPhone experience helps to establish a visual language for the user to interact with to quickly categorize entries.On the S60 application, the wallet photo in the upper-right corner helps communicate the message of the application to the user.

Iconography

The most common form of graphics used in mobile design is icons. Iconography is useful to communicate ideas and actions to users in a constrained visual space. The challenge is making sure that the meaning of the icon is clear to the user. For example,looking at figure, you can see some helpful icons that clearly communicate an idea and some perplexing icons that leave you scratching your head.

Photos and images

Photos and images are used to add meaning to content, often by showing a visual display of a concept, or to add meaning to a design. Using photos and images isn't as common in mobile design as you might think. Because images have a defined height and width, they need to be scaled to the appropriate device size, either by the server,using a content adaptation model, or using the resizing properties of the device. In the latter approach, this can have a cost in performance. Loading larger images takes longer and therefore costs the user more.Using graphics to add meaning to a design can be a useful visual, but you can encounter issues regarding how that image will display in a flexible UI—for example, when the device orientation is changed.

10 Discuss about

a. Mobile Design Tools

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.

In table you can see each of the design tools and what interface toolkits are available for it.

rk	Design tool	Interface toolkits

Java ME	Photoshop, NetBeans	JavaFX, Capuchin
BREW	Photoshop, Flash	BREW UI Toolkit, uiOne, Flash
iPhone	Photoshop, Interface Builder	iPhone SDK
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	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

b. Designing for the Right Device

With the best possible experience at hand, take a moment to relish it. Remind yourself that you are working with a rapidly evolving medium and though it might not be possible for every user to experience things exactly the way you've intended, you've set the tone and the vision for how the application should look. 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. Now is the time to ask, "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. You will learn which devices you need to design for, what users really want, and how well your design works in the mobile context .This knowledge will help you develop your porting and/or adaptation strategy, the most expensive and riskiest part of the mobile equation. For example, if you know that 30 percent of your users have iPhones, then that is a market you can exploit to your advantage. 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 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.

c. 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. Of course, there are some devices by default in a horizontal orientation, and many smartphones that can switch between the two orientations, but most people use their mobile devices in portrait mode. This is a big shift in thinking if you are coming from interactive design, as up to this point, screens have been getting wider, not taller. we've become used to placing less-crucial information along the sides of web pages. In software, tasks flow from left to right. With vertical designs, the goal is to think of your design as a cascade of content from top to newspaper. The most contextual information lives at the top, and the content

consumes the majority of the screen. Any exit points live at the bottom. Mobile is no different.

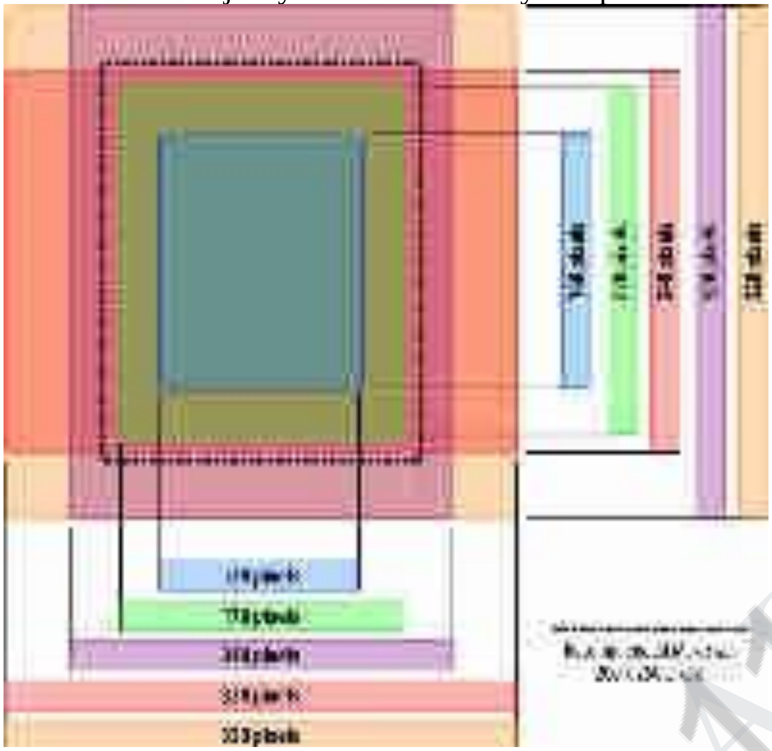


Fig: *Comparing the various screen sizes*

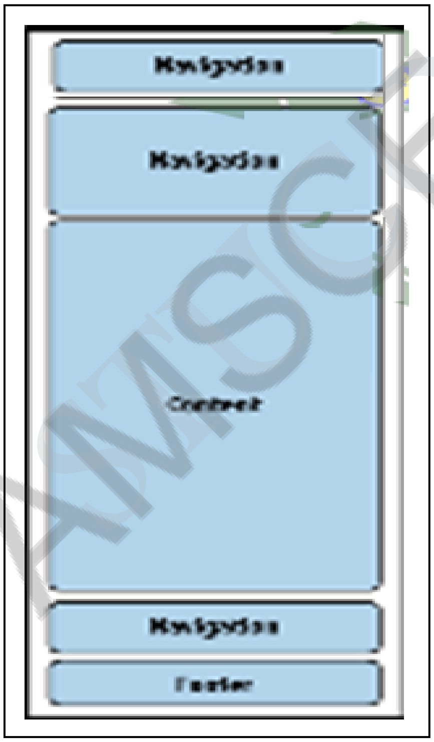


Fig: *The typical flow of information on mobile devices*

The greatest challenge to creating a design that works well on multiple screen sizes is filling the width. For content-heavy sites and applications, the width of mobile devices is almost the perfect readability, presenting not too many words per line of text. The problem is when you have to present a number of tasks or actions. The easiest and most compatible way is to present a stacked list of links or buttons, basically one action per line. It isn't the most effective use of space, but presenting too many actions on the horizontal axis quickly clutters the design—not to mention that it is more difficult to adapt to other devices.

Unfortunately, it isn't always reasonable to implement fluid or flexible designs that stretch to fit en. Although most mobile web browsers and device framework APIs enable it in principle, its execution across multiple devices is a little anticlimatic. Mobile websites usually employ a fixed-width layout for the lowest common denominator, and native applications are often resized for multiple screen sizes during development. As devices get larger, denser screens, you will see an increase in the use of touch, forcing the size of content to increase to fingertip size—typically 40 pixels wide and 40 pixels tall . This actually solves part of the horizontal axis problem, simply by making content larger for larger screens.

UNIT-V WEB INTERFACE DESIGN

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

PART-A

1 **List any five events available for cueing the user during a drag and drop interaction?**

- Page Load
- Mouse Hover
- Mouse Down
- Drag Initiated
- Drag Leaves Original Location.

2 **Define Grid.**

The grid is a handy tool for planning out interesting moments during a drag and drop interaction. It serves as a checklist to make sure there are no "holes" in the interaction.

3 **Explain Placeholder targeting and Midpoint boundary.**

Placeholder targeting - Most explicit way to preview the effect.

Midpoint boundary - Requires the least drag effort to move modules around.

4 **Explain Full-size module dragging and Ghost rendering.**

Full-size module dragging - Coupled with placeholder targeting and midpoint boundary detection, it means drag distances to complete a move are shorter.

Ghost rendering - Emphasizes the page rather than the dragged object. Keeps the preview clear.

5 **What do you mean by drag lens**?

A drag lens provides a view into a different part of the list that can serve as a shortcut target. It could be a fixed area that is always visible, or it could be a miniature view of the list that provides more rows for targeting. The lens will be made visible only during dragging. Example: Dragging the insertion bar while editing text on the iPhone.

6 **When a drop will be invalid in Yahoo! Mail?**

• The dragged object's icon becomes a red invalid sign.

• If over an invalid folder, the folder is highlighted as well.

7 **When a drop will be valid in Yahoo! Mail?**

• The dragged object's icon changes to a green checkmark.

• The drop target highlights.

8 **Write the good rule of thumb on drag initiation from the Apple Human Interface Guidelines.**

Your application should provide drag feedback as soon as the user drags an item at least three pixels. If a user holds the mouse button down on an object or selected text, it should become draggable immediately and stay draggable as long as the mouse remains down.

9 **What do you mean by 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.

areas where Drag and Drop may be employed.

	Module, List, Object, Action
11	**What do you mean by Object Selection?** On the desktop, the most common approach is to initiate a selection by directly clicking on the object itself. We call this selection pattern Object Selection. Object Selection is used for initiating a drag drop.
12	**What is meant by 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. Example: Yahoo! Mail Classic. Toggle Selection is used for selecting bookmarks for editing, deleting, etc.
13	**Define Collected Selection.** Collected Selection is a pattern for keeping track of selection as it spans multiple pages. 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. The "Delete Forever" action will operate on all 2785 conversations, not just the 25 selected on the page.
14	**Explain Hybrid Collection.** Hybrid Selection brings with it the best of both worlds. You can use the checkbox selection model as well as normal row selection. You get the benefit of explicit selection and simplified multiple selection that Toggle Selection brings. And you get the benefit of interacting with the message itself and direct object highlighting.
15	**Define 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.
16	**What do you mean by 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.
17	**What are the 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."
18	**Define Mystery Meat and Soft Mode.** **Mystery Meat** - It is a common anti-pattern that occurs when you have to hover over an item to understand how to use it. **Soft Mode** - If a mode is *soft* it is usually acceptable. By "soft" we mean the user is not trapped in the mode.
19	**Define Muttons.** Muttons (menu + button = mutton) are useful when there are multiple actions and we want one of the actions to be the default. Yahoo! Mail uses a mutton for its "Reply" button. It is a variation on Multi-Level Tools. Muttons are used to:

	• Provide a default button action ("Reply to Sender") • Provide a clue that there are additional actions. • Provide additional actions in the drop-down.
20	**Define overlays and inlays.** **Overlays -** Instead of going to a new page, a mini-page can be displayed in a lightweight layer over the page. **Inlays -** Instead of going to a new page, information or actions can be inlaid within the page.
21	**List the three specific types of overlays.** • Dialog Overlays • Detail Overlays • Input Overlays
22	**When should an overlay be used?** • Use an overlay when there may be more than one place a dialog can be activated from • Use an overlay to interrupt the process. • Use an overlay if there is a multi-step process.
23	**When should an inlay be used?** • Use an inlay when you are trying to avoid covering information on the page needed in the dialog. • 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.
24	**What are the Patterns that support virtual pages?** • Virtual Scrolling • Inline Paging • Scrolled Paging • Panning • Zoomable User Interface
25	**Compare paging and scrolling.** • 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 the users want to sort the data, 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. • If the content is really continuous, scrolling is more natural than paging. • 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.

26 **List out the process flow patterns.**

- Interactive Single-Page Process
- Inline Assistant Process
- Configurator Process
- Overlay Process
- Static Single-Page Process

PART- B

1 **What is the purpose of drag and drop? Briefly explain drag and drop module. (April/May 2019) (April/May 2017)**

One of the most useful purposes of drag and drop is to allow the user to directly place objects in the desired place on the page. A typical pattern is Drag and Drop Modules on a page. **Example**: Netvibes (Figure 2-3).

Considerations

Netvibes allows its modules to be rearranged with drag and drop. 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:

a. Placeholder target
b. Insertion target

a. Placeholder target

Netvibes uses a placeholder (hole with dashed outline) as the drop target. The hole serves as a placeholder and always marks the spot that the dragged module will land when dropped. For module drag and drop, the other modules only slide up or down within a vertical column to make room for the dragged module.

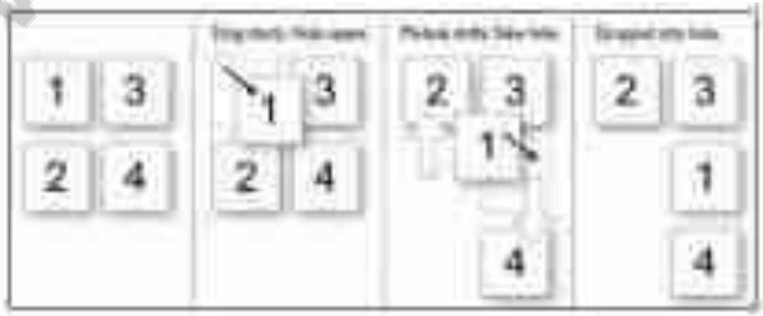


Figure 2-3. A placeholder target always shows where the dragged module will end after the drop; module 1 is being dragged from the upper right to the position between module 3 and 4

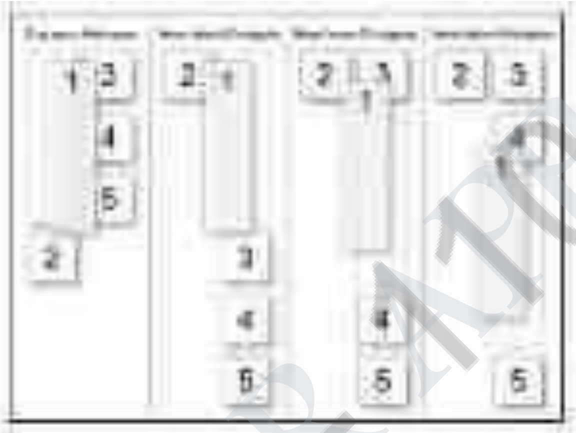
Boundary-based placement

Approach by Netvibes

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.

Approach by iGoogle

A more desirable approach is that taken by iGoogle. Instead of basing the drag on the *title bar*, iGoogle calculates the placeholder targeting on the dragged-over object's *midpoint*. As Figure 2-11 illustrates, module 1 is dragged from the first column to the second column, the placeholder moves above module 3. As module 1 is dragged downward, the placeholder moves below 3 and 4 as the bottom of module 1 crosses their midpoints.



[illegible]

b. Insertion target

While the module is dragged, the page remains stable. No modules move around. Instead an insertion bar marks where the module will be placed when dropped. This technique is illustrated in Figure 2-14. When module 1 is dragged to the position between 3 and 4, an insertion bar is placed there. This indicates that if 1 is dropped, then 4 will slide down to open up the drop spot.

Of the various approaches for Drag and Drop Modules, iGoogle combines the best approaches into a single interface:

- ***Placeholder targeting***
  - Most explicit way to preview the effect.
- ***Midpoint boundary***
  - Requires the least drag effort to move modules around.
- ***Full-size module dragging***
  - Coupled with placeholder targeting and midpoint boundary detection, it means drag distances to complete a move are shorter.
- ***Ghost rendering***

  Emphasizes the page rather than the dragged object. Keeps the preview clear.

2 **Explain the various types of selection patterns with examples.**

- **a. Toggle Selection**
  - a. Checkbox or control-based selection.
- **b. Collected Selection**
  - a. Selection that spans multiple pages.
- **c. Object Selection**
  - a. Direct object selection.
- **d. Hybrid Selection**

○ Combination of Toggle Selection and Object Selection.

a. Toggle Selection

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).

Example:

Considerations

- 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.

Scrolling versus paging

Yahoo! Mail uses a scrolled list to show all of its mail messages. While not all messages are visible at a time, the user knows that scrolling through the list retains the currently selected items. Sometimes the need for clarity of selection will drive the choice between scrolling and paging.

b. Collected Selection

Toggle Selection is great for showing a list of items on a single page. Collected Selection is a pattern for keeping track of selection as it spans multiple 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. 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 (Figure 3-9).

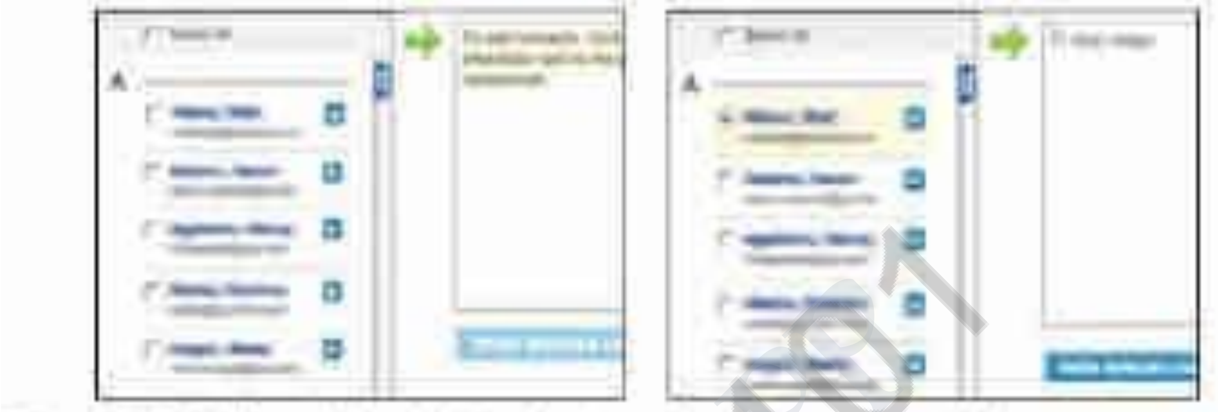


Figure 3-9. LinkedIn provides a holding place for saving selections across multiple pages

The list of potential invitees is shown in a paginated list on the left hand side. Clicking the checkbox adds them to the invite list. The invite list becomes the place where selected contacts across multiple pages are remembered. Any name in the invite list can be removed by clicking the "X" button beside it. Once the complete list of invitees is selected, clicking the "Invite selected contacts" sends each selected contact a LinkedIn invitation.

Collected Selection and actions

In order to support Collected Selection, Yahoo! Photos introduced the concept of a "tray" into the interface On any page, photos can be dragged into the tray. The tray keeps its contents as the user moves from page to page. So, adding a photo from page one and three more from page four would yield four items in the tray. As a nice touch, the tray would make itself visible (by sliding into view) even when the user was scrolled down below the fold.

Figure 3-10. Yahoo! Photos used a "tray" to implement a form of Collected Selection; the confusing aspect was which actions in the menu operated on the tray versus the photos selected on the page

c. Object Selection

Object Selection, is when selection is made directly on objects within the interface.

Example: Laszlo's WebTop mail

- It allows the user to select messages by clicking anywhere in the row.
- The result is that the whole row gets highlighted to indicate selection

Considerations

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.

Flickr is a simple example of the keyboard being used to extend the selection in a web application.

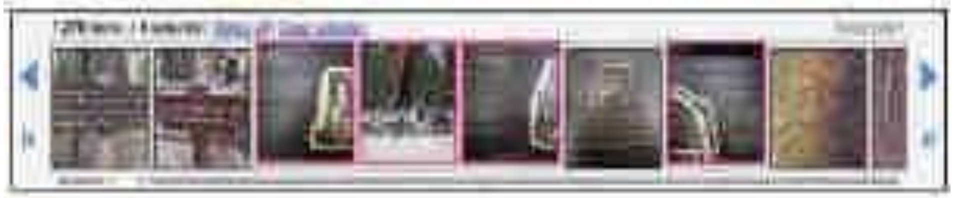


Figure 3-12. Flickr allows for discontinuous selection by using the Command/Control key to extend selection

d. Hybrid Selection

Hybrid Selection brings with it the best of both worlds. You can use the checkbox selection model as well as normal row selection. You get the benefit of explicit selection and simplified multiple selection that Toggle Selection brings.

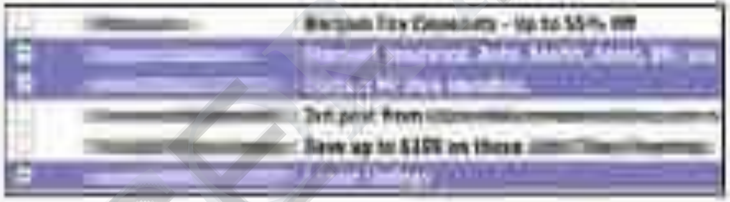


Figure 3-17. Yahoo! Mail now uses a hybrid approach: it incorporates both the Toggle Selection and the Object Selection patterns (messages can be selected with checkboxes or by clicking on the row); Toggle Selection selects the message without loading the message in the viewing pane

3 **Briefly explain the various contextual tools with examples.**

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:

a. **Always-Visible Tools**
   -Place Contextual Tools directly in the content.
b. **Hover-Reveal Tools**
   -Show Contextual Tools on mouse hover.
c. **Toggle-Reveal Tools**
   -A master switch to toggle on/off Contextual Tools for the page.
d. **Multi-Level Tools**
   -Progressively reveal actions based on user interaction.
e. **Secondary Menus**
   -Show a secondary menu (usually by right-clicking on an object).

a. Always-Visible Tools

The simplest version of Contextual Tools is to use Always-Visible Tools.

Example : Digg

Clear call to action

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. An alternate approach would be to hide them and show them on mouse hover.

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

Gmail provides a single Always-Visible Tool in its list of messages—the star rating—for flagging emails (Figure 4-5). 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.

Figure 4-5. Google Mail uses Contextual Tools to flag favorites

b. Hover-Reveal Tools

Instead of making Contextual Tools always visible, we can show them on demand. One way to do this is to reveal the tools when the user pauses the mouse over an object.

Example: 37 Signal's Backpackit

- To-do items may be deleted or edited directly in the interface.
- The tools to accomplish this are revealed on mouse hover.

Considerations

The gray bar on the left is a nice visual reinforcement for the interaction. By allowing the tools to "cut" into the sidebar, the designers draw your eye to the available tools. The light yellow background draws attention to the to-do item being acted on.

Visual noise

Showing the items on hover decreases the visual noise in the interface.

Discoverability

A serious design consideration for Hover-Reveal Tools is just how discoverable the additional functionality will be. In the earlier Backpackit example (Figure 4-8), 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

Sometimes 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."

Anti-pattern: Hover and Cover

Hover and Cover is a common anti-pattern that occurs when exposing an overlay on hover and hiding important context or further navigation.

Anti-pattern: Mystery Meat

Mystery Meat is a common anti-pattern that occurs when you have to hover over an item to understand how to use it.

c. Toggle-Reveal Tools

Toggle a tool mode for an area or page when the actions are not the main flow, but you want to provide the most direct way to act on these objects when the need arises.

Keep the transition between display and edit as seamless as possible to provide a "soft mode" for editing.

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

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 Basecamp, 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.

d. Multi-Level Tools

Contextual Tools can be revealed progressively with Multi-Level Tools.

Example: 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

Considerations

Songza reveals the four options "play", "rate", "share", and "add to playlist" after the user clicks on a song title. Hovering over "share" or "rate" reveals a secondary set of menu items (Figure 14-21, center).

Radial menus

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 menu easy to navigate since the revealed menu items are all equally close at hand (recall Fitts's Law).

Downside

- 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.

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.

r

Picnik is an online photo-editing tool that integrates with services like Flickr. In all, there are six sets of tools, each with a wide range of palette choices. Picnik uses Multiple-Level Tools to expose additional functionality. By wrapping the photo with tools in context and progressively revealing the levels of each tool, Picnik makes editing straightforward

Figure 4.22: Picnik wraps layers of Contextual Tools around the image being edited

Muttons

Another variation on Multi-Level Tools is the “mutton” (menu + button = mutton). Muttons are useful when there are multiple actions and we want one of the actions to be the default. Yahoo! Mail uses a mutton for its “Reply” button

Muttons are used to:

- Provide a default button action (“Reply to Sender”)
- Provide a clue that there are additional actions
- Provide additional actions in the drop-down

Anti-pattern: Tiny Targets

Providing Tiny Targets makes interaction much more difficult. An alternate approach would be to always show a “more info” link. Clicking it could toggle the additional profile information. Alternatively, providing a larger target for the arrow would improve its findability and targeting.

Secondary Menu

Desktop applications have provided Contextual Tools for a long time in the form of Secondary Menus. These menus have been rare on the Web. Google Maps uses a secondary menu that is activated by a right-click on a route. It shows additional route commands

Considerations

Secondary Menus have not been common in web applications.

Conflict with browser menu

One problem is the browser inserts its own right-click menu. Replacing the menu in normal content areas can confuse users, as they may not know if the standard browser menu or the application-specific menu will be shown.

Discoverability

As a general rule, never put anything in the Secondary Menu that can't be accomplished elsewhere. Secondary Menus are generally less discoverable. More advanced items or shortcuts, however, can be placed in the Secondary Menu as an alternate way to accomplish the same task.

Accessibility

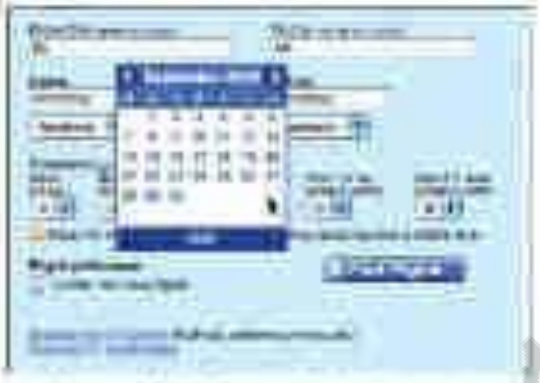
Right-click is not the only way to activate a Secondary Menu. You can activate the menu by holding down the mouse for about one second. This provides a more accessible approach to popping up a Secondary Menu.

4

Explain in detail about Dialog Overlays and inlays. (April/May 2019)

Overlays

Instead of going to a new page, a mini-page can be displayed in a lightweight layer over the page. 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



[illegible]

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.

Three specific types of overlays

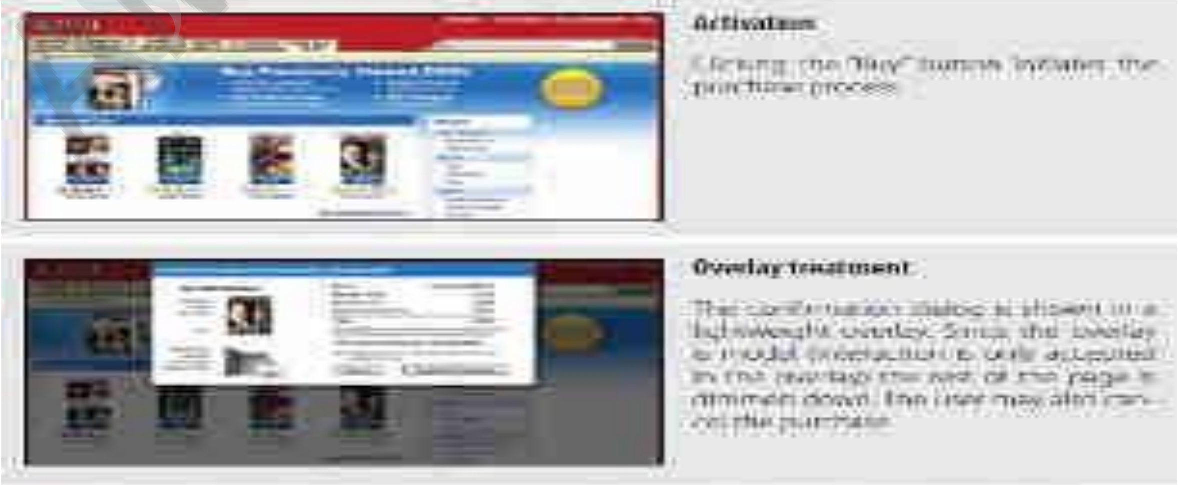
a. Dialog Overlays
b. Detail Overlays
c. Input Overlays.

a. Dialog Overlay

Dialog Overlays replace the old style browser pop ups.

Example: Netflix

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



Activation

Clicking the "Buy" button initiates the purchase process.

Overlay treatment

The confirmation dialog is shown in a lightweight overlay. Since the overlay is modal (interaction is only accepted in the overlay) the rest of the page is dimmed down. The user may also cancel the purchase.

Figure 5-5. *Netflix uses a lightweight pop up to confirm a previously viewed DVD purchase; in addition, it uses the Lightbox Effect to indicate modality*

Considerations

Because the overlay is a lightweight pop up, the confirmation can be displayed more lication has complete control over its look and placement.

Lightbox Effect

One technique employed here is the use of a Lightbox Effect. In photography a lightbox provides a backlit area to view slides. On the Web, this technique has come to mean bringing something into view by making it brighter than the background. In practice, this is done by dimming down the background.

Modality

Overlays can be modal or non-modal. A **modal overlay** requires the user to interact with it before she can return to the application. Sometimes overlays are **non-modal**.

Example: Netflix site.

When a DVD is added to the user's shipping list (queue), a confirmation overlay is shown (Figure 5-5). While it may appear that the only way to dismiss the overlay is by clicking the "Close" box in the upper-right corner, in reality the user can click anywhere outside the overlay (in the dimmed area) and the overlay will dismiss.

Staying in the flow

Overlays are a good way to avoid sending a user to a new page. This allows the user to stay within the context of the original page. However, since overlays are quick to display and inexpensive to produce, sometimes they can be tempting to use too freely, and in the process, may actually break the user's flow.

Anti-pattern: Idiot Boxes

This is a clear anti-pattern that should be avoided. We call these types of overlays Idiot Boxes. One of the clearest examples of Idiot Boxes is the way certain confirmation overlays were used in Yahoo! Photos.

b. Detail Overlay

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.

Example: Netflix - Information about a specific movie is displayed as the user hovers over the

movie's box shot

Activation

The overlay is displayed when the mouse hovers over a box shot. There is about a half second delay after the user pauses over a movie. The delay on activation prevents users from accidentally activating the overlay as they move the cursor around the screen. Once the user moves the cursor outside the box shot, the overlay is removed immediately. Removing it quickly gives the user a fast way to dismiss it without having to look for a "Close" box.

Anti-pattern: Mouse Traps

It is important to avoid activating the Detail Overlay too easily. We have seen usability studies that removed the delay in activation, and users reported that the interface was "too noisy" and "felt like a trap". We label this anti-pattern the Mouse Trap.

Anti-pattern: Non-Symmetrical Activation/Deactivation

When the user moves her mouse over the link, the overlay springs up immediately. The only way she can remove the overlay is by clicking the small close button in the upper right. Using Non-Symmetrical Activation/Deactivation is also a general anti-pattern that should be avoided. It should take the same amount of effort to dismiss an overlay as it took to open it.

Example: Yahoo! Foods (Figure 5-11).

To see all main ingredients for a recipe, the user clicked a red arrow. This activated an overlay with the ingredients. However, clicking on the arrow again did not collapse the overlay. Instead, the user had to click on the close button (red X).

Figure 5-11. Yahoo! Foods All Main Ingredients drop-down is activated by clicking on the arrow and can only be deactivated by clicking the close button (X)

Anti-pattern: Needless Fanfare

One of the advantages of a lightweight overlay is the ability to pop it up quickly. After a slight delay in activation (recall the half-second delay used by Netflix), you would not want or need the overlay to come up slowly.

But in the case of Borders online, this is precisely the approach taken (Figure 5-12).

- First the activation is immediate (no delay).
- Second, there's a needless animation that zooms the box up into place and then back down when the mouse moves away from a book.

eedless Fanfare is an anti-pattern to avoid. The animation takes a full second to complete. Instead of delaying before activation, it delays after activation.

c. 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 cards such as its gold card

Considerations

There are a few things to keep in mind when using Input Overlays.

Clear focus

The overlay creates focus on the given input field. Instead of seeing an ocean of inputs, the user is focused on just entering one field.

Display versus editing

Additionally, when the Input Overlay is shown, the prompt is displayed in exactly the same manner as when the overlay doesn't show. This is critical, as it makes the overlay feel even more lightweight. If the overlay prompt were bold, for example, the change would be slightly distracting and take the focus away from input. The only difference between the non-overlay field and the overlay version is a slightly thicker input field border. This draws the eye to the task at hand—input.

Anti-pattern: Hover and Cover

In Figure 5-15 ("Obscuring fields"), the "Name on Card" overlay hides the "Home Apt/Suite#" and "Home Phone Number Fields" fields below it.

5. **Briefly explain the following:**

(i) Detail Overlays (ii) Input Overlays **(April / May 2017)**

Detail Overlays

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.

Example:

Netflix - Information about a specific movie is displayed as the user hovers over the **movie's box shot**

Activation

The overlay is displayed when the mouse hovers over a box shot. There is about a half second delay after the user pauses over a movie. The delay on activation prevents users from accidentally activating the overlay as they move the cursor around the screen. Once the user moves the cursor outside the box shot, the overlay is removed immediately. Removing it quickly gives the user a fast way to dismiss it without having to look for a "Close" box.

Anti-pattern: Mouse Traps

It is important to avoid activating the Detail Overlay too easily. We have seen usability studies that removed the delay in activation, and users reported that the interface was "too noisy" and "felt like a trap". We label this anti-pattern the Mouse Trap.

Anti-pattern: Non-Symmetrical Activation/Deactivation

When the user moves her mouse over the link, the overlay springs up immediately. The only way she can remove the overlay is by clicking the small close button in the upper right. Using Non-Symmetrical Activation/Deactivation is also a general anti-pattern that should be avoided. It should take the same amount of effort to dismiss an overlay as it took to open it.

Example: Yahoo! Foods (Figure 5-11).

To see all main ingredients for a recipe, the user clicked a red arrow. This activated an overlay with the ingredients. However, clicking on the arrow again did not collapse the overlay. Instead, the user had to click on the close button (red X).

Figure 5-11. Yahoo! Foods All Main Ingredients drop-down is activated by clicking on the arrow and can only be deactivated by clicking the close button (X)

Anti-pattern: Needless Fanfare

One of the advantages of a lightweight overlay is the ability to pop it up quickly. After a slight delay in activation (recall the half-second delay used by Netflix), you would not want or need the overlay to come up slowly.

But in the case of Borders online, this is precisely the approach taken (Figure 5-12).

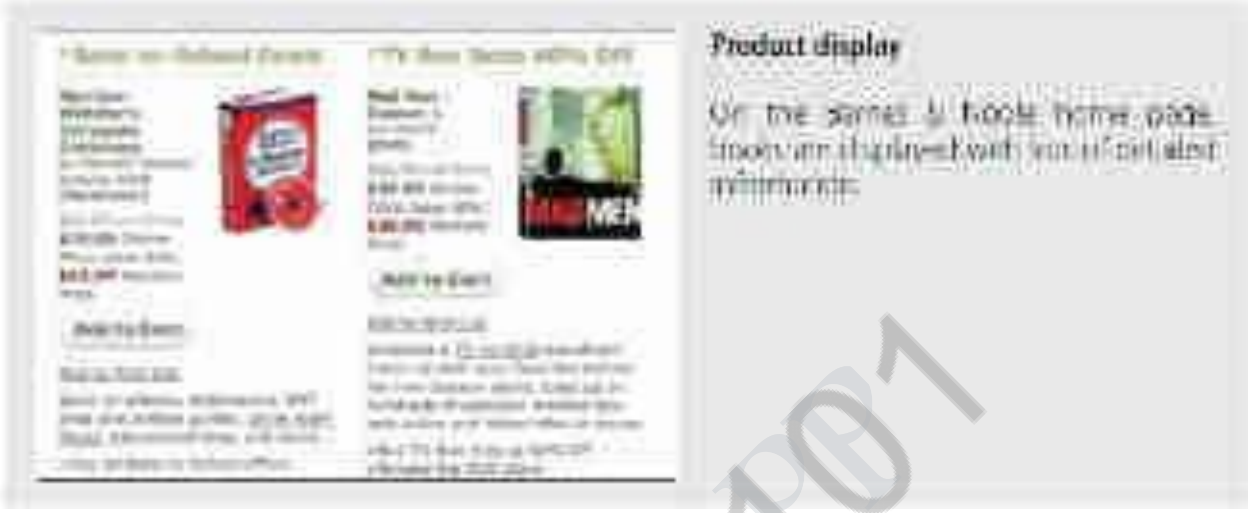
- First the activation is immediate (no delay).
- Second, there's a needless animation that zooms the box up into place and then back down when the mouse moves away from a book.

Needless Fanfare is an anti-pattern to avoid. The animation takes a full second to complete. Instead of delaying before activation, it delays after activation.

Anti-pattern: Hover and Cover

In Netflix example (Figure5-7), the overlay does not get in the way of moving to the next box shot. Even though it covers the neighboring box shot, moving the mouse outside the original one removes the overlay immediately, providing a clear path to get an overlay on the next box shot.

Compare this to the Detail Overlay provided by barnesandnoble.com



Product display

On the same B Noble home page, books are displayed with lots of detailed information.

Barnes & Noble uses a Detail Overlay in a completely useless manner. The overlay contains exactly the same information that the page already contained! Not only that, but the overlay almost completely obscures the other item displayed on the page. It also creates another common problem—the book image (dictionary) in the overlay is not positioned the same as the book image on the page (about 12 pixels difference). The shift causes the annoying illusion that the book is wiggling back and forth, which detracts from the experience.

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 cards such as its gold card

Considerations

There are a few things to keep in mind when using Input Overlays.

Clear focus

The overlay creates focus on the given input field. Instead of seeing an ocean of inputs, the user is focused on just entering one field.

Display versus editing

Additionally, when the Input Overlay is shown, the prompt is displayed in exactly the same manner as when the overlay doesn’t show. This is critical, as it makes the overlay feel even more lightweight. If the overlay prompt were bold, for example, the change would be slightly distracting and take the focus away from input. The only difference between the non-overlay field and the overlay version is a slightly thicker input field border. This draws the eye to the task at hand—input.

Anti-pattern: Hover and Cover

In Figure 5-15 (“Obscuring fields”), the “Name on Card” overlay hides the “Home Apt/Suite#” and “Home Phone Number Fields” fields below it.

6 **Explain in detail about list and detail 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

Considerations

Showing one item at a time focuses the reader on the current content.

Accordion: One-at-a-time expand

Accordion is an interface element that employs the List Inlay pattern to show only one open panel in a list at a time.

The following example to display real-time weather information in an accordion (Figure 6-6).

One panel at a time

This weather demo uses an accordion to display a single panel of weather-related information at a time.

In this example, the accordion allows only one weather pane to be visible at a time. When a pane is activated it slides open, simultaneously closing the previously open panel.

Accordions work best when:

- Each pane of content is independent
- Only one pane is visible at a time
- Each pane can be logically titled

Accordion: More than one pane visible at a time

We may want to allow multiple panes to be visible at a time if:

- Detail panes provide context for one another
- The vertical height of the accordion area is not fixed
- Different panels of content may be different heights

Parallel content

The Yahoo! Autos Car Finder tool (Figure 6-7) uses an accordion-style interaction for search filters that allows more than one pane to be open at a time. This choice makes sense because the decisions needed for one detail pane may be affected by the details of another pane. However, one problem with this specific implementation is the lack of information when a pane is closed. For example, no summary information is given for the "Price" tab.

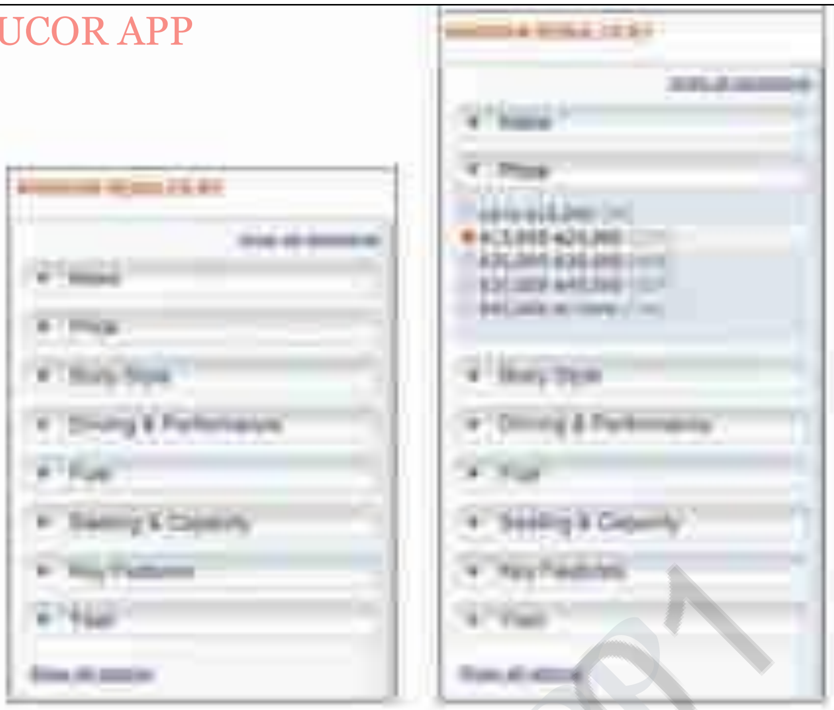


Figure 6-7. Yahoo! Autos places filter criteria in Accordion panes; when panes are hidden, no summary information is provided

Wine.com takes the opposite approach (Figure 6-8). Each area is initially open. All filters are displayed. As the user selects filters, the area collapses partially to show the results of filtering. This strategy accomplishes a few things. First, it gets rid of detail progressively—unnecessary information is hidden. Second, it invites the user to interact with the hidden content (with the “view all” link). Third, it provides summary information for panels that do not need to be opened as often.

Detail Inlay

A common idiom is to provide additional detail about items shown on a page. 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

Considerations

One of the more difficult things to do on most real estate sites is get a view of the house in context without having to navigate from page to page. The curb appeal, inside view, and backyard are all key factors in driving interest for a house. Knowing this, the team at Roost wanted to make it really easy to get to the photos quickly.

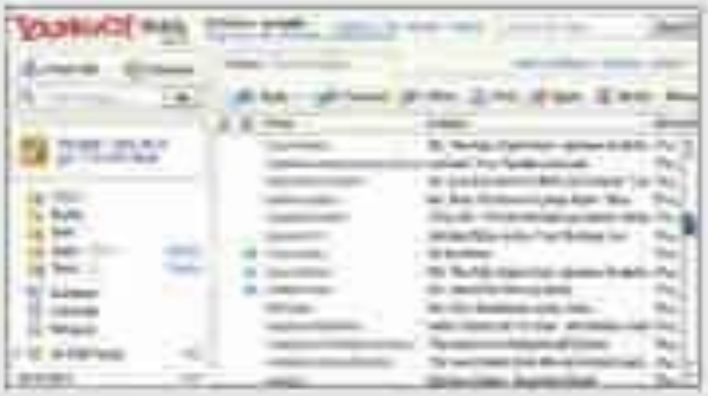
Combining inlays and overlays

Roost's solution was to combine several patterns.

1. It uses the Hover Reveal, a Contextual Tools pattern, to reveal a set of tools when the user hovers over a listing.
2. It uses the Detail Inlay pattern to show a carousel of photos when the user clicks on the "View photos" link.
3. It uses a Detail Overlay to blow up a thumbnail when clicked on.

Compare this to the traditional approach, one that requires the user to navigate from the listing

	page to a photo page and back again. The Roost team actually expended a Herculean effort in setting up this convenience, as it is dealing with hundreds of MLS listings with different contractual requirements for displaying real estate photos. The Roost team worked out the difficulties behind the scenes to create a nice user experience. **Note:** • Use Detail Inlay to provide additional information in context without hiding other information. • Use Detail Inlay to avoid the anti-pattern Hover and Cover. • Make it easy to dismiss the Detail Inlay.
7	**Briefly explain Virtual Scrolling, Inline Paging and Scrolled Paging.** **Virtual Scrolling** Every implementation of websites pagination was the key way to get to additional content. 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 (Figure 7-1). Another approach is to remove the artificial page boundaries created by paginating the data with Virtual Scrolling. In Yahoo! Mail, mail messages are displayed in a scrolled list that loads additional messages on demand as the user scrolls



Scroll completes

Messages are displayed based on where the user scrolled to.

Figure 7-2. Instead of showing just the messages that can be displayed on a single page, Yahoo! Mail dynamically loads messages as the user scrolls

Considerations

Virtual Scrolling turns the scrollbar into a “pagination control.” But instead of a page refresh, the messages are seamlessly brought into the message-list pane, giving the illusion of a larger virtual space.

Desktop-style applications

The more the web application looks and behaves like a desktop application, the more intuitive desktop idioms (like Virtual Scrolling) are to the user.

Example: Yahoo! Mail Web application looks very similar to desktop web applications, the expectation for scrolling already exists in the user’s mind.

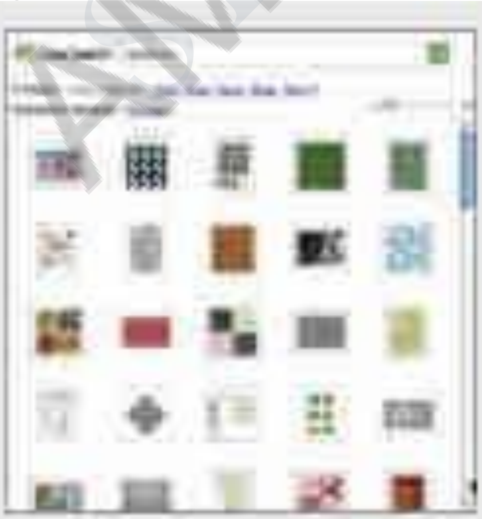
Loading status

if the loading is slow, it spoils the illusion that the data is continuous. Second, since the scrollbar does not give any indication of where users are located in the data, they have to guess how far down to scroll.

A remedy would be to apply a constantly updating status while the user is scrolling.

Progressive loading

Microsoft has applied Virtual Scrolling to its image search. However, it implements it in a different manner than Yahoo! Mail. Instead of all content being virtually loaded, the scrollbar reflects what has been loaded. Scrolling to the bottom causes more content to load into the page (Figure 7-3).



Scrolled list

12,500,000 image results are represented as a scrolled list. Obviously there is no way to accurately represent that many items in a list with a scrollbar. Notice the scrollbar shows size relative to the amount of data that has been loaded.

The Live Image Search approach is eliminating paging. It gets around the "Where am I?" issue by progressively loading and dynamically adapting the scrollbar to reflect how much has been loaded.

This type of Virtual Scrolling works well for search results since relevance starts dropping off the further you move through the data.

Example: An endless wall of pictures and uses a novel approach to a scrollbar control for Virtual Scrolling. PicLens is a Firefox add-on that allows viewing images from Google Search, Flickr, and other services to be displayed in the browser as a wall of photos that can be scrolled through

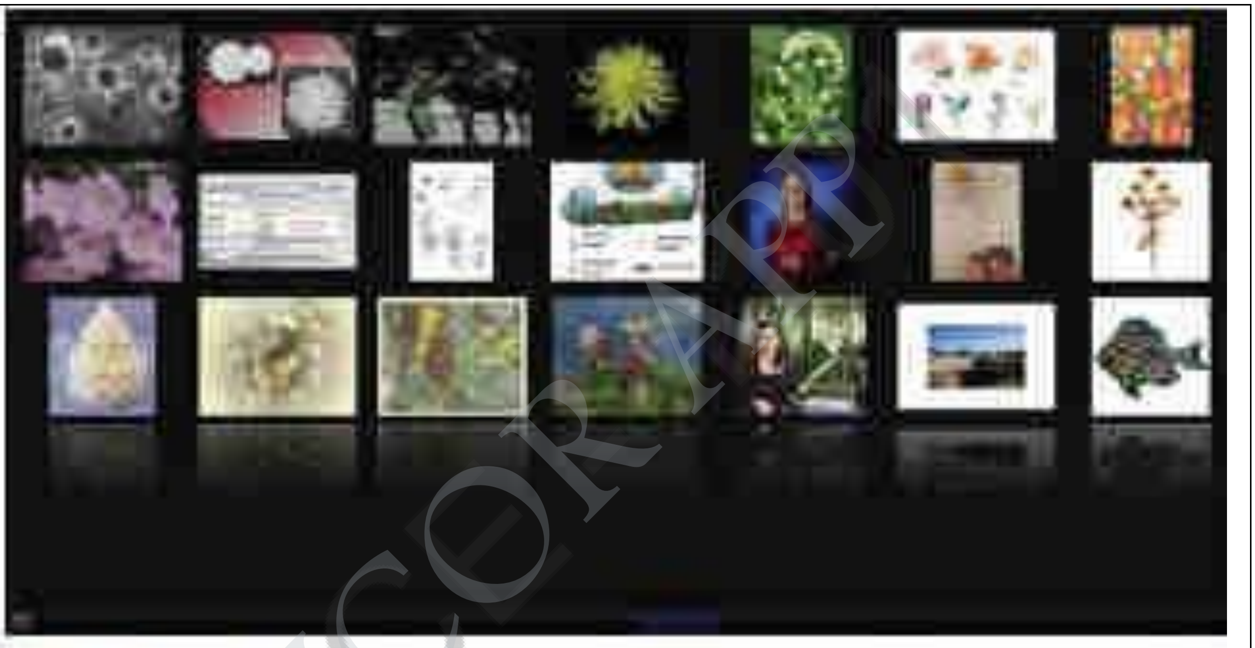


Figure 7-5. PicLens provides an endless wall of photos; the scroller at the bottom continues to expand with more content

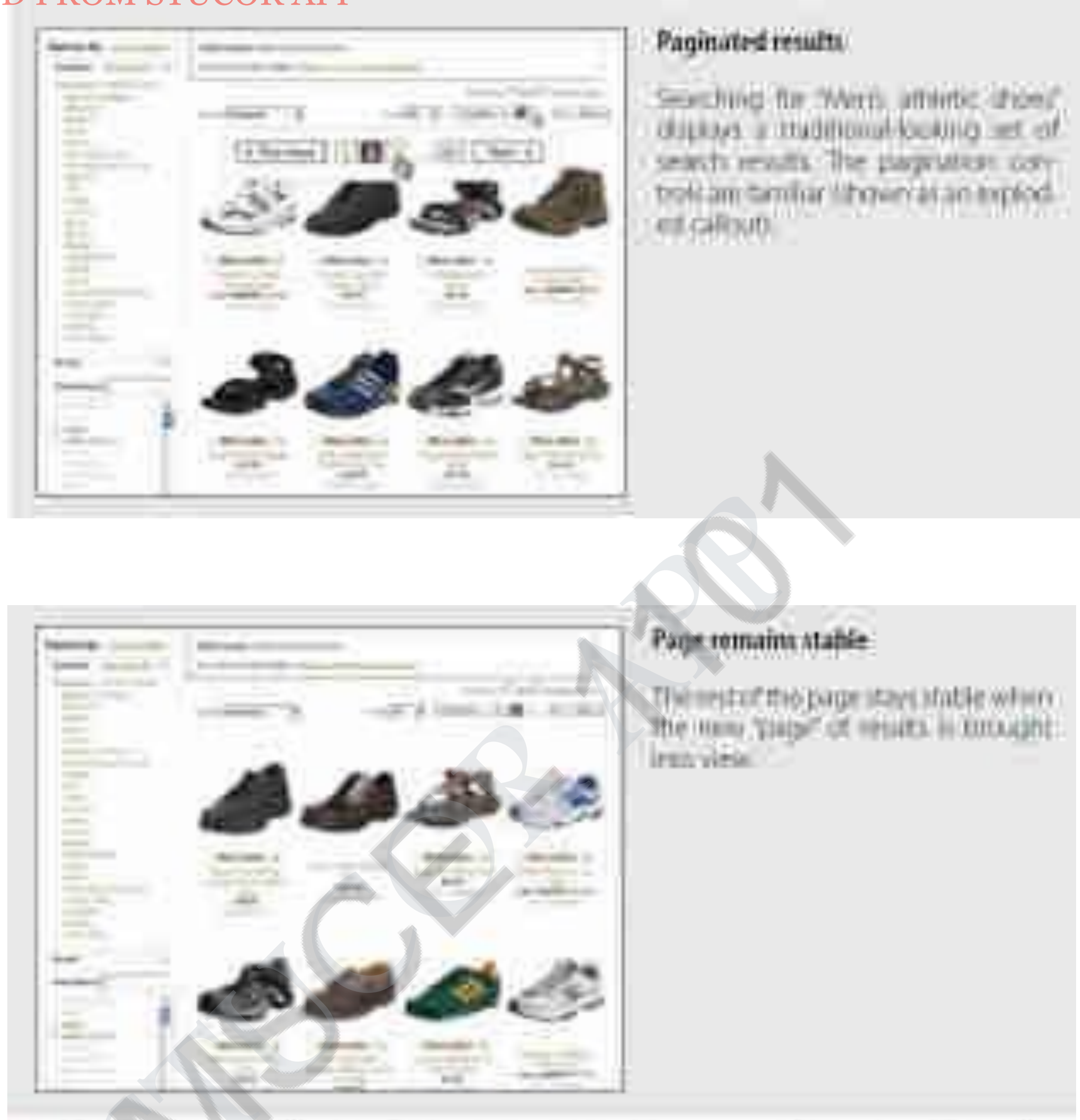
The basic idea is to give users good navigation no matter where they are in the gallery of images.

Virtual Scrolling demonstrate three different ways to manage the virtual space:

1. Yahoo! Mail creates the illusion that all data has been loaded up-front by having the scrollbar reflect the total virtual space.
2. Microsoft Live Search creates the virtual illusion as the user moves down through the search results.
3. PicLens does the same with the caveat that it • shows a virtual window in the larger virtual space

Inline Paging

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 (Figure 7-6).



Paginated results

Searching for "Men's athletic shoes" displays a traditional-looking set of search results. The pagination controls are familiar (shown in an exploded callout).

Page remains stable

The rest of the page stays stable when the new "page" of results is brought into view.

Figure 7-[illegible]. Zappos uses Inline Paging to create a seamless experience moving through search results

Considerations

There are some issues to consider with Inline Paging.

In-page update

Endless.com provides the normal pagination controls. But instead of the whole page refreshing, only the results area is updated. Keeping the context stable creates a better flow experience. With Inline Paging it feels like the user never leaves the page even though new virtual pages of results are being brought into view.

Natural chunking

Inline Paging can also be useful when reading news content online. The *International Herald Tribune* applied this as a way to page through an article while keeping the surrounding context visible at all times

Gmail also uses Inline Paging.. A set number of messages is displayed on the page. Clicking the "Older" or "Newer" links moves the user to a new set of messages. However, instead of refreshing the complete page, just the message area updates.

Inline Paging can be as simple as clicking on a button to load more items into the page on demand. The newly loaded content can be added to the current page. This allows the content to be scrolled, but places the control outside the scrollbar.

Back button

The biggest issue with Inline Paging is whether the back button works correctly. One criticism of Endless.com is that if the user pages through search results and then hits the back button, it jumps to the page just before the search. This unexpected result could be fixed by making the back button respect the virtual pages as well. This is the way Gmail handles the back button. Clicking back moves you through the virtual pages.

Scrolled Paging: Carousel

We 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.

Considerations

There are some issues to consider when using Scrolled Paging.

Time-based

Carousels work well for time-based content. Flickr employs a Carousel to let users navigate back and forth through their photo collection (Figure 7-12).

Animation direction

Inexplicably, AMCtheatres.com animates its Carousel the opposite way. This leads to a confusing experience, and it's harder to know which control to click (Figure 7-13).

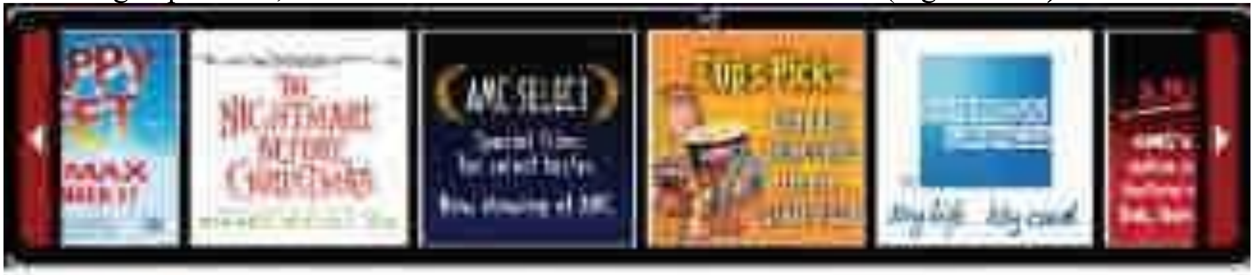


Figure 7-13. AMC animates its carousel backward from the accepted standard; clicking the right arrow should scroll in content from the right (not the left)

8 **Explain the following:**

(i) Virtual Panning

(ii) Zoomable User Interface

(i)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 (Figure 7-14).

Considerations

There are some issues to consider while using Virtual Panning.

Natural Visual Construct

in Google Maps, we pointed out that drag and drop worked well since it fit with the natural visual representation of routes on the map. In the same way, panning around in a map is a natural visual metaphor. Extending the visual space to a larger virtual space is a natural fit.

Gestures

The idea of Virtual Panning has been extended to other devices like gesture-based interfaces. With the introduction of the iPhone, the user can simply "flick" through weather locations, photos, or an iTunes playlist.

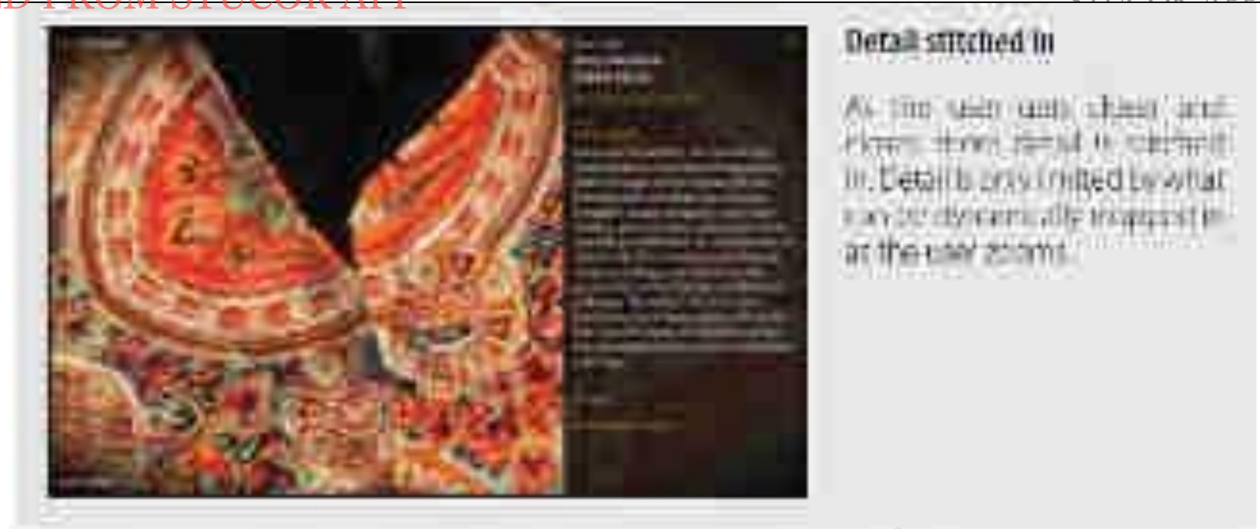
Flicking is similar to panning yet has some differences. With Virtual Panning the canvas only moves while the mouse is dragging it around. With flicking, if the user starts the dragging operation and releases, the canvas will continue moving with some momentum. The canvas slows in such a way as to mimic real-world forces.

(ii) 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.

Considerations

The memorabilia application uses a ZUI interface to move around from artifact to artifact and zoom in to see details on each item



Detail stitched in

As the user zooms in and zooms down, detail is stitched in. Detail is only limited by what can be dynamically mapped in as the user zooms.

Figure 1-15. Hard Rock Cafe uses a zoomable user interface (ZUI) to allow its memorabilia collection to be viewed online.

In this concept prototype, two key ideas are presented:

1. Additional interface elements are just off the page to the top, left, bottom, and/or right. Pulling the content to one side or the other reveals these tools.
2. Moving between the canvas (desktop) and windows is accomplished with a ZUI interface.

ZUIs provide the ultimate virtual canvas. By extending the concept of the page, the user never actually leaves the virtual page.

9. Briefly explain Configurator Process, Overlay Process and Static Single-Page Process.

Configurator Process

Process Flow is meant to invoke delight. We can see Configurator Process interfaces on the Web. 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.

Considerations

There are some issues to consider when using a Configurator Process.

Immediate feedback

In the case of the Porsche site, when the user clicks on various body colors, the car is reconfigured in the selected color and rendered in photorealistic shading. Most configurators allow a 360-degree view of the car as it is configured.

Porsche color configuration experience is actually part of a multi-page process flow. This highlights the fact that the decision between a single-page experience and a multi-page experience are not mutually exclusive. Single-page workflows can live within a multi-page workflow.

Out of view status

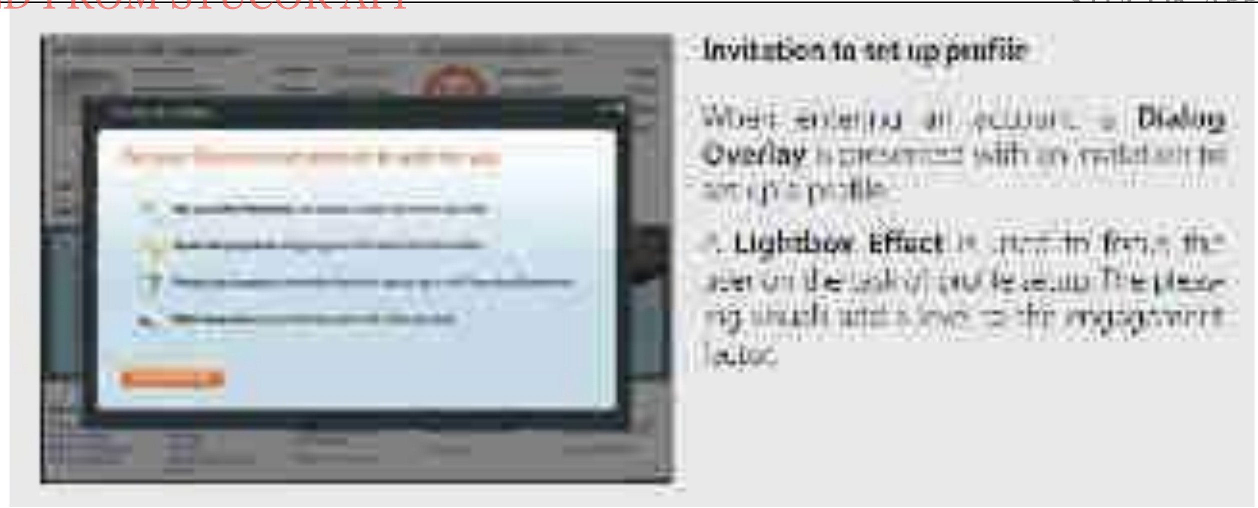
Apple has a Configurator Process for purchasing a Macintosh computer

There is a dynamic nature to this Configurator Process. As the user selects items to add to the computer, the price is dynamically updated using a Spotlight technique. In the upper-right corner of the page, the system configuration and price are highlighted to indicate updated information. The downside is that if this user is scrolled too far down, she can't see the update Spotlight since the summary panel is not visible.

Dialog Overlay Process

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 (Figure 8-11). The overlay pops up when you first enter your account.



Invitation to set up profile

When entering an account, a **Dialog Overlay** is presented with an invitation to set up a profile.

A **Lightbox Effect** is used to focus the user on the task of profile setup. The pleasing visuals add a lot to the engagement factor.

Considerations

There are some issues to consider when using the Dialog Overlay Process.

Making it feel lightweight

The Lightbox Effect is nice because it focuses the user on the task. But what is smart about the flow is the first page is colorful and rendered in a simple, pleasing manner. The process looks like it might be simple and feels engaging. The in-between steps are usually a single action. There is a whole page dedicated just to selecting the due date for payments (Figure 8-12).

Figure 8-12. Discover dedicates a complete "page" to just selecting when you would like to pay your bill

Making the in-between steps clear and visually appealing with a single call to action makes the process feel lightweight. The last step is the biggest. By this time the user has committed to the process to some degree. Most of the user information is already filled in from the account, so the step does not feel too involved.

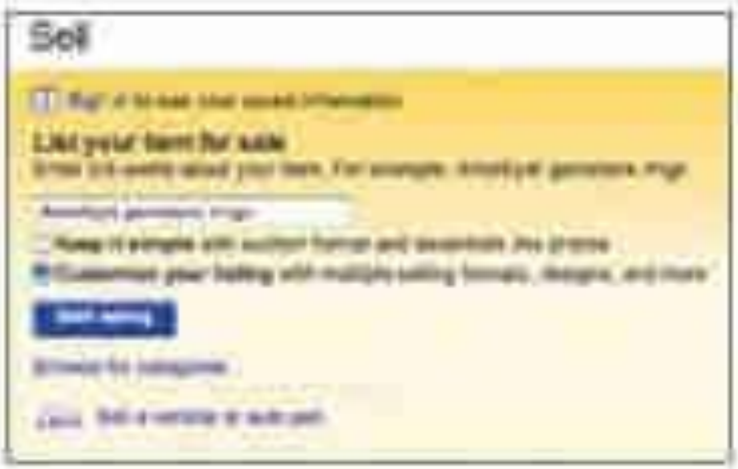
Clear status

Figure 8-13 give some indication of what the users are dealing with when they start. Usually three steps are ideal. In this case, there are five steps. But as we mentioned, the early steps are single actions.

Static Single-Page Process

The user sees all the tasks needed to complete the full process called Static Single-Page Process 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 (Figure 8-17) gathers the description of the item for sale. The "Customize your listings…" option takes the user through a traditional multi-page process (Figure 8-18).



Sell

List your item for sale

Figure 8-17 eBay's simplified one-page flow

Figure 8-18. The customized flow is a traditional multi-page process

Customized flow contains many options for listing an item, it requires a good deal of handholding. A multi-page process fits well with guiding the user through a complex set of tasks. The other flow provided by eBay is a simplified Static Single-Page Process

Figure 8-18. The customized flow is a traditional multi-page process

Customized flow contains many options for listing an item, it requires a good deal of handholding. A multi-page process fits well with guiding the user through a complex set of tasks. The other flow provided by eBay is a simplified Static Single-Page Process (Figure 8-19).

Considerations

There are some issues to keep in mind when using a Static Single-Page Process.

Making it feel lightweight

In this Static Single-Page Process, many options are defaulted and a simplified form is presented to the user. Multiple pages are compressed into a single page. Of course a long page like this can also be daunting. But eBay did a good job of getting the essentials into a single page. Each step is clearly called out with a clear border and a large step number. Multiple pages are not necessarily evil. The eBay example (Figures 8-16 through 8-18) illustrates that there is more than one way to deal with a step-by-step Process Flow. For a very complex flow, a Static Single-Page Process may work well.

Sometimes it is good to break what could be a Static Single-Page Process into a multipage process. Multiple pages can provide the natural chunking needed. They say “You are done with that step, now move onto the next.” Netflix has a problem-reporting interface that does just that. When reporting a scratched disc, clicking on the “DVD is damaged…” link takes the user to a secondary page

9 **10.** Explain the following:

(i) Interactive Single-Page Process (ii)Inline Assistant Process

(i) Interactive Single-Page Process

The Gap accomplishes in product selection in shopping in a single page (Figure 8-3) using Interactive Single-Page Process.

Example:

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. The purple shirt is available in all sizes from XS to XXXL. Hovering over the dark blue shirt immediately discloses that this color is only available in XS and S sizes.

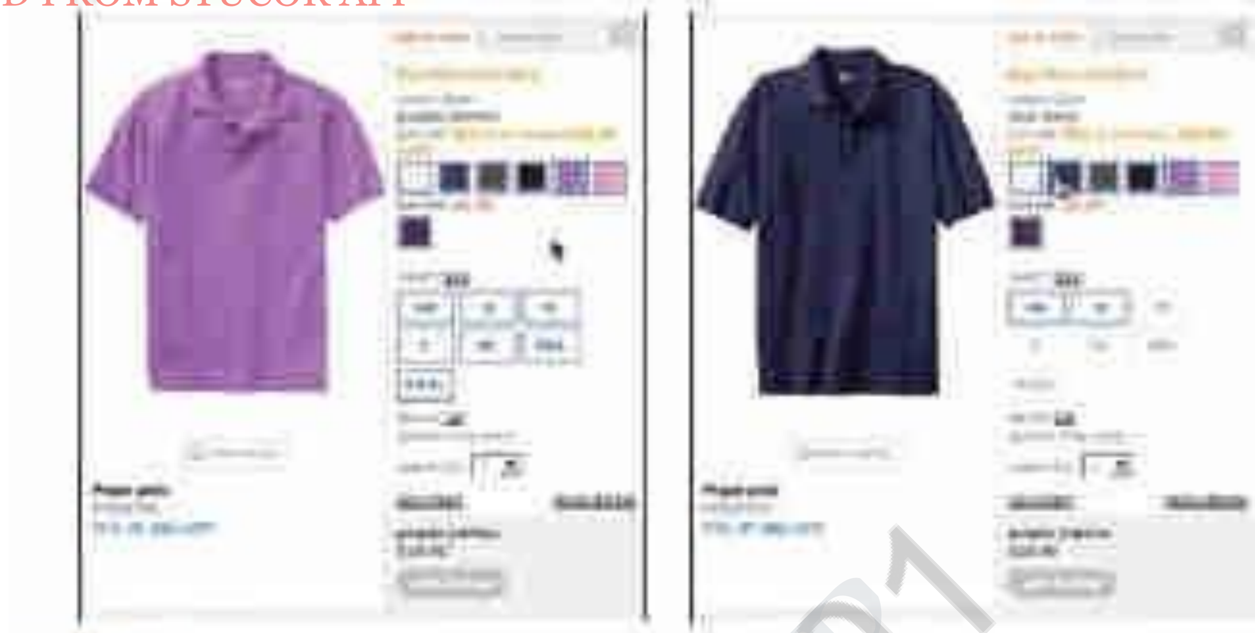


Figure 8-3. The Gap uses Interactive Single-Page Process to reflect the sizes for each product color choice in real time

Considerations

There are some issues to consider when using an Interactive Single-Page Process.

Responsiveness

The user's taste preference comes first. Either the color or the size can be chosen. If the item is out of stock for any color/size combination, it is displayed as unavailable. By placing this process in a few simple interactions, the user can quickly find something available to buy. With any online shopping experience, the potential for the user to bail out is a real concern. In-place interactions like this reduce these *bailout moments*.

Amazon's interface for selecting a shirt also uses **Interactive Single-Page Process** with a slightly different interface

Interactive, single-page process flows improve user engagement and increase conversion rates.

Keeping users engaged

Broadmoor Hotel uses Interactive Single-Page Process for room reservations

Each column represents what would normally be presented on a separate page. In the first column, a calendar discloses availability up front. This prevents scheduling errors. Selecting the room from the second column updates both the room picture and the pricing. The pricing is reflected back on the calendar days (Figure 8-6) as well as in the third column where credit card and contact information is entered.

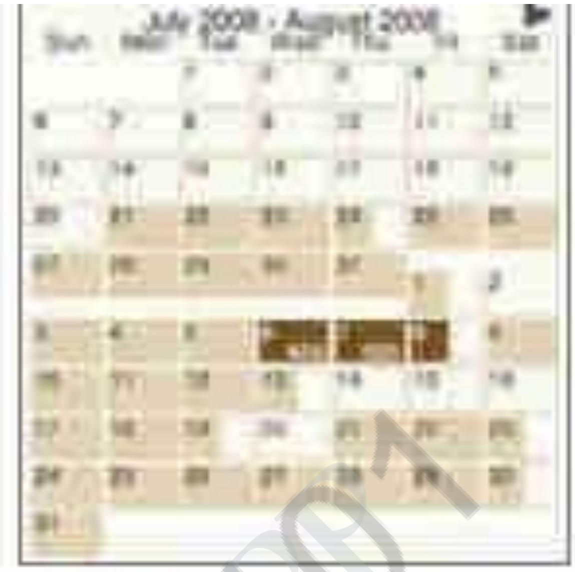


Figure 8-6. *The Broadmoor calendar shows availability and pricing information*

Benefits

Adobe calls out the Broadmoor one-page reservation interface in its Adobe Showcase. It states the benefits of this method:

- Reduces entire reservation process to a single screen.
- Reduces the number of screens in the online reservation process from five to one. Other online reservation applications average 5 to 10 screens.
- Seventy-five percent of users choose One Screen in favor of the HTML version.
- Allows users to vary purchase parameters at will and immediately view results.
- Reduces the time it takes to make a reservation from at least three minutes to less than one.

Inline Assistant Process

Common place where multiple pages are used to complete a process is when adding items to a shopping cart. 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 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 (Figure 8-7).

Considerations

There are some things to consider when using the Inline Assistant Process.

Quick and easy

The Gap integrates the shopping cart into its entire site as a drop-down shade. In fact, the Gap, Old Navy, Banana Republic, and PiperLime all share the same Inline Assistant Process-style shopping cart. The Gap is betting that making it quick and easy to add items to the cart across four stores will equal more sales.

Additional step

Amazon is betting on its recommendation engine. By going to a second page, Amazon can display other shirts like the one added—as well as advertise the Amazon.com Visa card (Figure 8-8).

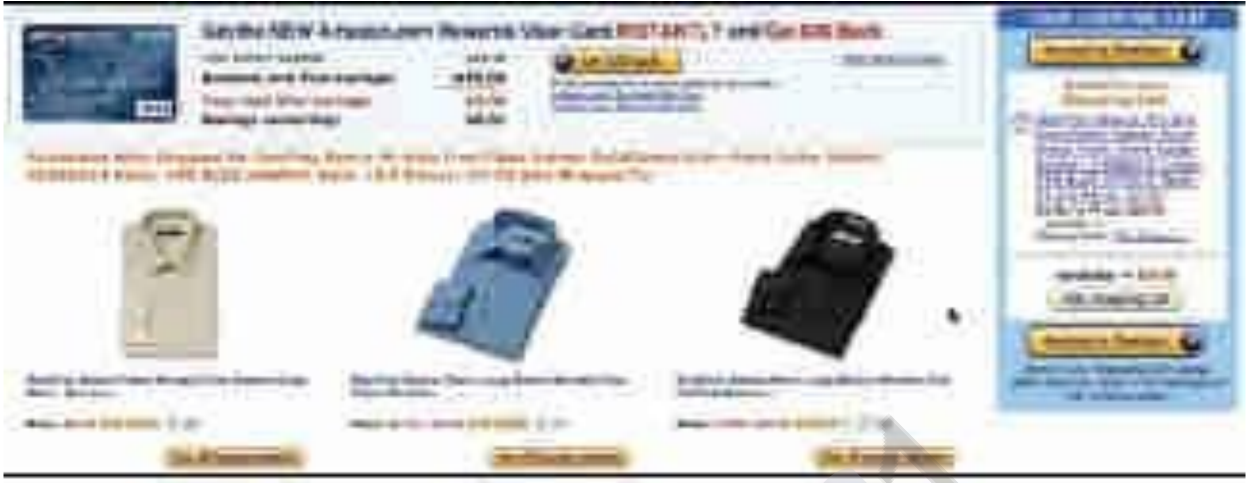


Figure 8-8. Amazon shows recommendations when confirming an add to its shopping cart

Blending quick and easy with the additional step

Netflix does when a user adds movies to his shipping queue (Figure 8-9).

Each movie on the site has an "Add" button. Clicking "Add" immediately adds the movie to the user's queue. As a confirmation and an opportunity for recommendations, a Dialog Overlay is displayed on top of the movie page.

Just like Amazon, Netflix has a sophisticated recommendation engine. The bet is that since the user has expressed interest in an item (shirt or movie), the site can find other items similar to it to suggest. Amazon does this in a separate page. Netflix does it in an overlay that is easily dismissed by clicking anywhere outside the overlay (or by clicking the close button at the top or bottom).

In a previous version of Netflix (or if JavaScript is disabled), this becomes a multiple-page experience.

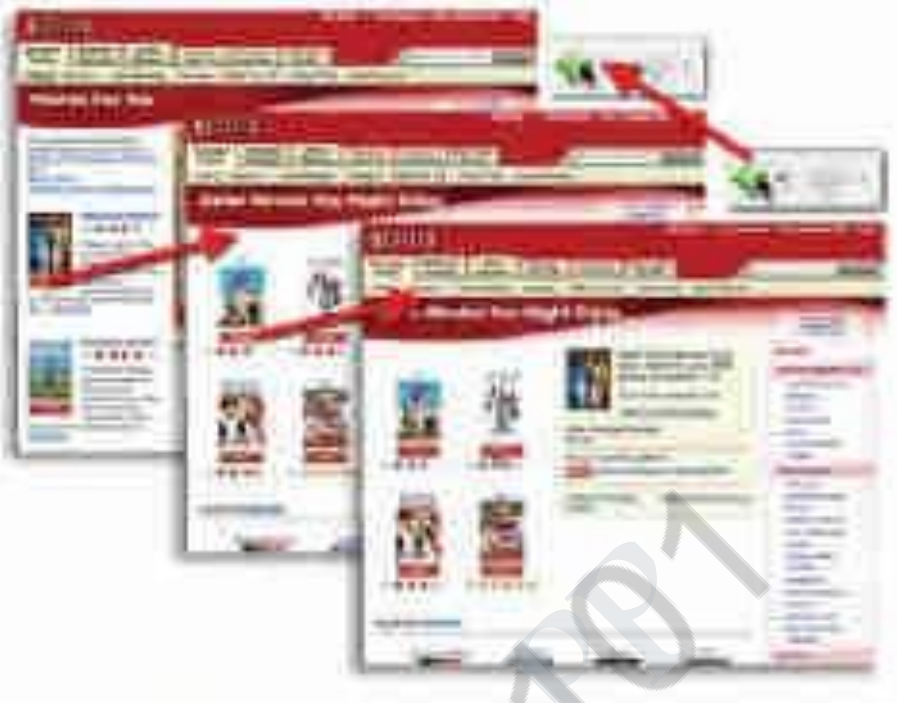


Figure 8-10. Without JavaScript enabled, you can see the original multi page flow for the "Add to Queue" process on the Netflix site

Each movie add leads to a separate recommendation page. Clicking on "Add" for a movie on a recommendation page takes the user to a secondary recommendation page. This process can continue, on and on. Eventually, the user has to hit the back button a number of times to get back to the original context.

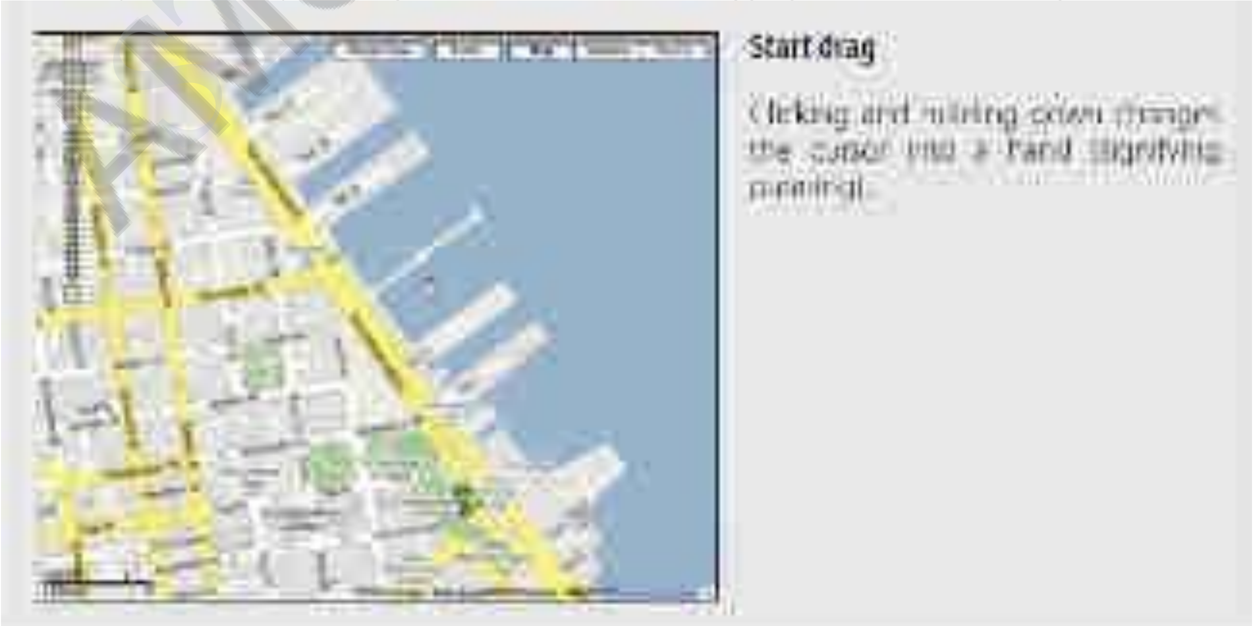
10 **8. Explain the following:**

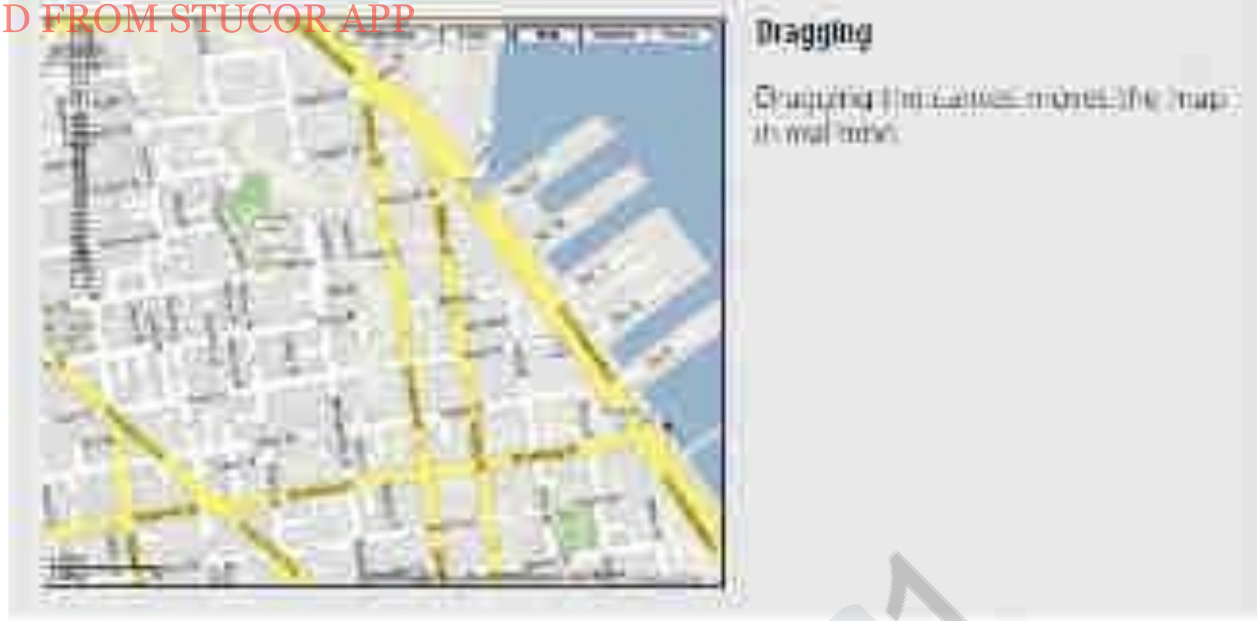
(i) Virtual Panning

(ii) Zoomable User Interface

(i)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 (Figure 7-14).


Start drag
Clicking and holding down changes the cursor into a hand signifying [illegible]



Dragging

Dragging the canvas moves the map in real time.

Figure 7-14. Google Maps creates a virtual canvas; one tool that helps with that illusion is the ability to pan from area to area

Considerations

There are some issues to consider while using Virtual Panning.

Natural Visual Construct

in Google Maps, we pointed out that drag and drop worked well since it fit with the natural visual representation of routes on the map. In the same way, panning around in a map is a natural visual metaphor. Extending the visual space to a larger virtual space is a natural fit.

Gestures

The idea of Virtual Panning has been extended to other devices like gesture-based interfaces. With the introduction of the iPhone, the user can simply “flick” through weather locations, photos, or an iTunes playlist.

Flicking is similar to panning yet has some differences. With Virtual Panning the canvas only moves while the mouse is dragging it around. With flicking, if the user starts the dragging operation and releases, the canvas will continue moving with some momentum. The canvas slows in such a way as to mimic real-world forces.

(ii) 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.

Considerations

The memorabilia application uses a ZUI interface to move around from artifact to artifact and zoom in to see details on each item

In this concept prototype, two key ideas are presented:

1. Additional interface elements are just off the page to the top, left, bottom, and/or right. Pulling the content to one side or the other reveals these tools.
2. Moving between the canvas (desktop) and windows is accomplished with a ZUI interface.

ZUIs provide the ultimate virtual canvas. By extending the concept of the page, the user never actually leaves the virtual page.