CHAPTER 1: Introduction

1. Compare and contrast the role of a systems analyst, business analyst and infrastructure analyst.

These three roles emphasize different perspectives on the system. The **business** analyst identifies and represents the users interests, while the **systems** analyst knows how to apply IS to support business needs. Together, the systems analyst and the business analyst can design a system that conforms to the IS standards while adding value to the business. The **infrastructure analyst** has more technical knowledge and provides the team with technical constraints, or identifies infrastructure changes that the new system will require.

System Analyst: A System Analysts role is to:

- Identifying how technology can improve business processes
- Designing the new business processes
- Designing the information system
- Ensuring that the system conforms to information systems standards

Business Analyst: A Business Analyst's role is to:

- Analyzing the key business aspects of the system
- Identifying how the system will provide business value
- Designing the new business processes and policies

Infrastructure Analyst: The Infrastructure Analyst's role is to:

- Focuses on the systems technical architecture that includes networking, hardware, software
- Ensuring the system conforms to infrastructure standards
- Identifying infrastructure changes needed to support the system

2. Compare and contrast phases, steps, techniques and deliverables.

The Systems Development Life Cycle "SDLC" has a set of four fundamental phases: planning, analysis, design, and implementation. Different projects may emphasize different parts of the SDLC or approach the SDLC phases in different ways, but all projects have elements of these four phases.

Each phase is itself composed of a series of steps, which rely upon techniques that produce deliverables (specific documents and files that provide understanding about the project).

Example: When you apply for admission to a university, there are several phases that all students go through: information gathering, applying, and accepting. Each of these phases has steps—information gathering includes steps like searching for schools, requesting information, and reading brochures. Students then use techniques (e.g., Internet searching) that can be applied to steps (e.g., requesting information) to create deliverables (e.g., evaluations of different aspects of universities).

3. Describe the major phases in the systems development life cycle (SDLC).

There are four major phases in SDLC.

a) Planning: The Planning phase is the fundamental process of understanding why an information system should be built and determining how the project team will go about building it.

b) Analysis: The Analysis phase answers the questions of who will use the system, what the system will do, and where and when it will be used. During this phase, the project team investigates any current system(s), identifies improvement opportunities, and develops a concept for the new system.

c) Design: The design phase decides how the system will operate, in terms of the hardware, software, and network infrastructure; the user interface, forms and reports; and the specific programs, databases, and files that will be needed.

d) Implementation: The final phase in the SDLC is the implementation phase, during which the system is actually built (or purchased, in the case of a packaged software design). This is the phase that usually gets the most attention, because for most systems it is longest and most expensive single part of the development process.

4. Which phase in the SDLC is most important and why?

All systems development projects follow essentially the same fundamental process called the system development life cycle (SDLC). **The planning phase in SDLC** is the most important phase because during this phase the project team identifies the business value of the system, conducts a feasibility analysis, and plans the project. This phase is the fundamental process of understanding why an information system should be built and determining how the project team will go about building it.

5. What does gradual refinement mean in context of SDLC?

Generally, the clarity of understanding and the depth of detail of the new system are gradually refined during the phases of the SDLC. Initially, the requirements are only vaguely understood. This understanding is improved during the Analysis phase. Further detail is developed during Design, and then is fully expressed during Implementation. Deliverables produces from each phase are developed on to the next the phases that follow up. This could be seen that on planning, deliverable produced would be a work plan. With out having a work plan, it would be difficult to conduct analyses phase, as there would not be a guideline of conducting SDLC

6. What are the key factors in selecting a methodology?

There are several factors that influence the choice of a methodology:

- Clarity of the user requirements
- Familiarity with the base technology
- System complexity
- Need for system reliability
- Time pressures &
- Need to see progress on the time schedule.

7. Compare and contrast process-centered methodologies with data-centered methodologies.

Process-centered methodologies: Process-centered methodologies emphasize process models as the core of the system concept.
Example: Process-centered methodologies would focus first on defining the processes (e.g., assemble sandwich ingredients).

Data-centered methodologies**:** Data-centered methodologies emphasize data models as the core of the system concept.
Example: Data centered methodologies would focus first on defining the contents of the storage areas (e.g., refrigerator) and how the contents were organized.

8. Compare and contrast structured-design based methodologies in general to RAD-based methodologies in general.

Structured design methodologies are usually fairly formal, step-by-step approaches to SDLC phases. The project moves through the phases in a systematic way. The emphasis in most of these methodologies is development of paper-based specifications for the new system prior to implementation.

RAD methodologies, on the other hand, tend to emphasize quick creation of a limited-capability version of the system or a system model. These methodologies focus on refining this preliminary system or model rather than trying to fully describe it on paper prior to implementation.

9. Describe the major elements and issues with prototyping.

A prototyping-based methodology performs the analysis, design, and implementation phases concurrently, and all three phases are performed repeatedly in a cycle until the system is completed. Prototype is usually the first part of the system that the user will use and the key advantage of a prototyping-based methodology is that it *very* quickly provides a system for the users to interact with, even if it is not ready for widespread organizational use at first.

The major problem with prototyping is that its fast-paced system releases challenge attempts to conduct careful, methodical analysis. Often the prototype undergoes such significant changes that many initial design decisions become poor ones. This can cause problems in the development of complex systems because fundamental issues and problems are not recognized until well into the development process.

Example: Imagine building a car and discovering late in the prototyping process that you have to take the whole engine out to change the oil (because no one thought about the need to change the oil until after it had been driven 10,000 miles).

10. Describe the major elements and issues with throwaway-prototyping.

In **throw-away prototyping**, there is a more careful and rigorous analysis performed as compared to Prototyping (more similar to the traditional SDLC). However, in the Design phase, models of various elements of the system are developed to help explore design alternatives and refine system requirements. These prototypes help users clarify their needs, options, and priorities. Once user requirements are established through reaction to the design prototype, it is discarded; however, the requirements it embodied are incorporated into the new system.

Throw-away prototyping is very useful in situations where users are uncertain about key elements of the system. Throw-away prototypes can help focus users on design issues and improve understanding of business needs. The approach helps create a system that suits the users' needs even though those needs may have been poorly understood initially.

11. Compare and contrast extreme programming and throwaway prototyping.

Extreme programming (XP): Extreme Programming is founded on four core values which provide a foundation on which XP developers use to create any system. The four core values are Communication, Simplicity, Feedback and Courage.

Throwaway Prototyping: Throwaway prototyping-based methodologies balance the benefits of well thought out analysis and design phases with the advantages of using prototypes to refine key issues before a system is built. Each of these issues is examined by analyzing, designing, and building a design prototype.

12. Describe the principal steps in the planning phase. What are the major deliverables?

The planning phase is the fundamental process of understanding why an information system should be built and determining how the project team will go about building it. It has two steps:

1. Project Initiation: This step involves the identification of the system's business value. A s*ystem request* presents a brief summary of a business need, and it explains how a system that supports the need will create business value. The IS department works together with the person or department that generated the request (called the *project sponsor*) to conduct a feasibility analysis. The *feasibility analysis* examines key aspects of the proposed project like the technical feasibility, economic feasibility and organizational feasibility. Work plan, staffing and standards list are the deliverables.

2. Project Management: Once the project is approved, it enters—project management. During project management, the project manager creates a work plan, staffs the project, and puts techniques in place to help the project team control and direct the project through the entire SDLC. The deliverable for project management is a project plan that describes how the project team will go about developing the system.

13 Describe the principal steps in the analysis phase. What are the major deliverables?

The analysis phase answers the questions of who will use the system, what the system will do, and where and when it will be used. During this phase, the project team investigates any current system(s), identifies improvement opportunities, and develops a concept for the new system. This phase has three steps:

1. Analysis Strategy: Developed to guide the project team's efforts. Such a strategy usually includes an analysis of the current system (as-is system) and its problems, and then ways to design a new system (to-be system).

2. Requirements Gathering: The analysis of this information—in conjunction with input from project sponsor and many other people—leads to the development of a concept for a new system. The system concept is then used as a basis to develop a set of business *analysis models* that describes how the business will operate if the new system were developed.

3. System Proposal: The analyses, system concept, and models are combined into a document called the system proposal, which is presented to the project sponsor and other key decision makers.

The system proposal is the initial deliverable that describes what business requirements the new system should meet. This is really the first step in the design of the new system.

14. Describe the principal steps in the design phase. What are the major deliverables?

The design phase decides how the system will operate, in terms of the hardware, software and network infrastructure; the user interface, forms and reports; and the specific programs, databases, and files that will be needed. The design phase has four steps:

1. Developing design strategy: This clarifies whether the system will be developed by the company's own programmers, whether it will be outsourced to another firm (usually a consulting firm), or whether the company will buy an existing software package.

2. Basic architecture design: This describes the hardware, software, and network infrastructure that will be used. In most cases, the system will add or change the infrastructure that already exists in the organization.

3. Database and file specifications: These define exactly what data will be stored and where they will be stored.

4. Program Design: The analyst team develops the program design, which defines the programs that need to be written and exactly what each program will do.

This collection of deliverables (architecture design, interface design, database and file specifications, and program design) is the *system specification* that is handed to the programming team for implementation. At the end of the design phase, the feasibility analysis and project plan are reexamined and revised, and another decision is made by the project sponsor and approval committee about whether to terminate the project or continue.

15. Describe the principal steps in the implementation phase. What are the major deliverables?

The final phase in the SDLC is the implementation phase, during which the system is actually built (or purchased, in the case of a packaged software design). This phase has three steps:

1. System construction: The system is built and tested to ensure it performs as designed. Since the cost of bugs can be immense, testing is one of the most critical steps in implementation.

2. Installation: This is the process by which the old system is turned off and the new one is turned on. It may include a direct cutover approach, a parallel conversion approach, or a phased conversion strategy.

3. Support plan: This plan usually includes a formal or informal post-implementation review, as well as a systematic way for identifying major and minor changes needed for the system.

16. What are the roles of a project sponsor and the approval committee?

Project sponsor: Project Sponsor could be represented either by an individual or a department or a team. The project sponsor is from where the request is generated. They are basically involved through out the various phases of the SDLC.

a. The IS department works hand in hand with the project sponsor to conduct a feasibility analysis, be it a technical, organizational or an economical feasibility.

b. During the Analysis phase the project team investigates any current system, identifies improvement opportunities and develops a concept for the new system in conjunction with the input from the project sponsor.

c. At the end of the design phase the feasibility analysis and the project plan are reexamined and revised and the decision whether the project is to continue or terminated is made by the project sponsor along with the approval committee.

d. The key deliverables for each phase are typically very long and are presented to the project sponsor for approval as the project moves from phase to phase.

e. The project sponsor approves the key deliverables for each phase only after which the project moves from phase to phase.

f. The project sponsor along with the users provides valuable comments which are used to re-analyze, re-design and re-implement a second prototype which provides few more features.

Approval Committee: Sometimes referred to as the steering committee is the final decision maker regarding the fate of the project after carefully reviewing the system request, feasibility analysis, system proposal etc.

a) In planning phase after feasibility analysis the approval committee decides whether the project should be undertaken.

b) After Analysis the system proposal is forwarded to the approval committee to decide whether the project should continue to move forward

c) During the Design phase the feasibility analysis and project plan are reexamined and the approval committee will decide whether to terminate or continue with the project

17. Describe the major elements and issues with waterfall development.

Waterfall Development follows the phases of the life cycle in sequence (Planning, Analysis, Design, and Implementation). Each phase is thoroughly documented and approval is required before proceeding to the subsequent phase. It is difficult, though not impossible, to go backwards in the SDLC under Waterfall Development.

Waterfall Development requires that the system requirements be precisely specified prior to implementation and also often "freezes" those requirements during development. The high degree of effort devoted to specifying user requirements is a strength of Waterfall Development but specifying those requirements on paper is laborious and may lead to errors and omissions. "Freezing" the requirements during development helps assure that the system is developed according to specifications, but in a dynamic business environment, the system that is ultimately developed may bear little resemblance to what is actually needed at the time the project is completed. Therefore, extensive maintenance may be needed after implementation to revise the system to meet current conditions.

The two key advantages of the structured design waterfall approach are that:

- It identifies system requirements long before programming begins
- It minimizes changes to the requirements as the project proceeds.

The two key disadvantages are that:

a. The design must be completely specified before programming begins.
b. A long time elapses between the completion of the system proposal in the analysis phase and the delivery of the system.

18. Describe the major elements and issues with phased development.

Phased Development is a RAD methodology that does not attempt to develop the complete system initially. Instead, the system is generally specified. User requirements are organized into a series of versions. The first version includes the essential system components and is delivered to the users quickly. Subsequent versions add features and refinements to the system based on the initial specification plus the users' feedback and reaction to using the system.

The critical issue with Phased Development is to accurately specify the initial requirements so that the first version provided to the users is useful, although incomplete. If this is done well, Phased Development will provide value to the organization by getting the users a system to use quickly. New requirements may be identified through user feedback as well, which improves the "fit" of the system to the business needs.

19. Describe the major elements and issues with parallel development

The parallel development methodology attempts to address the problem of long delays between the analysis phase and the delivery of the system. Instead of doing design and implementation in sequence, it performs a general design for the whole system and then divides the project into a series of distinct subprojects that can be designed and implemented in parallel. Once all subprojects are complete, there is a final integration of the separate pieces, and the system is delivered

The major issues in this approach are that it still suffers from problems caused by paper documents. It also adds a new problem such as sometimes the subprojects are not completely independent; design decisions made in one subproject may affect another, and the end of the project may require significant integration efforts.

20. Describe the major elements and issues with an object-oriented approach to developing information systems.

Object-oriented systems analysis and design (OOSAD) is most associated with a phased development RAD-based methodology, where the time spent in each phase is very short. OOSAD uses a use-case-driven, architecture-centric, iterative, and incremental information systems development approach. It supports three different views of the evolving system: functional, static, and dynamic. OOSAD allows the analyst to decompose complex problems into smaller, more manageable components using a commonly accepted set of notations. Additionally, object-oriented systems analysis and design allows the analyst to interact with the user with objects from the user's environment instead of a set of separate processes and data.

21. What are the phases and workflows of the Unified Process?

The phases of the Unified Process are:

1. Inception: a business case is made for the proposed system.
2. Elaboration: requirements are determined and the system is designed.
3. Construction: focuses heavily on programming the evolving information system.
4. Transition: addresses aspects typically associated with the implementation phase of a traditional SDLC approach

The workflows of the Unified Process are:

1. Engineering Workflows
   a. Business Modeling Workflow
   b. Requirements Workflow
   c. Analysis Workflow
   d. Design Workflow
   e. Implementation Workflow
   f. Testing Workflow
   g. Deployment Workflow
2. Supporting Workflows
   a. Project Management Workflow
   b. Configuration and Change Management Workflow
   c. Environment Workflow

22. Compare the phases of the Unified Process with the phases of the waterfall model.

The Inception phase of the Unified Process is similar to the Planning phase of the Waterfall methodology.
The Elaboration phase of the Unified Process encompasses most of the activities in both the Analysis and Design phases of the Waterfall methodology.
The Construction and Transition phases of the Unified Process together comprise the set of activities normally performed during the Implementation phase of the Waterfall methodology.

23. Who is the Object Management Group?

The Object Management Group is a consortium of organizations whose purpose it is to set standards for object-oriented systems. The OMG is the group that ratified the UML as an industry standard.

24. What is the Unified Modeling Language?

The Unified Modeling Language (UML) is a standard set of diagramming techniques. The objective of UML is to provide a common vocabulary of object-oriented terms and diagramming techniques rich enough to model any systems development project from analysis through implementation.

25. What does it mean for an OOSAD approach to be incremental and iterative?

Iterative and incremental development means that the system undergoes continuous testing and refinement throughout the life of the project. As each increment and iteration is completed, a more complete representation of the user's real functional requirements is uncovered.

26. Why is it important for an OOSAD approach to be architecture centric?

A development methodology is said to be architecture centric if the architecture is the main artifact upon which the design and implementation of the system is built. In other words, first the developers decide upon a framework and then the system is built around and upon it. The architecture becomes both an enabler and a constraint on how the system is developed.

It is important of an OOSAD approach to be architecture centric because this approach allows the development team to iterate repeatedly without undermining the stability of the system's core. If the architecture were to be revisited and modified over time, then there would be a very good chance that support to some (or most) system components would eventually be compromised.

27. What is a use case?

A use case is a description of a system's behavior as it responds to stimuli from its environment. These stimuli come from the various users of the system as well as from external systems. The use cases are used to identify and to communicate the requirements for the system to the programmers who must write the system.

A use case diagram is a type of behavioral diagram that is part of the Unified Modeling Language (UML) and that graphically depicts a use case or a collection of use cases.

28. What is the primary purpose of structure diagrams? Give some examples of structure diagrams.

Structure diagrams provide a way to represent the data and static relationships in an information system. The structure diagrams include class, object, package, deployment, component, and composite structure diagrams.

29. For what are behavior diagrams used? Give some examples of behavior diagrams.

Behavior diagrams provide the analyst with a way to depict the dynamic relationships among the instances or objects that represent the business information system. They also allow the modeling of the dynamic behavior of individual objects throughout their lifetime. The behavior diagrams support the analyst in modeling the functional requirements of an evolving information system. The behavior modeling diagrams include activity, sequence, communication, interaction overview, timing, behavior state machine, protocol state machine, and use-case diagrams.

Chapter 4: REQUIREMENTS ANALYSIS

1. **What are the three basic steps of the analysis process? Which step is sometimes skipped or done in a cursory fashion? Why?**

   The basic process of analysis is divided into three steps:

   a) Understanding the existing system: as-is system
   b) Identifying improvements
   c) Developing requirements for the new system: to-be system.

   Sometimes the first step (i.e., understanding the as-is system) is skipped or done in a cursory manner. This happens when no current system exists, if the existing system and processes are irrelevant to the future system, or if the project team is using a RAD or agile development methodology in which the as-is system is not emphasized.

2. **What are the important factors in selecting an appropriate analysis strategy?**

   The potential business value varies with analysis strategy. While BPA has the potential to improve the business, most of the benefits from BPA are tactical and small in nature. Since BPA does not seek to change the business processes, it can only improve their efficiency. BPI usually offers moderate potential benefits, depending upon the scope of the project, because it seeks to change the business in some way. It can increase both efficiency and effectiveness. BPR creates large potential benefits because it seeks to radically improve the nature of the business.

3. **What are the key deliverables that are created during the analysis phase? What is the final deliverable from the analysis phase, and what does it contain?**

   The Analysis Phase takes the general ideas in the system request and refines them into a detailed requirements definition and answers the questions of *who* will use the system, *what* the system will do, and *where* and *when* it will be used. During this phase, the project team will learn about the system. The team then produces the Functional Model (Activity Diagrams, Use Case Descriptions and Diagram), Structural Model (CRC Cards and Class and Object Diagrams), and Behavioral Models (Sequence Diagrams, Communication Diagrams, and Behavioral state machines) that together form the system proposal.

   The system proposal also includes revised project management deliverables, such as the feasibility analysis and the work plan. The system proposal is presented to the approval committee, who decides if the project is to continue.

4. What is the purpose of the requirements definition?

The requirements definition report—usually just called the requirements definition—is a straightforward text report that simply lists the functional and nonfunctional requirements in an outline format. The most obvious purpose of the requirements definition is to provide the information needed by the other deliverables in analysis, which include functional, structural, and behavioral models, and to support activities in the design phase. The most important purpose of the requirements definition, however, is to define the scope of the system. The document describes to the analysts exactly what the system needs to end up doing. When discrepancies arise, the document serves as the place to go for clarification.

5. Compare and contrast the business goals of BPA, BPI, and BPR

BPA, BPI & BPR techniques are tools used by analysts when they need to guide the users in explaining what is wanted from a system. The three kinds of techniques work similarly. They help users critically examine the current state of systems and processes (the as-is system), identify exactly what needs to change, and develop a concept for a new system (the to-be system).

Business Process Automation (BPA)
Means leaving the basic way in which the organization operates unchanged and using computer technology to do some of the work. BPA can make the organization more efficient but has the least impact on the business. BPA projects spend a significant time understanding the current as-is system before moving on to improvements and to-be system requirements. Problem analysis and root cause analysis are two popular BPA techniques.

Business Process Improvement (BPI),
Means making moderate changes to the way in which the organization operates to take advantage of new opportunities offered by technology or to copy what competitors are doing. BPI can improve efficiency (i.e., doing things right) and improve effectiveness (i.e., doing the right things). BPI projects also spend time understanding the as-is system, but much less time than BPA projects; their primary focus is on improving business processes, so time is spent on the as-is only to help with the improvement analyses and the to-be system requirements. Duration analysis, activity-based costing, and information benchmarking are three popular BPI activities.

Business Process Reengineering (BPR)
Means changing the fundamental way in which the organization operates—"obliterating" the current way of doing business and making major changes to take advantage of new ideas and new technology. BPR projects spend little time understanding the as-is, because their goal is to focus on new ideas and new ways of

doing business. Outcome analysis, technology analysis, and activity elimination are three popular BPR activities.

6. What is the difference between an as-is system and a to-be system?

An as is system is an existing or a current state of system or process. The As-Is system is the combination of people, processes, data, and technology that currently perform the tasks and functions of the system under study. The As-Is system may or may not incorporate computers.
A to be system is the one which is developed on the basis of the analysis done on the current system which needs to be changed. The To-Be system is the combination of people, processes, data, and technology that will perform the required tasks and functions of the system under study.

8. Compare and contrast problem analysis and root cause analysis. Under what conditions would you use problem analysis? Under what conditions would you use root cause analysis?

Problem Analysis and Root-Cause Analysis are two different techniques to be employed in Business Process Automation to determine improvements to the current system. The two strategies vary in the emphasis of the analysis performed.

Problem Analysis
Problem Analysis asks the users and managers of the As-Is system to identify system problems and to suggest problem solution. Problem analysis would be suitable when the problems being experienced with the As-Is system are relatively minor, and the changes needed are primarily 'touch-ups'.

Root-Cause Analysis
Root-Cause analysis focuses on being sure that the problem's underlying cause is understood, rather than just assuming that cause is known. This emphasis helps ensure that solutions chosen will solve real business problems rather than solving a problem symptom.

Root-Cause Analysis is appropriate when the problems of the As-Is system are more significant, and the team needs assurance that they are designing a solution that really solves the true problems.

9. Compare and contrast duration analysis and activity based costing

Duration analysis and activity-based costing are techniques used in Business Process Improvement to help identify system improvement opportunities. These two techniques focus on existing business processes in the As-Is system.

"Duration Analysis" assesses the time requirements to complete a process. First, the total time required to complete a business process is determined. Then the process is broken down into individual steps, and the time required to complete each step is determined. The total time of all steps is calculated and compared to the total time of the process. If these totals are substantially different (total time of process > total time of process steps), then significant inefficiencies exist, and the process needs major revision.

"Activity-Based Costing" assesses the costs required to perform a process. In this technique, the cost of each major process or each step in a business function is measured. The most costly processes are then the targets of the development team's improvement efforts. Although straightforward in concept, this technique is complex in practice due to the difficulty of determining the indirect costs to apply to the business process (es). Incorrectly assigned indirect costs may bias the results of the analysis.

17. Describe the five major steps in conducting JAD sessions.

The five major steps to conducting JAD sessions are:

a) Selecting participants - determine who should be included and why they should be included (what contribution will they make to the project?).
b) Design the JAD session - plan the activities and techniques that will be incorporated into the JAD session in order to accomplish the session goals.
c) Prepare for the JAD session - review related material; review JAD plan; inform participants about the process and the contributions they'll be expected to make.
d) Conduct the JAD session - establish rapport with the participants; define session ground rules; Follow session plan and have facilitator conduct various activities and techniques; record information.
e) Follow up the session by preparing a post-session report and circulating it among session attendees.

18. How are participants selected for interviews and JAD sessions?

The interview is the most commonly used requirements-gathering technique. The people who appear on the interview schedule are selected based on the analyst’s information needs. The project sponsor, key business users, and other members of the project team can help the analyst determine who in the organization can best provide important information about requirements. These people are listed on the interview schedule in the order in which they should be interviewed.

Joint application development “JAD” is an information gathering technique that allows the project team, users, and management to work together to identify requirements for the system. Participants are selected based on the information they can contribute, to provide a broad mix of organizational levels, and to build political support for the new system.

19. What are the three primary things that a facilitator does in conducting the JAD session?

a) The facilitator is responsible for guiding the group through the planned activities on the JAD agenda. The facilitator must keep the group on track and try to prevent it from digressing in unproductive directions.
b) The facilitator helps foster communication among the group members and assists them through the analysis techniques that are being used.
c) The facilitator records the group's ideas on a public display area, organizes the information, and assists the group in working with the information.

20. How does a JAD facilitator differ from a scribe?

The facilitator is responsible for conducting the session. He/she is generally knowledgeable about IS issues, JAD facilitation, and group behavior management. The scribe merely records information as the session proceeds, perhaps using CASE tools

21. How does designing questions for questionnaires differ from designing questions for interviews or JAD sessions?

The major difference is the fact that questions on questionnaires need to be very carefully stated in order to avoid misunderstanding by the recipient. If a question during an interview or a JAD session is misunderstood, the misunderstanding can be immediately detected and the question clarified. A poorly worded question on a questionnaire may confuse the recipient, causing him/her to answer with incorrect information, or may antagonize the recipient, causing him/her to forego completing the questionnaire.

Chapter 5: USE CASE DIAGRAMS

1. What is the purpose of an activity diagram?

Activity diagrams are used to model the behavior in a business process independent of objects. Activity diagrams can be viewed as sophisticated data flow diagrams that are used in conjunction with structured analysis. Activity diagrams include notation that addresses the modeling of parallel, concurrent activities and complex decision processes. Activity diagrams can be used to model everything from a high-level business workflow that involve many different use cases, to the details of an individual use case, all the way down to the specific details of an individual method. In a nutshell, activity diagrams can be used to model any type of process.

2. What is the difference between an activity and an action?

An Activity is used to represent a set of actions. In other words, an activity can be decomposed further into a set of activities and/or actions, whereas an action represents a simple non-decomposable piece of the overall behavior being modeled.
An Action is a simple, non-decomposable behavior and is labeled by its name

3. Why is business process modeling important?

Business process models describe the different activities that when combined together support a business process. Business processes typically cut across functional departments (e.g., the creation of a new product will involve many different activities that will combine the efforts of many employees in many departments). Furthermore, from an object-oriented perspective, they cut across multiple objects.

Business processes is a very constructive activity that can be used to make sense out of the gathered requirements. They are very powerful tools for communicating the analyst's current understanding of the requirements with the user.

5. What is an object node?

Object nodes represent the flow of information from one activity to another activity. Activities and actions typically modify or transform objects. Object nodes model these objects in an activity diagram.

- Object node is used to represent an object that is connected to a set of Object Flows
- Object node is labeled by its class name

7. What is the purpose of a fork node?

The fork node is used to split the behavior of the business process into multiple parallel or concurrent flows. Unlike the decision node, the paths are not mutually exclusive, in other words both the paths are executed concurrently.

8. What is CRUD? Why is this useful?

CRUD stands for create, read, update, or delete. This is useful because these are the very basic elemental actions that are needed regarding almost all tasks particularly where the gathering and storage of data is involved.

10. Every association must be connected to at least one _______ and one _________. Why?

Every association must be connected to at least one use case and one actor. This is the definition of an association—it shows two-way communication between the use case and the actor.

11. Explain the following terms. Use layman's language as though you were describing them to a user: (a) actor; (b) use case; (c) system boundary; (d) relationship.

a) <u>Actor</u>: The actor is someone (or occasionally some thing) outside the system that provides information or things (inputs) that the system needs, receives some or all of the things or information (outputs) that the system creates, or the actor can both supply and receive things or information from the system.

b) <u>Use case</u>: A use case is a description of what the system does from the perspective of the user. Since systems can often do many things, the use case often includes a central activity of the system but also shows the many variations that can occur under different circumstances or when varied users interact with it in different ways.

c) <u>System boundary</u>: The system boundary is the dividing point between those items that are included in the system and those that are not. In terms of information technology, the boundary separates the parts of a job that are done by the application from those that are done by humans. In terms of the entire information system that includes both the humans and applications working together to solve business problems, the boundary can be fuzzier. For the efficient development of new systems, however, the boundary shows what the programmers must include and what they should not include in the framework of the project.

d) <u>Relationship</u>: Use cases are not independent and stand alone items in most examples. Pairs (or more) of use cases can be related can be associated -- for example there can be communication between an actor and a use case. Use cases can be extended such that a normal flow of events can be extended to another use case showing an alternate or exceptional flow. An include relationship shows that one or more use case is a part of another more encompassing use case. Finally, a generalization relationship shows that a use case is a specialized example of a more general case

12. Which of the following could be an actor found on a use case diagram? Why?

Ms.Mary Smith
Supplier
Customer
Internet customer
Mr. John Seals
Data entry clerk
Database administrator

In the above cases either of the ones listed below could perform the role of an actor.

a) Ms. Mary Smith
b) Customer
c) Internet Customer
d) Mr. John Seals.

The reason being, an *actor* is not a specific user, but a role that a user can play while interacting with the system. An actor can also represent another system in which the current system interacts. In this case, the actor optionally can be represented by a rectangle with <<actor>> and the name of the system. Basically, actors represent the principal elements in the environment in which the system operates. Actors can provide input to the system; receive output from the system, or both. Sometimes an actor plays a specialized role of a more general type of actor. For example, there may be times when a new patient interacts with the system in a way that is somewhat different than a general patient. In this case, a *specialized actor* (i.e., new patient) can be placed on the model, shown using a line with a hollow triangle at the end of the more general actor (i.e., patient). The specialized actor will inherit the behavior of the more general actor and extend it in some way

13. What are some guidelines for designing a set of use cases? Give two examples of the extend associations on a use case diagram. Give two examples for the include association.

(Assuming that the first question is a broad overview of the chapter) There are several levels of detail from which the student can approach this question. At its highest level, the analyst must consider the major case and its multitude of possible variations; must consider the boundaries, must consider all the stakeholders and all inputs and outputs of the system; and must translate the descriptions accurately into diagrams.

The extend association can include; out-of-stock situation activities during an order; the customer getting a discount for the purchase of two rather than just one of an item in stock.

The uses association can include: a customer uses a shopping cart to gather products from a website shopping center; a "calculate invoice total" use case could use a "look up tax from table" use case.

20. How do you create use cases?

a) Write each step in the SVDPI form
b) Make clear the initiator and receiver of action in each step
c) Write the step from the perspective of an independent observer
d) Write all steps at the same level of abstraction
e) Ensure that each use case represents a sensible set of actions
f) Keep all steps as simple as possible
g) Repeat all actions as necessary.

21. What are some heuristics for creating a use case diagram?

No order is implied by the diagram, so it is not important to sequence the various use cases. It is usually necessary to redraw the diagram several times to make the diagram easy to read. There should be no more than three to nine use cases on the model to keep it simple yet informative. Place the use-cases to minimize crossing lines.

Chapter 6

CLASS DIAGRAMS

1. Give three examples of derived attributes that may exist on a class diagram. How would they be denoted on the class diagram?

Derived attributes, which are attributes that can be calculated or derived; these special attributes are denoted by placing a slash (/) before the attribute's name. For example:

a) -/ age (derived from the difference between birth date and current date.
b) -/ line-item cost (derived from multiplying product price by quantity)
c) -/ grade point average (derived from the dividend of total grade points by total units)

2. Give two examples of aggregation, generalization, and association relationships. How is each type of association depicted on a class diagram?

Aggregation: The spark plug is part of the engine; the engine is part of the car
Generalization: The graduate student is a student; the rock music CD is a CD
Association: customer orders book; UPS/FedEx/Post Office delivers book

Generalization shows that one class (subclass) inherits from another class (superclass), meaning that the properties and operations of the superclass are also valid for objects of the subclass. The generalization path is shown with a solid line from the subclass to the superclass and a hollow arrow pointing at the superclass. generalization relationship occurs when you need to use words such as "is a-kind-of" to describe the relationship.

Aggregation is used when classes actually comprise other classes. For example, think about a doctor's office that has decided to create health care teams that include doctors, nurses, and administrative personnel. As patients enter the office, they are assigned to a health care team that cares for their needs during their visits. Typically, you can identify these kinds of associations when you need to use words like "is a part of" or "is made up of" to describe the relationship.

3. Contrast the following set of terms:

a) Object; class; instance
b) Property; method; attribute
c) Super class; subclass
d) Concrete class; abstract class

Note to instructors, I did not find the term property in this chapter -- nor in the list of key terms. Unless you plan to discuss this in lecture, I'd advise dropping that part of this question.

A class is a general template that we use to create specific instances, or objects, in the application domain. There are two different general kinds of classes of interest during the analysis phase: concrete and abstract.

An attribute represents a piece of information relevant to the instance/class in the domain of the particular application. Method is generally used as another name for operation. Therefore, a method is something that the object can do, a task that can be performed. The term property is not presented as far as I can tell in this chapter, but should be something of a synonym for attribute.

The super class will contain the attributes and operations shared by a number of subclasses. The subclasses inherit the characteristics of the super class and may contain further attributes and operations that belong to each of themselves alone.

A concrete class is a template actually used to create objects. An abstract class may be a generalization of multiple concrete classes that is useful for indicating common features/attributes/methods for the individual concrete classes, but is not used to generate objects itself.

4. Describe to a Business Person the multiplicity of a relationship between two classes.

When a relationship exists between classes, the multiplicity documents how many objects may or must be associated between these classes in the form of minimum and maximum numbers. In general, multiplicity reflects business rules regarding the nature of these relationships among classes.

5. What is an association class?

These are classes formed when a relationship itself has associated properties, especially when its classes share a many-to-many relationship. In these cases, a class is formed, called an association class that has its own attributes and operations.

6. For what purpose is an association class used in a class diagram? Give an example of an association class that may be found in a class diagram that captures students and the courses they have taken.

The association class is used to show a relationship where the relationship itself has attributes -- in other words which has information regarding the relationship itself that should be captured. Students enroll in courses. There is information about that enrollment itself -- when it occurs and the grade received, for example, that would be of interest to keep track of.

7. What are the different types of visibility? How would they be denoted on a class diagram?

Visibility relates to the level of information hiding to be enforced for the attribute. The visibility of an attribute can be public (+), protected (#), or private (-).

8. How do you designate the reading direction of a relationship on a class diagram?

A primary purpose of the class diagram is to show the relationships, or associations, that classes have with one another. When multiple classes share a relationship (or a class shares a relationship with itself), a line is drawn and labeled with either the name of the relationship or the roles that the classes play in the relationship.

For example, the two classes Patient and Appointment can be associated with one another whenever a patient schedules an appointment. Thus, a line labeled "schedules" connects patient and appointment, representing exactly how the two classes are related to each other. Also there would be a small solid triangle beside the name of the relationship. The triangle allows a direction to be associated with the name of the relationship. (Please refer to Fig. 6-2 in the Chapter)

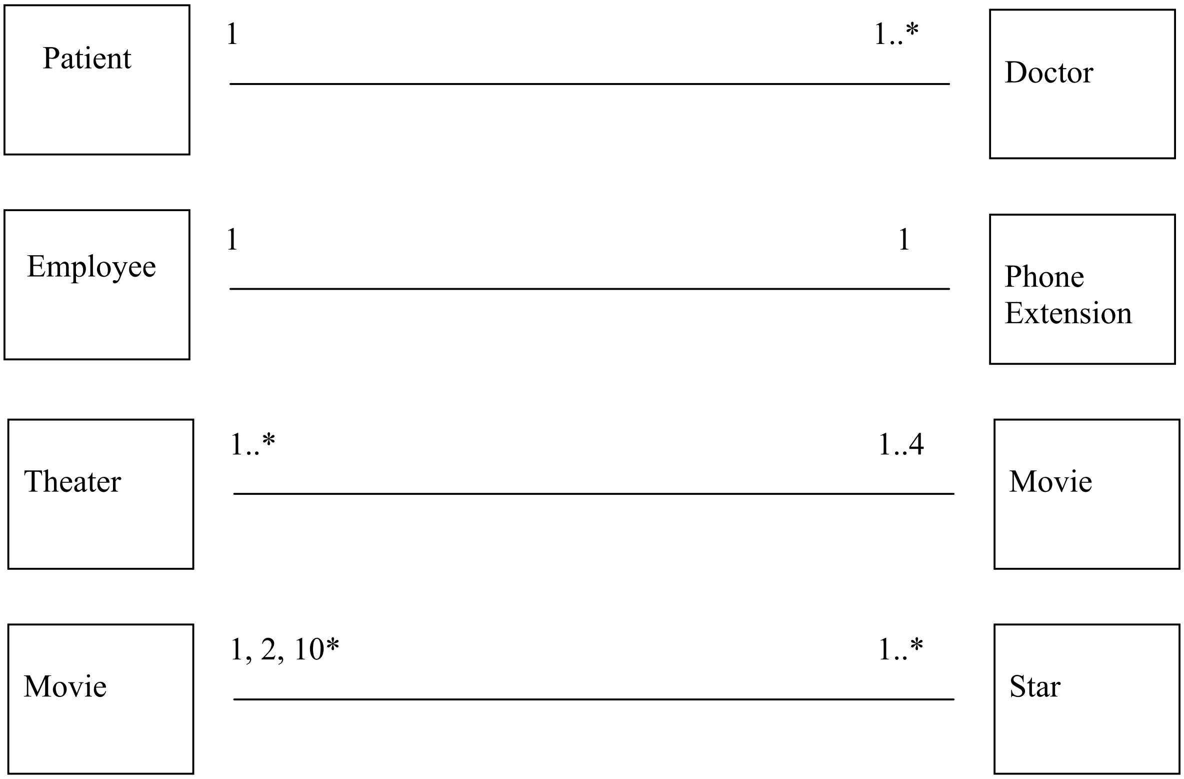
One can also infer that while the classes or objects are generally presented from left to right and top to bottom, there really is no definite sequence of these, so one could start anywhere and move in any direction.

9. Why are assumptions important to a structural model?

The process of developing a structural model involves much learning about the increasing levels of details about a new system. In many cases, the analyst must continue to progress in developing the model, even without knowing all of the business rules. Sometimes the firm may not have encountered situations that cannot be handled informally. Assumptions are important for continuing to work when faced with uncertainty. However, the developer risks having to do a great deal of rework if the assumptions prove wrong. Therefore, the developer should check these assumptions at the earliest possible occasion with users and/or sponsors of the project.

10. Draw the relationships that are described by the following business rules. Include the multiplicities for each relationship.

a) A patient must be assigned to only one doctor, and a doctor can have one or more patients.
b) An employee has one phone extension, and a unique phone extension is assigned to an employee.
c) A movie theater shows at least one movie, and a movie can be shown at up to four other movie theaters around town.
d) A movie either has one star, two co-stars, or more than 10 people starring together. A star must be in at least one movie.



11. Identify the following operations as constructor, query, or update. Which operations would not need to be shown in the class rectangle?

Calculate employee raise (raise percent)
Calculate sick days ( )
Increment number of employee vacation days ( )
Locate employee name ( )
Place request for vacation ( )
Find employee address ( )
Insert employee ( )
Change employee address ( )
Insert spouse ( )

Constructor -- generally available to all classes and not shown

Insert employee ( )

Insert spouse ( ) (assuming that spouse is a class; if spouse is an attribute of employee, then, in effect, inserting spouse would be changing an attribute value and more appropriately called "update spouse" or "add spouse-name/spouse identifier")

Update

Calculate employee raise (raise percent)
Calculate sick days ( )
Increment number of employee vacation days ( )
Place request for vacation ( )
Change employee address ( )

Query

Locate employee name ( )
Find employee address ( )

Chapter 10

DATABASE DESIGN

1. How are a file and database different from each other?

Files are essentially an electronic list of information that is formatted for a particular transaction. Any programs that are written must be developed to work with the file exactly as it is laid out. If there is a need to combine data in a new way, a new file must be created (usually by extracting data from other files) and a program written to work specifically with that new file. Databases, on the other hand, are made up of a collection of data sets that are related to each other in some way. Database management system software creates these data groupings. The DBMS provides access to the data and can usually provide access to any desired subset of data. It is not necessary to write new programs to build a new file in order to retrieve data from the database in a new way.

2. Name five types of files and describe the primary purpose of each file.

1. Master files store the business's or application's core data. The data in a master file is considered fairly permanent, does not change much, and is usually retained for long periods.
2. Look-up files contain reference information that is used primarily during validation processing. A list of valid code values for a field might be referred to during data entry to ensure that a valid code was entered.
3. Transaction files contain information that will be used to update a master file. These files are usually temporary in nature; they are used to collect transactions, the transactions update the master file, and then the transaction files are archived.
4. Audit files are used to help trace changes made to files. An image of a record before and after a change may be written to an audit file so that the change is documented.
5. History files serve as archives for older information that is rarely used. These files can be accessed if necessary, but are usually stored on tape to remove the little-used data from the active data storage areas.

3. What are the differences between sequential and random access files?

Sequential access files allow only sequential file operations to be performed because they actually store the data in sequence according to an identifier. These are very efficient when reports are needed for the whole set of data, but are quite inefficient in locating a specific object of interest. In contrast a random access file is optimized for location of specific objects of interest but are less efficient for reports regarding the entire set of data. Generally, these are stored according to some formula or algorithm rather than based on the sequence of some identifier.

4. **What is the most popular kind of database today? Provide three examples of products that are based on this database technology**

   Relational databases are most popular today due to their ease of use and conceptual simplicity. Examples of relational DBMSs on the market include MS Access, Oracle, DB@, Sybase, Informix, and MS SQL Server.

5. **What is the difference between an end-user database and an enterprise database? Provide an example of each one.**

   An end-user database is one that is designed to run on a PC and is used to create personal database applications. An end-user in sales might develop a Microsoft Access database, for example, to keep track of current and prospective client contacts. An enterprise database is one that is capable of handling huge volumes of information for the entire organization. Applications that serve the entire enterprise can be built upon these enterprise databases. These databases are fast, high capacity, but also complex. Oracle is a vendor of enterprise database management systems.

9. **Why is it important to understand the initial and projected size of a database during design?**

   The design team needs to be sure that the hardware that is specified for the system is adequate to support the size of the database. If inadequate hardware is chosen, the performance of the system will be poor regardless of the 'tuning' techniques that are applied.

10. **Describe what should be considered when estimating the size of a database.**

    The size of the database will be based on the amount of raw data expected, the growth rate of raw data that is expected, and the overhead requirements of the DBMS.

11. **What is referential integrity and how is it implemented in a relational database?**

    Referential integrity refers to the need to make sure that the values linking the table together through the primary and foreign keys are valid and correctly synchronized. For example, if a customer is placing an order, we need to have information on the customer in the customer table, The RDBMS will check to see if there is a record for that customer in the Customer table before it will let an order be entered. Checking for known required relationships helps assure referential integrity.

15. What is the purpose of normalization?

Normalization is a process that optimizes relational data storage for storage efficiency and for minimization of update anomalies. The rules of normalization help assure that the data is stored as efficiently as possible.

21. What are the two dimensions in which to optimize a relational database?

The two dimensions in which to optimize a relational database are for storage efficiency and for speed of access

26. What are some of the nonfunctional requirements that can influence the design of the data management layer?

- Operational Requirements: DAM layer technologies that must be used
- Performance Requirements: DAM layer speed and capacity
- Security Requirements: Access controls, encryption, and backup
- Political & Cultural Requirements: Date formats, currency conversions

30. What is an index and how can it improve the performance of a system?

An index is a small, quickly searchable table that contains values from the table and indicates where in the table those values can be found. System performance is improved with an index because it is no longer necessary to search the entire table for the desired values. The small index table can be quickly searched to reveal exactly where the desired values are stored.

Chapter 11

User Interface Design

Tutorial week 10 of 12
(UI week 1of 2)

1. How can a system be designed to be used by experienced and first time users?

Experienced users prefer systems that focus on ease of use, while novice users prefer systems that are easy to learn. These two goals are not necessarily mutually exclusive. Generally, systems should be set up so that the commonly used functions can be accessed quickly, pleasing the experienced users. To assist the novice users, guidance should be readily available, perhaps through the "show me" functions that demonstrate menus and buttons.

9. Explain three important user interface design principles.

The authors list six principles of user interface design:

1. Layout - the interface should be a series of areas on the screen that are used consistently for different purposes.
2. Content Awareness - the user is always aware of where they are in the system and what information is being displayed.
3. Aesthetics - interfaces should look inviting and should be easy to use.
4. User Experience - experiences users prefer ease of use, while inexperienced users prefer ease of learning.
5. Consistency - users can predict what will happen before a function is performed.
6. Minimize Effort - interface should be simple to use.

10. Describe the basic process of user interface design.

First, identify 'use cases' that describe commonly used patterns of actions that users will perform. These use cases will be valuable in ensuring that the interface permits the users to enact these use cases quickly and smoothly. Next, develop the interface structure diagram, defining the basic structure of the interface (screens, forms, and reports) and how the interface components connect. Third, develop interface standards, the basic design elements that will be used throughout the interface. Fourth, create prototypes of the various interface components (navigation controls, input screens, output screens, forms, and reports). Finally, evaluate the prototypes and make changes as needed.

11. Why do we prototype the user interface design?

Prototyping helps the users and programmers understand how the system will perform. Prototypes can be very useful in helping the users conceptualize how they will actually work with the system, and prototypes can help identify problems or misconceptions in the interface before it is actually implemented.

12. Compare and contrast the three types of interface design prototypes.

Storyboards are really just pictures or drawings of the interface and how the system flows from one interface to another. HTML prototypes are web pages that show the fundamental parts of the system. Users can interact with the system by clicking buttons and entering data, moving from page to page to simulate navigating through the system. Language prototypes create models of the interface in the actual language that will be used to implement the system. These will show the user exactly what the interface will look like, which is not possible with the other two methods.

20. How do you improve the form in Figure 11-4?

The form shown in Fig 11-4 is very unpleasant to users and makes it difficult as its density is too high and it has too much information packed into too small of space.

It is important to have a good font size, which is no less than 8 points, and it's often preferred to be 10 points. Never use all capital letters, except possibly for titles. Color and patterns should be used carefully and sparingly and only when they serve a purpose. (About 10 percent of men are color blind, so the improper use of color can impair their ability to read information.)

The goal is to have a pleasant readability, not art; color and patterns that are use to strengthen the message, not overwhelm it. It's important to focus on:

- User Experience
- Consistency and
- Minimize user efforts

21. What is white space and why is it important?

White space refers to areas on an interface that are intentionally left blank. The more white space on an interface, the less dense the information content. Designer need to try and strike a balance between information content and white space. Some white space is necessary to help the users find things on the interface. Generally, more experienced users need less white space than novice users.

22. Describe the four types of navigation controls.

There are two traditional hardware devices that can be used to control the user interface: the keyboard and a pointing device such as a mouse. There are three basic software approaches for defining user commands: languages, menus, and direct manipulation.

1. Languages: With a command language, the user enters commands using a special language developed for the computer system (e.g., UNIX and SQL both use command languages). Command languages sometimes provide greater flexibility than other approaches because the user can combine language elements in ways not predetermined by developers.
2. Menus: The most common type of navigation system today is the menu. A menu presents the user with a list of choices, each of which can be selected. Menus are easier to learn than languages because a limited number of available commands are presented to the user in an organized fashion.
3. Direct Manipulation: With direct manipulation, the user enters commands by working directly with interface objects.

23. Describe three basic principles of navigation design.

The navigation component of the interface enables the user to enter commands to navigate through the system and perform actions to enter and review information it contains. The three basic principles of navigation design are:

1. Prevent Mistakes: The first principle of designing navigation controls is to prevent the user from making mistakes. Mistakes can be reduced by labeling commands and actions appropriately and by limiting choices.
2. Simplify Recovery from mistakes: No matter what the system designer does, users will make mistakes. The system should make it as easy as possible to correct these errors.
3. Use consistent grammar order: One of the most fundamental decisions is the grammar order. The grammar order should be consistent throughout the system, both at the data element level as well as at the overall menu level.

26. Why are menus the mostly commonly used navigation control?

The most common type of navigation system today is the menu. A menu presents the user with a list of choices, each of which can be selected. Menus are easier to learn than languages because a limited number of available commands are presented to the user in an organized fashion. Clicking on an item with a pointing device or pressing a key that matches the menu choice (e.g., a function key) takes very little effort. Therefore, menus are usually preferred to languages.

33. What do you think are three common mistakes that novice analysts make in navigation design?

One of the hardest things about using a computer system is learning how to manipulate the navigation controls to make the system do what you want. Analysts usually think that:

- The users have read the manual
- The users have attended proper training classes
- The users can seek or rather have external help readily available.

34. What are some of the nonfunctional requirements that can influence the design of the human computer interaction layer?

Perhaps more so than any other aspect of the system, the human-computer interface is affected by a host of nonfunctional requirements. Among them are the following.

- Operational requirements, such as choice of hardware and software platforms
- Performance requirements
- Security requirements such as access controls implemented to protect the objects from unauthorized access
- Cultural and political requirements that include multilingual requirements and unstated social norms

35. What are three fundamental parts of most user interfaces?

1. Navigation mechanism - the way the user gives instructions to the system and tells it what to do.
2. Input mechanism - the way in which the system captures information.
3. Output mechanism - the way the system provides information to the user or to other systems.

36. How can different parts of the interface be consistent?

The navigation controls can be consistent, using the same icon or command to trigger an action throughout the system. Terminology can be consistent throughout the interface. The content portion of the screen that contains forms and reports should also present consistently designed reports and forms. Messages and information in the status area should be specified consistently throughout the system.

37. Why is consistency in design important? Why can too much consistency cause problems?

Consistency means that all parts of the same system work in the same way. This enables the users to predict what will happen because a function in one part of the system works the same way in other parts of the system. Users will be confident as they work with different parts of the system if they can predict the behavior of functions throughout the system. The problem with too much consistency is that sometimes the users don't differentiate forms or reports that look very similar to each other, and inadvertently use the wrong one. So, in these cases, there should be enough unique characteristics to distinguish each form and report from the others.

38. What are use scenarios and why are they important?

Use scenarios describe commonly used patterns of actions that users will perform. Use cases describe how users will interact with the system. Use cases are developed for the most common ways of working through the system. These use cases will be valuable in ensuring that the interface permits the users to enact these use cases quickly and smoothly

CHAPTER 11: User Interface Design
YOUR TURN

Practical Week 11 of 12

1. **Web Page Critique - practical**
   Visit the Web home page for your university and navigate through several of its Web pages. Evaluate the extent to which they meet the six design principles.

   Each university's web site is, obviously, unique. However, your students being from the same university should be in close agreement. At a minimum, they should all consider the following basic principles.
   1. Layout
   2. Content awareness
   3. Aesthetics
   4. User experience
   5. Consistency
   6. Minimal user effort

2. **Search the web to find answers to the following:**

   A. four types of menus found on a web page

The types of menus are Menu bar, drop-down menu, pop-up menu, tab menu, tool bar, and image map.

<u>Menu bar</u>: This is the main menu of the system. It gives a list of commands at the top of the screen and is always displayed on the screen.

- Use the same organization as the operating system and other packages.
- Menu items are always one word, never two.
- Menu items lead to other menus rather than perform action.
- Never allow users to select actions they can't perform

<u>Drop-down menu</u>: This is the second level of menu often from the main menu. This is the menu that drops down immediately below another menu and disappears after one use.

- Menu items are often multiple words
- Avoids abbreviations
- Menu items perform action or lead to another cascading drop-down menu, pop-up menu, or tab menu.

<u>Pop-up Menu</u>: This mainly serves as a short cut to commands for experienced users. This menu pops up, floats over the screen and then disappears after one use.

- Pop-up menus often (not always) invoked by a right click in Windows-based systems.
- These are often overlooked by novice users so usually they should duplicate functionality provided in other menus.

Tab Menu: This menu is often used when the user needs to change several settings or perform certain related commands. This is a multipage menu with one tab for each page that pops up and floats over the screen and continues to remain on the screen until it is closed.

- Menu items should be short to fit on the tab label.
- Avoid more than one row of tabs, because clicking on a tab to open it can change the order of the tabs and in virtually no other case does selecting from a menu rearrange the menu itself.

B. What is meant by the following terms:-

- Direct manipulation

In computer science, **direct manipulation** is a human-computer interaction style which involves continuous representation of objects of interest, and rapid, reversible, incremental actions and feedback. The intention is to allow a user to directly manipulate objects presented to them, using actions that correspond at least loosely to the physical world. An example of direct-manipulation is resizing a graphical shape, such as a rectangle, by dragging its corners or edges with a mouse.

Wiki 22/11/2011

- Interface Metaphor

An **Interface metaphor** is a set of user interface visuals, actions and procedures that exploit specific knowledge that users already have of other domains. The purpose of the interface metaphor is to give the user instantaneous knowledge about how to interact with the user interface. They are designed to be similar to physical entities but also have their own properties (e.g., desktop metaphor and web portals). They can be based on an activity, an object, or a combination of both and work with users' familiar knowledge to help them understand 'the unfamiliar,' and placed in terms the user may better understand.

An example of an interface metaphor is the folders and the file cabinet representation of the file system of an operating system. Another example is the tree view representation of a file system, as in a file manager, that helps a user to use it, given some previous knowledge of recursive structures.

- Wiki 22/11/2011

CHAPTER 11: User Interface Design
YOUR TURN

Practical Week 11 of 12

1. **Web Page Critique - practical**
   Visit the Web home page for your university and navigate through several of its Web pages. Evaluate the extent to which they meet the six design principles.

   Each university's web site is, obviously, unique. However, your students being from the same university should be in close agreement. At a minimum, they should all consider the following basic principles.
   1. Layout
   2. Content awareness
   3. Aesthetics
   4. User experience
   5. Consistency
   6. Minimal user effort

2. **Search the web to find answers to the following:**

   A. four types of menus found on a web page

The types of menus are Menu bar, drop-down menu, pop-up menu, tab menu, tool bar, and image map.

Menu bar: This is the main menu of the system. It gives a list of commands at the top of the screen and is always displayed on the screen.

- Use the same organization as the operating system and other packages.
- Menu items are always one word, never two.
- Menu items lead to other menus rather than perform action.
- Never allow users to select actions they can't perform

Drop-down menu: This is the second level of menu often from the main menu. This is the menu that drops down immediately below another menu and disappears after one use.

- Menu items are often multiple words
- Avoids abbreviations
- Menu items perform action or lead to another cascading drop-down menu, pop-up menu, or tab menu.

Pop-up Menu: This mainly serves as a short cut to commands for experienced users. This menu pops up, floats over the screen and then disappears after one use.

- Pop-up menus often (not always) invoked by a right click in Windows-based systems.
- These are often overlooked by novice users so usually they should duplicate functionality provided in other menus.

Tab Menu: This menu is often used when the user needs to change several settings or perform certain related commands. This is a multipage menu with one tab for each page that pops up and floats over the screen and continues to remain on the screen until it is closed.

- Menu items should be short to fit on the tab label.
- Avoid more than one row of tabs, because clicking on a tab to open it can change the order of the tabs and in virtually no other case does selecting from a menu rearrange the menu itself.

B. What is meant by the following terms:-

- Direct manipulation

In computer science, **direct manipulation** is a human-computer interaction style which involves continuous representation of objects of interest, and rapid, reversible, incremental actions and feedback. The intention is to allow a user to directly manipulate objects presented to them, using actions that correspond at least loosely to the physical world. An example of direct-manipulation is resizing a graphical shape, such as a rectangle, by dragging its corners or edges with a mouse.

Wiki 22/11/2011

- Interface Metaphor

An **Interface metaphor** is a set of user interface visuals, actions and procedures that exploit specific knowledge that users already have of other domains. The purpose of the interface metaphor is to give the user instantaneous knowledge about how to interact with the user interface. They are designed to be similar to physical entities but also have their own properties (e.g., desktop metaphor and web portals). They can be based on an activity, an object, or a combination of both and work with users' familiar knowledge to help them understand 'the unfamiliar,' and placed in terms the user may better understand.

An example of an interface metaphor is the folders and the file cabinet representation of the file system of an operating system. Another example is the tree view representation of a file system, as in a file manager, that helps a user to use it, given some previous knowledge of recursive structures.

- Wiki 22/11/2011

Chapter 11

User Interface Design

3. Why is input validation important?

Input validation is important as all data which is entered into the system needs to be validated in order to ensure their accuracy.

4. Describe five types of input validation methods.

There are six types of validation methods. Five of which are described below:

1. Completeness check: When several fields need to be entered before the form can be processed, completeness check ensures that all required data have been entered. If the completeness check is not done and the information provided id incomplete then the form is returned to the user unprocessed.
2. Range check: A range check permits only numbers between correct values. It ensures that all numeric data entered are within correct minimum and maximum values.
3. Consistency check: Data fields are often interrelated. When this happens, consistency check ensures that the combination of the data is valid. Although it is impossible for the system to know which data are incorrect, it can report the error to the user for correction.
4. Format check: When fields are numeric and contain coded data, format check ensures that the data are of right type. Ideally numeric fields should not permit users to type text data but if this is not possible, then the data entered should be checked to ensure that it is numeric.
5. Database checks: Usually data are compared against information in the database to ensure that they are correct. When this occurs, the database check ensures that the comparison of the data against the data base is correct.

5. What do you think are three common mistakes that novice analysts make in input design?

- Failure to perform proper input validation.
- Use of overly cumbersome forms
- Using the wrong controls for a given input item

6. Compare and contrast check boxes and radio buttons. When do you use one versus the other?

Check boxes are used when several items need to be selected from the list. This presents a complete list of choices each with a square box in front. These are not mutually exclusive.

Radio buttons are used when only one item is selected from a mutually exclusive item list. They are represented by a complete list of mutually exclusive items with a circle in front.

7. Explain three principles in the design of inputs.

Input design means designing the screens used to enter the information, as well as any forms on which users write or type information (e.g., timecards, expense claims). The goal of the input mechanism is to simply and easily capture accurate information for the system. The fundamental principles for input design reflect the nature of the inputs (whether batch or online) and ways to simplify their collection.

- Online versus Batch Processing: There are basically two methods of entering inputs into a system online processing and batch processing. With online processing or transaction processing each input item is entered into the system individually usually at the same time as the event or transaction prompting the input. With batch processing, all the inputs collected over some time period are gathered together and entered into the system at one time in a batch.
- Capture data at the source: Perhaps the most important principle of input design is to capture the data in an electronic format at its original source or as close to the original source as possible.
- Minimize Keystrokes: Another important principle is to minimize keystrokes. Keystrokes cost time and money, whether a customer, user, or trained data-entry operator performs them.

13. Explain three principles in the design of outputs.

The fundamental principles for output design reflect how the outputs are used and ways to make it simpler for users to understand them.

Understand Report Usage: The first principle in designing reports is to understand how they are used. Reports can be used for many different purposes. In some cases—but not very often— reports are read cover to cover because all information is needed.

Manage information load: Most managers get too much information, not too little (i.e., the information load that the manager must deal with is too great). The goal of a well-designed report is to provide all of the information needed to support the task for which it was designed. This does not mean that the report needs to provide all the information available on the subject—just what the users decide they need in order to perform their jobs.

Minimize Bias: No analyst sets out to design a biased report. The problem with bias is that it can be very subtle; analysts can introduce it unintentionally. Bias can be introduced by the way in which lists of data are sorted because entries that appear first in a list may receive more attention than those later in the list.

14. Describe five types of outputs.

There are many different types of reports, such as detail reports, summary reports, exception reports, turnaround documents and graphs.

Detailed report: Lists detailed information about all the items requested.
Summary report: Lists summary information about all items
Turn around document: Turn around documents is a special type of report that is both outputs and inputs. In other words, outputs which turn around and become inputs.
Graphs: Charts that are used in addition to and instead of tables of numbers.
Exception report: Lists detailed information about certain specific items.

15. Compare and contrast batch processing and online processing. Describe one application that would use batch processing and one that would use online processing.

There are basically two methods of entering inputs into a system online processing and batch processing. With online processing or transaction processing each input item is entered into the system individually usually at the same time as the event or transaction prompting the input. Online processing is most commonly used when it is important to have real-time information about the business process. For example, when you reserve an airline seat, the seat is no longer available for someone else to use.

With batch processing, all the inputs collected over some time period are gathered together and entered into the system at one time in a batch. Some business processes naturally generate information in batches. For example, most hourly payrolls are done using batch processing because time cards are gathered together in batches and processed at once. Batch processing also is used for transaction processing systems that do not require real-time information.

16. Why is capturing data at source important?

Perhaps the most important principle of input design is to capture the data in an electronic format at its original source or as close to the original source as possible. In the early days of computing, computer systems replaced traditional manual systems that operated on paper forms. Many business processes still operate this way today. For example, most organizations have expense claim forms that are completed by hand and submitted to an accounting department, which approves them and enters them into the system in batches. There are three problems with this approach. *First*, it is expensive because it duplicates work (the form is filled out twice, once by hand, once by keyboard). *Second*, it increases processing time because the paper forms must be physically moved through the process. *Third*, it increases the cost and probability of error, because it separates the entry from the processing of information; someone may misread the handwriting on the input form, data could be entered incorrectly, or the original input may contain an error that invalidates the information.

17. Describe four devices that can be used for source data automation.

Source data automation refers to using special hardware devices to automatically capture data without requiring anyone to type it. Four devices that can be used are:

1. Bar codes: These automatically scan products and that enter data directly into the computer system.
2. Optical character recognition: This can read printed numbers and text. (e.g., checks)
3. Magnetic stripe readers: This can read information encoded on a stripe of magnetic material similar to a diskette. (e.g., credit cards)
4. Smart cards: These contain microprocessors, microchips and batteries, very similar to credit card sized calculators.

18. When would you use electronic reports rather than paper reports and vice versa?

There are many different types of media used to produce reports. The two dominant media in use today are paper and electronic.

1. Paper report: Being more traditional and permanent, paper is easy to use and is accessible in most situations. Highly portable at least in short reports. Considering the fact that they are inflexible, once information printed of any changes done then the entire report needs to be reprinted which is expensive affair. They also require large storage space and are hard to be moved quickly on long distances.

2. Electronic report: Considering the above-mentioned drawbacks about paper, more organizations are moving to electronic production of reports where reports are printed but stored electronically on file servers or web servers. Electronic reports can be produced on demand; they also enable users to search for certain words. They also provide a means to support ad-hoc reports where users customize the contents of the report at the time the report is generated. Some users still print the electronic report on their own printers, but the reduced cost of electronic delivery over distance and the ease of enabling more users to access the reports than when they were only in paper form usually offsets the cost of local printing.

19. What do you think are three common mistakes that novice analysts make in output design?

- Presenting the user with too much information
- Using reports that are inconsistent with existing business processes
- Using reports that are difficult for the user to read

25. Compare and contrast four types of menus.

The types of menus are Menu bar, drop-down menu, pop-up menu, tab menu, tool bar, and image map.

Menu bar: This is the main menu of the system. It gives a list of commands at the top of the screen and is always displayed on the screen.

- Use the same organization as the operating system and other packages.
- Menu items are always one word, never two.
- Menu items lead to other menus rather than perform action.
- Never allow users to select actions they can't perform

Drop-down menu: This is the second level of menu often from the main menu. This is the menu that drops down immediately below another menu and disappears after one use.

- Menu items are often multiple words
- Avoids abbreviations
- Menu items perform action or lead to another cascading drop-down menu, pop-up menu, or tab menu.

Pop-up Menu: This mainly serves as a short cut to commands for experienced users. This menu pops up, floats over the screen and then disappears after one use.

- Pop-up menus often (not always) invoked by a right click in Windows-based systems.
- These are often overlooked by novice users so usually they should duplicate functionality provided in other menus.

Tab Menu: This menu is often used when the user needs to change several settings or perform certain related commands. This is a multipage menu with one tab for each page that pops up and floats over the screen and continues to remain on the screen until it is closed.

- Menu items should be short to fit on the tab label.
- Avoid more than one row of tabs, because clicking on a tab to open it can change the order of the tabs and in virtually no other case does selecting from a menu rearrange the menu itself.

29. Describe five types of messages.

Messages are the way in which the system responds to a user and informs him or her of the status of the interaction. There are 5 types:

1. Error message: This pops up when the user does something that is not permitted to inform the user that he or she has attempted to do something to which the computer cannot respond.
2. Confirmation message: When a user selects a potentially dangerous operation such as deleting a file, a confirmation message asks the user if he or she is sure of performing this operation.
3. Acknowledgment message: Seldom used this informs the user that the operation or task was successfully completed.
4. Delay message: More common when the action requested is delayed and informs the user that the computer is working properly.
5. Help message: This is present in all systems and provides additional information about the system and its components.

30. What are the key factors in designing an error message?

All messages should be crafted with care, especially the error messages. The key factors, which need to be kept in mind while designing an error message, are:

- Should always explain the problems in polite succulent terms
- Should explain corrective action as clearly and as explicitly as possible
- In case of complicated errors, the error message should display what the user has entered and suggest probable causes for error.
- When in doubt provide either more information than the user needs or ability to get additional information.
- Error message should provide error number

35. What are three fundamental parts of most user interfaces?

1. Navigation mechanism - the way the user gives instructions to the system and tells it what to do.
2. Input mechanism - the way in which the system captures information.
3. Output mechanism - the way the system provides information to the user or to other systems.

Chapter 12

Architecture

1. What is distributed object computing?

From an object-oriented perspective, distributed objects computing (DOC) is the next version of client–server computing. DOC represents a software layer that goes between the clients and servers; hence, it is known as middleware. Middleware supports the interaction between objects in a distributed computing environment.

7. What are the three primary hardware components of any physical architecture?

The three primary hardware components of any physical architecture are:

- Servers
- Client computers
- Network

9. Describe the major nonfunctional requirements and how they influence physical architecture layer design.

The nonfunctional requirements developed in early during analysis (see Chapter 5) play a key role in physical architecture layer design. These requirements are reexamined and refined into more detailed requirements that influence the system's architecture.

There are four primary types of nonfunctional requirements that can be important in designing the architecture:

1. Operational requirements: Specify the operating environment(s) in which system must perform and how those may change over time. That is:

    - Technical environment
    - System Integration
    - Portability
    - Maintainability

2. Performance requirements: Focus on performance issues such as response time, capacity and reliability.

    - Speed
    - Capacity
    - Availability

- Reliability

3. Security requirements: Ability to protect the information system from disruption and data loss, whether caused by international act or a random event.

    - System value
    - Access control
    - Encryption
    - Authentication
    - Virus control

4. Cultural/political requirements: Are specific to the countries in which the system will be used.

    - Multilingual
    - Customization
    - Unstated norms
    - Legal

11. What are the four basic functions of any information system?
The four basic functions of any information system are:

- Data storage
- Data access logic
- Application logic
- Presentation logic

12. What do you think are the most important security issues for a system?

Security is the ability to protect the information system from disruption and data loss, whether caused by an intentional act (e.g., a hacker or a terrorist attack) or a random event (e.g., disk failure, tornado). Security is primarily the responsibility of the operations group—the staff responsible for installing and operating security controls, such as firewalls, intrusion detection systems, and routine backup and recovery operations. Nonetheless, developers of new systems must ensure that the system's security requirements produce reasonable precautions to prevent problems; system developers are responsible for ensuing security within the information systems themselves.

Some of the most important security issues are:

- Access Control Requirements: This means limitations on who can access what data.

- Encryption and Authentication: Defines what data will be encrypted where and whether authentication will be needed for user access.
- Virus Control Requirements: Requirements to control the spread of virus.

15. What six criteria are helpful to use when comparing the appropriateness of computing alternatives?

The six important criteria to use are:

1. Cost of the infrastructure
2. Cost of development.
3. Ease of development
4. Interface capabilities
5. Control and security and
6. Scalability.

18. What is the biggest problem in client-based computing?

The fundamental problem in client-based networks is that all data on the server must travel to the client for processing.

20. Describe the differences between two-tiered, three-tiered and n-tiered architectures.

Since the application has four components, these can be organized between two layers, three layers, or four (and conceivably more) layers. Generally the presentation logic will reside with the client and the database in a more centralized server; however application and data access logic will vary in terms of where they are located.

Two-tiered architecture: A two-tiered architecture is one of the most common. In this case the server is responsible for the data and the client is responsible for the application and presentation. It is called two-tiered because uses only two sets of computers, clients and servers.

Three-tiered architecture: A three-tiered architecture uses three sets of computers In this case, the software on the client computer is responsible for presentation logic, an application server(s) is responsible for the application logic, and a separate database server(s) is responsible for the data access logic and data storage.

N-tiered architecture: An n-tiered architecture uses more than three sets of computers. In this case, the client is responsible for presentation, a database server(s) is responsible for the data access logic and data storage, and the application logic is spread across two or more different sets of servers.