BAOU
Education for All

DR.BABASAHEB AMBEDKAR
OPEN UNIVERSITY

BCA

BACHELOR OF COMPUTER APPLICATION

BCAR-205
System Analysis and Design

SYSTEM ANALYSIS AND DESIGN

BAOU
Education
for All

DR. BABASAHEB AMBEDKAR OPEN UNIVERSITY
AHMEDABAD

Editorial Panel

Author : Prof. (Dr.) Nilesh Modi
Professor and Director,
School of Computer Science,
Dr. Babasaheb Ambedkar Open University,
Ahmedabad

Editor : Dr. Himanshu Patel
Assistant Professor,
School of Computer Science,
Dr. Babasaheb Ambedkar Open University,
Ahmedabad

Language Editor : Dr. Jagdish Vinayakrao Anerao
Associate Professor,
Smt A. P. Patel Arts and
N. P. Patel Commerce College,
Ahmedabad.

ISBN 978-93-91071-01-1

Edition : 2020

Acknowledgment

Every attempt has been made to trace the copyright holders of material reproduced in this book. Should an infringement have occurred, we apologize for the same and will be pleased to make necessary correction/amendment in future edition of this book.

ROLE OF SELF–INSTRUCTIONAL MATERIAL IN DISTANCE LEARNING

The need to plan effective instruction is imperative for a successful distance teaching repertoire. This is due to the fact that the instructional designer, the tutor, the author (s) and the student are often separated by distance and may never meet in person. This is an increasingly common scenario in distance education instruction. As much as possible, teaching by distance should stimulate the student's intellectual involvement and contain all the necessary learning instructional activities that are capable of guiding the student through the course objectives. Therefore, the course / self–instructional material is completely equipped with everything that the syllabus prescribes.

To ensure effective instruction, a number of instructional design ideas are used and these help students to acquire knowledge, intellectual skills, motor skills and necessary attitudinal changes. In this respect, students' assessment and course evaluation are incorporated in the text.

The nature of instructional activities used in distance education self–instructional materials depends on the domain of learning that they reinforce in the text, that is, the cognitive, psychomotor and affective. These are further interpreted in the acquisition of knowledge, intellectual skills and motor skills. Students may be encouraged to gain, apply and communicate (orally or in writing) the knowledge acquired. Intellectual–skills objectives may be met by designing instructions that make use of students' prior knowledge and experiences in the discourse as the foundation on which newly acquired knowledge is built.

The provision of exercises in the form of assignments, projects and tutorial feedback is necessary. Instructional activities that teach motor skills need to be graphically demonstrated and the correct practices provided during tutorials. Instructional activities for inculcating change in attitude and behaviour should create interest and demonstrate need and benefits gained by adopting the required change. Information on the adoption and procedures for practice of new attitudes may then be introduced.

Teaching and learning at a distance eliminate interactive communication cues, such as pauses, intonation and gestures, associated with the face–to–face method of teaching. This is particularly so with the exclusive use of print media. Instructional activities built into the instructional repertoire provide this missing interaction between the student and the teacher. Therefore, the use of instructional activities to affect better distance teaching is not optional, but mandatory.

Our team of successful writers and authors has tried to reduce this.

Divide and to bring this Self–Instructional Material as the best teaching and communication tool. Instructional activities are varied in order to assess the different facets of the domains of learning.

Distance education teaching repertoire involves extensive use of self–instructional materials, be they print or otherwise. These materials are designed to achieve certain pre–determined learning outcomes, namely goals and objectives that are contained in an instructional plan. Since the teaching process is affected over a distance, there is need to ensure that students actively participate in their learning by performing specific tasks that help them to understand the relevant concepts. Therefore, a set of exercises is built into the teaching repertoire in order to link what students and tutors do in the framework of the course outline. These could be in the form of students' assignments, a research project or a science practical exercise. Examples of instructional activities in distance education are too numerous to list. Instructional activities, when used in this context, help to motivate students, guide and measure students' performance (continuous assessment)

PREFACE

We have put in lots of hard work to make this book as user–friendly as possible, but we have not sacrificed quality. Experts were involved in preparing the materials. However, concepts are explained in easy language for you. We have included many tables and examples for easy understanding.

We sincerely hope this book will help you in every way you expect.

All the best for your studies from our team!

SYSTEM ANALYSIS AND DESIGN

Contents

BLOCK 1 : SYSTEM DEVELOPMENT LIFECYCLE AND MODELLING

Unit 1 OVERVIEW OF SYSTEM ANALYSIS AND DESIGN

Introduction, Constraints of a System, Properties of a System, Elements of a System, Types of Systems, Systems Models, Categories of Information

Unit 2 SYSTEM AND WORKING WITH TECHNOLOGY

Introduction, System, Computer Based Business System, Personal Traits of a System Analyst, System Life Cycle, Working with Technology

Unit 3 MODELLING TOOLS FOR SYSTEM ANALYST

Introduction, Role of Data in Business, Modelling with DFD, DFD'S with CASE, Structured Methodology

BLOCK 2 : SYSTEM ANALYSIS AND PHOTOTYPING

Unit 4 SYSTEM DEVELOPMENT LIFE CYCLE

Introduction, Stages of System Development Life Cycle, Project Selection, Feasibility Study, Analysis, Design, Implementation, Post – Implementation and Maintenance, Considerations for Candidate System, Planning and Control for System Success

Unit 5 PRELIMINARY SYSTEM ANALYSIS

Introduction, Fact Finding and Interview, Detailed Analysis, Review and Assignment, Working with People with Technology

Unit 6 SYSTEM REQUIREMENT SPECIFICATIONS & ANALYSIS

Introduction, What is Requirements Determination ?, Fact – Finding Techniques, What is Structured Analysis ?, Pros and Cons of Each Tool

Unit 7 PROTOTYPING AND 4GLS

Introduction, Prototyping, 3GLs and 4GLs, Object Oriented Analysis, Working with People and Technology, System Design

BLOCK 3 : FILE DESIGNING AND TESTING

Unit 8 FILE DESIGN

Introduction, File Design, Database Design, Overview of Implementation, Scheduling and Assigning Tasks

Unit 9 TESTING AND MAINTENANCE

Introduction, Testing, Training, System Maintenance, Management Issues

Unit 10 SYSTEM ADMINISTRATION AND TRAINING

Introduction, Training, Training Systems Operators, User Training, Training methods, Vendor and In–Service Training, In–house Training, Conversion, Conversion Methods, Conversion Plan, Operating Plan

Unit 11 SYSTEM SECURITY, AUDIT AND QUALITY ASSURANCE

Introduction, System Audit, Quality Assurance in SDLC, Specifications for Quality Factors, Software Requirement Specification, Software Design Specification, Software Testing and Its Implementation, Software Support and Maintenance

BLOCK 4 : STRUCTURED SYSTEM DESIGN AND DATA ORIENTED SYSTEM DESIGN

Unit 12 STRUCTURED SYSTEMS ANALYSIS AND DESIGN

Introduction, Procedure Specifications in Structured English, Examples and Cases, Decision Tables For Complex Logical Specifications, Specification Oriented Design Vs Procedure Oriented Design

Unit 13 DATA ORIENTED SYSTEMS DESIGN

Introduction, Entity Relationship Model, E–R Diagrams, Relationships Cardinality and Participation, Normalizing Relations, Various Normal Forms and their Need, Examples of Relational Data Base Design

Unit 14 **OBJECT ORIENTED ANALYSIS AND DESIGN**

Introduction, Object–Oriented Analysis and Design, Basic Terms of Object–Oriented Analysis, UML Diagrams, Use–Case Diagram, Class Diagram, Sequence Diagram, Analysis Modeling

Dr. Babasaheb Ambedkar Open University Ahmedabad

BCAR-205/ DCAR-205

System Analysis and Design

BLOCK 1 : SYSTEM DEVELOPMENT LIFECYCLE AND MODELLING

UNIT 1 OVERVIEW OF SYSTEM ANALYSIS AND DESIGN

UNIT 2 SYSTEM AND WORKING WITH TECHNOLOGY

UNIT 3 MODELLING TOOLS FOR SYSTEM ANALYST

SYSTEM DEVELOPMENT LIFECYCLE AND MODELLING

Block Introduction :

System analysis and design for information system theory emphasizes on closer aspect of all parts of a system. Too often, analyses focuses on only one component and overlooks other equally important components.

In this block, we will learn and study about basic of system and design features with idea on development model. We will cover the topics related to Computer based system, System Life Cycle, Modelling of DFD, Structural methodology and Business data. The student will be given with the knowledge about different rules and criteria's of System life Cycle Model.

In this block, we will learn and study the roles and functions of System Analyst and explain how a model can be better design and implemented. Here the student will be well versatile with Data Flow Analysis and structured modelling concepts. The student will be given with the knowledge about different rules and criteria's of ER diagram and on decision table.

Block Objectives :

After learning this block, you will be able to understand :

- Idea about System
- Concept of Computer based Business system
- Familiarization with personal traits of System Analyst
- Detailed about System Life Cycle
- Role of data in Business
- Knowledge about modelling related to DFD
- Idea about Structured Methodology

Block Structure :

Unit 1 : Overview of System Analysis and Design

Unit 2 : System and Working with Technology

Unit 3 : Modelling Tools for System Analyst

Unit

01

OVERVIEW OF SYSTEM ANALYSIS AND DESIGN

UNIT STRUCTURE

1.0 Learning Objectives

1.1 Introduction

1.2 Constraints of a System

1.3 Properties of a System

1.4 Elements of a System

1.5 Types of Systems

1.6 Systems Models

1.7 Categories of Information

1.8 Let Us Sum Up

1.9 Answer for Check Your Progress

1.10 Glossary

1.11 Assignment

1.12 Activities

1.13 Case Study

1.14 Further Readings

1.0 Learning Objectives :

After learning this unit, you will be able to understand :

- Defining a system
- The role of computer in information systems
- What are the characteristic and element of information system
- What are the various types of information system and models
- What are the different types of specialised information system

1.1 Introduction :

In business, System Analysis and Design refers to the process of examining a business situation with the intent of improving it through better procedures and methods. System analysis and design relates to shaping organizations, improving performance and achieving objectives for profitability and growth. The emphasis is on systems in action, the relationships among subsystems and their contribution to meeting a common goal. Looking at a system and determining how adequately it functions, the changes to be made and the quality of the output are parts of system analysis.

Organizations are complex systems that consist of interrelated and interlocking subsystems. Changes in one part of the system have both anticipated and unanticipated consequences in other parts of the system. The systems approval is a way of thinking about the analysis and design of computer–based applications. It provides a framework for visualizing the organizational and environmental factors that operate on a system. When a computer is introduced into an organization, various functions' and dysfunction's operate on the user as well as on the organization. Among the positive consequences are improved performance and a feeling of achievement with quality information. Among the unanticipated consequences might be a possible threat to employee's job, a decreased morale of personnel due to back of involvement and a feeling of intimidation by users due to computer illiteracy. The analyst's role is to remove such fears and make the system a success. System analysis and design focus on systems, processes and technology.

Systems development can generally be thought of as having two major components: Systems analysis and Systems design. System design is the process of planning a new business system or one to replace or complement an existing system. But before this planning can be done, we must thoroughly understand the old system and determine how computers can best be used to make its operation more effective. System analysis, then, is the process of gathering and interpreting facts, diagnosing problems, and using the information to recommend improvements to the system. This is the job of the systems analyst.

Systems development is systematic process which includes phases such as planning, analysis, design, deployment, and maintenance. Here, in this tutorial, we will primarily focus on ?

- Systems analysis
- Systems design

Systems Analysis : It is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components.

System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem–solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose.

Analysis specifies **what the system should do.**

Systems Design : It is a process of planning a new business system or replacing an existing system by defining its components or modules to satisfy the specific requirements. Before planning, you need to understand the old system thoroughly and determine how computers can best be used in order to operate efficiently.

System Design focuses on **how to accomplish the objective of the system.**

System Analysis and Design (SAD) mainly focuses on ?

- Systems
- Processes
- Technology

❑ **Check Your Progress – 1 :**

1. ________ refers to the process of examining a business situation with the intent of improving it through better procedures and methods.

   a. Computer Graphics b. Database Management System

   c. Programming Languages d. System Analysis and Design

2. System ________ is the process of planning a new business system or one to replace or complement an existing system

   a. analysis b. design c. overview d. error

3. ________ is the process of gathering and interpreting facts, diagnosing problems, and using the information to recommend improvements to the system.

   a. analysis b. design c. overview d. error

What is a System ? The word System is derived from Greek word Systema, which means an organized relationship between any set of components to achieve some common cause or objective.

A system is "an orderly grouping of interdependent components linked together according to a plan to achieve a specific goal."

1.2 Constraints of a System :

A system must have three basic constraints ?

- A system must have some **structure and behaviour** which is designed to achieve a predefined objective.
- **Interconnectivity** and **interdependence** must exist among the system components.
- The **objectives of the organization** have a **higher priority** than the objectives of its subsystems.

For example, traffic management system, payroll system, automatic library system, human resources information system.

1.3 Properties of a System :

A system has the following properties.

Organization : Organization implies structure and order. It is the arrangement of components that helps to achieve predetermined objectives.

Interaction : It is defined by the manner in which the components operate with each other.

For example, in an organization, purchasing department must interact with production department and payroll with personnel department.

Interdependence : Interdependence means how the components of a system depend on one another. For proper functioning, the components are coordinated and linked together according to a specified plan. The output of one subsystem is the required by other subsystem as input.

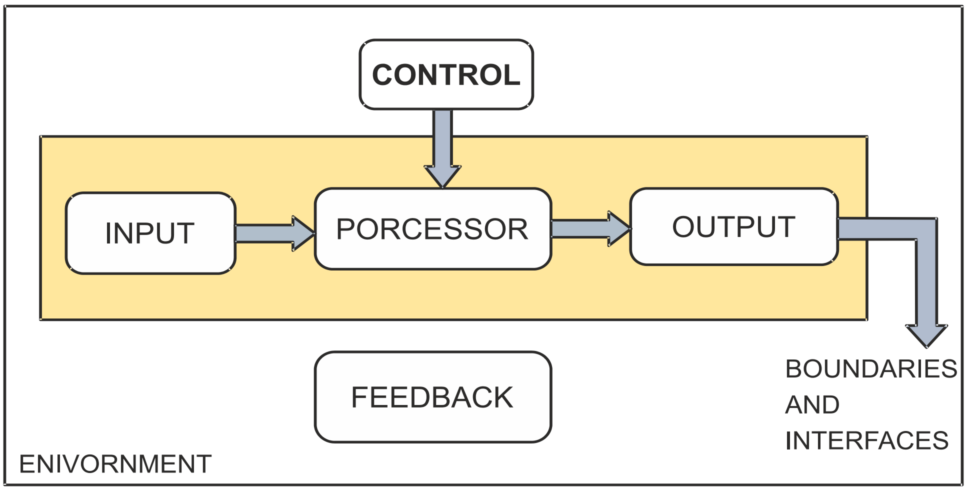
Integration : Integration is concerned with how system components are connected together. It means that the parts of the system work together within the system even if each part performs a unique function.

Central Objective : The objective of system must be central. It may be real or stated. It is not uncommon for an organization to state an objective and operate to achieve another.

The users must know the main objective of a computer application early in the analysis for a successful design and conversion.

1.4 Elements of a System :

The following diagram shows the elements of a system.



❖ **Outputs and Inputs :**

- The main aim of a system is to produce an output which is useful for its user.
- Inputs are the information that enters into the system for processing.
- Output is the outcome of processing.

❖ **Processor(s) :**

- The processor is the element of a system that involves the actual transformation of input into output.
- It is the operational component of a system. Processors may modify the input either totally or partially, depending on the output specification.
- As the output specifications change, so does the processing. In some cases, input is also modified to enable the processor for handling the transformation.

❖ **Control :**

- The control element guides the system.
- It is the decision–making subsystem that controls the pattern of activities governing input, processing, and output.
- The behaviour of a computer System is controlled by the Operating System and software. In order to keep system in balance, what and how much input is needed is determined by Output Specifications.

❖ **Feedback :**

- Feedback provides the control in a dynamic system.
- Positive feedback is routine in nature that encourages the performance of the system.
- Negative feedback is informational in nature that provides the controller with information for action.

❖ **Environment :**

- The environment is the "super system" within which an organization operates.

- It is the source of external elements that strike on the system.
- It determines how a system must function. For example, vendors and competitors of organization's environment may provide constraints that affect the actual performance of the business.

❖ **Boundaries and Interface :**

- A system should be defined by its boundaries. Boundaries are the limits that identify its components, processes, and interrelationship when it interfaces with another system.
- Each system has boundaries that determine its sphere of influence and control.
- The knowledge of the boundaries of a given system is crucial in determining the nature of its interface with other systems for successful design.

❑ **Check Your Progress – 2 :**

1. ________ means an organized relationship between any set of components to achieve some common cause or objective.

   a. Process　　b. System

   c. Technology　　d. All of the above

2. From the given below, ________ is not the property of the System.

   a. Organization　　b. Interdependence

   c. Structure　　d. Integration

3. From the given below, identify the elements of the system.

   a. Input　　b. Feedback

   c. Output　　d. All of the above

1.5 Types of Systems :

The systems can be divided into the following types ?

❖ **Physical or Abstract Systems :**

- Physical systems are tangible entities. We can touch and feel them.
- Physical System may be static or dynamic in nature. For example, desks and chairs are the physical parts of computer center which are static. A programmed computer is a dynamic system in which programs, data, and applications can change according to the user's needs.
- Abstract systems are non–physical entities or conceptual that may be formulas, representation or model of a real system.

❖ **Open or Closed Systems :**

- An open system must interact with its environment. It receives inputs from and delivers outputs to the outside of the system. For example, an information system which must adapt to the changing environmental conditions.
- A closed system does not interact with its environment. It is isolated from environmental influences. A completely closed system is rare in reality.

❖ **Adaptive and Non–Adaptive System :**

- Adaptive System responds to the change in the environment in a way to improve their performance and to survive. For example, human beings, animals.
- Non–Adaptive System is the system which does not respond to the environment. For example, machines.

❖ **Permanent or Temporary System :**

- Permanent System persists for long time. For example, business policies.
- Temporary System is made for specified time and after that they are demolished. For example, A DJ system is set up for a program and it is dissembled after the program.

❖ **Natural and Manufactured System :**

- Natural systems are created by the nature. For example, Solar system, seasonal system.
- Manufactured System is the man–made system. For example, Rockets, dams, trains.

❖ **Deterministic or Probabilistic System :**

- Deterministic system operates in a predictable manner and the interaction between system components is known with certainty. For example, two molecules of hydrogen and one molecule of oxygen make water.
- Probabilistic System shows uncertain behavior. The exact output is not known. For example, Weather forecasting, mail delivery.

❖ **Social, Human–Machine, Machine System :**

- Social System is made up of people. For example, social clubs, societies.
- In Human–Machine System, both human and machines are involved to perform a particular task. For example, Computer programming.
- Machine System is where human interference is neglected. All the tasks are performed by the machine. For example, an autonomous robot.

❖ **Man–Made Information Systems :**

- It is an interconnected set of information resources to manage data for particular organization, under Direct Management Control (DMC).
- This system includes hardware, software, communication, data, and application for producing information according to the need of an organization.

❖ **Man–made information systems are divided into three types :**

- **Formal Information System :** It is based on the flow of information in the form of memos, instructions, etc., from top level to lower levels of management.
- **Informal Information System :** This is employee–based system which solves the day–to–day work–related problems.
- **Computer Based System :** This system is directly dependent on the computer for managing business applications. For example, automatic library system, railway reservation system, banking system, etc.

❑ **Check Your Progress – 3 :**

1. ________ systems are tangible entities. We can touch and feel them.

   a. Abstract　　b. Physical

   c. Virtual　　d. None of the above.

2. ________ system do not interact with the environment.

   a. Close　　b. Open

   c. Adaptive　　d. Non-Adaptive

3. ________ System responds to the change in the environment in a way to improve their performance and to survive.

   a. Close　　b. Open　　c. Adaptive　　d. Non-Adaptive

1.6 Systems Models :

❖ **Schematic Models :**

- A schematic model is a 2–D chart that shows system elements and their linkages.
- Different arrows are used to show information flow, material flow, and information feedback.

❖ **Flow System Models :**

- A flow system model shows the orderly flow of the material, energy, and information that hold the system together.
- Program Evaluation and Review Technique (PERT), for example, is used to abstract a real–world system in model form.

❖ **Static System Models :**

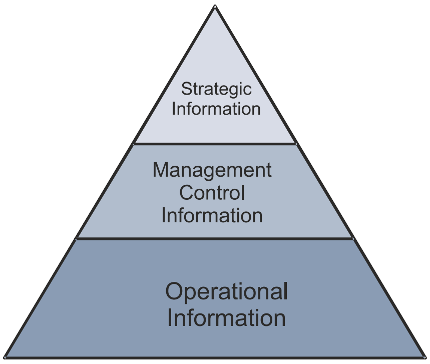
- They represent one pair of relationships such as *activity–time or cost–quantity.*
- The Gantt chart, for example, gives a static picture of an activity–time relationship.

❖ **Dynamic System Models :**

- Business organizations are dynamic systems. A dynamic model approximates the type of organization or application that analysts deal with.
- It shows an ongoing, constantly changing status of the system. It consists of ?
  - o Inputs that enter the system
  - o The processor through which transformation takes place
  - o The program(s) required for processing
  - o The output(s) that result from processing.

1.7 Categories of Information :

There are three categories of information related to managerial levels and the decision managers make.

Volume of Information	Type of Information	Information Level	Management Level	System Support
Low Consensed	Unstrucuted	 Strategic Information	Upper	DSS
Medium Moderately Processed	Moderately Structured	Management Control Information	Middle	MIS
Large Detail Reports	Highly Structured	Operational Information	Lower	DPS

❖ **Strategic Information :**

- This information is required by topmost management for long range planning policies for next few years. For example, trends in revenues, financial investment, and human resources, and population growth.
- This type of information is achieved with the aid of Decision Support System (DSS).

❖ **Managerial Information :**

- This type of Information is required by middle management for short and intermediate range planning which is in terms of months. For example, sales analysis, cash flow projection, and annual financial statements.
- It is achieved with the aid of Management Information Systems (MIS).

❖ **Operational Information :**

- This type of information is required by low management for daily and short–term planning to enforce day–to–day operational activities. For example, keeping employee attendance records, overdue purchase orders, and current stocks available.
- It is achieved with the aid of Data Processing Systems (DPS).

❑ **Check Your Progress – 4 :**

1. A ________ model is a 2–D chart that shows system elements and their linkages.

   a. systematic b. flow–system

   c. static system d. None of the above

2. PERT is an example of ________.

   a. systematic b. flow–system

   c. static system d. None of the above

3. ________ represents activity-time or cost-quantity.

   a. systematic b. flow–system

   c. static system d. None of the above

4. MIS stands for ________

   a. Manage Internet Services b. Main Information System

   c. Management Information Service d. Management Information System

1.8 Let Us Sum Up :

A system is orderly grouping of interdependent components linked together according to a plan to achieve a specific objective. Its main characteristics are organization, interaction, interdependence, integration and a central objective. To construct a system, system analyst must consider its elements– input and output, processors, control, feedback, and environment. Systems are classified as physical or abstract, open or closed, and man–made information systems. A system may be schematic, static or dynamic. An information system is an open system that allows inputs and facilitates interaction with the user. The main characteristic of an open system is input from outside, processing, output, and operation in cycles through feedback, differentiation, and equifinality. Three level of information in organization that require a special type of information system. Strategic information system for long range planning policies and upper management. Managerial information system helps middle management and department heads in policy implementation and control. Operational information system helps the daily information needed to operate the business. Future emphasises on the decision support system not on information processing, it requires a computer aided environment and accentuates a combined man and machine and decision environment.

1.9 Answer for Check Your Progress :

❑ **Check Your Progress 1 :**

1 : d   **2 :** b   **3 : a**

❑ **Check Your Progress 2 :**

1 : b   **2 :** c   **3 :** d

❑ **Check Your Progress 3 :**

1 : b   **2 :** a   **3 :** c

❑ **Check Your Progress 4 :**

1 : a   **2 :** b   **3 :** c   **4 :** d

1.10 Glossary :

1. **System :** A system is "an orderly grouping of interdependent components linked together according to a plan to achieve a specific goal."
2. **Systems Analysis :** It is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components.
3. **Systems Design :** It is a process of planning a new business system or replacing an existing system by defining its components or modules to satisfy the specific requirements.
4. **Formal Information System :** It is based on the flow of information in the form of memos, instructions, etc., from top level to lower levels of management.
5. **Informal Information System :** This is employee–based system which solves the day–to–day work–related problems.

1.11 Assignment :

1. Define system. Give examples.
2. What is man–made information system ?
3. Explain the features of a system.

1.12 Activities :

- Elaborate the different types of systems.
- A system leads to a lot of planning and less of implementation. Do you agree, justify your answer ?

1.13 Case Study :

- Elaborate the following systems in brief :
  - o Payroll System.
  - o Library Automation System

1.14 Further Readings :

1. Analysis and Design and Information System, Rajaraman.
2. Analysis and Design of Information System 2nd Ed. Senn.
3. Introducing Systems Analysis and Design, Lee.

Unit 02

SYSTEM AND WORKING WITH TECHNOLOGY

UNIT STRUCTURE

2.0 Learning Objectives

2.1 Introduction

2.2 System

2.3 Computer Based Business System

2.4 Personal Traits of a System Analyst

2.5 System Life Cycle

2.6 Working with Technology

2.7 Let Us Sum Up

2.8 Answers for Check Your Progress

2.9 Glossary

2.10 Assignment

2.11 Activities

2.12 Case Study

2.13 Further Readings

2.0 Learning Objectives :

After learning this unit, you will be able to understand :

- The fundamentals of System.
- Explain Computer Based Business system.
- Outline Personal traits of System Analyst.
- Explain working of System Life cycle.
- Outline basics of working with technology.

2.2 Introduction :

We are surrounded by systems. There are many systems such as the transportation system, the distribution of goods and services system, education system, manufacturing and almost every other human economic activity. From the conceptual point, we can view the economy and business as a set of interrelated systems. While learning about system analysis and design, we will discuss certain applications that will lead to solve many queries and problems. Such mechanism can be applied to achieve extra knowledge and experience about real system analysis and design principles. A good example of an important technique is learning how to represent a system using dataflow diagrams. Information is an organisational resource that must be managed as carefully as other resources. Costs are associated with information processing. It must be managed to take full advantage of its potential. In this unit, we will discuss the concept of system and system life cycle.

2.2 System :

A system is a set or group of components that interact to accomplish some purpose. There are a number of systems around us. For example, complex nervous system, which comprises a set of parts or components that is brain, spinal cord, nerves, etc. A system is a set of interrelated components working together for a common purpose.

System = Data + Processing of data

A system is a combination of resources working together to transform inputs into usable outputs.

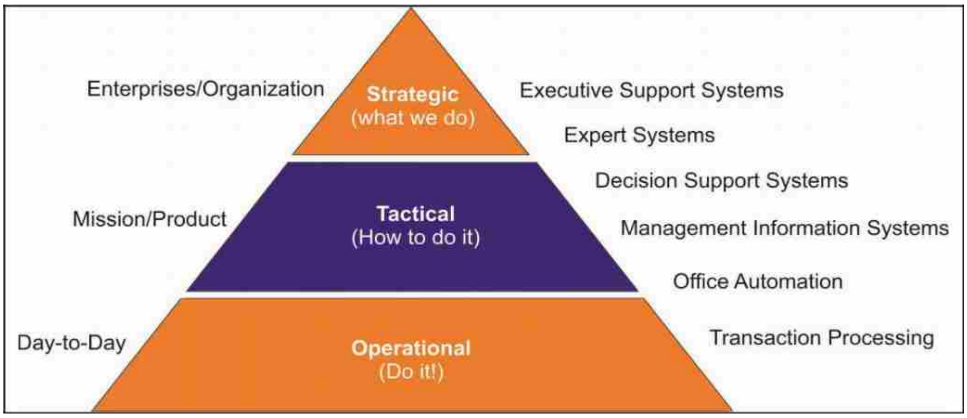


Fig. 2.1 : Overview of System

Information is an organisational resource, which must be managed as carefully as other resources. Organisations are complex systems composed of interrelated and interdependent subsystems. Organisational subsystems are said to be interrelated and interdependent when a change in one subsystem affects other subsystems. An organisational boundary separates the system from its environment. The systems and subsystems boundaries and environments influence information system analysis and design.

The three levels of management in organisations are :

1. Strategic management
2. Middle management
3. Operations management

Take basic control model of the system, which consists of :

- **Standards :** What comprises acceptable performance
- **Measurement :** Measuring actual performance
- **Compare :** Comparing actual performance
- **Feedback :** A method for feedback

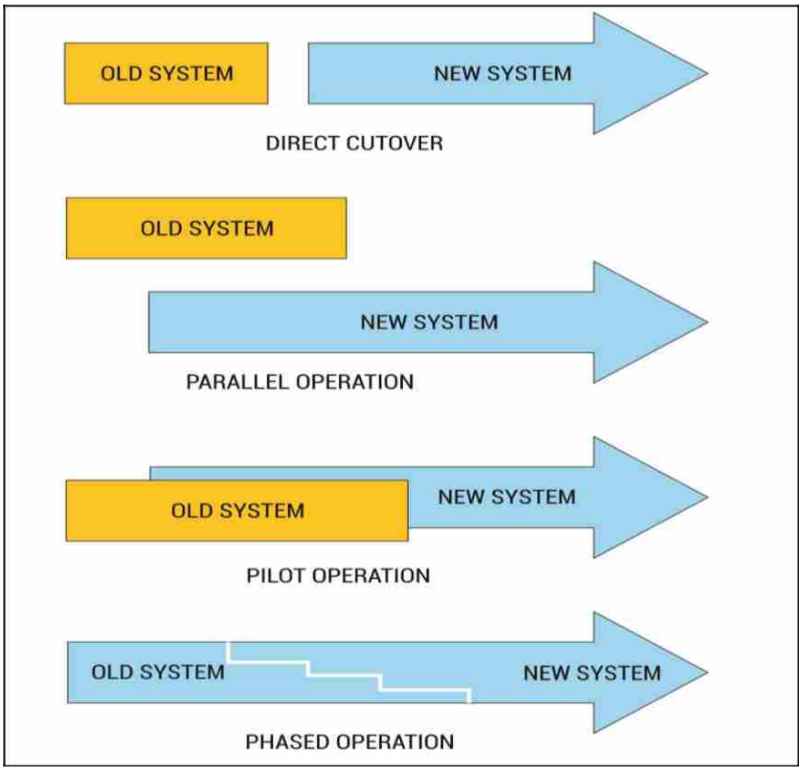


Fig. 2.2 : System Changeover

- It seems that personal systems do not share information and cannot be run on distributed mechanism as these are framed to work and run only on personal computer.
- There are embedded systems which can work with only single processor or can be further run with integrated processors.
- It is found that distributed systems able to run system software on distributed group of linked processors which are connected through network system analysis.

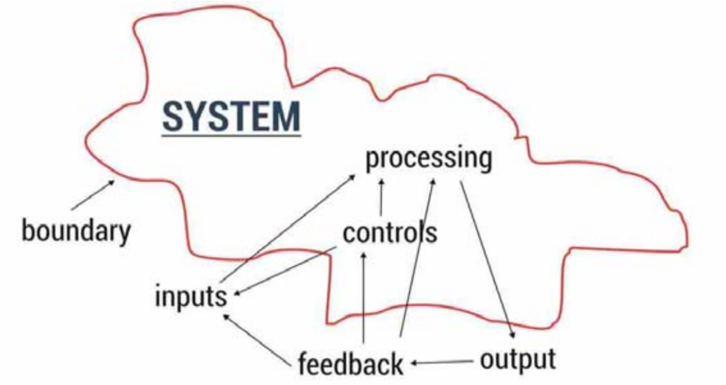


Fig. 2.3 : Generic System Model (with Six Components)

Examples :

- Automobile
- Student registration system

SAD stands for System Analysis and Design, which gives answers to questions :

- What to be done ?
- Who will do ?

- When it will be done ?
- How it will be done ?

❖ **Definition :**

A system is a way of thinking about organisation and their problems. It also involves set techniques that help in solving problems.

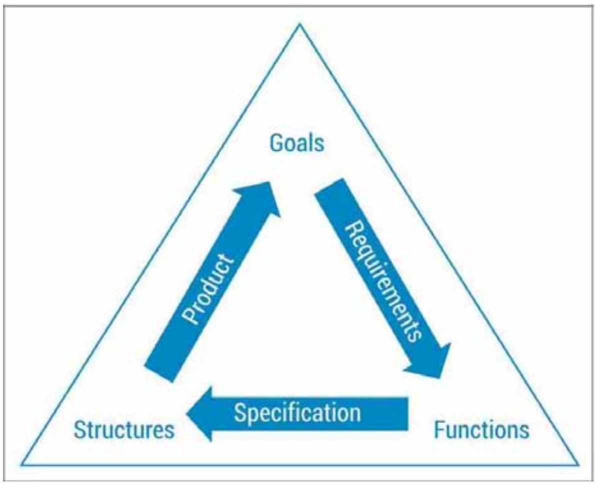


Fig 2.4 : Structure of System

A system is a mixture of several processes which when combined together will do activities or will complete particular task.

It is an arrangement of organized mechanism which are mixed together as per the plan to obtain certain ideas. It is seen that while studying system concepts, there exists following basic complications:

- A system should be such produce that it should carry fixed activities.
- In this, interrelationships as well as interdependence both occurs as per the components.
- A system must be user friendly and it should maintain security from unauthorised users as well as making selected data.

According to the definition of system, some characteristics are present in all systems.

The important characteristics of a system are as follows :

- **Organisation :** It shows arrangement of structure and order where components are arranged so as to help in getting objectives.

  Similarly, a computer system is so produced that should contain input device, a processing unit and output device having storage units. If all these components are joined, they work as complete system that will generate information.
- **Interaction :** In a system, interaction relates to coordinating activities in which every component gets mixed with each other. So formally, interaction explains about the working of every component and their interaction and shows how such components can communicate among each other.

In case of an organisation, we can say that purchasing relates to interaction with production, advertising, sales and payrolls that will serve as input device which can do many problems.

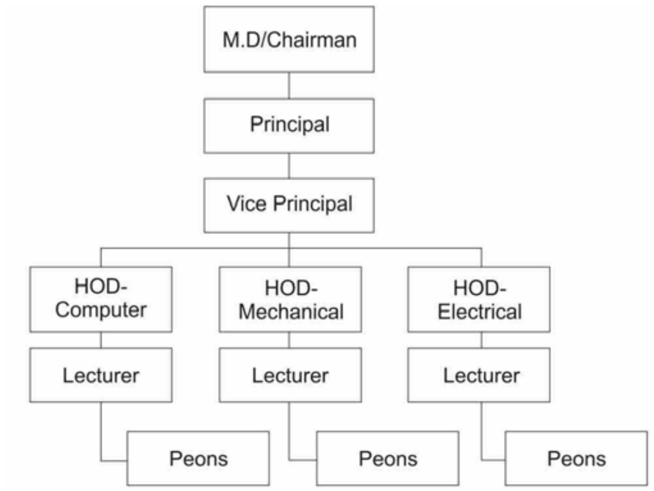


Fig. 2.5 : Organisation Structure for College System

So we finally say that interdependence refers to such parts of an organisation which depends on one another. These parts are coordinated as well as linked as per the prescribed plan. It is analysed that a subsystem mainly dependent on input of other subsystem in order to work well. So practically, here the output of particular subsystem acts as an input for other subsystem.

So functionally, it is seen that in a computer system, there are three individual units input, system unit and output. No subsystem can function in isolation because it is designed by analyst, programmed and tested by programmer and run by computer operator as shown in below fig.1.6

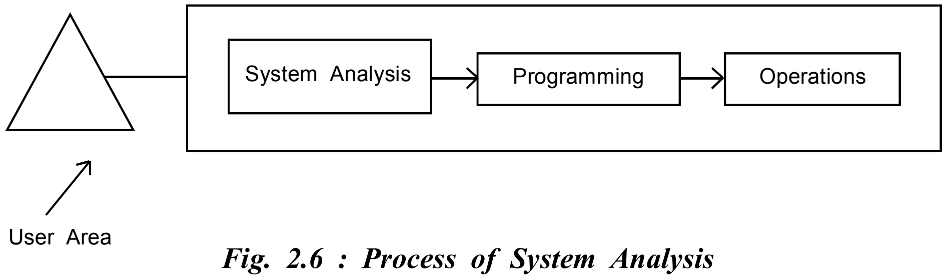


Fig. 2.6 : Process of System Analysis

- **Integration :** Integration is concerned with how a system is tied together in order to achieve common goal, thus forming integration. Integration refers to holism of system. Synthesis follows analysis to achieve the central objective of organisation. It means that parts of system work together within system even though each part performs unique function.
- **Central Objective :** The stated object of the system could differ based on the policy of the company. The user should develop a central objective by considering real objectives and stated objectives.

The important point is that the users must know the central objective of the computer application in analysis for successful design and conversion. This means that the analyst must work around the obstacles to identify the real objective of proposed change.

❖ **Elements of System :**

To reconstruct a system following elements of system should be considered :

- **Outputs and Inputs :** A major function of a system is to produce the output as per the user's requirement. In systems, output is projected as finished products and information in shape of services.

Similarly, inputs are elements which will allow the system to work to generate an output which could be in shape of material or information. The elements are feed inside the system for processing which will show output as processed information.

- **Processors :** It is the main organiser of input output operations that takes care about all operations inside the system. The work of the processor should be such that it should accept input in given form and output the result in desired format.

It is the operational component of the system. Processors may modify input totally or partially depending on specifications of output.

- **Control :** The control elements guide the system. The control element controls the working of the system at all stages. It is necessary to control input, process and output, continuously, in order to get desired results.
- **Feedback :** Feedback is a method that helps to compare output produced with expected results and make necessary changes in the process or input in order to reduce the difference between output produced and output expected. Input information is fed back to input for deliberation. After the output is compared against performance standards, changes can result in input or output processing and output. Feedback may be positive or negative.
- **Environment :** All the things that are outside the system are called environment of the system. The environment does affect working or progress of the system. The system should be sensitive to the changes in its environment. It determines how a system functions.

❖ Types of System :

There are different types of systems as per their use and component structure. Following is the detailed study of different types of systems:

❖ Physical and Abstract System :

Physical System :

The physical system could be static or dynamic in nature. Static means systems that do not change as far as working or life of the system is concerned. On the other hand, dynamic system may change due to processing of the system.

For example, in computer system, the hardware parts are static but the data that changes due to processing is dynamic. These together form physical system along with the programs controlling the data.

Abstract System :

The systems that are represented conceptually (i.e. non–existing) non – physical systems are called abstract systems. Abstract systems are prepared for studying the physical system. The computer itself is a physical system and its block diagram is called the abstract system.

❖ Open and Closed System :

Open System :

Another classification of system is based on their degree of independence. An open system is a one, which does not provide for its own control or modification. It does not supervise itself so it needs to be supervised by people.

For example, if the high–speed printer used with computer system did not have a switch to sense whether paper is in printer, then a person would have to notice when the paper runs out and signal the system (push a switch) to stop printing. It has many interfaces to its environment. It permits interaction across its boundary.

Closed System

A closed system is one, which automatically controls or modifies its own operation by responding to data generated by the system itself.

For example, high–speed printers used with computer systems usually have a switch that senses whether there is paper in the printer. If the paper runs out, the switch signals to stop printing.

An Information System :

- Supports policies and procedures
- Contains 3 components such as data, people, procedures along with 6 general system components

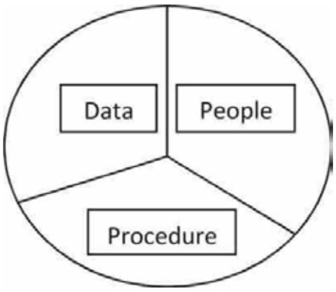


Fig. 2.7 : An Information System

An Automated Information System :

It is a sort of fabricated system which can be applied with one or more persons so as to help them while doing certain work by using certain hardware as well as software

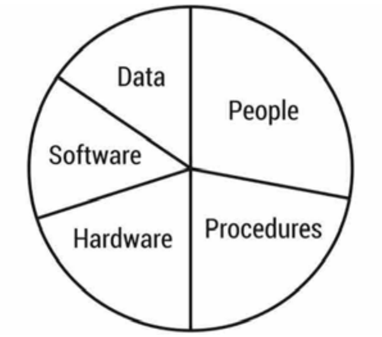


Fig. 2.8 : An Automated Information System

It is studied that an automated information system contains three characteristics data, input and output as shown :

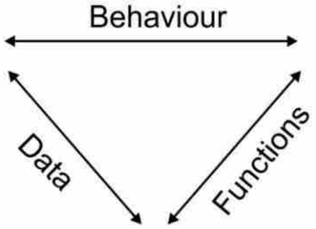


Fig. 2.9 : Characteristics of Automated Information System

From the automated information system, the following features exist :

- The data be considered as input, output and can be kept on computer
- The system will perform certain transformational action
- The system will observe certain effects on interaction

❑ **Check Your Progress – 1 :**

1. The most important factor in respect to management information system is _________.

   a. System c. Process b. Data d. None of these

2. Which among the following is not a characteristic of well–designed system ?

   a. Practical c. Expensive b. Effective d. Flexible

2.3 Computer Based Business System :

Many information systems now use computers for manipulating information. They are sometimes called Computer–Based Information Systems.

The general view of the computer based information system is as shown in below figure.

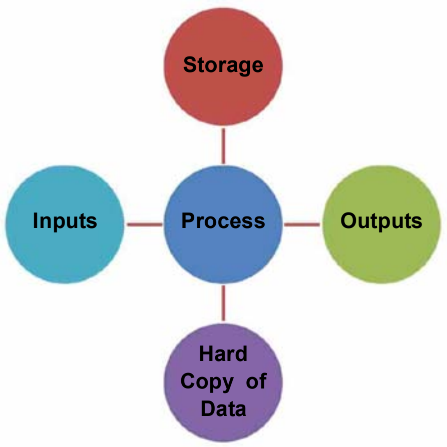


Fig. 2.10 : Basic Components of a Computer – Based Information System

A computer based information system involves six– interdependent elements :

- Hardware (machine)
- Software (instructions or programs)
- People (programme mangers or users)
- Procedures (rules)
- Data (input)
- Output information

People of course are an essential component of any information system. Information is produced and used by people in an organisation. People use this information in their activities to make decisions about what is to be done.

For example, Railway reservation system, which is a computer–based information system, involves all components mentioned above.

For RS (railway reservation system) input data in reservation forms is given by different railway passengers, then these forms are processed one by one according to rules and procedures, using some standard software on computer and at last it gives output information such as reservation tickets, reservation tickets and reservation charts according to different train numbers.

Advantages of Computer – Based Information System By using computer based information systems, it is possible :

- To store large volume of data
- To perform different operations speedily
- To improve processing power of organisation
- To maintain security of data from unauthorised users
- To produce different output information easily

❑ Check Your Progress – 2 :

1. Which is not the feature of Computer– Based Information System ?

   a. Storing small volume of data

   b. Performing several operations speedily

   c. Maintaining security

   d. All of these

2.4 Personal Traits of a System Analyst :

System analysis is central to the whole of the system development. It includes gathering the necessary data and developing plans for new systems. This is not an easy task because many people need to be satisfied and many conflicts resolved. The system analyst is the person (or persons) who guide the analysis, design and implementation and maintenance of a given system.

In design, system analyst is the person who will take care of all activities related to development and design of information system. While doing all jobs, a system analyst will always match the objectives of designing with goals of organisation. The system analyst should have following qualities :

- Ability to solve problems
- Good communication skills
- Should have an adequate computer/IT experience
- Should be self–disciplined and motivated
- Have qualities of project management

❖ Responsibilities of System Analyst :

Roles and responsibilities of system analyst differ from organisation to organisation. They should have following responsibilities:

1. **Analysis of particular system :** This quality will cover know how of system. It is more with respect to collecting information and dealing with particular requirements.
2. **Designing and Analysis of System :** Apart from analysing a system, the system analysis needs to design certain new system/applications.

3. **Analysis, design and programming of system :** This is a crucial activity of a system analyst where he/she needs to analyse, design and program an existing or new system. With this, a system analyst should have sufficient qualities and qualifications to handle certain responsibilities at various stages of life cycle. Apart from technical knowledge and experience, a system analyst should possess knowledge on following mentioned domains :

- **Business knowledge :** An analyst should be well versatile with required business applications, so that he/she can develop and understand new applications.
- **Interpersonal skills :** These are required skills that an analyst should have as it can be used at various stages of development process in order to interact with customers and to extract their requirements.
- **Problem solving skills :** To solve system related analysis and designing problems, a system analyst should have certain problem solving skills that should be suggested as an alternative of problems which occurs at various stages of development.
- **System analysts can also be known as business analysts :** Business analysts study the overall business and information needs of an organisation in order to develop solutions to business and related technology problems. Business analysts are the liaisons between technology staff and management. A business analyst's role is usually undertaken prior to the system design, building and programming stages of the systems development process. Business analysts are also needed for your changing business environment as upgrades, new software, or new departments that need to be integrated into the existing systems are always part of the ever–changing world.

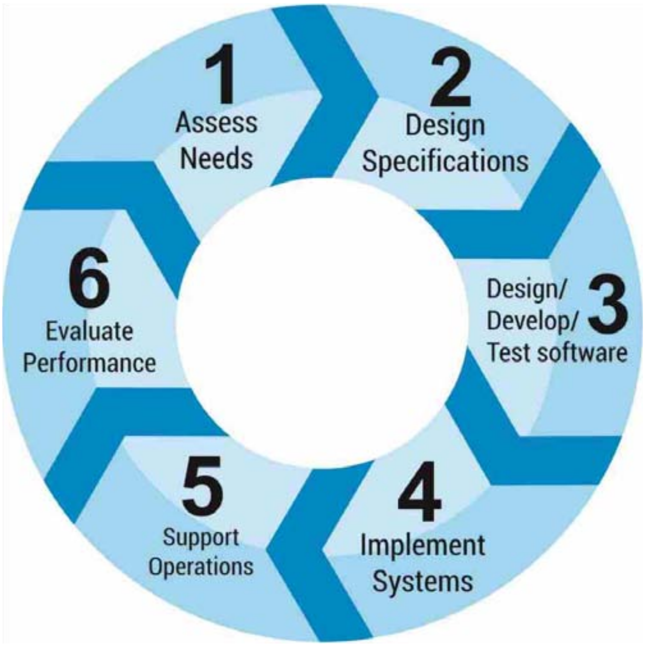


Fig. 2.11 : Responsibilities of a Business Analyst

The figure above shows the responsibilities of a business analyst, which is same as that of system analyst. System in other words can be known as a business or an organisation. Progress of any system (business) depends upon the analyst and her/his capabilities to understand the situation or problem perfectly.

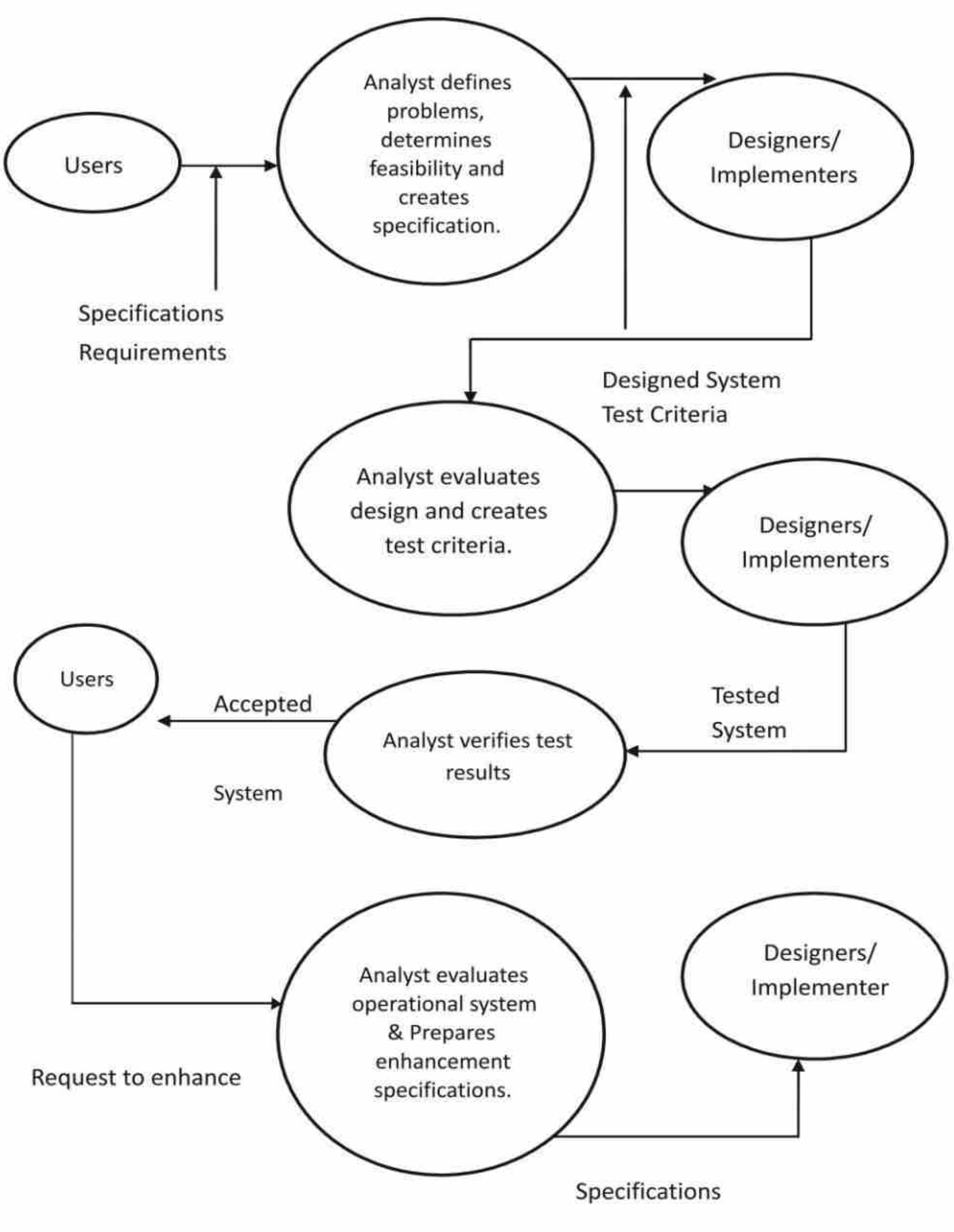

Users
Analyst defines problems, determines feasibility and creates specification.
Designers/ Implementers
Specifications Requirements
Designed System Test Criteria
Analyst evaluates design and creates test criteria.
Designers/ Implementers
Users
Accepted
System
Analyst verifies test results
Tested System
Analyst evaluates operational system & Prepares enhancement specifications.
Designers/ Implementer
Request to enhance
Specifications

Fig. 2.12 : Stages of System Analysis

❖ **Skills expected in a System Analyst :**

The system analyst is a key person in system analysis and design. He should be a knowledgeable person. He should possess certain indispensable skills.

The skills required in a system analyst can be divided into two categories:

1. Interpersonal skills
2. Technical skills

Interpersonal skills deal with the relationship of the system analyst with the people in the business. They are useful in resolving conflicts, establishing trust and communicating information.

The interpersonal skills relevant to system work include the following :

- **Communication :** The system analyst should be able to communicate with users in such a way that he/she can collect all the required information that is necessary for system design from the users He should be ready to interact with top–level management as well as workers in the industry.
- **Understanding :** The analyst should understand the problem for which the system is to be designed. Once he/she understands a problem, he/she must be able to produce different solutions to meet user requirement.
- **Teaching :** The analyst should train the different people such as programmers, technical writers and end users. He/she should be able to train programmers about how to code the system using programming language. Analysts should be able to train the technical writer about how to create different documents and analysts should give training to the end user about how to use the system.
- **Selling :** The analyst should introduce the people with the way to solve the problem by using computers by putting ideas before them.

The technical skills relevant to system work include following :

- **Creativity:** The system analyst must be creative and imaginative in producing new solutions meet be creative and imaginative in producing new solutions to meet user requirements.
- **Problem solving :** The analyst must be able to solve problems and must be able to provide different solutions, to the given problem.
- **Project management :** The analyst should prepare a plan/schedule for the design and development of the system. He should assign the jobs to the respective persons accordingly and keep coordination among them. He should also take care of costs and expenditures.
- **Dynamic interface :** The analyst should properly co–ordinate technical and technical activities into functional specifications and general design.
- **Questions attitude and inquiring mind :** This involves questions related to what, when, why, where, who and how a system can work.

❖ **Academic Qualification for a System Analyst :**

The system analyst is a person, no doubt but he should have good academic qualifications, which are important for system work.

- The system analyst may have knowledge in computer science, engineering, business administration, information systems or economics.
- The analyst should have background in system theory and organisation behaviour.
- He should be familiar with the areas such as financial, accounting, personal administration, marketing and sales, operation management, model building and production control.
- He should be competent in system tools and methodologies and should have practical knowledge of one or more programming and database languages.
- He should have experience in hardware and software specifications.

❑ Check Your Progress – 3 :

1. One of the main work of systems analyst is to :

   a. Introduced an information system that will take care of all organization requirements

   b. Design a specification code

   c. Repair the system

   d. None of the above

2. Which among the following is an important attribute of systems analyst ?

   a. good coding knowledge

   b. good hardware maintenance skills

   c. excellent technical management qualities

   d. all of these

2.5 System Life Cycle :

Before we describe the system development cycle, let us look at what actually happens during system development. Many activities take place in system development such as :

- Understanding user requirement
- Designing and decision making about the system
- Building a system
- Forming group memory
- Developing software
- System testing
- Implementation and evaluation

First, analysts have discussions with users to familiarise themselves with the system and user requirements. They then determine how these requirements can fit in with the system. Previous knowledge is used to develop an understanding of the system and to become familiar with its problems.

Ideas about the new system are proposed and evaluated. There may be some experimentation to find out if some of the various proposed ideas used in the design are useful.

During this a group memory of what was discussed and what conclusions were reached about the system are developed.

"System Development Life Cycle (SDLC) is a set of steps, which are used for building a system". The SDLC has been widely used in the design of a system that can be easily understood and which have well defined workflows.

SDLC is a software lifecycle management procedure which had 5 different phases which includes :

- gathering requirements
- design
- construction/test

- implementation
- post–implementation support

Basically, SDLC shows the software development process how to work and describes documents and deliverables for certain SDLC phases. In this every alternate phase show leverage with write and setting of documents as well as gain knowledge from earlier SDLC phases. In SDLC process, comprehensive business requirements should be known to find functional design that carries design and coding.

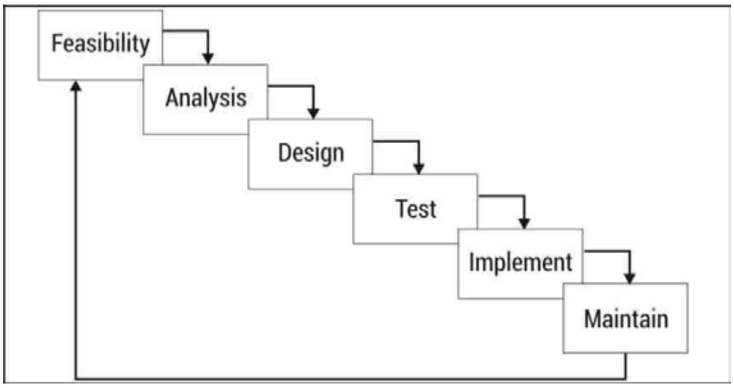


Fig. 2.13 : Stages of SDLC

In above figure two stages that are combined are the system–coding step and system design.

Table Seven Steps of SDLC

Sr. No.	Stage	Key Question	Result
1	Recognition of need – Preliminary survey/ Initial investigation	What is the problem or opportunity ?	Statement of scope and objectives
2	Feasibility, study Evaluation of existing system and analysis of alternative candidate system, cost, estimates.	What are the user's needs ? How can the problem be solved ?	Technical/behavioural feasibility Cost/benefit analysis
3	Analysis : Detailed evaluation of present system	What must be done to solve problems ?	Logical model of system
4	Design : General design specifications, detailed design specifications–output/ input, files, Procedures	In general, how the problem should be solved ?	Design of alternative Solutions Cost/ Benefit analysis Implementation of schedule
5	Programme construction testing	How will individual modules tested out ?	Test plans, operating procedures
6	Implementation user training	What is an actual operation ?	Training programme user-friendly documentation
7	Post Implementation	Where is the key system running ?	User requirements wet, user standard wet.

The analyst must progresses from one stage to another methodically, answering key questions and achieving the results in each stage.

❖ Purpose of SDLC :

The base of SDLC is to lunch a particular practice or method which can be a copy or duplicate repetition with high software quality. It will convert to low cost of ownership with more output on amount invested by way of direct savings to corporate 'bottom line. In SDLC, cost effectiveness is required output with ensuring high software quality. If the SDLC model in good, then robust software as well as project management practice will only work. Basically SDLC is not software engineering process nor can show software engineering quality standards. There are certain steps involved in SDLC:

- Recognition of need/requirements
- System analysis
- System design
- System coding
- System testing
- System implementation
- System maintenance

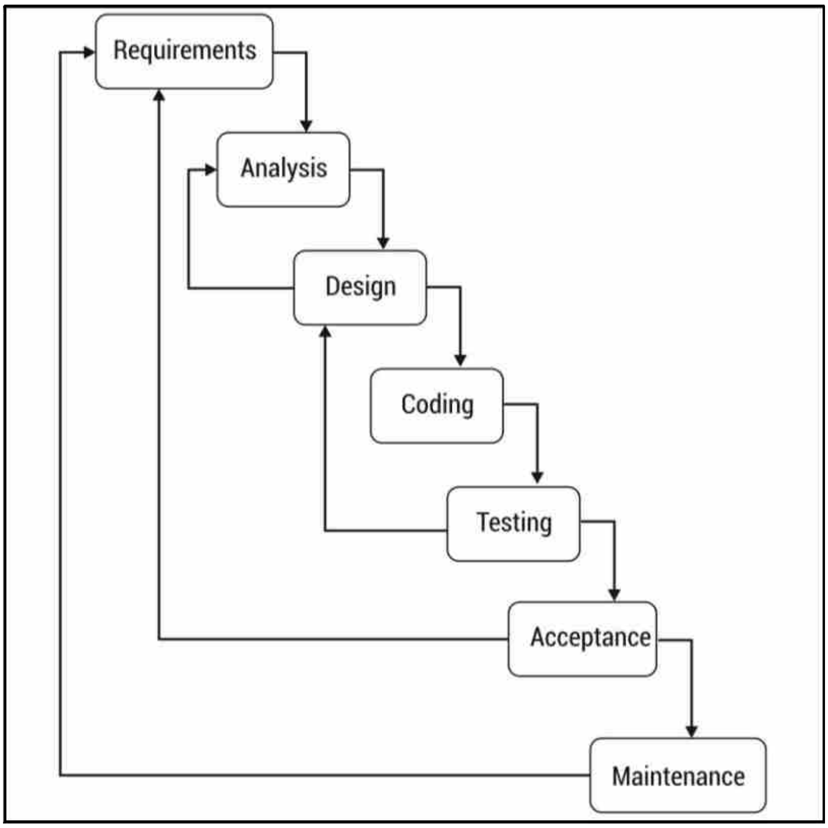


Fig. 2.14 : Steps involved in SDLC

❑ Check Your Progress – 4 :

1. ________ refers to the process which involves in production of reliable and workable information systems

   a. Case tools b. System Development Life Cycle

   c. Decision table d. Success factors

2. In System Development Life Cycle, up gradation, improvement and bug fixation are normally performed at the time of ________ stage.

   a. Maintenance b. Evaluation

   c. Design d. Maintenance and Evaluation

2.6 Working with Technology :

The process of analysing and designing of system involves step–by–step procedure which can create high quality information systems. The information system is a mixture of information technology, people as well as data, which will take care about all business requirements. It is studied that an information system will take care of :

- daily business transaction
- improves company productivity
- help managers making sound decisions

In case of IT department, the team of system analyst includes the persons that will be responsible for planning, development and maintenance of information systems.

In a department, the work of systems analyst is to study and analyse information related problems and to work and coordinate with people, methods and technology in order to solve problems in an organisation. Further it is seen that an analyst will design a system that can accommodate and process efficient data for business applications. The analyst will decide about data and information which can be feed inside the computer for processing that can be further saved in storage device located inside computer.

❑ **Check Your Progress – 5 :**

1. Which of the following shows the features of information?

   a. accuracy b. advance c. completeness d. all of these

2.7 Let Us Sum Up :

That system is a set of components which interacts to do certain task. The basic of system is deals with interrelationship and integration of operations using certain components.

There are many systems that which could be static or dynamic in nature. Static system does not change whereas dynamic system changes due to processing of the system. System can be abstract, open and closed.

It is seen that computer based information system carries six interdependent elements such as Hardware, Software, People, Procedures, Data and Output.

With the help of Computer based information system, you can store large volume of data, can perform different operations speedily, improves processing power of organisation, maintain security of data from unauthorised users and produce different output information easily

It is found that System Development Life Cycle contains a software management methodology having five phases which includes gathering, design, construction/test, implementation and post–implementation.

The main aim of SDLC is to introduce a practice or methodology which can be both duplicable and repeatable in order to have high software quality.

2.8 Answers for Check Your Progress :

- ❑ **Check Your Progress 1 :**

  1 : a **2 :** c

- ❑ **Check Your Progress 2 :**

  1 : a

- ❑ **Check Your Progress 3 :**

  1 : d

- ❑ **Check Your Progress 4 :**

  1 : a **2 :** d

- ❑ **Check Your Progress 5 :**

  1 : d

2.9 Glossary :

1. **Database design –** It is a process of producing detailed data model of a database.
2. **Scheduling –** It is a process of deciding how to commit resources between a variety of possible task.
3. **Data Flow Diagram –** It is a graphical representation of data flow inside an information system.
4. **Embedded System –** It is a computer system having focused function present inside mechanical or electrical system which deals with real time computing constraints.
5. **Abstract System** – Systems that is shown as non–existing or non–physical.
6. **Closed System** – System that directly controls its own operation by interfering with data which is generated by system itself.
7. **Automated Information System –** A fabricated system applicable with one or more persons that works centrally with hardware and software.

2.10 Assignment :

1. Collect information on the use and advantages of a computer–based information system and prepare a report on the same.

2.11 Activities :

- What are the activities that take place in system development ?

2.12 Case Study :

- Collect information on the skills expected in a system analyst and list them down in your own words.

2.13 Further Readings :

1. Analysis and Design and Information System, Rajaraman.
2. Analysis and Design of Information System 2nd Ed. Senn.
3. Introducing Systems Analysis and Design, Lee.
4. System Analysis and Design, Edwards, Tata McGraw–Hill.
5. Systems Analysis and Design, Elias Awad.
6. System Analysis and Design, J.F. Gerald, Tats McGraw–Hill.

Unit 03

MODELLING TOOLS FOR SYSTEM ANALYST

UNIT STRUCTURE

3.0 Learning Objectives

3.1 Introduction

3.2 Role of Data in Business

3.3 Modelling with DFD

3.4 DFD'S with CASE

3.5 Structured Methodology

3.6 Let Us Sum Up

3.7 Answers for Check Your Progress

3.8 Glossary

3.9 Assignment

3.10 Activities

3.11 Case Study

3.12 Further Readings

3.0 Learning Objectives :

After learning this unit, you will be able to understand :

- The role of data in business
- Explain concept of Modelling with DFD
- Outline DFD's
- Discuss structured methodology
- Explain relevant CASE technologies

3.1 Introduction :

In the previous unit, we discussed the system development cycle and stages of the same. After knowing about the system development life cycle stages, let us now discuss various modelling tools, which are used by system analyst. After going through this unit, you will come to know about data flow diagrams, E–R diagrams and other specification tools which are used by the system analyst. Data is the raw material that an information system transforms into useful information. An information system can store data in various locations, called table. By linking the tables, the system can extract specific information.

3.2 Role of Data in Business :

In business, the importance of data plays a major role. Without data, nothing is possible in business. Each and everything in business involves usage of data, there are different processes where data is transformed from one form to another. Data has to be stored for future access; it can be received or sent to a system.

❑ **Check Your Progress – 1 :**

1. The information system outputs :

   a. raw data b. clear data c. refine data d. processed data

3.3 Modelling with DFD :

Data flow diagram is a modelling technique which is applied to find and develop information processes. It signifies the course or flow of information inside a process. A DFD will show flow of information process inside inputs and outputs which is often known as Process Model.

Data flow analysis is not flow charting. It is a specific technique that focuses on :

- Processes (business processes performed)
- Entities external to the process (the people with whom the agents or operators interact)
- Data stores (where the information is stored)
- Data flows (which information is passed back and forth)

The results of data flow analysis are best–documented using data flow diagrams. These graphically display the four components, in a simple, direct way. Part of the clarity in data flow diagrams is due to intentionally omitting some aspects, which then feature later, elsewhere in the analysis process. It shows the straight representation of any business system or its associated functions. In these techniques a rough sketch based on information about a company is prepared which is further analysed in terms of respective functional domains. It is taken out precisely with different level detail as on demand.

Data flow diagrams simply show the transformation of system information and describe how the how the information is carried out and is kept. In order to develop a data flow diagram, we need to consider the following:

- Data inflow or outflow information.
- Position of input data before leaving the system.
- Causes of delay between input and output of data.

It is an important modelling procedure where the information process is analysed and further developed. It shows an illustration which describes about the movement of information that is travelling inside a process.

Additionally, a DFD is utilised to visualise data processing or a structured design. A DFD illustrates technical or business processes with the help of the external data stored, the data flowing from a process to another, and the results.

A designer usually draws a context–level DFD showing the relationship between the entities inside and outside of a system as one single step. This basic DFD can be then disintegrated to a lower level diagram demonstrating smaller steps exhibiting details of the system that is being modelled. Numerous levels may be required to explain a complicated system.

The output obtained comprises of sequence of diagrams which shows the business activities that is clear and concise in order to communicate. The business model essentially contains one or more data flow diagrams. During the process of development of DFD, a context diagram is drawn carries related presentation of complete system which is under inspection. In a level 1 diagram, the complete information about main functional areas of the business is highlighted.

Here the level 1 diagram signifies main business processes at high level which can be analysed by giving rise to certain level 2 business process diagrams. Under this process, more detailed analysis continues by considering levels 3, 4 and so on. Many times the inspection stops at level 2 diagram that is quiet unusual to move ahead to level 3 diagram.

Determining the subsisting business approaches, exercising a approach similar data flow diagrams, continues an imperative antecedent to business approach re–engineering, migration to current technology, or advancement of an subsisting business mechanism. Although, the category of detail will necessarily depends on category of change approximated.

❖ **Data Flow Diagrams Symbols :**

Some symbols are used in the drawing of business process diagrams (data flow diagrams). These are explained below together with the rules that apply to them.

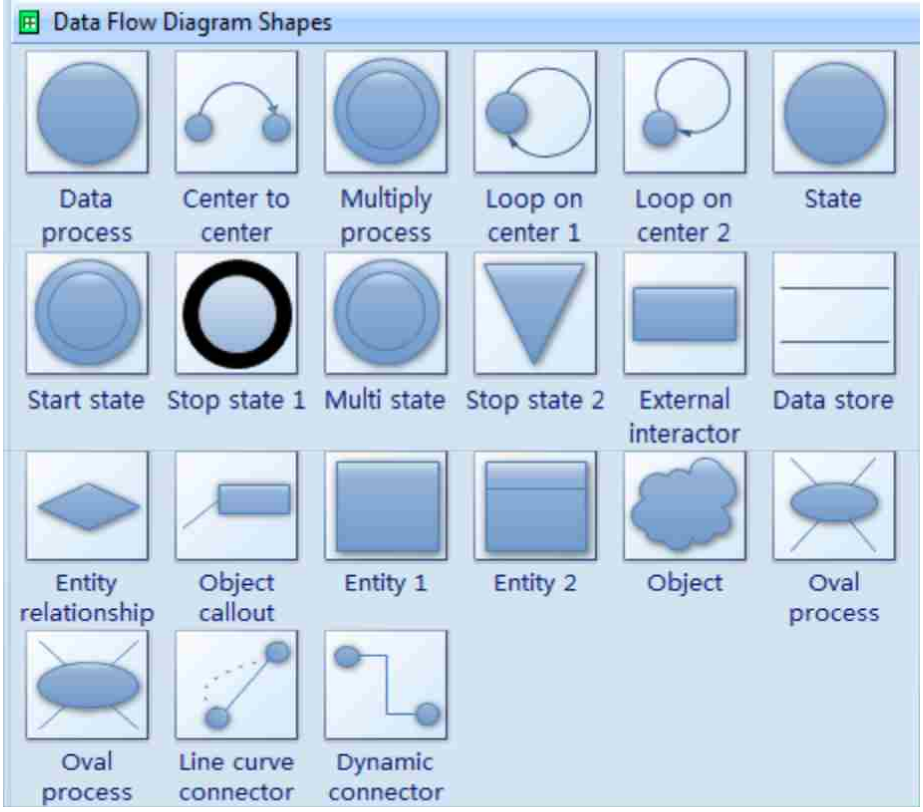


Fig. 3.1 : Data Flow Diagram Symbols

❖ **Process :**

The process is represented by various shapes. These shapes show certain task which can be controlled inside an application. Such task will process the data or action as per the data present.

❖ **Multiple Processes :**

Multiple process shape is applied to show collection of sub processes. Such types of multiple processes can further be split down into sub processes inside different data flow diagram.

❖ **External Entity :**

Such shape will show any entity which lies outside an application which will further interact with certain applications through entry point.

❖ **Data Flow :**

The shape of data flow will demonstrate the flow of data available inside an application. The arrow will show the direction of movement of data flow.

❖ **Data Store :**

Shape related to data store will show locations where data is kept. In this, data is not modified, but is simply stored.

❖ **Privilege Boundary :**

Such shape will highlight the occurrence of changes which occurs at various levels during the flow of data inside an application. In this, the following steps are needed while designing data flow diagram.

❖ **Steps :**

- Consider a context diagram and locate for parent process along with external entities showing their total inputs/outputs
- Put external entities on diagram and draw its boundary.
- Find the flow in data that will produce total input/output to external entities
- Locate the business processes that will work for producing input/output data flow.
- Now join data flow with external entities in order to work
- Locate data stores
- Join the processes and data store with data flows
- Use Process Model Paradigm, so as to check for diagram addresses and processing requirements for complete external entities
- Use External Control Paradigm after Process Model Paradigm to check for flow of external entities
- Continuously apply till nth level of data flow diagram.
- After that, draw all data flow diagrams on certain level before moving to next level. The process of decomposing will initially start from horizontally to nth level to make sure that all the processes gets correct partition. After horizontally, decompose it vertically.

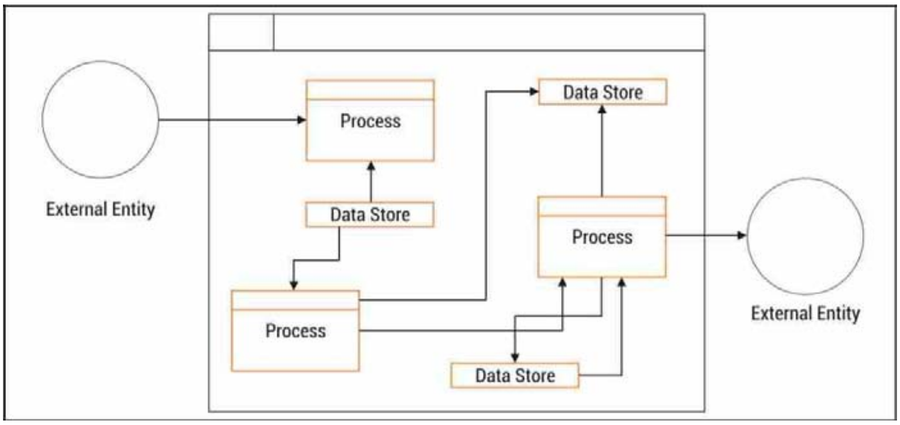


Fig. 3.2 : Process of Data Flow

❖ **Context Diagram :**

- Top–level view of IS
- A data flow diagram (DFD) of the scope of an organisational system that shows the system boundaries, external entities that interact with the system and the major information flows between the entities and the system

❑ **Check Your Progress – 2 :**

1. Data that is kept in DFD shows :

   a. sequential file   b. storage disk

   c. repository of data   d. memory

2. In case of DFD, which among the following symbol will highlight a process?

   a. square   b. circle   c. rectangle   d. rhombus

3.4 DFD'S with CASE :

The DFD endures an exemplary communication medium for analysts to prototype procedures along with appropriate obligations. It continues appraised one of the foremost modelling approaches for effecting as well as approximating the processing necessities of a system.

Applied consequentially, it is a practical as well as comfortable to believe modelling facility. It grasps bulky approaches as well as usability against maximum software development approaches. It is efficiently connected with data modelling, workflow modelling appliances, along with textual specs. Together with these, it assigns analysts as well as developers with integrated models as well as specs. By itself, furthermore, it keeps restricted usability. It is efficient as well as comfortable to accept by users additionally can be efficiently annexed as well as finished with additionally specification into a corporal explanation for the design as well as development teams. Further, the use of CASE tool will help for constructing a data flow diagram which can also explain its correct use.

❖ **CASE Tools as Toolsets :**

In certain quarters, CASE Tools are also referred as 'Toolsets' and the Toolsets could be :

- **The planning Toolsets** – This helps to begin the development process with information strategy planning from a high level, business vantage point.
- **The analysis Toolset** – This focuses on correctly capturing detailed business requirements early in the development process.
- **The design Toolset** – This provides detailed specifications of the system solution.
- **The information integrator** – This integrates system specifications, checks them for consistency and correctness and records in local or central repositories.
- **The code generation Toolset** – This produces a higher language programme code based on system specifications.

- **The database generation Toolset** – This generates system control information needed for data storage and access.
- **The public interface Toolset** – This provides for file transfer and query reporting.

Thus, we have seen that using various toolsets of CASE tools we can design DFD's.

❑ **Check Your Progress – 3 :**

1. Which toolset will focus on business requirements early in the development process ?

a. planning Toolsets	b. analysis Toolset
c. design Toolset	d. code generation Toolset

3.5 Structured Methodology :

The Structured System analysis and Design method is a technique which can find and construct of computer systems. The Central Computer and Telecommunications Agency in the United Kingdom developed the structured systems analysis and design method in the early 1980s. The technique adopts a structured methodology toward systems development using data flow, logical data, and entity event modelling. Core development stages include feasibility study; requirements analysis; requirements specification; logical system specification; and physical design. All the steps and tasks within each stage must be complete before subsequent stages can begin.

Structured analysis depends on group of process models which can explain the system by graphical representation. With the help of process modelling, we can find data flow in a process, business rules which convert data and gives output as data flow. Structured analysis is developing into a technique called Information Engineering.

❖ **Need :**

SDLC is an olden systems development process which is time tested as well as simple to understand. It originates in 1960, when mainly systems are based on mainframe and file processing is done individually. Since it explains about processing of raw data in useful information, the structured analysis is mainly concerned with data organisation and structure, relational database design and user interface issues. SDLC employs a sequence of phase that can plan, analyse, design, implement as well as handle information system.

❖ **Relevant Case Technology :**

Computer–aided system engineering (CASE) uses powerful programs called as CASE tools which will make system analysts to frame and maintain information systems. Such techniques will focus on framework employed for systems development as well as helped several technology used for designing such as structured analysis along with object oriented analysis.

Traditionally, it is found that the systems developers used to differentiate among two CASE categories that are upper CASE and lower CASE tools. Upper CASE tools will able to handle modelling process and will generate logical design of information system. While the lower CASE tools will use development process by forming source code based which can be available on logical modelling. Presently several popular CASE tools today coordinates with upper as well as lower CASE features that can be converted to single product.

By using CASE tools, the productivity of IT can be improved and hence produces extremely high quality output. We see that developers makes use of such tools to safe guard design integrity in handling complicated projects and can produce code modules which can help to implement fast.

CASE is a category of software tools, which aids a developer in creating and maintaining software. A CASE tool is a software tool that automates a particular task within the system development life cycle.

The software packages available in the areas/activities of the automating the systems development activities are known as CASE Tools.

CASE tools are not manual but is microcomputer based software packages applied in case of systems analysis as well as design. CASE tools are used for:

- Increasing the analyst productivity
- Easy interaction that happens among the analysts as well as users
- To provide continuity between life cycle phases
- To assess the impact of maintenance

❖ **Stages of the CASE life cycle :**

1. Strategic Planning
2. Analysis
3. Logical Design
4. Physical Design
5. Construction

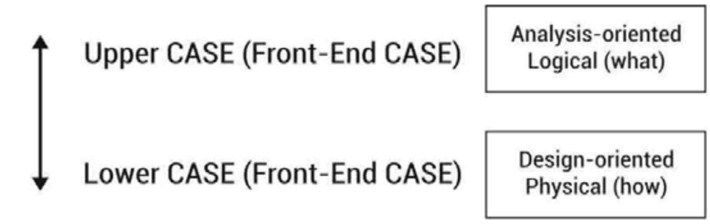

Upper CASE (Front-End CASE)
Analysis-oriented Logical (what)
Lower CASE (Front-End CASE)
Design-oriented Physical (how)

Fig. 3.3 : CASE Tools

❖ **Goals of Automated Development Aids (CASE) :**

- Increase analyst productivity
- Increase the quality of system deliverables
- Ensure cohesiveness among tools/procedures
- Standardize and coordinate efforts of development team members
- Facilitate system maintenance

It is seen that upper CASE tools are actually employed in study as well as design. In case of lower CASE tools, such tools are used to produce computer language source code from CASE design. The advantages of producing source code will cover:

- Actual time to frame any fresh system
- Actual time to handle generated code which is lower so as to maintain earlier systems

- Generation of computer programs in several programming language
- Purchasing of CASE design from third–party vendors Production of code which is free from any coding errors

CASE tools are available for the varying purposes/activities as follows :

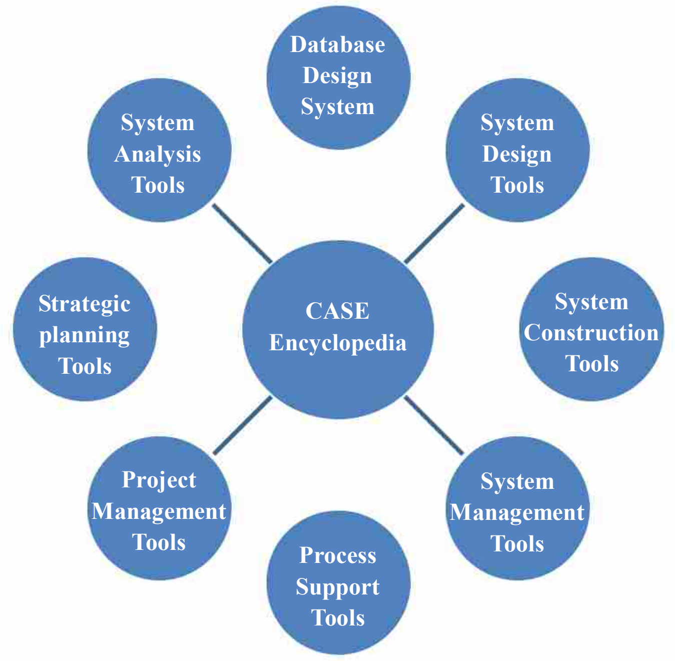


Fig. 3.4 : Purposes of the Availability of CASE tools

Maximum benefits can be obtained by integrating various CASE tools. Integrated CASE refers to a CASE product that includes support for the full life cycle. Such a product would be repository–based and repository–driven. The repository is a centralised database, the nucleus of the CASE environment, holding the all of the information needed to create, modify and evolve a software system from project initiation and planning to code generation and maintenance

❖ CASE Tools Capabilities :

It is obvious that CASE does not and cannot do anything and everything. It must be remembered that CASE is, at best, only a facilitator. It can certainly accelerate analysis and design and promote iterative design. However, it still remains a methodology and set a set of tools for developing systems with, as stated earlier, enhanced productivity, reliability and versatility. CASE is not automated systems development and it does not enable systems to be designed automatically to meet business requirements.

CASE tools are capable of performing some functions. The table shows the functions that CASE tools can and cannot perform.

Table Functions of CASE tools

Functions that CASE tools can perform	Functions that CASE tools cannot perform
Automate the number of the manual task involved in system development.	Automatically provide a functional, relevant system
Promote standardization based on a single methodology	Easily interface with the databases and fourth-generation languages

Promote greater consistency and coordination during a development project	Automatically force analysts to use a prescribed methodology when one does not exist
Generate a large portion of documentation for a system, such as Data Flow Diagrams, Data Models and/or other specifications	Radically transform the systems analysis and design process

❖ **Other Specification Tools :**

E–R Diagrams :

Entity–relationship diagram (ERD): A relationship diagram is a graphical arrangement of database development. Such type of relationship diagrams will clearly show the dynamics involved different databases relations immaterial of the fact that whether you are using latest design software or not. We say that an entity–relationship model (ERM) is an abstract and conceptual representation of data. Entity–relationship modelling is a database modelling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database and its requirements in a top–down fashion. Diagrams created by this process are called entity–relationship diagrams, ER diagrams or ERDs. Such type of model will find certain concepts or entities which occur in system and carries relationships among particular entities.

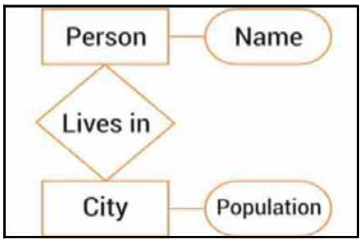


Fig. 3.5 : ER Diagram

Such diagram is applied to imagine relational database, where every single entity shows database table along with relationship lines that shows keys in particular table with particular records located in tables. The diagram is more theoretical which does not cover each table required inside a database which serves as main concepts and relationships.

Such diagram will present an abstract which could be a theoretical view of main entities as well as relationships that is required for electronic resources. This will help the database design process for the purpose of an e–resource management system which does not locate each table which is compulsory for electronic resource management database

Examples :

In a database it contains the details about residents of a particular city. As per the ER diagram shown having two entities such as people and cities, it seems that there exists single live–in relationship. In our example, due to space constraints, there is only one attribute associated with each entity. People have names and cities have populations. In a real–world example, each one of these would likely have many different attributes Entity Relationship Diagrams (ERDs) illustrate the logical structure of databases.

ERD components are:

- Rectangles representing entity sets
- Ellipses representing attributes
- Diamonds representing relationship sets
- Lines linking attribute to entity sets and entity sets to relationship sets

In the text, lines may be directed (have an arrow on the end) to signify mapping cardinalities for relationship sets. Figures show some examples.

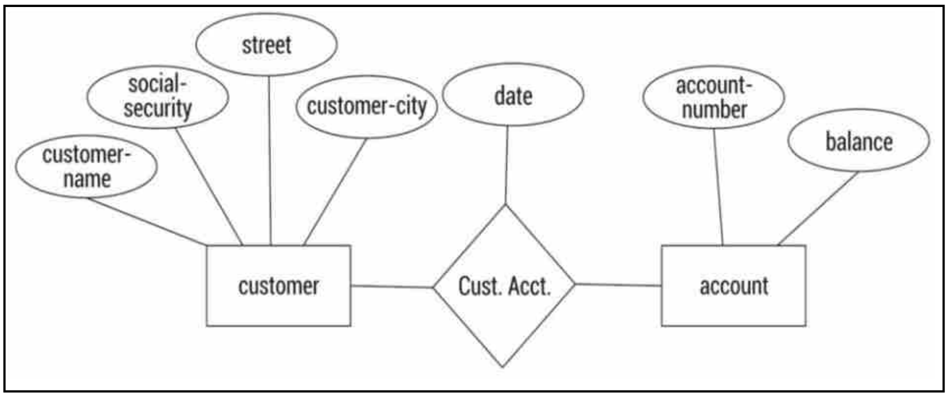


Fig. 3.6 : An E–R diagram

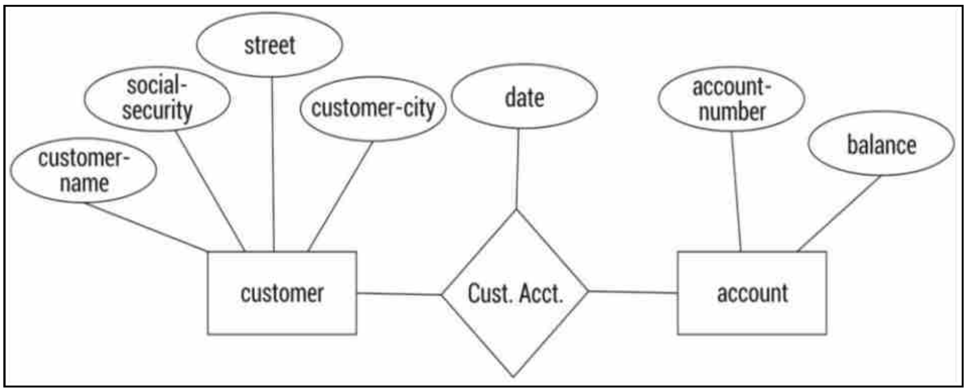


Fig. 3.7 : Many–to–one from customer to account

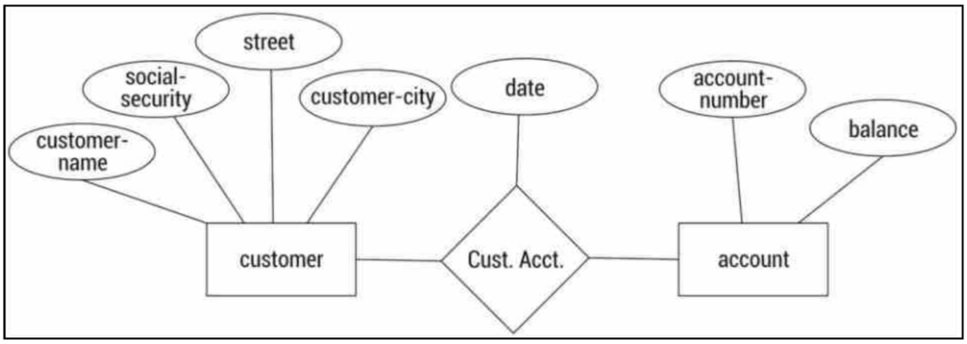


Fig. 3.8 : One–to–one from customer to account

Relationship: The relationship that exists among data items having different entities and entities themselves will result in main action of database normalisation. We found that there exist three types of data relationships such as:

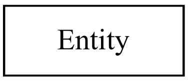
One–to–one (1:1) : The one–to–one relationship shows that at a particular time an entity will show accurately single example of an additional entity. In an example of class result, each student achieves one grade record and every grade record is for single student.

One–to–many (1:M) : The one–to–many relationship shows that every example of required entity will show one or more occurrence of an additional entity. Consider an example of class study where single teacher entity is located to be teaching of certain classes where it is seen that all class is handled by single teacher.

Many–to–many (M:N) : In case of many–to–many relationship, there exists several example that will talk about several occurrence of additional entity. It is finding that in a school, a timetable can be prepared for several classes whereas a class can be conducted within different timetables.

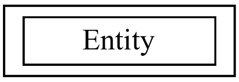
❖ **Entity Relationship Diagram Notations :**

Entity is object or concept regarding which you require to keep your information.



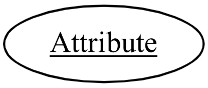
❖ **Weak Entity :**

Attributes relates to several features or characteristics of an entity.



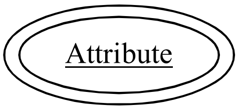
❖ **Key attribute :**

Key attribute is only one of its kinds a unique attribute of an entity. It is seen that in key attribute, an employee's social security number could be his employee's key attribute.



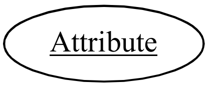
❖ **Multi–valued attribute :**

Multi–valued attribute carries multiple values. It is seen that an employee entity will result in multiple proficiency values.



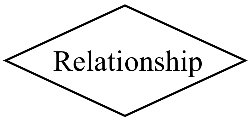
❖ **Derived attribute :**

Derived attribute depends on another attribute. As we can say that an employee's monthly salary depends on employee's gross salary.



❖ **Relationships :**

Relationships concerns with the sharing of information among two entities in database arrangement. Here we have to initially connect two entities and further drop relationship notation on a line.



❖ Cardinality :

Cardinality shows how many cases of an entity will speak about in relation to one example of a new entity. Further we see that ordinality is closely in relation with cardinality. As cardinality describes the amount of relationship, so ordinality also shows relationship that can be mandatory or optional. We can say that cardinality shows maximum number of relationships while ordinality shows fully less number of relationships.

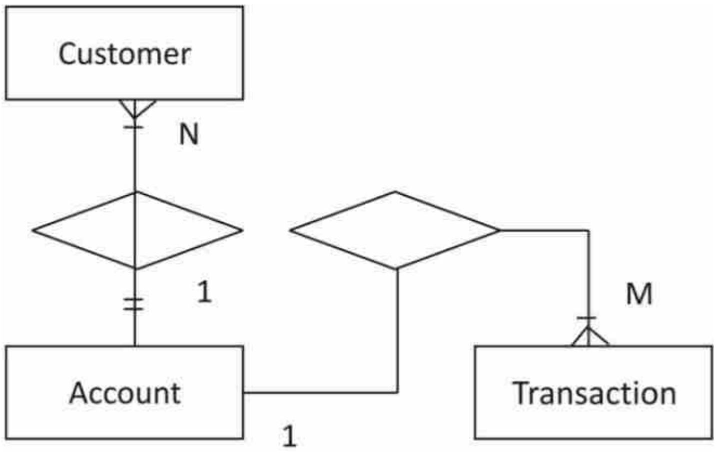


Fig 3.9 : Cardinality

❖ Recursive Relationship :

We see that there occur entities that are linked by it. In an organisation, we see that employees can supervise other employees.

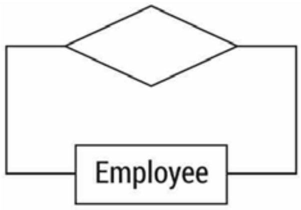


Fig 3.10 : Recursive Relationship

The ER diagram is specific graphical arrangement that shows interrelationships that exists among entities present in database. Such diagram explains with the help of symbols so as to represent various types of information. In a common ER diagram, it is seen that boxes are mostly applied to show entities. In the ER diagram, the diamonds shaped is used to show relationships whereas ovals in ER diagram shows attributes. This is also known as ER Diagram, E–R Diagram and entity–relationship model.

Example of ER – diagram :

1. **E–R Diagram for Purchase Order Application :**

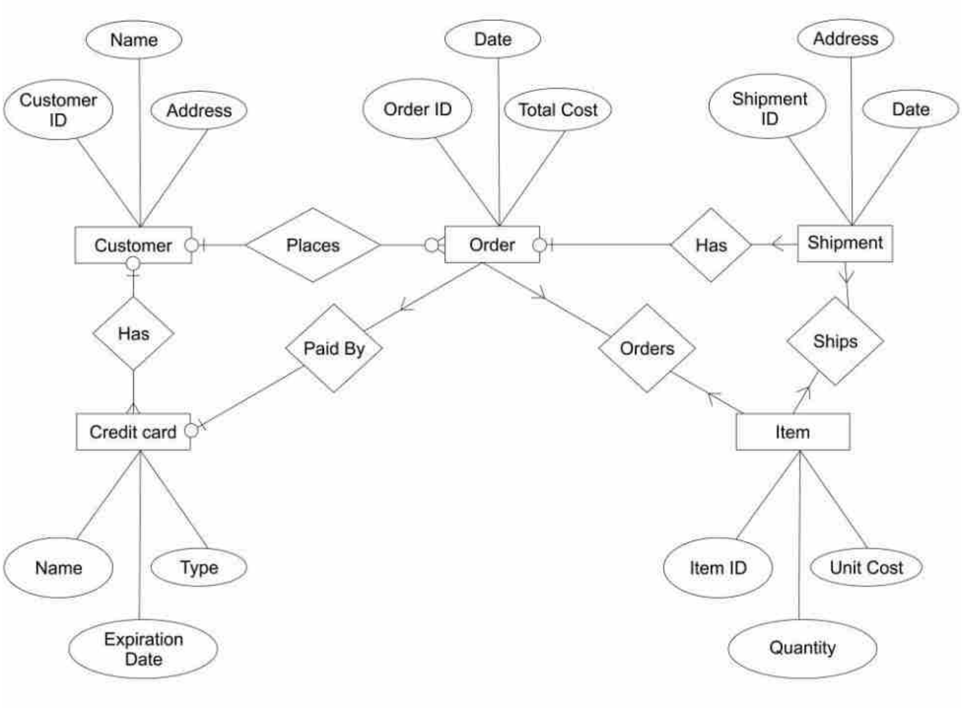


Fig. 3.11 : E–R Diagram for Purchase Order application

While seeing the purchase order application, a customer will have one–to–many relationship as customer can put several orders but single purchase order can only be applicable by single customer. Such type of relationship results as an optional as no customer can place any order.

Also, it seems in a purchase order, a customer has many–to–many relationship with stock item since the purchase order is related to several stocks which can be called as several purchase orders. This is usual that nobody knows which purchase orders refer to which items.

For knowing this, we study about line item, since purchase order carries one–to–many relationship along with line item as purchase order can show several line items but at the same time a given line item shows only single purchase order.

The Line Item carries many–to–one relationship along with Stock item since the line item indicates only single stock item where a given stock item means many line items. It is also optional as zero–line items can relate to given stock item.

2. **The Flight Database :**

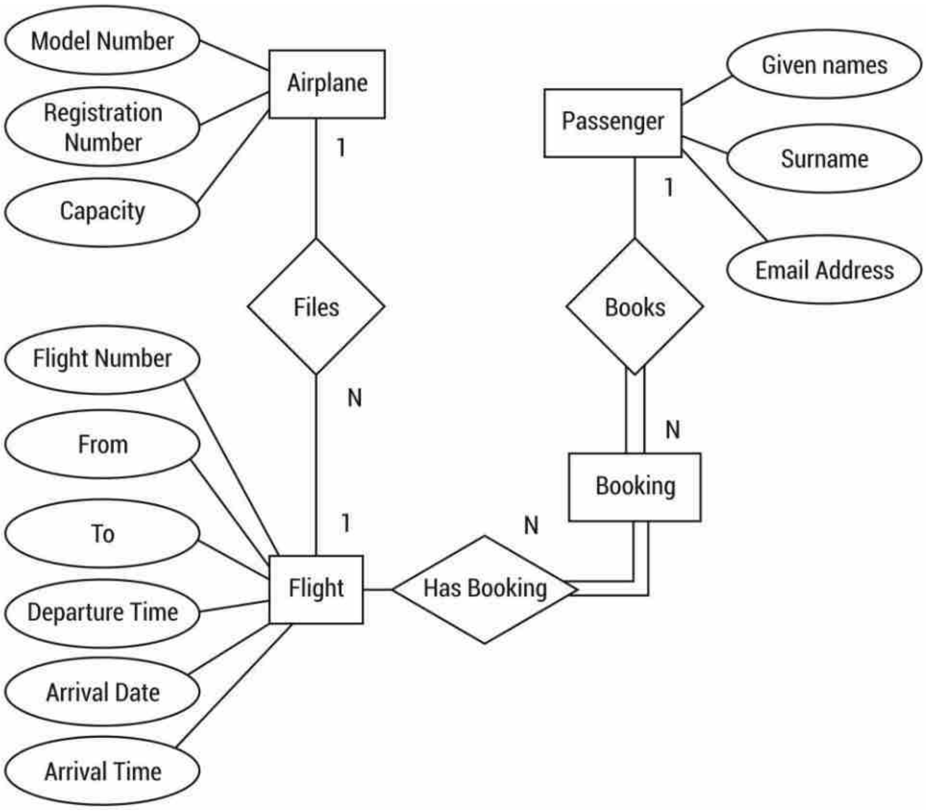


Fig. 3.12 : The ER Diagram of the Flight Database

The flight database stores details about an airline's fleet, flights and seat bookings. Again, it is a hugely simplified version of what a real airline would use but the principles are the same.

3. **ER Diagram for Hotel Management :**

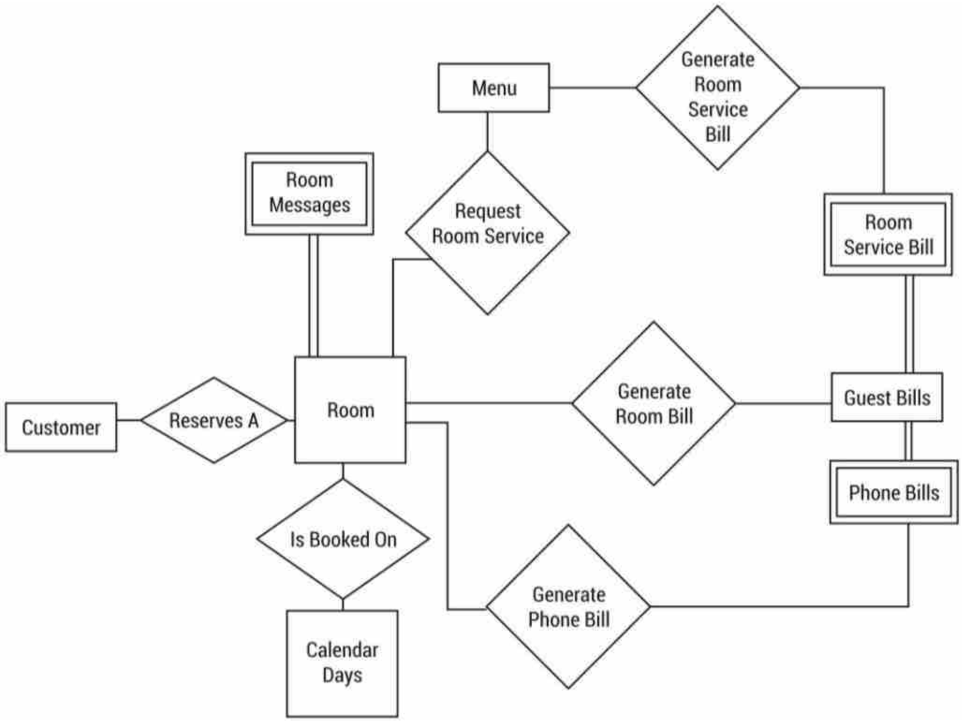


Fig. 3.13 : E–R Diagram for Hotel Management

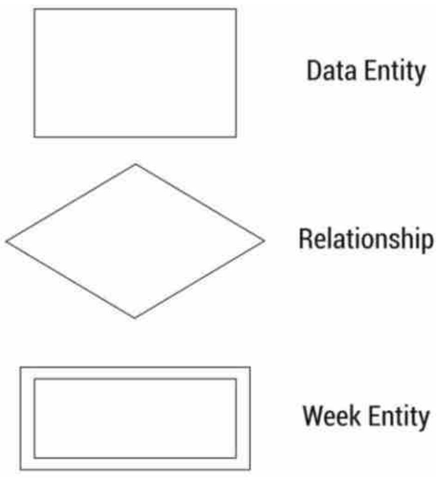


Fig. 3.14 : E–R Diagram for Hotel Management

❖ **Structure Charts :**

It is seen that structure charts shows high–level design or architecture of a computer program. Being a designing tool, it will help the programmer in solving and working on heavy software problem. This can be done by splitting the problems in parts and then solving bit wise by using his own knowledge. Such process relates to top–down designing.

Programmers will apply structural charts to prepare a programme similarly like architect prepares a blueprint for a particular building. During the time of designing, the chart is prepared that can be utilised by the clients with several software designers in order to communicate. At time of implementation, the structural chart will serve as master plan for a particular programme which can be sometimes altered by programmers on demand.

Once a programme is done, these charts can be utilised for fixing bugs which further can be altered for maintenance. After all amendments, a final chart gets prepared which serves as navigation tool for several programmers who are working on the programme. If you have ever experienced the joy of modifying someone else's program, you know how important this blue print can be.

Structure charts are used for procedural programs to illustrate the following information about a programme in a visual format :

- Partitioning of a programme into named modules (functions)
- Top–down hierarchy and organisation of modules
- Linkages between modules
- Flow of data, control or exception information

The structure chart is pictorially graphed tool that demonstrate the relation that exist among processing modules in computer software. It shows various stages of component modules in which the data flows. It contains analysis of input–to–output transformations as well as analysis of transaction.

Such charts will demonstrate relation exists among processing modules available in computer software. This design tool visually shows relationships that exist among programme modules. The chart will demonstrate the type of module present inside a system, which cooperates additionally, shows data what is travelled in different modules.

These charts are prepared before the write–up of programme code, which identifies type of data or information, which passes between single modules, which work together with one another.

The first step in creating a structure is to place class main in the root of an upside–down tree, which forms the structure chart. The next step is to conceptualise the main sub–tasks that must be performed by the programme to solve the problem. These sub–tasks are placed in nodes below the root, and connecting lines are drawn from the root to each sub–task.

Next, the programmer focuses on each sub–task individually, and conceptualises how each can be broken down into even smaller tasks. Eventually, the programme is broken down to a point where the leaves of the tree represent simple methods that can be coded with just a few programme statements.

❖ **Notation used in Structure Charts :**

Programme modules are identified by rectangles with the module name written inside the rectangle. Arrows indicate calls, which are any mechanism used to invoke a particular module.

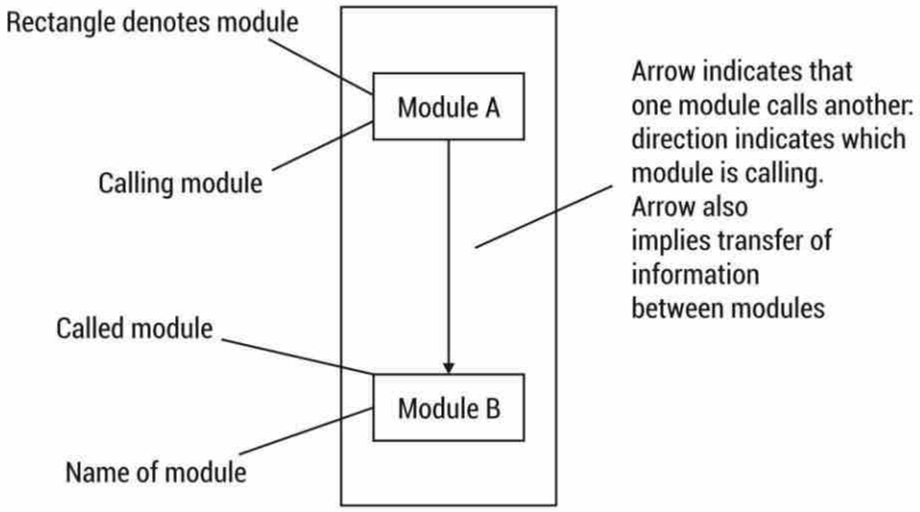

Rectangle denotes module
Module A
Calling module
Arrow indicates that one module calls another: direction indicates which module is calling. Arrow also implies transfer of information between modules
Called module
Module B
Name of module

Fig. 3.15 : Notation used in Structure Charts

Once the structure chart has been designed, it is used to drive the software implementation. Branches of the tree are assigned as modules to be programmed by various members of the development team. Often, a process called bottom–up implementation and testing is used to implement each module. This process will be discussed in detail later in this document.

Consider the following sample structure chart. Many information–processing applications can be modelled with a chart similar to this sample :

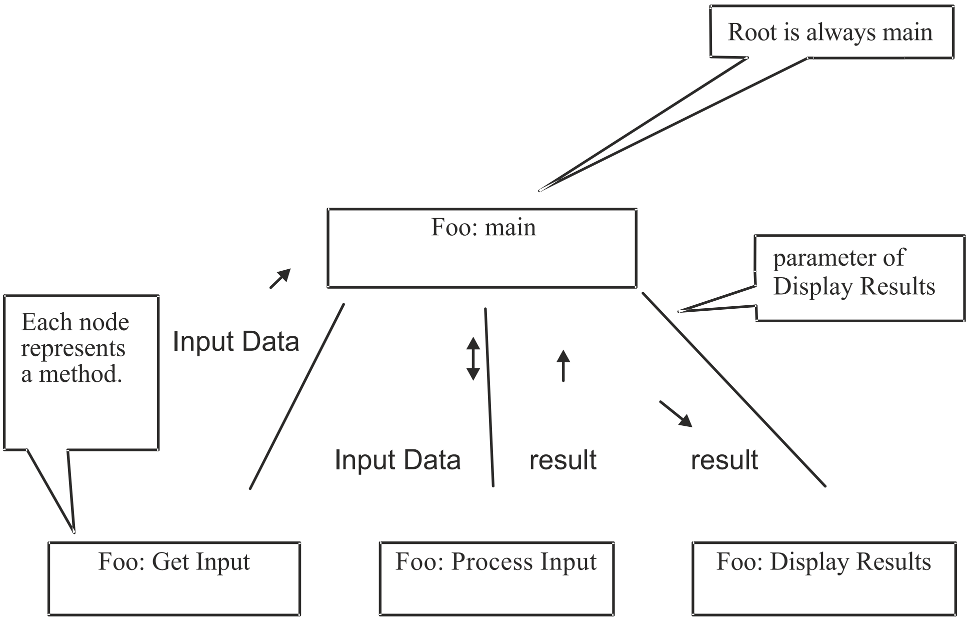


Fig. 3.16 : a Sample Structure Chart

Note that in this sample, all methods are from the class Foo. In general, the methods of a structure chart will come from various classes. In addition, structure charts for real–world applications tend to be many levels deep and contain hundreds or thousands of methods.

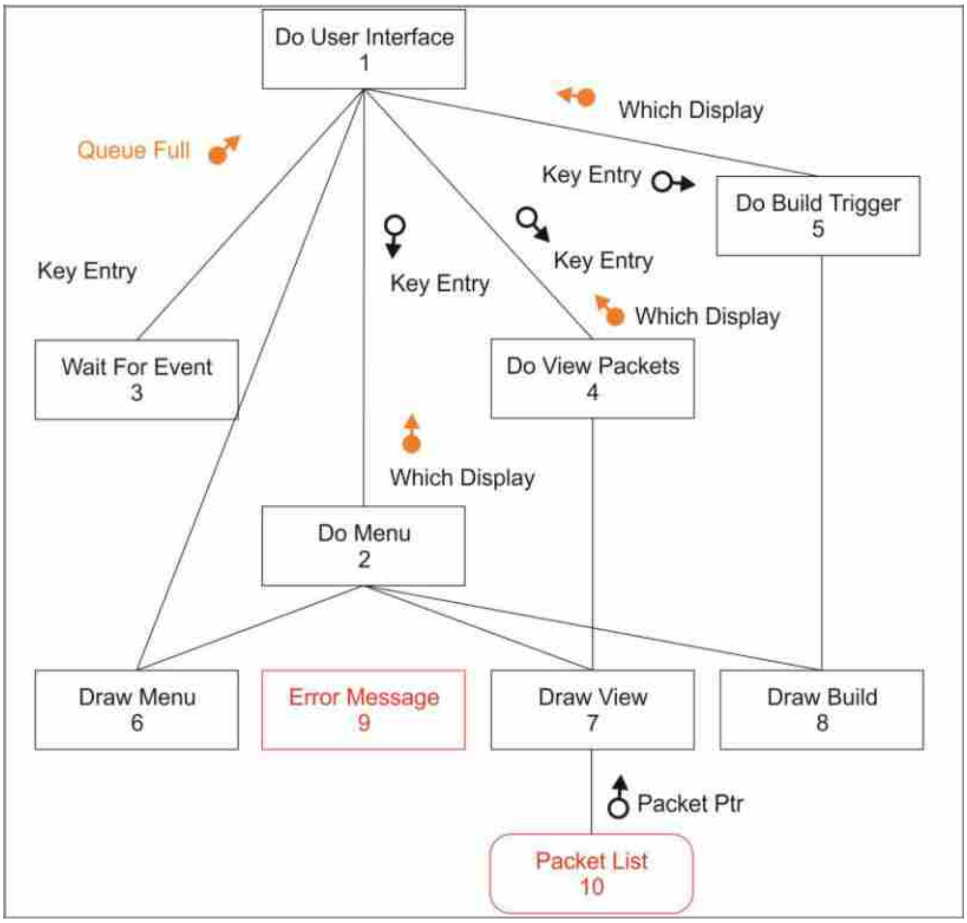


Fig. 3.17 : Structure Chart

Examples of Structure Charts :

The structural chart shows different structure of modules which can work as a customer order.

EXAMPLE OF A HIERARCHICAL NUMBERING SCHEME ON A STRUCTURE CHARG

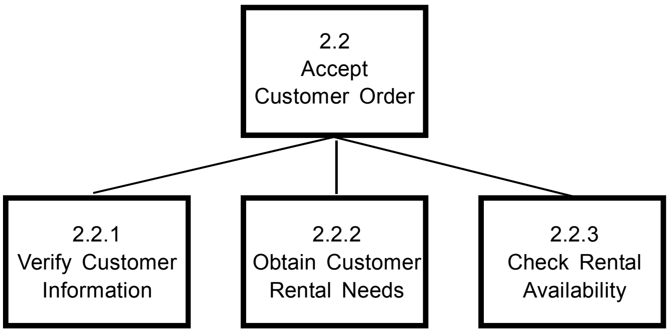


Fig. 3.18 : Example of a Hierarchical Numbering Scheme on a Structure Chart

Normally it is studied that structure charts be created with the help of CASE tool which produces the layout of particular code from structure chart. The codes if not generated from ER diagram, then the advantage of structure chart will not holds good for computer assisted drawings as compared to simple hand sketches. It is found that structure chart will not represent module size, execution time, or other operational features.

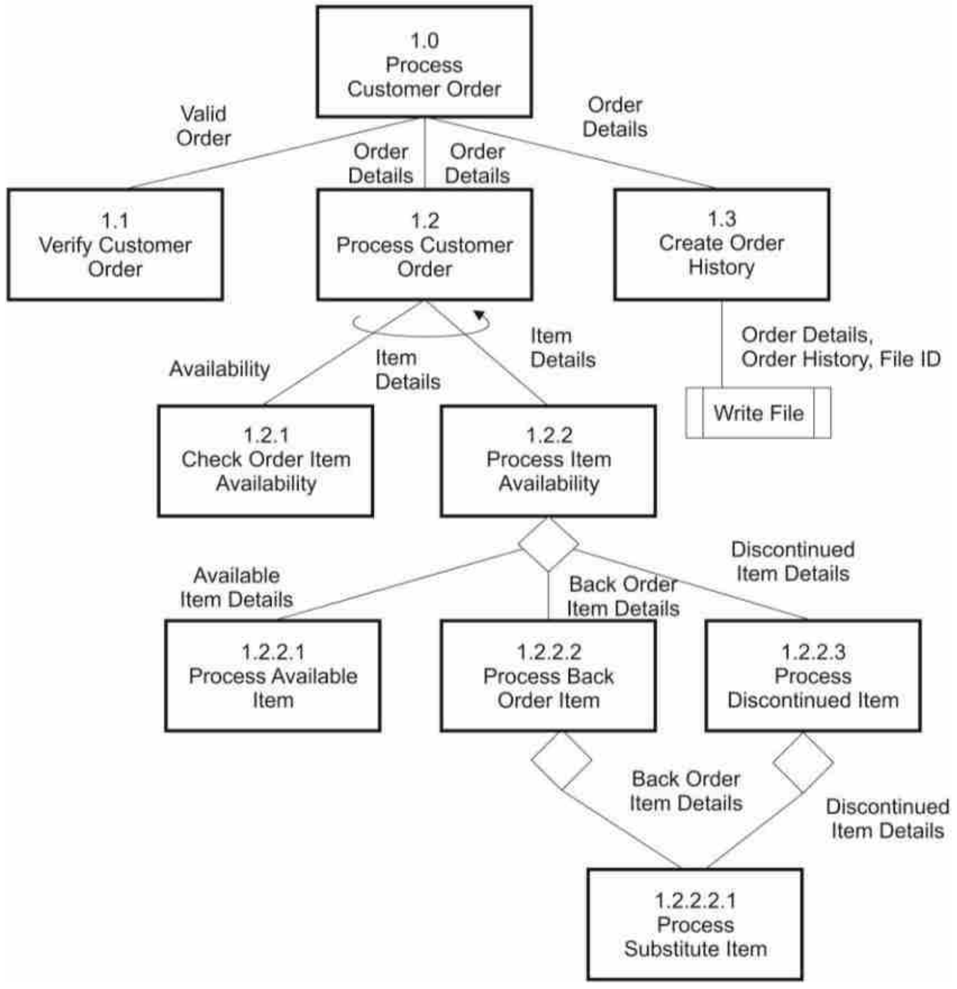


Fig. 3.19 : Example of Structure Charts

❖ Structured Flowcharts (Nassi–Shneiderman Diagrams) :

This is the traditional method of representing the flow of data in a system in schematic form. The flowchart shows the points of input and output, the logic or sequence of the various processing steps in the system, and the relationship of one element of the system to the other parts of the system or to other information systems.

❖ Data Dictionary :

A Data Dictionary is documentation that supports data flow diagrams. It contains all the terms and their definitions for data flows and data stores that relate to a specific system. The purpose of a data dictionary is to define the contents of the data flow and data stores, with the exception of the process description.

It also contains definitions for the data and control items on structure charts, which will be discussed later. Data dictionaries are needed because, while the data flow diagrams are useful for understanding what is happening, complete understanding of the data flow diagrams is not possible until you know what is meant by the various terms used on them.

In addition, you must have some methods that prevents you from calling the data flow or data store by two different names or two different data flows by same names. In other words, data dictionaries are necessary to provide consistency.

The project dictionary has a large number of different entries. A description of some possible entries is as follows:

- Data flow diagrams
- Data structure
- Process description
- Other kind of entries
- Maintenance of project dictionary

❖ Importance of Data Dictionary :

- To locate errors in the system
- To allow communication between systems elements
- To handle large system
- To determine the change in the system
- To locate omissions in the system
- To maintain document features of the system

Data dictionary consists of the three classes as follows :

- Data element
- Data structure
- Data flows and Stores

❖ Decision tree :

Decision tree is a tool, which is used to portray the logic of the policy. It shows the detail request by using a tree like structure, in which number of branches and level are used.

Such decision trees are easy to conduct, easy to tread and easy to update and understand.

Characteristics of decision tree

- Decision tree is a diagram that gives conditions sequentially.
- It also gives the sequentially action that means consider 1st which means second and second means 3rd.
- The root of the tree is at the left of the diagram.
- The starting point of system is in the form of decision sequences.
- Nodes of decision tree represent conditions of the system.

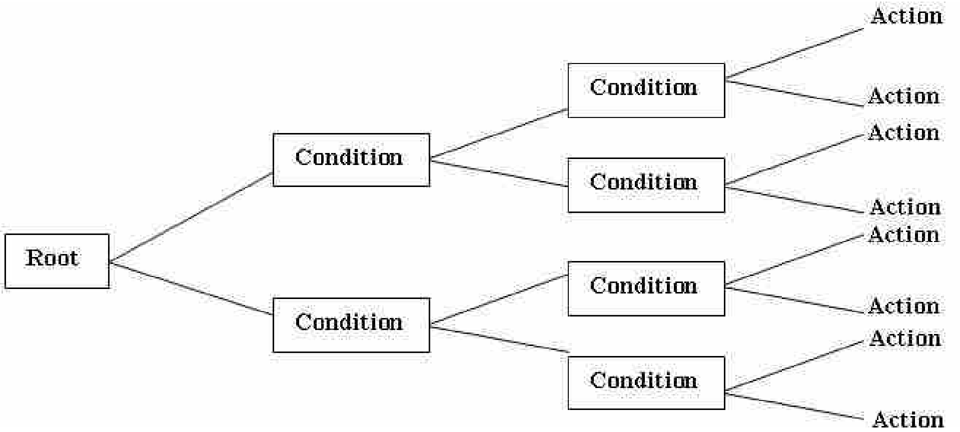


Fig. 3.20 : Decision Tree Sequence of Decisions in Form of Left to Right

❖ **Decision Table :**

A decision table represents the condition and action to be taken against that condition in the form of table. The table consists of two parts: stub and entry. The stub part is divided into an upper quadrant called the condition stub and a lower quadrant called the action stub.

Header

Condition stub	Condition entry
Action stub	Action entry

Characteristics of Decision Table

1. It is made up of four sections as follows:
   a. Condition statement
   b. Condition entries
   c. Action statement
   d. Action entries

CONDITION	DECISION RULES
Condition Statements	Condition entries
Action Statements	Action entries

Fig. 3.21 : General Forms for Decision Tables

2. Conditional statements identify the relevant condition. Condition entities are the value applied.
3. Action statement consists of all steps when certain condition occurs
4. Right side column of the table is taken as linking conditions and action form decision rules

❖ **Primary uses of Decision Table**

A decision table is useful when :

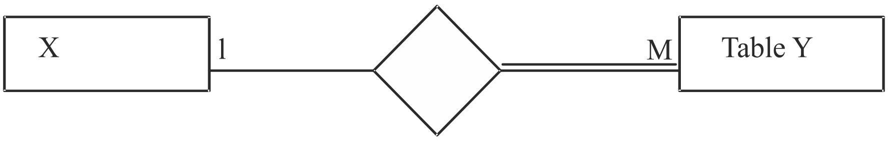
- Information has to be represented in the form of problem and action to be taken against it.
- All possible combinations of problem and the actions to be taken
- Information has to be presented in tabular format

The entry part is also divided into an upper quadrant called the conditions entry and a lower quadrant called the action entry. The condition stub part contains conditions in the form of questions; answer for which in form of Yes/ No are given in condition entry part.

Similarly, condition action part contains the statements of the actions to be taken, which action has to be taken against each of the condition, it is shown in "X" mark in entry block. The format for decision table is as shown below.

❑ **Check Your Progress – 4 :**

1. In the E–R diagram, which statement is true ?



a. Both tables carry equal primary key attributes.

b. Table X carries more key attributes than Y.

c. Table Y carries more key attributes than X.

d. None of above.

2. The entity relationship diagram is graphically representation of ________ database model.

a. condensed b. physical c. logical d. conceptual

3.6 Let Us Sum Up :

In this unit we have learnt that, system is group of components which will be able to do some work. It carries set of interrelated components working together for a common purpose.

It is found that system analysis and design for information system theory will focus on closer look of all system parts.

A computer based information system involves six– interdependent elements such as Hardware, Software, People, Procedures, Data and Output.

System analysis is the main of system development which will cover necessary data and developing plans for new systems.

It is found that there are certain activities that take place in system development that can understand of user requirement, designing and decision making, constructing a system, formation of group memory along with development of software with testing, implementation and evaluation

3.7 Answers for Check Your Progress :

❑ **Check Your Progress 1 :**

1 : d

❑ **Check Your Progress 2 :**

1 : c **2 :** b

❑ **Check Your Progress 3 :**

1 : b

❑ **Check Your Progress 4 :**

1 : d 2 : d

3.8 Glossary :

1. **Database Design –** It is a process of producing detailed data model of a database.
2. **System Life Cycle –** It is an examination of a system which shows all phases of its existence which covers system design, development, production, construction, distribution, operation, maintenance and support.
3. **System Analyst –** It is person who involved in research problems, plans solutions, recommends software and systems, and coordinates with development for business requirements.
4. **Scheduling –** It is a process of deciding how to commit resources between a varieties of possible task.

3.9 Assignment :

1. Explain the role of data in your own words.

3.10 Activities :

- Discuss DFD with relation to CASE and explain the same.

3.11 Case Study :

- Discuss the steps involved in drawing a data flow diagram and prepare a report in your own words.

3.12 Further Readings :

1. Analysis and Design and Information System, Rajaraman.
2. Analysis and Design of Information System 2nd Ed. Senn.
3. Introducing Systems Analysis and Design, Lee.
4. System Analysis and Design, Edwards, Tata McGraw–Hill.
5. Systems Analysis and Design, Elias Awad.
6. System Analysis and Design, J.F. Gerald, Tats McGraw–Hill.

BLOCK SUMMARY :

While studying this block, the user will achieve knowledge and understanding about Computer based system, System Life Cycle, Modelling of DFD, Structural methodology concepts. The block explains about System life Cycle and will explain detailed knowledge about System analyst and how students will be benefitted from this explanation.

The block has given the knowledge about Data Flow Analysis and structured modelling concepts which will help students in analysing the concept of flow of data in a design. The students or programmers will get benefit while reading this block as it gives shortcuts and related examples that will clear all doubts.

BLOCK ASSIGNMENT :

❖ **Short Questions :**

1. What is the responsibility of a system analyst ?
2. What are the various symbols of data flow diagram ?
3. What are the elements of a system ?
4. What is the use of DFD as a communication tool ?
5. Which are the areas a system analyst should have knowledge of ?

❖ **Long Questions :**

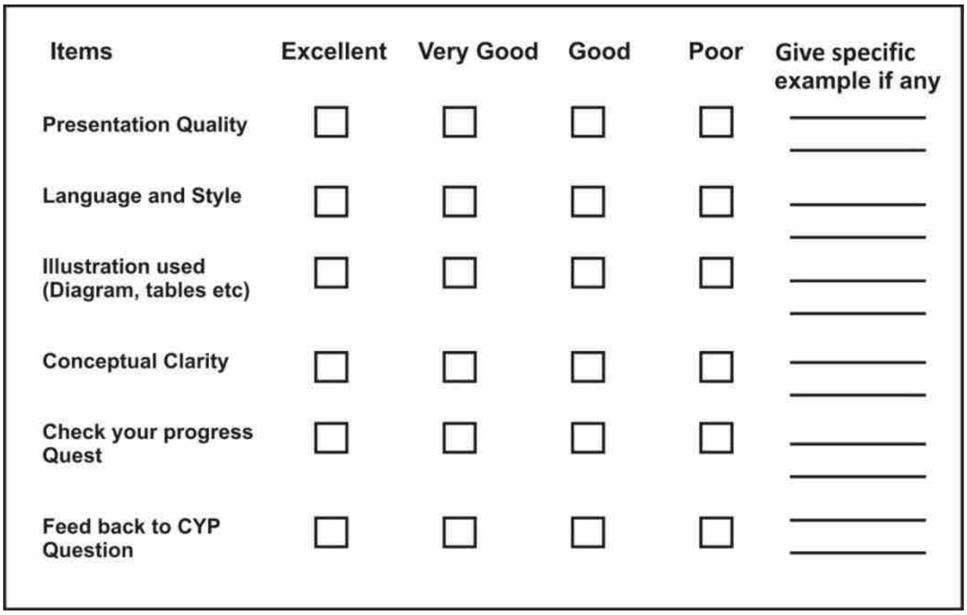
1. Describe the CASE toolsets ?
2. What do you mean by DFD and explain on what does data flow analysis focus on ?
3. Explain the concept of system and write down its important characteristics ?

❖ **Enrolment No. :**

1. How many hours did you need for studying the units ?

Unit No.	1	2	3
No. of Hrs.			

2. Please give your reactions to the following items based on your reading of the block :



Items	Excellent	Very Good	Good	Poor	Give specific example if any
Presentation Quality	□	□	□	□	______ ______
Language and Style	□	□	□	□	______ ______
Illustration used (Diagram, tables etc)	□	□	□	□	______ ______
Conceptual Clarity	□	□	□	□	______ ______
Check your progress Quest	□	□	□	□	______ ______
Feed back to CYP Question	□	□	□	□	______ ______

3. Any other Comments

....

....

....

....

....

....

....

....

Dr. Babasaheb Ambedkar
Open University Ahmedabad

BCAR-205/
DCAR-205

System Analysis and Design

BLOCK 2 : SYSTEM ANALYSIS AND PHOTOTYPING

UNIT 4 SYSTEM DEVELOPMENT LIFE CYCLE

UNIT 5 PRELIMINARY SYSTEM ANALYSIS

UNIT 6 SYSTEM REQUIREMENT SPECIFICATIONS AND ANALYSIS

UNIT 7 PROTOTYPING AND 4GLS

SYSTEM ANALYSIS AND PHOTOTYPING

Block Introduction :

To investigate a particular system, you need to gone through with its fact finding activities. Under this factual stage detection, working of system needs to be understood by system analyst in order to frame a proposed system. For this, many methods are employed and such mechanism is known as fact–finding techniques. The analyst needs to understand the current system completely. 3GLs are generally universal functions, allowing you to create an extensive range of applications. The input process should be efficient and logical. The System analysts by applying business process engineering techniques while studying transactions and business operations to determine how and when the data should be entered into the system.

In this block, we will learn and study about basic of System Analysis with basic of Detailed Analysis of a system. We will cover the topics related to Prototyping, 3GLs, 4GLs and Object Oriented Analysis. The student will be given with the knowledge about different rules and criteria's of system design.

In this block, we will learn and study about Prototyping and generation programming languages. The student will be given with the knowledge about different rules and criteria's of normalization.

Block Objectives :

After learning this block, you will be able to understand :

- The basic about analysis tools.
- Concept of fact finding and Interview.
- Idea about Review and Assignment.
- Concept of Prototyping.
- Detailed about 3GLs and 4GLs.
- Knowledge about Object Oriented Analysis.
- Features of System Design Analysis.

Block Structure :

Unit 4 : System Development Life Cycle

Unit 5 : Preliminary System Analysis

Unit 6 : System Requirement Specifications and Analysis

Unit 7 : Prototyping and 4GLS

Unit
04

SYSTEM DEVELOPMENT LIFE CYCLE

UNIT STRUCTURE

4.0 Learning Objectives

4.1 Introduction

4.2 Stages of System Development Life Cycle

4.3 Project Selection

4.4 Feasibility Study

4.5 Analysis

4.6 Design

4.7 Implementation

4.8 Post – Implementation and Maintenance

4.9 Considerations for Candidate System

4.10 Planning and Control for System Success

4.11 Let Us Sum Up

4.12 Glossary

4.13 Answer for Check Your Progress

4.14 Assignment

4.15 Activities

4.16 Case Study

4.17 Further Readings

4.0 Learning Objectives :

After learning this unit, you will be able to understand :

- How to build the computer–based information system.
- What are the different steps in system development life cycle ?
- What prompts users to change their request ?
- What are the various components of feasibility study ?
- What are the factors to consider in a candidate system ?
- How to plan and control for the system success.

4.1 Introduction :

The system analyst gives a system development project meaning & direction. A candidate system is approached after the analyst has a thorough understanding of user needs & problems. A viable solution is worked out and then communicates the same. Candidate systems often cut across the boundaries of users in the organization. For example, a billing system may involve users in the sales order department, the credit department, the warehouse and the accounting department. To make sure that all users' needs are met, a project from that represents each user works with the analysis to carry out a system development project.

4.2 Stages of System Development Life Cycle :

The system development life cycle method is classically thought of as the set of activities that analysts, designers and users carry out to develop and implement an information system. The various stages in the business are closely related to each other, even the order of the steps in these activities is difficult to determine.

4.3 Project Selection :

One must know what the problem is before it can be solved. The basis for a candidate system is recognition of a need for improving an information system or a procedure. For example, a supervisor may want to investigate the system flow in purchasing, or a bank president has been getting complaints about the long lines in the drive – in. This need leads to a preliminary survey or an initial investigation to determine whether an alternative system can solve the problem. It entails looking into the duplication of effort, bottlenecks, inefficient existing procedures, or whether parts of the existing system would be candidates for computerization.

If the problem is serious enough, management may want to have an analyst look at it. Such an assignment implies a commitment, especially if the analyst is hired from the outside. In larger environments, where formal procedures are the norm, the analyst's first task is to prepare a statement specifying the scope and objective of the problem. He/She then reviews it with user for accuracy. At this stage, only a rough "ball park" estimate of the development cost of the project may be reached. However, an accurate cost of the next phase– the feasibility study – can be produced.

Impetus for system Change: The idea for change originates in the environment or from within the firm (See Figure 4.1). Environment–based ideas originate from customers, vendors, government sources, and the like. For example, new unemployment compensation regulations may make it necessary to change the restructures. Customer complaints about the delivery of orders may prompt an investigation of the delivery schedule, the experience of truck drivers, or the volume of orders to be delivered. When investigated, each of these ideas may lead to a problem definition as a first step in the system life cycle process.

Ideas for change may also come from within the organization– top management, the user, and the analyst. As an organization changes its operations or faces advances in computer technology, someone within the organization may feel the need to update existing applications or improve procedures. Here are some examples :

- An organization acquires another organization.
- A local bank branches into the suburbs.
- A department spends 80 percent of its budget in one month.
- Two departments are doing essentially the same work, and each department head insists the other department should be eliminated.
- A request for a new form discloses the use of bootleg (unauthorized) forms.

Serious problems in operations, a high rate of labour turnover, labour intensive activities, and high reject rates of finished goods, also prompt top management to initiate an investigation. Other examples are :

- A report reaches a senior vice president and she suspects the figures.
- The company comptroller reads an IRS audit report and starts thinking.
- An executive read about decision support systems for sales forecasting and it gives him an idea.

Many of these ideas lead to further studies by management request, often funnelled downward and carried out by lower management.

User – originated ideas also prompt initial investigations. For example, a bank's head teller has been noticing long customer lines in the lobby. She wants to know whether they are due to the computer's slow response to inquiries, the new teller's limited training or just a sudden increase in bank business. To what extent and how quickly a user – originated idea is converted to a feasibility study depend on several factors :

- The risks and potential returns.
- Management's bias toward the user.
- Financial costs, and the funds, available for system work.
- Priorities of other projects in the firm.
- The persuasive ability of the user.

All these factors are crucial for a prompt response to a user request for change. A systems analyst is in a unique position to detect and even area of operations makes him/ her convenient resource for ideas. The role and status of the analyst as a professional add credibility to the suggestions made.

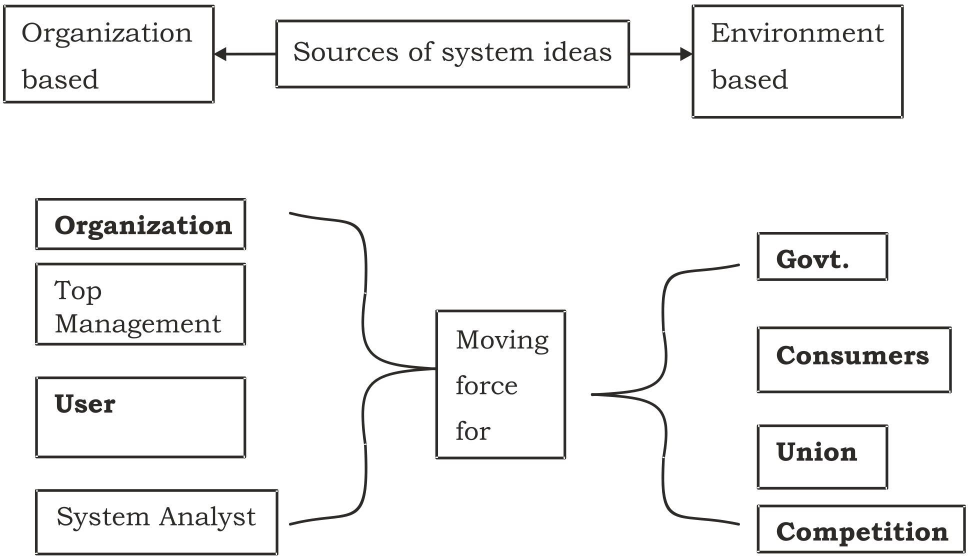


Figure 4.1 : Major Sources of Change

4.4 Feasibility Study :

Depending on the results of the initial investigation, the survey is expanded to a more detailed feasibility study. A feasibility study is a test of a system proposal according to its workability. Impact on the organization, ability to meet user needs, and effective use of resources. It focuses on three major questions :

1. What are the user's demonstrable needs and how does a candidate system meet them ?
2. What resources are available for given candidate systems ? Is the problem worth solving ?
3. What is the likely impact of the candidate system on the organization? How well does it fit within the organization's master MIS plan ?

Each of these questions must be answered carefully. They revolve around investigation and evaluation of the problem, identification and description of candidate systems, specification or performance and the cost of each system and final selection of the best system.

The objective of feasibility study is not to solve the problem but to acquire a sense of its scope. During the study the problem definition is crystallized and aspects of the problem to be included in the system are determined. Consequently, costs and benefits are estimated with greater accuracy at this stage.

The result of the feasibility study is a formal proposal. This is simply a report – a formal document detailing the nature and scope of the proposed solution. The proposal summarizes what is known and what is going to be done. It consists of the following:

1. Statement of the problem – a carefully worded statement of the problem that led to analysis.
2. Summary of findings and recommendations – a list of the major findings and recommendations of the study. It is ideal for the user who requires quick access to the results of the analysis of the system under study. Conclusions are stated followed by a list of the recommendations and a justification for them.
3. Details of findings – an outline of the methods and procedures undertaken by the existing system followed by coverage of the objectives and procedures of the candidate system. Included are also discussions of output reports, file structures, and costs and benefits of the candidate system.
4. Recommendations and conclusions – specific recommendations regarding the candidate system including personnel assignments, costs, project schedules, and target dates.

After management reviews the proposal, it becomes a formal agreement that paves the way for actual design and implementations. This is a crucial decision point in the life cycle. Many projects die here, whereas the more promising ones continue through implementations. Changes in the proposal are made in writing, depending on the complexity size, and cost of the project. It is simply common sense to verify changes before committing the project design.

4.5 Analysis :

Analysis is a detailed study of the various operations performed by a system and their relationships within and outside of the system. A key question is, what must be done to solve the problem ? One aspect of analysis is defining the boundaries of the system and determining whether or not a candidate system should consider other related systems. During analysis, data are collected on the available files, decision points, and transactions handled by the present system. Data flow diagrams interviews, on – site observations, and questionnaires are examples of the analysis tools. The interviews are a commonly used tool in analysis, it requires special skills and sensitivity to the subjects being interviewed. Bias in data collection and interpretation can be a problem. Training, experience, and common sense are required for collection of the information needed to do the analysis.

Once analysis is completed the analyst has a firm understanding of what is to be done. The next step is to decide how the problem might be solved. Thus, in systems, design we move from the logical to the physical aspects of the life cycle.

❑ **Check Your Progress – 1 :**

1. The objective of ________ is not to solve the problem but to acquire a sense of its scope.

   a. Analysis b. Design

   c. Testing d. Feasibility Study

2. Formal proposal is the outcome of ________.

   a. Analysis b. Design

   c. Testing d. Feasibility Study

3. ________ is a detailed study of the various operations performed by a system and their relationships within and outside of the system.

   a. Analysis b. Design

   c. Implementation d. Feasibility Study

4. Information gathering is mainly focused during ________ phase of SDLC.

   a. Analysis b. Design

   c. Implementation d. Feasibility Study

5. ________ activity(ies) should be conducted during Analysis phase of SDLC.

   a. On site observation b. Interviews

   c. Questionaries d. All of the above

4.6 Design :

The most creative and challenging phase of the system life cycle is system design. The term design describes a final system and the process by which it is developed. It refers to the technical specifications (analogous to the engineer's blueprints) that will be applied in implementing the candidate system. It also includes the construction of programs and program testing. The key question here is: How should the problem be solved ? The major steps in design are shown in Figure 4.2.

The first step is to determine how the output is to be produced and in what format. Samples of the output (and input) are also presented. Second, input data and master files (database) have to be designed to meet the requirements of the proposed output. The operational (processing) phases are handled through program construction and testing including a list of the programs needed to meet the system's objectives and complete documentation. Finally, details related to justification of the system and an estimate of the impact of the candidate system on the user and the organization are documented and evaluated by management as a step toward implementation.

The final report prior to the implementation phase includes procedural flowcharts, record layouts, report layouts, and a workable plan for implementing the candidate system. Information on personnel, money, hardware, facilities, and their–estimated cost must also be available. At this point, projected costs must be close to actual costs of implementation.

In some firms, separate groups of programmers do the programming, whereas other firms employ analyst– programmers who do analysis and design as well as code programs. For this discussion, we assume that two separate persons carry out analysis and programming. There are certain functions, though, that the analyst must perform while programs are being written. Operating procedures must also be developed.

4.7 Implementation :

The implementation phase is less creative than system design. It is primarily concerned with user training site preparation, and file conversion. When the candidate system is linked to terminals or remote sites, the telecommunication network and tests of the network along with the system are also included under implementation.

During the final testing, user acceptance is tested, followed by user training. Depending on the nature of the system, extensive user training may be required. Conversion usually takes place at about the same time the user is being trained or later.

In the extreme, the programmer is falsely viewed as someone who ought to be isolated from other aspects of system development. Programming is itself design work, however. The initial parameters of the candidate system should be modified as a result of programming efforts. Programming provides a "reality test" for the assumptions made by the analyst. It is therefore a mistake to exclude programmers from the initial system design.

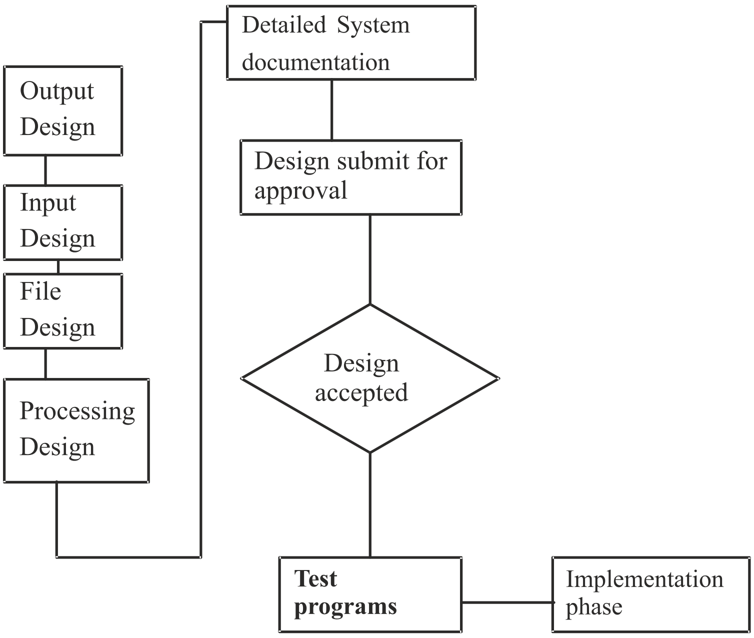


Figure 4.2 : Steps in Systems Design

System testing checks the readiness and accuracy of the system to access, update and retrieve data from new files. Once the programs become available, test data are read into the computer and processed against the file(s) provided for testing. If successful, the program(s) is then run with "live" data. Otherwise, a diagnostic procedure is used to locate and correct errors in the program. In most conversions, parallel run is conducted where the new system runs simultaneously with the "old" system. This method, though costly, provides added assurance against errors in the candidate system and also gives the user staff an opportunity to gain experience through operation. In some cases, however, parallel processing in not practical. For example, it is not plausible to run parallel two online point–of–sale (POS) systems for a retail chain. In any case, after the candidate system proves itself, the old system is phased out.

4.8 Post – Implementation and Maintenance :

After the installation phase is completed and the user staff is adjusted to the changes created by the candidate system, evaluation and maintenance begin. Like any system there is an aging process that requires periodic maintenance of hardware and software. If the new information is inconsistent with the design specifications, then changes have to be made. Hardware also requires periodic maintenance to keep in tune with design specifications. The importance of maintenance is to continue to bring the new system to standards.

User priorities, changes in organizational requirements, or environmental factors also call for system enhancements. To contrast maintenance with enhancement, if a bank decided to increase its service charges on checking accounts from Rs. 3.00 to Rs. 4.50 for a minimum balance of Rs. 300, it is maintenance. However, if the same bank decided to create a personal loan on negative balances when customers overdraw their account, it is enhancement. This change requires evaluation program modifications, and further testing.

❑ Check Your Progress – 2 :

1. During the ________ phase of SDLC, blueprint of the system is prepared.

   a. Analysis b. Design

   c. Testing d. Feasibility Study

2. During design phase of SDL, manly ________ activity(ies) is done.

   a. Preparation of Input and Database

   b. Preparation of Flowcharts and Procedures

   c. Estimation of Implementation Cost

   d. All of the above

3. ________ is primarily concerned with user training site preparation, and file conversion.

   a. Design b. Analysis

   c. Implementation d. Feasibility Study

4. User priorities, changes in organizational requirements, or environmental factors kind of optimization is done during ________ phase of SDLC.

   a. Design b. Analysis

   c. Implementation d. Post-Implementation

❖ **Project Termination :**

A system project may be dropped at any time prior to implementation although it becomes more difficult (and costly) when it goes past the design phase. Generally, projects are dropped if, after a review process, it is learned that :

- Changing objectives or requirements of the user cannot be met by the existing design.
- Benefits realized from the candidate system do not justify commitment to implementation.
- There is a sudden change in the user's budget or an increase in design costs beyond the estimate made during the feasibility study.
- The project greatly exceeds the time and cost schedule. In each case, a system project may be terminated at the user's request. In contrast project termination is new system failure. There are many reasons a new system does not meet user requirements:
- User requirements were not clearly defined or understood.
- The user was not directly involved in the crucial phases of system development.
- The analyst, programmer, or both were inexperienced.
- The systems analyst (or the project team) had to do the work under stringent time constraints. Consequently, not enough thought went into the feasibility study and system design.
- User training was poor.
- Existing hardware proved deficient to handle the new application.
- The new system left users in other departments out of touch with information that the old system had provided.
- The new system was not user–friendly.
- Users changed their requirements.
- The user staff was hostile.

The list can be expanded to include many more causes. The important point is that although advances in computer systems and software make life easier for the analyst, the success of a system project depends on the experience, creative ability, and knowledge of the analyst and the support from the user staff. This suggests that the analyst be skilled in the state of the art (hardware and software) as well as in dealing with people.

4.9 Considerations for Candidate System :

In today's business, there is more demand for computer services than there are resources available to meet the demand. The demand is made up of the following :

1. Operations of existing system.
2. Maintenance that focuses on "patching" programs – often representing over 50 percent of maintenance.
3. Enhancements that involve major modifications in program structure or equipment.
4. Requests for candidate systems.

All these demands require resource – human, financial, and technological. On the human side, the computer department has to provide the following:

- Computer operators to run equipment.
- Data entry personnel.
- Systems analysts to define and design specifications.
- Application programmers to convert system specifications to computer programs
- Maintenance programmers to repair errors.
- Supervisors, project leaders, and managers to coordinate the jobs with the users.

Thus, the basic problem is to match the demands for service with the available resources. How much one project is favoured over another depends on technical, behavioural, and economic factors.

The technical factor involves the system department's ability to handle a project. Much depends on the availability of qualified analysts, designers, and software specialists to do the work. This is especially true in designing databases and implementing complex systems for large concerns. The alternative to abandoning a project because of limited talent on the inside is free – lancing it to an outside consulting firm. The cost of developing the project has to be weighed against the total benefits expected.

The behavioural factor involves (1) the user's past experience with an existing system (2) the success record of the analyst, and (3) the influence the user can exert on upper management to finance a candidate system. Political considerations that subjectively favour one project over another, the status of the department, and its performance record are additional factors that bear on funding a candidate system.

Perhaps the most important criterion in selecting a project is the economic factor. It focuses on the system's potential return on investment. What is considered an acceptable rate varies with different formulas, the variables chosen, and the like. System consultants suggest an annual rate of return of just over 20 percent.

4.10 Planning and Control for System Success :

What can the analyst do to ensure the success of a system ? First, a plan must be devised, detailing the procedure, some methodology, activities, resources, costs, and timetable for completing the system. Second, in larger projects, a project team must be formed of analysts, programmers, a system consultant, and user representatives. Shared knowledge, interaction, and the coordination realized through team effort can be extremely effective in contrast with individual analysts doing the same work. Finally, the project should be divided into manageable modules to reflect the phases of system development – analysis, design, and implementation.

Most of this work falls under project management and control. The main idea behind the system development life cycle is to formalize a means structured at three major levels for effective control of the project. At the lowest level, work assignments are broken down into small manageable tasks. A task is usually a well – defined, structured work unit that can be carried out by one individual. The task can be easily budgeted and scheduled and its quality measured. It can be easily completed independent of other tasks and other project team members. If rework is necessary, there is minimal loss or impact on other tasks, except where time is critical.

The second level at which work units are structured involves activities that have larger scope and are designed to produce substantial results. An activity is a group of logically related tasks that serve one phase of the system development life cycle.

A phase, a third level of control, is a set of activities that bring the project to a critical milestone. Milestones are steppingstones that make up the entire project.

In planning a project, the following steps should be taken :

1. Identify the activities in each phase and the tasks within each activity.
2. Calculate the budget for each phase and obtain agreement to proceed.
3. Review, record, and summarize progress on activities periodically.
4. Prepare a project progress report at the end of a reporting month.

In summary, system development should not be regarded merely as some procedure that deals with hardware and software. The original assumptions upon which system specifications were based should be tested and re–evaluated with the user in mind. Managing system projects includes the important responsibility of seeing to it that all features of the candidate system – technological, logical, and behavioural – are considered before implementation and maintenance.

❑ **Check Your Progress – 3 :**

1. From the given below options, which is not a project termination criterion ?

   a. Changing objectives or requirements of the user cannot be met by the existing design.

   b. Existing hardware proved sufficient to handle the new application.

   c. The analyst, programmer, or both were inexperienced.

   d. User training was poor.

2. From the given below options, which is not a project termination criterion ?

   a. The user staff was not hostile.

   b. The new system was not user-friendly.

   c. Users changed their requirements.

   d. User training was poor.

3. Behavioural factors includes.

   a. The user's past experience with an existing system.

   b. The success record of the analyst.

   c. The influence the user can exert on upper management to finance a candidate system.

   d. All of the above

4.11 Let Us Sum Up :

System analysis and design are keyed to the system development life cycle (SDLC). The stages are project selection, feasibility, analysis, Design, implementation, and post implementation stages. The idea for the project is originates in the environment or from within the organization. Once the problem is verified an initial investigation is conducted to determine whether change is feasible. If the answer is yes, a feasibility study is conducted. Analysis is a detailed study of the various operation performed by a system. System design refers to the technical specifications that will be applied in implementing the candidate system. Implementation is concerned with details of the candidate system. After implementation, maintenance begins includes enhancements, modifications, or any changes from the original specifications. To ensure the success of the system, careful and often extensive planning is required. The overall management process is crucial to the successful completion of system.

4.13 Answers for Check Your Progress :

- ❑ **Check Your Progress 1 :**

1 : d	**2 :** d	**3 :** a
4 : b		**5 :** d

- ❑ **Check Your Progress 2 :**

1 : b	**2 :** d	**3 :** c	**4 :** d

4.12 Glossary :

1. **SDLC :** SDLC stand of System Development Project.
2. **Feasibility Study :** It is a phase of SDLC, where the system under development is Technically and Economically feasible or not is tested.
3. **Analysis :** Analysis is a detailed study of the various operations performed by a system and their relationships within and outside of the system.
4. **Design :** The term design describes a final system and the process by which it is developed. It refers to the technical specifications (analogous to the engineer's blueprints) that will be applied in implementing the candidate system.

4.14 Assignment :

1. Why is a system proposal so crucial for system design ?
2. What is System Development Life Cycle ?
3. What is the difference between analysis and design ? Explain.
4. How would an analysis determine the users' needs for a system ? Explain.
5. Distinguish between initial investigation and feasibility study. In what way are they related ?

4.15 Activities :

- Find out what is Testing ? What is the need of it? How many types of testing are there ?
- Make a list of considerations in deciding on a candidate system.

4.16 Case Study :

- Do the details and Phase wise analysis of the following systems. List all the steps that you take during each phase of the SDLC during making the following systems as a Software Engineer :
  - o Payroll System.
  - o Library Automation System

4.17 Further Readings :

1. Analysis and Design and Information System, Rajaraman.
2. Analysis and Design of Information System 2nd Ed. Senn.
3. Introducing Systems Analysis and Design, Lee.
4. System Analysis and Design, Edwards, Tata McGraw-Hill.
5. Systems Analysis and Design, Elias Awad.
6. System Analysis and Design, J. F. Gerald, Tats McGraw-Hill.

Unit 05

PRELIMINARY SYSTEM ANALYSIS

UNIT STRUCTURE

5.0 Learning Objectives

5.1 Introduction

5.2 Fact Finding and Interview

5.3 Detailed Analysis

5.4 Review and Assignment

5.5 Working with People with Technology

5.6 Let Us Sum Up

5.7 Answers for Check Your Progress

5.8 Glossary

5.9 Assignment

5.10 Activities

5.11 Case Study

5.12 Further Readings

5.0 Learning Objectives :

After learning this unit, you will be able to understand :

- The preliminary system analysis
- Explain fact finding and interviews
- Describe detailed analysis
- Discuss reviews and assignments
- Outline tenets of working with people and technology

5.1 Introduction :

We have studied about the various modelling tools, which are used by system analyst. Now in this unit, we will discuss the preliminary and detailed system analysis techniques, which is the first stage of Software Development Life Cycle (SDLC).

The preliminary analysis includes fact–finding and conducting interviews with people to know their requirement and after that, the detailed analysis is done, where reviewing of the preliminary analysis is done and after that assignment of tasks is done.

5.2 Fact Finding and Interview :

❖ **Preliminary Investigation :**

Preliminary investigation is the first phase of System Development Life Cycle. The main objective of this phase is to identify deficiencies and requirements in the user's current environment. An important result of the preliminary investigation is whether the system to be developed is feasible or not.

Feasibility is determined on the following parameters :

- Whether current technical resources or technology is available in the developer's organisation or in the market which is capable of handling the user's requirements
- Whether economically or financially the system is cost effective
- How effectively the user will operate this software, once installed

Feasibility study report is produced at the end of this phase. The latest approval of desired system needs to be approved from the user. Once the proposal is agreed, then the planning of next phase will get started.

To be cleared with proposal and its limitations, the procedure allows to first investigating about the system. This is known as fact finding procedure where the working of the system can easily think off by system analyst who will design the layout of desired system. To investigate about a system, different methods are used but the system analyst needs to master on present system in order to work well.

To develop a layout of the proposed system, the system analyst needs to collect information and data and will decide for extra demands in order to conduct the system with various facts finding techniques. It is seen that different categories of techniques are applied with most common being the interviews, questionnaires, record reviews, CASE tools and personal observations.

Various kinds of techniques are used and the most popular among them are

- Interviews
- Questionnaires
- Record reviews
- Personal observations made by the analyst himself

Interview is way of collecting information or data where there occurs interaction between an analyst, the system and user of proposed system.

In interview, the interviewer initially set up a relationship with interviewee, where an interviewee may or may not be a technical handed which could be a hindrance for an analyst as it becomes difficult for analyst to use day–to–day language rather than terminology or technical terms.

To respond well in an interview, an analyst should prepare, as he needs to be in advance knowledge of what is required and until what level. Additionally, he should collect required information and data. The idea behind an interview is to allow analyst to work and pose anything as required with collecting details from required people by way of questionnaires, previous records, etc.

By adopting such means, the analyst will be in position to verify and validate information what he has collected. Interviewing should be logical with general point of view. For an interview to be successful, follow the steps:

- Setup interview period
- Create an understanding with the interviewee
- Ask questions clearly and briefly
- Become a listener and not a debater
- Find the net result of interview

The interviews are of two types namely structured and unstructured.

1. Structured Interview
2. Unstructured Interview

1. Structured Interview :

Structured interviews are such type of interview where an interviewee posed a standard set of questions in a desired order. In this type of interview, almost every interviewee poses the similar set of questions. In such interview, the questions are segmented in two formats. The first format is open response format where an interviewer can answer questions in his/her words.

2. Unstructured Interview :

In case of an unstructured interview, the interview is performed in question and answer format. It is easy and lucrative as compared to structured interview which can easily collect general information about system. In such interview, the respondent is free to answer the questions in his/her words. With this, the interviewer receives bigger domain of exploring and collecting data.

❖ **Structured Vs. Unstructured Interviews :**

It is studied that structured and unstructured interview methods contains its' own merits and demerits. In case of structured interview, the collection of information is easy and fast at it carries definite platform, whereas in case of unstructured interview, the questions are answered with own thinking, so collection and implementing different views requires time.

It is seen that in an unstructured interview, the respondents can answer in her/his words. This result in discussion as there are chances where some issue originates based on thinking. In such case, the respondent can express views on that issue also.

❖ **Questionnaires :**

Questionnaires is a technique of collecting information by forms or online where potential users of the system are required to fill the details that can be handed over to system analyst further for screening.

This type of technique is best applied for collecting information from group of people. The development of question depends on the requirement of system which is small and concise. Additionally, in short span of time, the questions are to be answered as all the questions are compulsory. In the system, if the analyst guarantees for secrecy of questions answered then the respondent will be able to answer the questionnaires honestly and correctly. The questionnaire will not result for such respondents who are busy or who are not giving priority to the questionnaire.

It is the responsibility of the system analyst to design and frame questionnaire with the idea that it should meet the entire objective and the questions so designed should match with the actual cost that appears while designing and distribution of system. According we see that questionnaires are of two types :

1. Open–Response Questionnaire
2. Closed–Response Questionnaire

1. Open–Response Questionnaire :

The idea behind the open response questionnaire is to collect data and information regarding the required and critical design features of particular system. In case of open ended question, there will be no requirement of response direction or particular response.

This type of questionnaire is applied to get knowledge on feelings, opinions and experiences of respondents. With this, the system can become much effective as the analyst can propose following amendment as per the knowledge gained.

2. Closed–Response Questionnaire :

The main aim behind closed response questionnaire is to gather necessary facts and data from required system. The main part of this response is that it shows in–depth information onto how the people will deal with such types of system and how comfortable they are adopting. For this, every respondent needs to select from group of given responses. So in that the respondents will express themselves from the most convenient possible options. The closed response questionnaire may contain questions based on:

- Fill–in–blanks
- Yes or No type
- Ranking scale questions where respondents have to rank the items as per the desired preferences.
- Multiple–choice questions where the respondents has to select the best option
- Rating scale questions which are more advanced than multiple choice questions, where the respondent has to rate the options out of given scale

❖ Open–Response Vs Closed–Response Questionnaire :

The questionnaire can be compared base on their respective formats. In case of an open form, the format is flexibility where the respondents are free to select and work while in case of closed form format, it is more specific.

The usability of open ended questionnaires exists when there is a need of identifying certain different situation. Such type of question format takes a lot of time for evaluating.

The closed questionnaires are applicable when the data or information needs to be obtained. Such type of questionnaire is fast and easy to analyse but requires more cost to get it prepared as it is best suited for actual and standard information.

For selecting the required question format, it is the duty of system analyst to decide the type of format required and what should be the required objectives. While designing and framing, it should be taken care that all the required parts of the questionnaire form should be simple for respondents which makes them to answer questionnaire with clarity and accurately.

❖ Working with People :

The primary purpose of an information system is to provide valuable information to users. Users, sometimes called end users, are the people who interact with an information system, both inside and outside the company. Internal users include administrators, managers, technicians, sales staff and corporate officers. External users include customer's system to plan their manufacturing schedules. The success or failure of a system usually depends on whether users are satisfied with the system's output and operations.

To serve users, successful information systems depend on skilled professionals, such as system analysts, programmers, network administrators and other IT staff members.

❑ **Check Your Progress – 1 :**

1. The main goal of an interview is to :

   a. allow analyst to work　　b. pose anything required

   c. collecting details　　d. all

2. Which information is correct in regards to closed questionnaires ?

   a. used to get information

   b. quick to analyse

   c. not costly to prepare

   d. suitable for factual and common information

5.3 Detailed Analysis :

Analysis is collective study of different operations that are worked out by a system along with its relationships inside and outside of system. The analysis will locate the system problems and amend it as per user requirements.

During analysis, the analysts will find and locate about the prevailing systems along with its procedures. They also indicate what input will be needed by the systems and what outputs will be produced i.e. in feasibility study the analyst has to do evaluation of existing systems and procedures. He has to present a number of alternative solutions to the user. After consulting with the user, the analyst has to finalise one alternative, which will be best for all the given solutions.

Such phase will lead to describe about a detailed model of system which explains about system functions, system data as well as system information flows. After that the analyst will decide how to solve the problems.

Table Initial Investigations Vs Feasibility Study

Sr. No.	Sr. No.	Sr. No.
1	Initial investigation is the first stage in system analysis	According to the results of initial investigation second stage feasibility
2	Initial investigation performs survey of present procedure	Feasibility study performs detailed survey of present procedures
3	Initial investigation does not give any clear cut alternative method	Feasibility study gives one most suitable method for system design

❑ **Check Your Progress – 2 :**

1. The terms analysis with respect to system means :

   a. detailed study of various operations

   b. study of system programming

   c. study about system analyst

   d. all of these

5.4 Review and Assignment :

Reviews can be :

- Formal or informal
- By management , peers or users
- By oral presentation, demonstration, formal document submission or a combination of all three.
- Interim, checkpoint, milestone or decision point.
- Overview or in–depth.

Review serves as an important aspect of life cycle. The hold flaws in logic, perception or design and advice corrective actions. It is noted that complete reviews should give out approval of finished or submitted product. It is found that the approved product should form the basis of continued work. Being different types and methods used in reviews, apart from this, there are certain items that are common to all reviews such as :

- It should be directed by any non–project worker.
- It should be carried off by people who are involved in project not necessarily with development.
- The product draft should be distributed to reviewers before final settlement.
- The aim of the review should be stated clearly.
- Any comment should be submitted and accepted for final decision.
- The comment, criticism and suggestion related to any product should be filled on review form.
- The comment, criticism and suggestion be evaluated by submitter with necessary action been taken should be noted.
- The reviewer should enclose a copy of final product with relevant changes highlighted.
- Finally, the review should be carried off by approval sheet with signature.
- The reviewer comments with all required approval sheets are enclosed along with the project documentation for improvement.

Designing and collecting reviews on particular product is a time consuming work as it requires lots of effort to prepare and conduct. The advantage of the review is that you can get correct views of people about the product and will save time, money and effort, if any fault is corrected or occurred. If the fault is corrected earlier, then it will cause less damage and if the fault persists longer, then the fault requires more time, more labour and more cost to amend.

Recurrent reviews will save the project from deflecting from its schedule and goals. With reviews, the project team will focus more on quality of the product, which will help them to meet their goals, direction and approach. Reviews can supply regular project corrections which are required to make sure about successful completion of project and its related activities.

❑ **Check Your Progress – 3 :**

1. Reviews can be conducted :

a. orally b. documenting c. verbally d. all

5.5 Working with People with Technology :

The given figure shows the importance of people, process and technology for making any software. People mean developers, users and clients and process means the network of activities that are expected to be conducted in software development. Technology is the tool used for creating software.

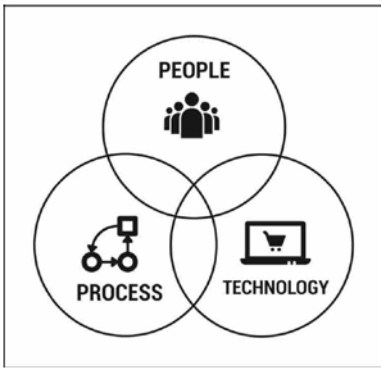


Fig. 5.1 : Combination of People, Process and Technology

❑ **Check Your Progress – 4 :**

1. To prepare software what is not compulsory ?

a. people b. office c. process d. technology

5.6 Let Us Sum Up :

In this unit we have learnt that the preliminary investigation occurs as the initial phase of System Development Life Cycle whose main idea is to find deficiencies and need of various users' environment.

Further it is seen that an analyst requires data related to need and demand of project that was initiated with several techniques that are used to collect such data. This type of techniques is known as fact–finding techniques.

This type of techniques covers several domains such as interviews, questionnaires, record reviews, case tools and other personal observations that were made by the analyst itself.

Questionnaires result as an important method to collect users view and to gather information about the system from several users which in turn will submit the details to the analyst. Such process is application for group and not for single individual.

It is found that analysing certain operations used by the system deals with scope and relationships that exists within or outside system by collecting relevant details. At the time of analysing information, the analysts have to find an existing systems and procedures.

Reviews play an important role in project life cycle as it can catch hold of any fault in logic, designing and if something is essential for a product.

5.7 Answers for Check Your Progress :

❑ **Check Your Progress 1 :**

1 : d **2 :** c

❑ **Check Your Progress 2 :**

1 : a

❑ **Check Your Progress 3 :**

1 : d

❑ **Check Your Progress 4 :**

1 : b

5.8 Glossary :

1. **System Life Cycle** – It is the screening of system where all phases of system exists which covers system design, development, production, construction, distribution, operation, maintenance and support

2. **System Analyst** – It is person who involved in research problems, plans solutions, recommends software and systems, and coordinates with development for business requirements.

3. **Scheduling –** It is a process of deciding how to commit resources between a variety of possible task.

4. **Model** – A model is an abstraction of some aspect of an existing or planned system.

5. **Preliminary Analysis** – The first phase of process which occurs at beginning of project which shows whether method is viable or not.

6. **CASE Tools** – These are computer aided tools used for designing of hardware.

7. **Structured Interview** – It is a type of interview that gathers data for statistical survey.

8. **Questionnaires –** A physical or electronic form of data collection format where the interviewers expresses their views.

5.9 Assignment :

1. Prepare a detailed report on reviews.

5.10 Activities :

- Explain preliminary investigation in your own words.

5.11 Case Study :

- Find all the various methods of carrying out preliminary analysis of Software Development Life Cycle (SDLC) and prepare a report

5.12 Further Readings :

1. Analysis and Design and Information System, Rajaraman.
2. Analysis and Design of Information System 2nd Ed. Senn.
3. Introducing Systems Analysis and Design, Lee.
4. System Analysis and Design, Edwards, Tata McGraw–Hill.
5. Systems Analysis and Design, Elias Awad.
6. System Analysis and Design, J.F. Gerald, Tats McGraw–Hill.

Unit 06

SYSTEM REQUIREMENT SPECIFICATIONS & ANALYSIS

UNIT STRUCTURE

6.0 Learning Objectives

6.1 Introduction

6.2 What is Requirements Determination ?

6.3 Fact – Finding Techniques

6.4 What is Structured Analysis ?

6.5 Pros and Cons of Each Tool

6.6 Let Us Sum Up

6.7 Answer for Check Your Progress

6.8 Glossary

6.9 Assignment

6.10 Activities

6.11 Case Study

6.12 Further Readings

6.0 Learning Objectives :

After learning this unit, you will be able to understand :

- What is the importance of system requirement specification ?
- How the facts are found and what are the methods
- What tools are used in structured analysis's
- How to construct a data flow diagram
- What are the advantages and uses of a data dictionary and structured English
- The elements and constriction of decision trees and decision tables

6.1 Introduction :

Analysis is the heart of the process. It is the key component of the first two phases of the cycle. In analysis the present system, the analyst collects a great deal of relatively unstructured data through interviews, questionnaires, on–site observations, procedures manuals, and the like. The traditional approach is to organize and convert the data though system flowcharts, which support future developments of the system and simplify communication with the user. But the system flowchart represents a physical rather than a logical system. It makes it difficult to distinguish between what happens and how it happens in the system.

There are other problems with the traditional approach.

1. The system life cycle provides very little quality control to ensure accurate communication from user to analyst. They have no language in common.

2. The analyst is quickly overwhelmed with the business and technical details of the system. Much of the time is spent gathering information. The details are needed and must be available, but the analyst does not have the tools to structure and control the details.

3. Present analytical tools have limitations.

   a. English narrative descriptions of a system are often too vague and make it difficult for the user to grasp how the parts fit together. Furthermore, English is inherently difficult to use where precision is needed.

   b. System and program flowcharts commit to a physical implementation of the system before on has complete understanding of its logical requirements.

4. Problems also relate to system specifications :

   a. System specifications are difficult to maintain or modify. A simple change in the user's requirements necessitates changes in several parts of the document.

   b. They describe user requirements inn terms of physical hardware that will implement the system rather than what the user wants the system to do.

   c. They are monolithic and redundant; that is, to find out information about a particular part of the system, the user has to search the entire document. Furthermore, the same information is found in numerous locations with no cross–reference.

Because of these drawbacks, the analyst needs something analogous to the architect's blueprint as a starting point for system design. It is a way to focus on functions rather than physical implementation. One such tool is the data flow diagram (DFD). There are other tools as well. The use of several tools in structured analysis, including the following :

1. Data flow diagram (DFD).
2. Data dictionary.
3. Structured English.
4. Decision trees.
5. Decision tables.

System analysis is about understanding situations, not solving problems. Effective analysts therefore emphasize investigation and questioning to learn how a system currently operates and to identify the requirements users have for a new or modified one. Only after analysts fully understand the systems are they able to analyze it and assemble recommendations for systems design.

The manner in which a systems investigation is conducted will determine whether the appropriate information is gathered. In turn, having the right information influences the quality of the application that follows. In other words, good system design, whether developed through the SDLC method, prototyping, or structured methods, begins by documenting the current system and properly diagnosing systems requirements.

❑ **Check Your Progress – 1 :**

1. SRS document is the outcome of ________ phase of SDLC.

   a. Analysis b. Design c. Testing d. Feasibility Study

2. From the given options, identify the tools used for structured analysis.

   a. DFD b. Data Dictionary

   c. Decision Table d. All of the above

3. From the given below identify the tool which not used for structured analysis.

   a. Decision Tree b. Data Dictionary

   c. Use Case d. Structured English

6.2 What is Requirements Determination ?

Requirements determination involves studying the current business system to find out how it works and where improvements should be made. Systems studies result in an evaluation of how current methods are working and whether adjustments are necessary or possible. These studies consider both manual and computer methods, they are not merely computer studies.

A requirement is a feature that must be included in a new system. It may include a way of capturing or processing data, producing information, controlling a business activity, or supporting management. The determination of requirements thus entails studying the existing system and collecting details about it to find out what these requirements are.

Since systems analysts do not work as managers or employees in user departments (such as marketing, purchasing, manufacturing, or accounting), they do not have the same base of acts and details as the managers and users in those areas. Therefore, an early step in analysts, investigation is to understand the situation. Certain types of requirements are so fundamental as to be common in most all situations. Developing answers to a specific group of questions (to be discussed in this section) will help you understand these basic depending on whether the system is transaction – or decision – oriented and whether the system cuts across several departments. For example, the need to inform the inventory manager of an unusually large order that is forthcoming underscores the importance of linking the sales, purchasing, and warehouse departments.

❖ **Activities in Requirement Determination :**

It is helpful to view requirements determination through the three major activities of and requirements specification.

❖ **Requirement Anticipation :**

Having had experience in a particular business area or having encountered systems in an environment similar to the one currently under investigation will influence systems analysts study. They may foresee the likelihood of certain problems or features and requirements for a new system. As a result, the features they investigate for the current system, questions they raise, or methods employed may be based on this familiarity.

Requirements anticipation can be a mixed blessing. On the one hand, experience form previous studies can lead to investigation of areas that would otherwise go unnoticed by an inexperienced analyst. Having the background to know what to ask or which aspect to investigate can be a substantial benefit to the organization.

On the other hand, if a bias is introduced or shortcuts are taken in conducting the investigation, requirements anticipation is a problem. We will point out guidelines for structuring an investigation around basic questions to avoid the undesirable consequences of requirements anticipation.

❖ **Requirements Investigation :**

This activity is at the heart of systems analysis. Using a variety of tools and skills, analysts study the current system and document its features for further analysis.

Requirements investigation relies on the fact–finding techniques and includes methods for documenting and describing system features.

❖ **Requirements Specifications**

The data produced during the fact–finding investigation are analyzed to determine requirements specifications, the description of features for a new system. This activity has three interrelated parts :

- **Analysis of Factual Data**

The data collected during the fact – finding study and included in data flow and decision analysis documentation are examined to determine how well the system is performing and whether it will meet the organization's demands.

- **Identification of Essential Requirements**

Features that must be included in a new system, ranging from operational details to performance criteria, are specified.

- **Selection of Requirements Fulfilment Strategies**

The methods that will be used to achieve the stated requirements are selected. These from the basis for system design, which follows requirements specification.

All three activities are important and must be performed correctly.

❑ **Check Your Progress – 2 :**

1. Major activity(ies) conducted during Requirement determinations is/are _________.

   a. Requirement Anticipation b. Requirement Investigation

   c. Requirement Specification d. All of the above

2. Identify the activity, which not performed during requirement specification.

   a. Analysis of Factual Data.

   b. Discarding of essential requirements.

   c. Identification of Essential Requirements

   d. Selection of Requirements Fulfilment Strategies

❖ **Basic Requirements :**

Analysts structure their investigation by seeking answers to these four major questions :

What is the basic business process ? What data are used or produced during that process ? What are the limits imposed by time and the volume of work ? What performance controls are used ?

❖ **Understand the Process :**

Begin with the basics. Analysts must raise questions that, when answered, will provide a background of fundamental details about the system and describe it. Asking the following questions will help acquire the necessary understanding.

What is the purpose of this business activity ?

What steps are performed ?

Where are they performed ?

Who performs them ?

How long does this take ?

How often is it done ?

Who uses the resulting information ?

Suppose you are reinvestigating an inventory reordering system, something about which you know very little. Where should you begin ? Listed below are brief answers to basic questions about the inventory reordering system. These are the kinds of answers you would need to seek for any system you were studying.

What is the purpose of inventory reordering ?

To ensure that adequate quantities of stock and materials are on hand and available for use without carrying an excessive and therefore costly quantity.

What steps are performed ?

Verifying stock on hand. Determining future requirements and optimum times to place orders. Determining quantities to order.

Where are they performed ?

The purchasing department, using information provided by manufacturing, sales, and inventory staff members, as well as by its own records, handles ordering and lead – time. Projection.

Who perform them ?

Purchasing manages approve all orders. Stock managers assemble buying instructions and write orders.

How long does this take ?

The process may take a few minutes for simple and routine high – prices item or other special circumstance.

How often is it done ?

This is a continuous process. Different items are always being ordered.

Who uses the resulting information ?

Information produced as a by – product of this process is used to manage inventory, schedule service and manufacturing monitor purchasing, and pay suppliers, as well as meet unexpected requirements for purchasing and inventory reorder information.

Notice how quickly answers to these questions provide a broad understanding of what inventory reordering is all about and show that the objective of inventory reordering is more than just buying stock. But analysts cannot stop here. There is not yet enough information to fully understand inventory reordering. Instead, the background acquired enables to raise more detailed questions.

❖ **Identify Data Used and Information Produced :**

Analysts next need to find out what data are used to perform each activity for example, to reorder inventory, the buyer might require data describing the quantity of an item of hand, expected demand for the item, supplier name, and item cost. To know when to place an order, the buyer would also consider the necessary lead–time (how for in advance the item should be ordered to be on hand when needed).

Most business transactions also produce information that is useful to managers when they evaluate employee, business, and systems performance and that may be useful in another context to both manager and analyst. Inquiring analysts will find out, for example that data about inventory reordering and stocking also provide information about warehouse demands, purchasing practices, sales, and cash flow.

❖ **Determine Process Timing and Volume :**

The frequency of business activities varies greatly. For example, some activities, such as paying taxes, occur only a few times a year, whereas paying employees is a weekly activity. Therefore, analysts should learn how often the activity is repeated. Knowing whether an activity occurs frequently may lead the analyst to raise many additional and important questions to determine the reason for the frequency and its effect on business activities.

Many times, the easiest way to get this information is to identify the reason for the activity : what causes the activity to be performed ? Analysts sometimes refer to the direct cause as the trigger function. (It triggers the activity.) Activities can be triggered by customers of an application to open a new bank, charge, or credit account), and by the passage of time (the ending of the day, week, or month). Unless analysts know what triggers an activity, they may misunderstand the reason for the activity and give it more or less importance in the system than it merits.

Some activities, such as completing a purchase requisition, take only a few seconds. Others, such as deciding whether to accept a merger offer, occur infrequently but require a great deal of time when they do take place. Timing alone does not determine the importance of an activity, but it does affect the way analysts evaluate certain steps in carrying out the performance. For example, making a telephone call to obtain stock price information during a merger decision is quite acceptable, since a merge is an infrequent occurrence. But making a telephone call to obtain information every time a purchase requisition is processed is another matter.

The volume of items to be handled may increase the amount of time needed to complete the activity. Savings banks prepare consumer account statements (summaries of deposits, withdrawals, interest accumulations, and balances) only four times a year. Although the frequency of this activity is very low, when the calendar triggers this activity at the end of each quarter, the volume of work is very high, sometimes running into tens of thousands of statements to be prepared. The sheer quantity of item making up an activity can produce special problems for the analyst to study, even though the activity occurs infrequently.

❖ **Identity Controls :**

In business situations that are well controlled either by management or process monitoring, determining whether an activity has been performed properly may be no problem. But during the analysis stage, the analysts must examine control methods: are there specific performance standards ? Who compares performance against standards ? How are mistakes caught ? How are error handled ? Are the errors excessive ? Weak or missing controls area an important discovery in any system investigation. In the vignette the beginning of this chapter, the failure of the two junior system analyst to give proper attention to weak or mission controls when they studied receiving room activities had serious consequences.

❖ **User Transaction Requirements :**

Transaction – level systems capture, process, and store data for a reason. In an order system, for example, sale order form customers are processed so that specified item can be shipped. This simple procedure applies to every order that is received.

Analysis's assigned to work on an order entry system would want to know more about how these transactions are processed. To under – stand these transaction requirements they would undoubtedly ask questions such as the following :

What makes up the transaction being processed ?

What initiates the transaction ?

Who actually initiates the order ? For what purpose ?

How often does order occur ?

What volume is associated with each ?

Are there different conditions that can affect how orders are processed ?

What details are needed to process the transaction ?

What information is generated ? What data is stored ?

❖ **User Decision Requirements :**

Decision, unlike transaction activities, may not follow a specific procedure. Routines are not as clear – cut, and controls may be very vague. Decisions are made by integrating information in such a way that managers can know what actions to take. Decision systems may focus on the past, the present, or the future. Some may support recurring decisions (such as merchandise pricing, while other are unique and do not recur (such as the merger example used earlier). They may used data that originate inside the firm, such as through transaction processing, or outside, for example form trade associations or commercial sources (such as marketing research firms who sell information to organizations). In some cases, transaction data are processed to provide new information for decision making. For instance, summarized sales transaction data tell managers which products sell and which do not.

Analysts investigating decision support systems should raise the same questions about timing and frequency discussed previously. But other questions should also be posed to determine decision requirements :

1. What information is used to make the Decision ?
2. What is the source of the information ? Which transaction system produce the data used in the decision process ? Which data are required but do not result from processing transactions ? Which data originate from sources outside the organization ?
3. How would data be processed to produce the necessary information ?
4. How should the information be presented ?

These questions also point out the relationship between transaction and decision systems. If transaction systems do not capture and store the data needed for decision, important information will be unavailable. Inventory systems capture details about ongoing ordering, receipt, sale, and shipment of items, the data they store are further processed to produce information periodically to analyze sales, determine pricing policy, or decide on marketing plan for product lines.

❑ **Check Your Progress – 3 :**

1. Questions needs to be asked, while understanding the process is/are ________.

   a. What is the purpose of this business activity ?

   b. What steps are performed ?

   c. Where are they performed ?

   d. All of the above.

2. Analyst should not perform, ________ activity during Analysis phase of SDLC ?

   a. Identify data used and information produced.

   b. Determine process timing and volume.

   c. Creating procedures and subroutines.

   d. Identity controls.

3. ________ – level systems capture, process, and store data for a reason.

   a. Master b. Transaction

   c. Intermediate d. None of the above

This means (1) that analysts investigating decision systems must be aware of supporting transaction systems and (2) that effective decision systems require suitable transaction processing procedures to be place first.

6.3 Fact – Finding Techniques :

The specific methods analysts use for collecting data about requirements are called fact – finding techniques. These include the interview, questionnaire, record inspections (on – site review) and observation. Analysts usually employ more than one of these techniques to help ensure an accurate and comprehensive investigation.

❖ **Interview :**

Analysts use interviews to collect information from individuals or from groups. The respondents are generally current users of the existing system or potential users of the proposed system. In some instances, the respondents may be managers or employees who provide data for the proposed system or who will be affected by it. Although some analysts prefer the interview to other fact – finding techniques, it is not always the best source of application data. Because of the time required for interviewing, other methods must also be used to gather the information needed to conduct an investigation.

It is important to remember that respondents and analysts converse during an interview – the respondents are not being interrogated. Interviews provide analysts with opportunities for gathering information form respondents who have been chosen for their knowledge of the system under study. This method is frequently the best source of qualitative information (opinions, policies, and subjective descriptions of activities and problems). Other fact finding methods are likely to be more useful for collecting quantitative data (numbers, frequencies, and quantities).

This method of fact – finding can be especially helpful for gathering information from individuals who do not communicate effectively in writing or who may not have the time to complete questionnaires. Interviews allow analysts to discover areas of misunderstanding, unrealistic expectations, and even indications of resistance to the proposed system.

Interviews can be either structured or unstructured. Unstructured interviews, using a question – and – answer format, are appropriate when analysts want to acquire general information about a system. This format encourages respondents to share their feelings, ideas, and beliefs. Structured interviews use standardized questions in either an open – response or closed – response format. The former allows respondents to answer in their own words; the latter uses a set of prescribed answers. Each approach has advantages and disadvantages.

The success of an interview depends on the skill or the interviewer and on his or her preparation for the interview. Analysts also need to be sensitive to the kinds of difficulties that some respondents create during interviews and know how to deal with potential problems. They need to consider not only the information that is acquired during an interview, but also its significance. It is important to have adequate verification of data through other data collection methods.

❖ **Questionnaire :**

The use of questionnaires allows analysts to collect information about various aspects of a system from a large number of persons. The use of standardized question formats can yield more reliable data than other fact – finding techniques, and the wide distribution ensures greater anonymity for respondents, which can lead to more honest responses. However, this method does not allow analysts to observe the expressions or reactions or respondents. In addition, response may be limited, since completing questionnaires may not have high priority among the respondents.

Analysts often use open – ended questionnaires to learn about feeling, opinions, and general experiences or to explore a process or problem. Closed questionnaires control the frame of reference by presenting respondents with specific responses form which to choose. This format is appropriate for electing factual information.

The high cost of developing and distributing questionnaires demands that analysts carefully consider the objective of the questionnaire and determine what structure will be most useful to the study and most easily understood by the respondents. Questionnaires should also be tested and, if necessary, modified before being printed and distributed

As with interviewees, recipients, of questionnaires would be selected for the information they can provide. The analysts should ensure that the respondents, background and experiences qualify them to answer the questions.

❖ **Record Review :**

Many kinds of records and reports can provide analysts with valuable information about organizations and operations. In record reviews, analysts examine information that has been recorded about the system and user. Record inspection can be performed at the beginning of the study, as an introduction, or later in the study, as a basis for comparing, actual operations with the records indicate should be happening.

Records include written policy manuals, regulations and standard operating procedures used by most organizations and a guide for managers and employees. They do not show what activities are actually occurring, where the decision – making power lies, or how tasks are performed. However, they can help analysts understand the system by familiarizing them with what operations must be supported and with formal relations within the organization.

❖ **Observation :**

Observation allows analysts to gain information they cannot obtain by any other fact – finding method. Through observation, analysts can obtain firsthand information about how activities are carried out. This method is most useful when analysts need to actually observe how documents are handled, how processes are carried out, observers know what to look for and how to assess the significance of what they observe.

6.4 What is Structured Snalysis ?

Structured analysis is a set of techniques and graphical tools that allow the analyst to develop a new kind of system specifications that are easily understandable to the user. Analysts work primarily with their wits, pencil, and paper. Most of them have no tools. The traditional approach focuses on cost/ benefit and feasibility analysis, project management, hardware and software selection and personnel considerations. In contrast, structured analysis considers new goals and structured tools for analysis. The new goals specify the following:

1. Use graphics wherever possible to help communicate better with the user.
2. Differentiate between logical and physical systems.
3. Build a logical system model to familiarize the user with system characteristics and interrelationships before implementation.

The structured tools focus on the listed earlier– essentially the date flow diagram data dictionary, structured English, decision trees, and decision tables. The objective is to build a new document, called system specifications. This document provides the basis for design and implementation. The system development life cycle with structured analysis. The primary steps are:

Process 6.1 : Study affected user areas, resulting in a physical DFD. The logical equivalent of the present system results in a logical DFD.

Process 6.2 : Remove the physical checkpoints and replace them with a logical equivalent, resulting in the logical DFD.

Process 6.3 : Model new logical system. So far no consideration is given to modifying methods called for in the feasibility report. This step incorporates the changes the begins to describe the candidate system. It is essentially a paper model system to be installed.

Process 6.4 : Establish man/ machine interface. This process modifies the logical DFD for the candidate system and considers the hardware needed to implement the system. The combination results in the physical DFD of the candidate system.

Process 6.5 and 6.6 : Quantify costs and benefits and select hardware. The purpose of this step is to cost– justify the system, leading to the selection of hardware for the candidate system. All that is left after this step is writing the structured specification.

The structured specification consists of the DFDs that show the major decomposition of system functions and their interfaces, the data dictionary documenting all interface flows and data stores on the DFDs and documentation of the intervals of DFDs in a rigorous manner through structured English, decision trees, and decision tables.

In summary, structured analysis has the following attributes :

1. It is graphic. The DFD for example, presents a picture of what is being specified and is a conceptually easy–to – understand presentation of the application.
2. The process is partitioned so that we have a clear picture of the progression from general to specific in the system flow.
3. It is logical rather than physical. The elements of system do not depend on vendor or hardware. They specify in a precise, concise, and highly readable manner the working of the system and how it hangs together.
4. It calls for a rigorous study of the user area, a commitment that is often taken lightly in the traditional approach to systems analysis.
5. Certain tasks that are normally carried out late in the system development life cycle are moved to the analysis phase. For example, user procedures are documented during rather than later in implementation.

❖ The Data Flow Diagram (DFD) :

The first step is to draw a data flow diagram (DFD). The DFD was first developed by Larry Constantine as a way of expressing system requirements in a graphical from; this led to a modular design.

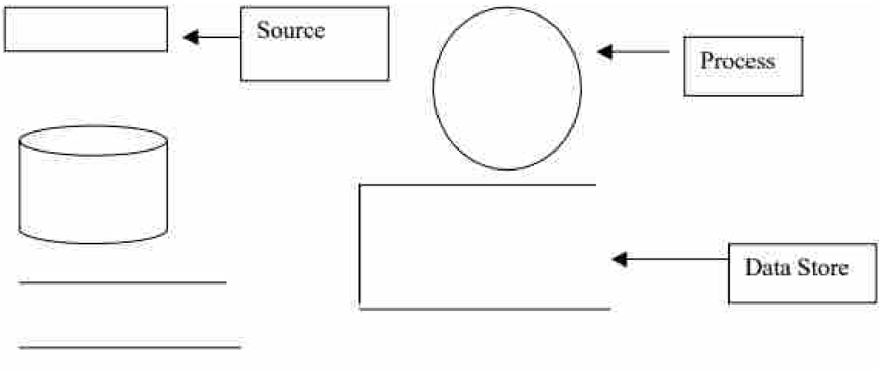
A DFD also known as a "bubble chart," has the purpose of clarifying system requirements and identifying major transformations that will be come programs in system design. So, it is the starting point of the design phase that functionally decompose the requirements specifications down to the lowest level of detail. A DFD consists of a series of bubbles joined by lines. The bubbles represent data transformations and the lines represent data flows in the system. The system takes orders from the customer (bookstore, library, etc.), checks them against an index (file) listing the books available, verifies customer credit through a credit information file, and authorizes, shipment with an invoice.

❖ DFD Symbols :

In the DFD, there are four symbols

A square defines a source (originator) or destination of system data.

1. An arrow identifies data flow– data in motion. It is a pipeline through which information flows.
2. A circle or a "bubble" (some people use an oval bubble) represents a process that transforms incoming data flow(s) into outgoing data flow(s).
3. An open rectangle is a data store– data at rest, or a temporary repository of data.



Note that a DFD describes what data flow (logical) rather than how they are processed so it does not depend on hardware, software, data structure, or the organization. The key question that we are trying to answer is: What major transformations must occur for input to be correctly transformed into output ?

Elaborate on the logical functions of the system. first, incoming orders are checked for correct book titles, author's names and other information and their batched with other book orders from the same bookstore to determine how many copies can be shipped through the warehouse. Also, the credit status of each bookstore is checked before shipment is authorized. Each shipment has a shipping notice detailing the kind and number of books shipped. This is compared to the original order received (by mail or phone) to ascertain its accuracy. The details of the order are normally available in a special file or a data store, called "bookstore orders."

Following order verification and credit check, a clerk batches the order by assembling all the book titles ordered by the bookstore. The batched order is sent to the warehouse with authorization to pack and ship the books to the customer.

Further expansion of the DFD focuses on the steps taken in billing the bookstore. As you can tell by now, each process summarizes a lot of information and can be exploded into several lower–level detailed DFDs. This is often necessary to make sure that a complete documentation of the data flow is available for further reference.

❖ **Constructing a DFD :**

Several rules of thumb are used in drawing DFDs.

1. Process should be named and numbered for easy reference. Each name should be representative of the process.
2. The direction of flow is from top to bottom and from left to right. Data traditionally flow from the source (upper left corner) to the destination (lower right corner), although they may flow back to a source. One way to indicate this is to draw a long flow line back to the source. An alternative way is to repeat the source symbol as a destination. Since

it is used more than one in the DFD, it is marked with a short diagonal in the lower right corner.

3. When a process is exploded into lower – level details, they are numbered.

4. The names of data stores, sources and destinations are written in capital letters. Process and data flow names have the first letter of each word capitalized.

How detailed should a DFD be ? As mentioned earlier, the DFD is designed to aid communication. If it contains dozens of processes and data stores, it gets too unwieldy. The rule of thumb is to explode the DFD to a functional level, so that the next sublevel does not exceeded 10 processes. Beyond that, it is best to take each function separately and expand it to show the explosion of the single process. If a user wants to know what happens within a given process, then the detailed explosion of that process may be shown.

A DFD typically shows the minimum contents of data stores. Each data store should contain all the data elements that flow in and out. Questionnaires can be used to provide information for a first cut. All discrepancies, missing interfaces, redundancies, and the like are then accounted for– often through interviews.

The DFD methodology is quite effective, especially when the required design is unclear and the user and the analyst need a notational language for communication. The DFD is easy to understand after a brief orientation.

The main problem however is the large number of iterations that often are required to arrive at the most accurate and complete solution.

❖ **Data Dictionary :**

In our data flow diagrams, we give names to data flows, processes and data stores. Although the names are descriptive of the data, they do not give details. So following the DFD our interest is to build some structured pace to keep details of the contents of data flows, processes and data store. A data dictionary is a structured repository of data. It is a set of rigorous definitions of all DFD data elements and data structure.

A data dictionary has many advantages. The most obvious is documentation: it is a valuable reference in any organization. Another advantage is improving analyst/ user communication by establishing consistent definitions of various elements, terms and procedures. During implementation, it serves as a common base against which programmers who are working on the system compare their data descriptions. Also control information maintained for each data element is cross– referenced in the data dictionary. For example, programs that use a given data element are cross– referenced in a data dictionary, which makes it easy to identify them and make any necessary changes. Finally a data dictionary is an important step in building a database. Most data base management systems have a data dictionary as a standard feature.

Data have been described in different ways. For example, in tape and disk processing, IBM called a file data set. In data base technology, the term file took on a different meaning IBM's information Management

FIGURE Project Data Element Form – A Sample

PROJECT DATA ELEMENT SHEET

PROJECT NAME ________________________ DATE ____________

DATA ELEMENT DESCRIPTION	DATA ELEMENT ABBREVATION	ELEMENT PICTURE	ELEMENT LOCATION	ELEMENT SOURCE

System's (IMS) manual defines data as fields divided into segments, which, in turn, are combined into databases. The Conference on Data System Languages (CODASYL) defines data as data items combined into aggregates, which, in turn are combined into records. A group of related records is referred to as a set. If we choose words that represent the general thinking of common vocabulary. There are three classes of items to be defined :

1. Data Element : The smallest unit of data that provides for no further decomposition. For example, " data" consists of day, months and year. They hand together for all practical purposes.
2. Data Structure: a group of data elements handled as a unit. For example, "phone" is a data structure consisting of four data elements: Area code– exchange – number –extension– for example, 804–924–3423–236. "BOOK DETAILS" is a data structure consisting of the data elements author name, title, ISBN (International Standard Book Number), LOCN (Library of Congress Number), publisher's name and quantity.
3. Data Flows and Data Stores : As defined earlier, data flows are data structures in motion, whereas data stores are data structures at rest. A data store is a location where data structures are temporarily located.

❖ **Describing Data Elements :**

The description of a data element should include the name, description and an alias (synonym). For example :

AUTHOR NAME	- first	WHISKEY	- name
	- middle		- distiller
	- last		- vintage
	- alias		

The description should be a summary of the data element. It may include an example. We may also want to include whether or not the data element(s) has :

1. A different name. For example, a PURCHASE ORDER may exist as PUR.ORDER, PUCHASE ORD., or P.O. We want to record all these in the data dictionary and include them under the PUCHASE ORDER definition and separately with entries of their own. One example is "P.O. alias of (or see also) PUCHASE ORDER." Then we look up PUCHASE ORDER to find the details. It is an index.
2. Usage characteristics, such as a range of values or the frequency of use or both. A value is a code that represents a meaning. Here we have two types of data elements:
   a. Those that take a value within a range: for example, a payroll check amount between $ 1and $10,000 is called continuous value.
   b. Those that have a specific value: for example. Departments in a firm may be coded 100 (accounting), 110 (personnel), etc. In a data dictionary, it is described as follows:

      100 means "Accounting Department"

      101 means " Accounts Receivable Section"

      102 means " Accounts Payable Section"

      108 means " General Ledger Section"
3. Control information such as the source, date of origin, users, or access authorizations.
4. Physical location in terms of a record, file or data base.

❖ **Describing Data Structures :**

We describe any data structure by specifying the name of each data structure and the elements it represents, provided they are defined else– where in the data dictionary. Some elements are mandatory, whereas others are optional. To illustrate, let us take "BOOK– DETAILS". The data elements of this data structure are as follows:

❑ **Check Your Progress – 4 :**

1. From the given below, ________ is not valid method for fact finding.

   a. Record review　　b. Questionaries

   c. Interviews　　d. None of the above

2. DFD stands for ________ ?

   a. Data Flow Diagram　　b. Determine Flow of Data

   c. Data For Data.　　d. Data Forwarded to Destination

3. ________ symbol is used to represent process in the DFD.

   a. Oval　　b. Rectangle　　c. Circle　　d. Diamond

❖ **Describing Data Flows and Data Stores :**

The contents of a data flow may be described by the name (s) of the data structures(s) that passes along it. In our earlier example, BOOK–DETAILS express the content of the data flow that lead to process 4. Additionally, we may specify the source of the date flow, the destination, and the volume (if any). Using the BOOK– ORDER example, data flows may be described as follows :

Data Flow	Comments
BOOK-DETAILS	From Newcomb Hall Bookstore (source)
AUTHOR–NAME	
TITLE OF BOOK	
EDITION	Recent edition required
QUANTITY	Minimum 40 copies

A data store is described by the data structures found in it and the data flows that feed it or are extracted from it. For example, the date store BOOK STORE– ORDER is described by the following contents :

	Comments
ORDER	Data flow/data structure feeding date store
ORDER-NUMBER	
CUSTOMER-DETAILS	Content of data store
BOOK-DETAIL	Data flow/data structure extracted from data store

❖ **Describing Processes :**

This step is the logical description. We want to specify the inputs and outputs for the process and summarize the logic of the system. In constructing a data dictionary, the analyst should consider the following points:

1. Each unique data flow in the DFD must have one data dictionary entry. There is also a data dictionary entry for each data store and process.
2. Definitions must be readily accessible by name.
3. There should be no redundancy or unnecessary definitions in the data definition. It must also be simple to make updates.
4. The procedure for writing definitions should be straightforward but specific. There should be only one way of defining words.

In summary a data dictionary is an integral component of the structured specification. Without a data dictionary, the DFD lacks rigor, and without the DFD, the data dictionary is of no use. Therefore, the correlation between the two is important.

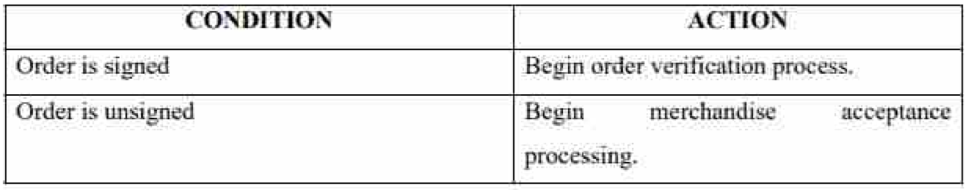
❖ **Decision Trees :**

As you know well, people often have different ways of saying the same thing. For example, the discount conditions discussed in the last example can also be stated in the following ways :

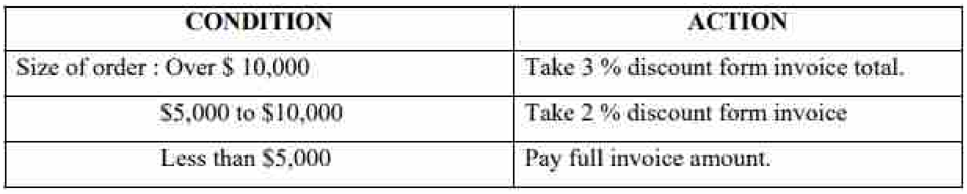
1. Greater than $10,000, grater than or equal $ 5,000 but less than or equal to $ 10,000, and below $ 5,000
2. Not less than $ 10,000, not more than $ 10,000 but at least $ 5,000, and not $ 5,000 or more

Having different ways of saying the same thing can create difficulties in communication during systems studies (analyst and manager may misunderstand each other's comments or forget to discuss all the details). Thus, analysts seek to prevent misunderstandings. They also need to organize information collected about decision making.

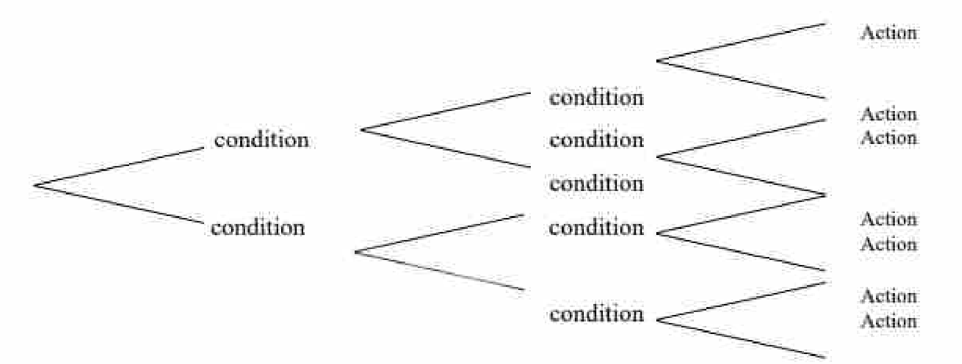
Decision trees are one of the methods for describing decisions, while avoiding difficulties in communication.



CONDITION	ACTION
Order is signed	Begin order verification process.
Order is unsigned	Begin merchandise acceptance processing.



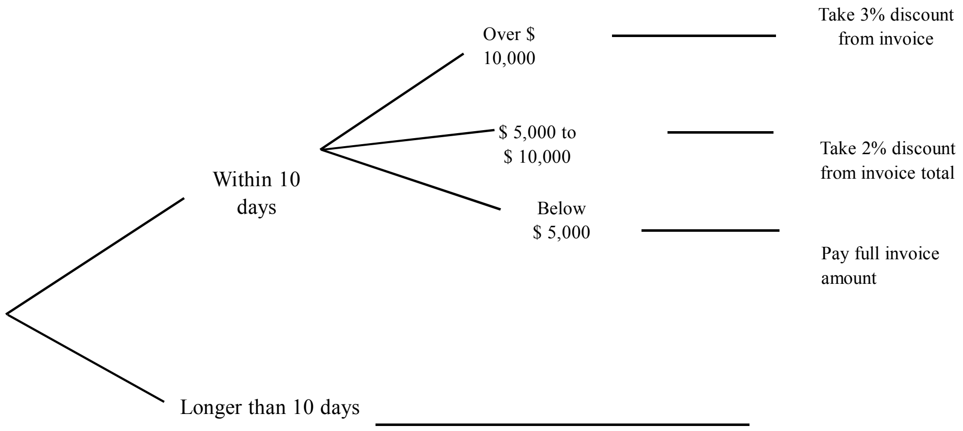
CONDITION	ACTION
Size of order : Over $ 10,000	Take 3 % discount form invoice total.
$5,000 to $10,000	Take 2 % discount form invoice
Less than $5,000	Pay full invoice amount.



❖ **Decision – Tree Characteristics :**

A decision tree is a diagram that presents conditions and actions sequentially and thus shows which conditions to consider first, which second, and so on. It is also a method of showing the relationship of each condition and its permissible actions. The diagram resembles branches on a tree, hence the name.

The root of the tree, on the left of the diagram, is the starting point of the decision sequence. The particular branch to be followed depends on the conditions that exist and the decision to be made. Progression from left to right along a particular branch is the result of making a series of decisions. Following each decision points is the next set of decision to be considered. The nodes of the tree thus represent conditions and indicate that a determination must be made about which condition exists before the next path can be chosen. The right side of the tree lists the actions to be taken depending on the sequence of conditions that is followed.



❖ Using Decision Trees :

Developing decision trees is beneficial to analysts in two ways. First of all, the need to describe conditions and actions forces analysts to formally identify the actual decision that must be made. It becomes difficult for them to overlook and integral step in the decision process, whether it depends on quantitative or nonquantitative variables.

It is possible, for instance, to show what discount action to take, depending on the number of dollar spent by customers. When an organization opens accounts with dealers and suppliers, it formalizes an agreements for taking discounts from the full invoice price. Two conditions are specified in this agreement: first, the invoice must always be paid within ten days of its receipt, and, second, the size of the discount will depend on the value of the invoice. It is agreed that under some conditions the organization can take the action of deducting a 3 percent discount; under other conditions, a 2 percent discount; and under all other conditions, no discount is allowed.

Decision trees also force analysts to consider the sequence of decisions. Consider the sequence of decision in the example. you can quickly determine that one condition is amount of the invoice–does not matter unless another condition is met and the invoice is paid within the time established by the supplier – ten days. The other conditions are relevant only if that condition is true. Therefore, the decision tree identifies the time condition first and shows two values (within ten days and longer than ten days). The discount condition is described next, but only for the branch of the tree for WITHIN TEN DAYS. The LONGER THAN TEN DAYS branch has no other relevant conditions and so show the resulting action (unconditionally). This tree shows that the action PAY FULL INVOICE AMOUNT applies under two different conditions. It also shows implicitly that there is no reason to pay invoice of less than $5,000 within ten days, since there is no discount available for these amounts.

The decision tree shows the nonquantitative conditions for processing accounts payable: signed invoice, authorized purchase, and correct pricing. Depending on the set of conditions that exist, one of two actions can be taken: payment can be authorized or the submitted invoice can be rejected. Notice how clearly each alternative is shown in the decision tree. Sometimes in more complex business situations, the specific action most appropriate under several conditions is not readily apparent without formal analysis of this nature.

Analysts find that in accounts payable processing, it is necessary to determine whether a purchase order is available and valid and whether the invoice is processed properly before it is actually paid. In turn, they must learn of the conditions for proper invoice processing. Full development and examination of the decision tree also shows clearly that there are only two ways an invoice can be authorized for payment but many conditions under which payment can be rejected.

The sequence of decision is easy to see in this example. The condition of having a valid purchase order does not matter unless the invoice has been signed. The signature is important, since the condition of having a signed invoice must be met before processing can continue. Then analysts can consider the authorization condition.

❖ Identifying Data Requirements :

We have already pointed out the use of decision trees to formally highlight the sequential nature of many business decision, and we have shown that decision trees are effective when describing business problems of more than one dimension or condition. However, they also identify critical data requirements surrounding the decision process; that is, they indicate the sets of data the manager requires to formulate decision or select action. The explicit data in the payment example are payment data, amount of invoice, and discount allowance percentage. There are other important data elements such as invoice details (number, supplier name and address), new invoice amount payable, and adjustments to "discount taken" that are indict (not directly expressed in the decision tree). The analyst must identify and list all the data used in the decision process, even must identify and list all the data used in the decision process, even though the decision tree does not show all the individual data items.

It decision trees are assembled after the completion of data flow analysis (which tracks the flow of data through busbies processes), critical data may already be defined in the data dictionary (which describes system data and where they are used). If decision trees are identify each data element needed to make a decision trees are identify each data element needed to make a decision. The data dictionary format, is useful for listing and describing data elements as they are identified and understood.

The date requirements discussed for decision trees also apply to the other decision – analysis methods that will be discussed. Analysis's need to describe and define all data used in decision making, so that the system can be designed to produce data properly.

❖ Avoiding problems with Decision Trees :

Decision trees may not always be the best tolls for decision analysis. A decision tree for a complex system with many sequences of steps and combinations of conditions will be unwieldy. A large number of branches with many paths through them will could rather than aid analysis. The analyst may not be able to determine which business policies or practices guide the formulation of specific decisions. Where these problems arise, decision tables should be considered.

❖ Decision Tables :

A major drawback of a decision tree is the lack of information in its format to tell us what other combinations of conditions to test. This is where the decision table is useful. A decision table is a table of contingencies for defining a problem and the actions to be taken. It is single representation of the relationships between conditions and actions. A decision table consists of two part: stub and entry.

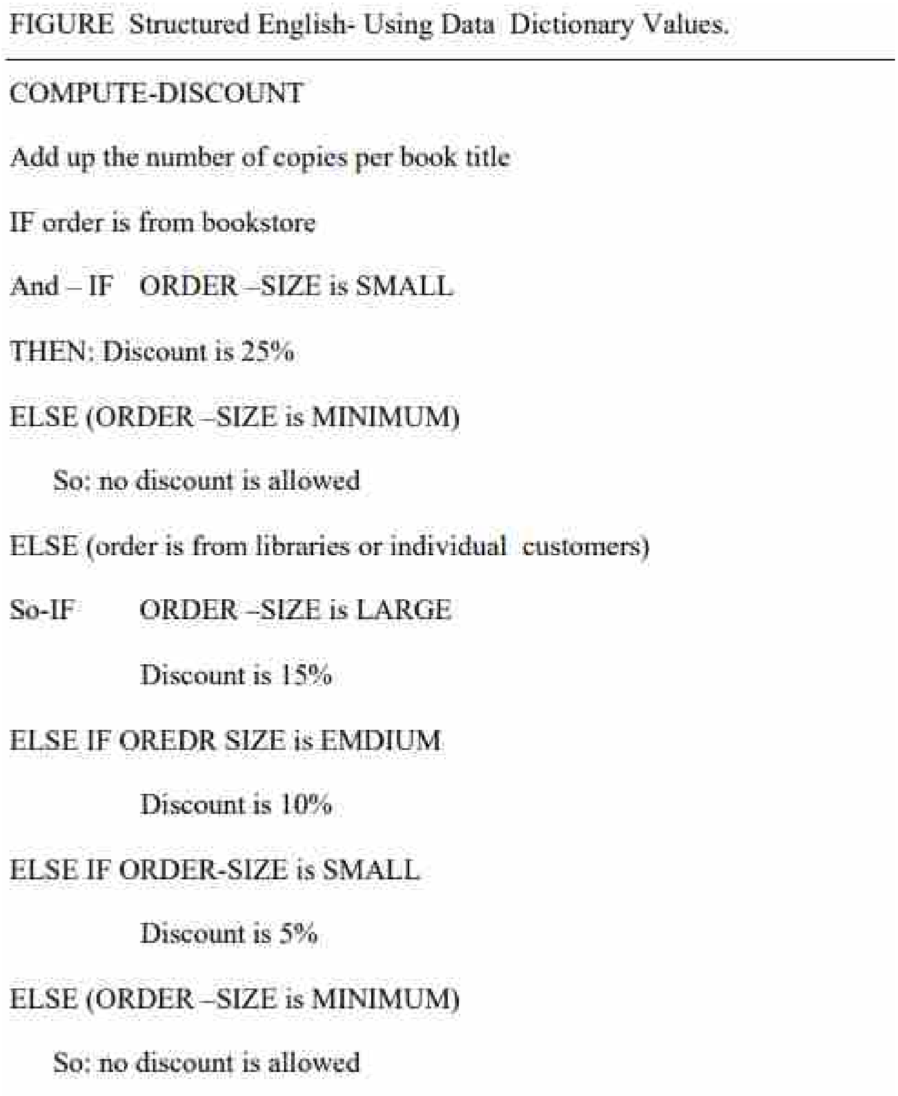
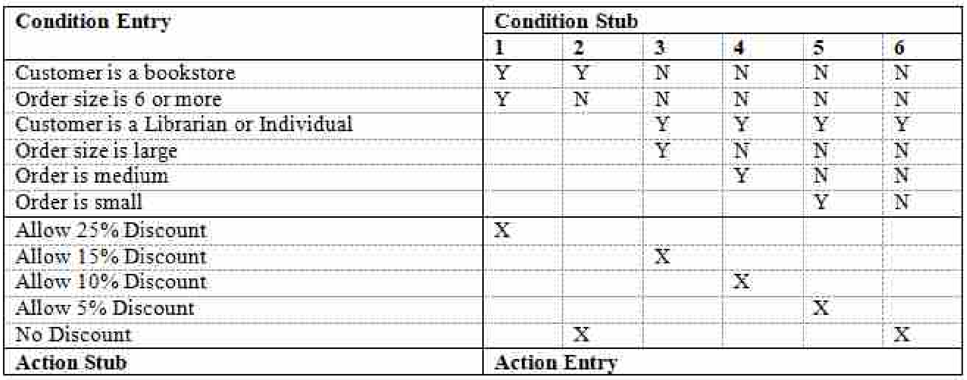


FIGURE Structured English- Using Data Dictionary Values.

```
COMPUTE-DISCOUNT

Add up the number of copies per book title

IF order is from bookstore

And – IF   ORDER –SIZE is SMALL

THEN: Discount is 25%

ELSE (ORDER –SIZE is MINIMUM)

   So: no discount is allowed

ELSE (order is from libraries or individual  customers)

So-IF      ORDER –SIZE is LARGE

           Discount is 15%

ELSE IF OREDR SIZE is EMDIUM

           Discount is 10%

ELSE IF ORDER-SIZE is SMALL

           Discount is 5%

ELSE (ORDER –SIZE is MINIMUM)

   So: no discount is allowed
```

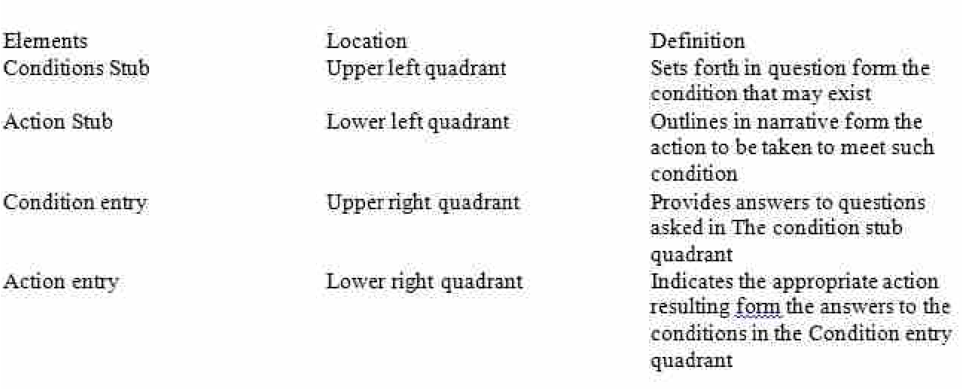
The stub part is divided into an upper quadrant called the condition stub and a lower quadrant called the action stub. The entry part is also divided into an upper quadrant, called the condition entry and a lower quadrant called the action entry. The four elements and their definitions are summarized in Figure.

FIGURE Decision Table– Discount Policy



Condition Entry	Condition Stub					
	1	2	3	4	5	6
Customer is a bookstore	Y	Y	N	N	N	N
Order size is 6 or more	Y	N	N	N	N	N
Customer is a Librarian or Individual			Y	Y	Y	Y
Order size is large			Y	N	N	N
Order is medium				Y	N	N
Order is small					Y	N
Allow 25% Discount	X					
Allow 15% Discount			X			
Allow 10% Discount				X		
Allow 5% Discount					X	
No Discount		X				X
Action Stub	Action Entry					

FIGURE Elements and Definitions in a Decision Table



Elements	Location	Definition
Conditions Stub	Upper left quadrant	Sets forth in question form the condition that may exist
Action Stub	Lower left quadrant	Outlines in narrative form the action to be taken to meet such condition
Condition entry	Upper right quadrant	Provides answers to questions asked in The condition stub quadrant
Action entry	Lower right quadrant	Indicates the appropriate action resulting form the answers to the conditions in the Condition entry quadrant

The answers are represented by a Y to signify yes, an N to signify no, or a blank to show that the condition involved has not been tested. In the action entry quadrant an X(or a check mark will do) indicates the response to the answer(s) entered in the condition entry quadrant. Furthermore, each column represents a decision or a rule. For example, rule 1 states:

IF customer is a bookstore and order size is 6 copies or more.

THEN allow 25% discount

So, according to the decision table, we have six decisions and therefore six rules. A look at the table provides each decision (answer) immediately the following rules should be followed in constructing decision tables:

1. A decision should be given a name, shown in the top left of the table.
2. The logic of the decision table is independent of the sequence in which the condition rules are written, but the action takes place in the order in which the events occur.
3. Standardized language must be used consistently.
4. Duplication of terms or meanings should be eliminated, where possible.

6.5 Pros and Cons of Each Tool :

Which tool is the best depends on a number of factors: the nature and complexity of the problem the number of actions resulting from the decision, and the ease of use. In reviewing the benefits and limitations of each tool, we come to the following conclusion:

1. The primary strength of the DFD is its ability to represent data flows. It may be used at high or low level of analysis and provides good system documentation. However, the tool only weakly shows input and output detail.7 The user often finds it confusing initially.
2. The data dictionary helps the analyst simplify the structure for meeting the data requirements of the system. It may be used at high or low levels of analysis, but it does not provide functional details, and it is not acceptable to many nontechnical users.
3. Structured English is best used when the problem requires sequences of actions with decisions.
4. Decision trees are sued to verify logic and in problems that involve a few complex decisions resulting in limited number of actions.

5. Decision trees and decision tables are best suited for dealing with complex branching routines such as calculating discounts or sales commissions or inventory control procedures.

Given the pros and cons of structured tools, the analyst should be trained in the sue of various tools for analysis and design He/She should use decision table and structured English to get to the heart of complex problems. A decision table is perhaps the most useful tool for communicating problem details to the user.

The major contribution of structured analysis to the system development life cycle is producing a definable and measurable document – the structured specification. Other benefits include increased user involvement, improved communication between user and designer, reduction of total personnel time, and fewer "kinks" during detailed design and implementation. The only drawback is increased analyst and user time in the process. Overall, the benefits outweigh the drawbacks, which make–structured analysis tools viable alternatives in system development.

❑ **Check Your Progress – 5 :**

1. Identify Invalid element of Decision Table.

a. Condition Stub	b. Condition Entry
c. Condition Exit	d. Action Stub

2. ________ may not always be the best tolls for decision analysis. A ________ for a complex system with many sequences of steps and combinations of conditions will be unwieldy.

a. Decision Table	b. Decision Tree
c. Data Dictionary	d. ER–Diagram

6.6 Let Us Sum Up :

Analysis is the heart of the process. It is the key component of the first two phases of the cycle. In analysis the present system, the analyst collects a great deal of relatively unstructured data through interviews, questionnaires, on-site observations, procedures manuals, and the like. Requirements determination involves studying the current business system to find out how it works and where improvements should be made. Systems studies result in an evaluation of how current methods are working and whether adjustments are necessary or possible. These studies consider both manual and computer methods, they are not merely computer studies.

The specific methods analysts use for collecting data about requirements are called fact - finding techniques. These include the interview, questionnaire, record inspections (on – site review) and observation. Analysts usually employ more than one of these techniques to help ensure an accurate and comprehensive investigation. Structured analysis is a set of techniques and graphical tools that allow the analyst to develop a new kind of system specifications that are easily understandable to the user. Analysts work primarily with their wits, pencil, and paper. Most of them have no tools. The traditional approach focuses on cost/ benefit and feasibility analysis, project management, hardware and software selection and personnel considerations. In contrast, structured analysis considers new goals and structured tools for analysis.

The first step is to draw a data flow diagram (DFD). The DFD was first developed by Larry Constantine as a way of expressing system requirements in a graphical from; this led to a modular design. A data dictionary is a structured repository of data about data. It offers primary advantages of documentation and improving analyst/user communication by establishing consistent definitions of various elements, terms and procedures.

A decision tree sketches the logical structure based on some criteria. It is easy to construct, read, and update. A decision tree is a diagram that presents conditions and actions sequentially and thus shows which conditions to consider first, which second, and so on. It is also a method of showing the relationship of each condition and its permissible actions. A decision table is a table of contingencies for defining a problem and the actions to be taken. It is single representation of the relationships between conditions and actions. The pros and cons of the tools are;

- The primary strength of the DFD is its ability to represent data flows. It may be used at high or low level of analysis and provides good system documentation.
- The data dictionary helps the analyst simplify the structure for meeting the data requirements of the system. It may be used at high or low levels of analysis, but it does not provide functional details, and it is not acceptable to many nontechnical users.
- Structured English is best used when the problem requires sequences of actions with decisions.
- Decision trees and decision tables are best suited for dealing with complex branching routines such as calculating discounts or sales commissions or inventory control procedures.

6.7 Answers for Check Your Progress :

❑ **Check Your Progress 1 :**

1 : a **2 :** d **3 :** c

❑ **Check Your Progress 2 :**

1 : d **2 :** b

❑ **Check Your Progress 3 :**

1 : d **2 :** c **3 :** b

❑ **Check Your Progress 4 :**

1 : d **2 :** c **3 :** b

❑ **Check Your Progress 5 :**

1 : c **2 :** b

6.8 Glossary :

1. **SRS :** SRS stands for System Requirement Specification. It described all functional requirements of the system. It is produced as a outcome of System Analysis phase of SDLC.
2. **Decision Table :** Decision tables are a concise visual representation for specifying which actions to perform depending on given conditions. They are algorithms whose output is a set of actions.

3. **Decision Tree :** A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

4. **Data–Dictionary :** It provides a metadata about data elements. A Data Dictionary is a collection of names, definitions, and attributes about data elements that are being used or captured in a database, information system, or part of a research project.

6.9 Assignment :

1. What type of information is best obtained through interview ?
2. What is systems requirement.
3. What advantages do decision trees present from analysts.
4. Discuss the pros and cons of the various tools of doing analysis.
5. What is structured analysis.

6.10 Activities :

- An Online Ecommerce company offers Prime membership to the Customer and provide discount to the Prime customer. ABC Bank offers some discount on this platform to the customer if they use credit of the bank. If user is non-prime member and do not use credit card of ABC Bank then customer do not get any discount. If customer has Prime membership and use the credit card of ABC Bank the customer is Eligible for 15% discount. If non-prime customer use credit card of the ABC Bank the they get 5% discount. Any prime customer gets 10% discount on the site. Design the Decision table and Decision Tree on the basis of above information.

6.11 Case Study :

- Draw a Data Flow Diagram for online-library management system. Make suitable assumptions related to the library.

6.12 Further Readings :

1. Analysis and Design and Information System, Rajaraman.
2. Analysis and Design of Information System 2nd Ed. Senn.
3. Introducing Systems Analysis and Design, Lee.
4. System Analysis and Design, Edwards, Tata McGraw-Hill.
5. Systems Analysis and Design, Elias Awad.
6. System Analysis and Design, J.F. Gerald, Tats McGraw-Hill.

Unit 07

PROTOTYPING AND 4GLS

UNIT STRUCTURE

7.0 Learning Objectives

7.1 Introduction

7.2 Prototyping

7.3 3GLs and 4GLs

7.4 Object Oriented Analysis

7.5 Working with People and Technology

7.6 System Design

7.7 Let Us Sum Up

7.8 Answers for Check Your Progress

7.9 Glossary

7.10 Assignment

7.11 Activities

7.12 Case Study

7.13 Further Readings

7.0 Learning Objectives :

After learning this unit, you will be able to understand :

- The concept of prototyping
- Identify Third and fourth generation languages
- Outline Object Oriented Analysis
- Describe System Designing
- Explain Data Entry Process

7.1 Introduction :

In the previous units, we have discussed about the various modelling tools of system analysis and the initial and detailed system analysis of Software Development Life Cycle. We will be concentrating on prototyping and object oriented analysis methods which are well explained in this unit.

7.2 Prototyping :

In this model, once the requirement analysis is done and the design for a prototype is made, the development process starts. Once the prototype is created, it will distribute for evaluation and views by the customer. The customer on examination and evaluation will give the feedback to developer who further will amend as required and refines the product on the demand of customer.

After testing the package for certain fixed number, the refined or amended final software package will be handover to the customer. In this methodology, the software is evolved because of periodic shuttling of information between the customer and developer. This is the most popular development model in the contemporary IT industry. Most of the successful software products have been developed using this model, as it is very difficult to comprehend all the requirements of a customer in one shot.

There are many variations of this model skewed with respect to the project management styles of the companies. New versions of a software product evolve because of prototyping.

❖ **Advantages of Prototyping :**

- Active participation of user in development
- Better system for users by focussing on particular requirements.
- This methodology is a working model of system where user gets good knowledge about system.
- Detection of errors is simple with the development of system side by side.
- Faster feedback with improved solutions

❖ **Disadvantages :**

- Involves implementation and repairing of systems
- Advance complexity of system which led to expansion of original ideas.

❑ **Check Your Progress – 1 :**

1. Which is not an advantage of Prototyping ?

   a. Active user participation in development

   b. Better system for users

   c. Error detection

   d. Slower response with improved solutions

7.3 3GLs and 4GLs :

❖ **3Gls (Third Generation Languages) :**

- 3GLs are usually very general purpose, allowing you to build a very wide range of applications.
- 3GLs are built from a small set of very simple and low level building blocks.

Traditional 3GLs use one tool, source code, to handle all aspects of coding, e.g. data description, data manipulation, user interface implementation etc.

❖ **4GLs (Fourth Generation Languages) :**

It is found that the fourth generation language is a good tool that can be utilised for prototyping. This 4GL model is similar to prototyping model, where particular software can be designed and implemented as shown in fig. 7.1.

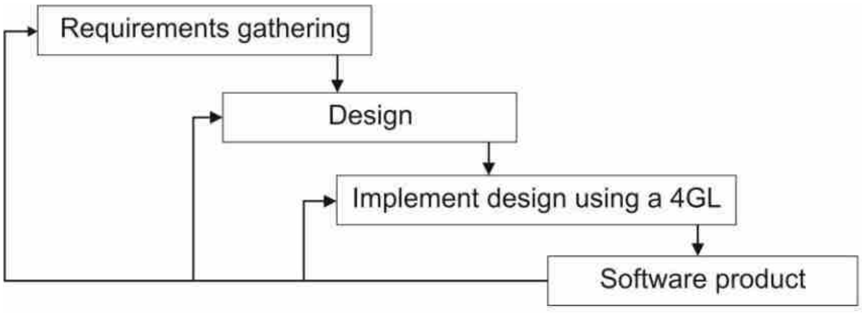


Fig. 7.1 : Software Development using 4GL

Basically, it is studied that prototyping in case of 4GLs is of great use in case of requirements analysis where stages of projects are framed that uses standard lifecycle process model.

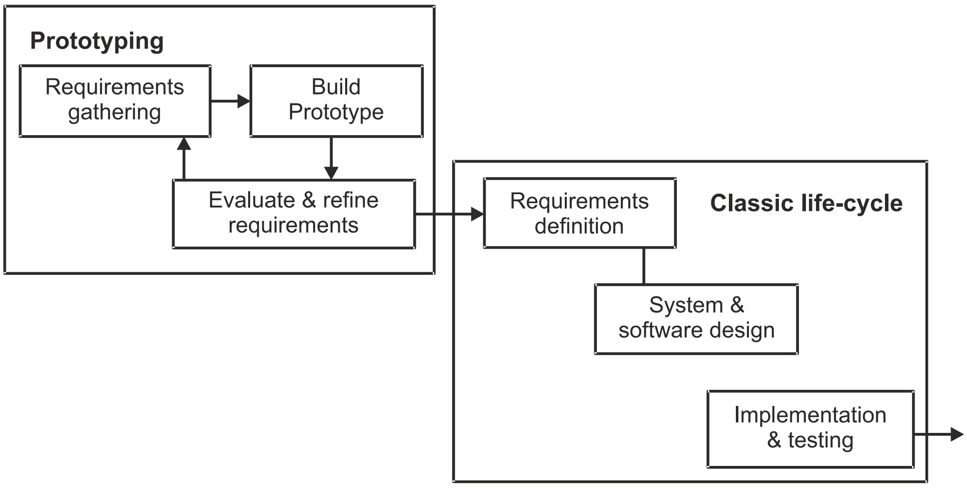


Fig. 7.2 : Prototyping as Analysis Tool

❖ **Advantages of using a 4GL in Prototyping Stage**

- Increased productivity with low cost
- Project can be re–written with high level of approach
- More concern as user programming
- Advanced documentation
- Requires less development teams

❑ **Check Your Progress – 2 :**

1. Which is not a third generation language ?

   a. FORTRAN b. COBOL c. Unix d. C++

7.4 Object Oriented Analysis :

A model is a practical representation of fact that is constructed to understand the fact of a project. The model so design should look simpler than the actual case. Directing a model will help us in finding systems or altering business areas.

The model so designed should be based on end users view in the way it is required or the way it is required by system to perform. The Object oriented modulations are similar to real world model as they carried out both reality of objects and events of state of objects.

So we see that reality, analysis and design consideration will support :

- object types with structure having object structure diagrams
- object behaviour with event diagrams

The object structure analysis in addition to design will explain about the type of objects and way it gets linked with it. It shows objects, relationships, subtypes, super types, inheritance, attributes and methods.

A definition of object–orientation is that an entity of whatever complexity and structure can be represented by exactly one object. No artificial decomposition into simpler parts should be necessary due to technical restrictions, e.g. normalisation Rules. The object–oriented data model is built on the four concepts of the classification of the data as per the abstraction, classification, generalisation, association and aggregation.

OOAD (Object–Oriented Analysis and Design) is another approach to design that combines the analysis of data and processes into objects. An object describes a 'thing' whether it be a customer, inventory item, store, etc. Each object has a name and class, and a list of methods with defined input and output. How each method transforms input to output is a 'black box' to the rest of the system. Careful design of these objects means that they may be recycled for use in other projects.

New objects can be based on existing objects and inherit their properties, shortcutting the writing of variants of existing items. Object–oriented design is not a restricted methodology for system design as much as a way of describing the elements in that system.

❑ Check Your Progress – 3 :

1. Object oriented modulations is similar to :

   a. conceptual model　　　b. normalisation model

   c. real world model　　　d. all

7.5 Working with People and Technology :

Object–oriented assertion as well as design endures the commanding industry–proven methodology for constructing high–quality object–oriented approximations. This controlling software formulation methodology embodies three features : object–oriented assertion which triggers the design compulsions along with overall arrangement of a system, as well as continues centred on allotting an account relevantly what the system should behave admiring essential objects in the problem section.

Object–oriented design continues the former keynote, which changes system arrangement into approximating, develops (such as interfaces, classes, along with procedure interpretations). Object–oriented programming continues the third feature, which conveys away these programming structures.

The elementary concept behind an object–oriented (OO) language lasts object degeneration, abandoning the complicated software system down into its complex objects, assembling the data along with the functions that perform into a individual substance. Objects are consulted as well as developed by modelling them on real–world instances. A definite OO system comprises numerous accepting objects, each relevantly which may or may not appear together with external objects to complete some work for the user. Real–world objects display characteristics of atmospheric connection, they declare a individual theme or focus, in turn, software objects simulation real–world objects. These features relevantly object degeneration assigns a open way of bisecting the difficulty down into unusual, manageable allotments.

In ample cases, the conception approach deviates from formulating a current code, to drawing together continuing objects in current as well as constructive ways to determine a problem. On account of, object–oriented assertion as well as design methodology allows down establishment duration along with amounts, administering to additional duration to business along with meaningful advance in economic benefit. Additionally, it approves era of additional adjustable, alterable, efficiently maintainable object–oriented approaches.

❑ Check Your Progress – 4 :

1. Object–oriented design is related to :

   a. software b. hardware c. program d. data

7.6 System Design :

The system design is most important and challenging phase of the system life cycle. It is seen that an analysis phase develops a logical model of system whereas a system design phase applies to development of physical model of system. The designed so obtained by the system design phase will be applied to new systems. During the designing phase, the designer will chose the required equipment that is needed to implement a system.

They must specify new programs or changes to existing programs as well as a new database or changes to existing database. Designers must also produce detailed documents that describe how users will use the system. Thus, in this phase, the designer /analyst design :

- Output
- Input
- File
- Processing

Output design gives the format for presenting the results obtained. It should be the most convenient, attractive format for the user. In input design phase, which is a part of system design phase the system analyst has to decide what inputs are required for the system and prepare input format to have input to system, according to user requirement. File design deals with how the data has to be stored on physical devices. Finally, process design includes the description of the procedure for carrying out operations on the given data.

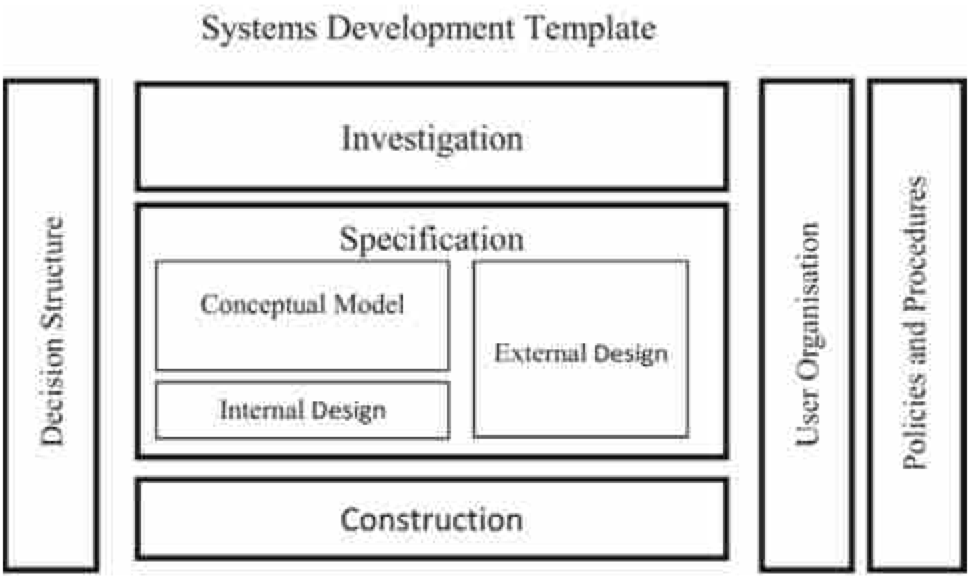


Fig. 7.3 : System Development Template

❖ **Guidelines for Output Design :**

The outputs obtained from computer systems are mainly used to communicate the result that is obtained after processing of data to the users. The output so obtained can be kept for further discussion. It seems that the output obtained from computer system is straight and serves as direct source of information for the users. The computer output so obtained is organized throughout and should be available to people for whom the system is easy to use.

It is analysed that the output of the system have been explained at the time of designing a logical stage. At that time, if the designed is not specified, then it should be planned at the starting of designing process and should include terms such as output connect, format, response etc. There exists different type of output obtained such as :

- External
- Internal
- Operational
- Interactive
- Turn around

It is found that nearly every screens obtained shows lots of information and are interactive for users to ask their needs by posting questions.

❖ **Choosing Output Technology :**

- Who will use/see the output ?
- How many people will need the output ?
- Where the output is needed (distribution/logistics) ?
- What is the purpose of the output ?
- How quickly is the output needed ?
- How frequently will the output be accessed ?
- How long must the output be stored ?
- What special regulations apply ?

- What is the cost ?
- What are the environmental factors (noise, temperature, space, hardware) ?

❖ **Guidelines :**

The system analyst should understand the logical design of the system before starting the physical design of any component. The first step is to review the system requirement document, after reviewing the system requirements; you can start the actual design process.

The best place to begin is with data design, which defines the physical data structures, elements and relationships. After completion of the data design, user interface is considered which affects the interaction between the user and the system. After the user interface is developed, work on specific input and output design tasks start. After that the work on architecture, which translates the design into code modules, is done.

You can design the programme to produce a report of all customers whose balance exceeds a specific amount entered by the user. For example, if a user wants to display customers with balances of more than Rs.8000, then in the parameter query 8000 is entered. A parameter is that value, which the user enters whenever the query is run, which provides flexibility, enables users to access information easily and costs less.

A good system design can combine both the approaches. For example, you can design the programme to accept a variable amount entered by the user but can also start with a default value of 3000. A default is that value which the system displays automatically.

❖ **Output Design Objectives :**

- Assure purposeful output
- Make meaningful to user
- Provide appropriate quantity
- Appropriate distribution
- Assure timeliness
- Choose effective output method

❖ **Formatting and Designing Report :**

A report is a business document, which contains only predefined data. It can also be considered a passive document for reading or viewing data, which contains data from many database records or transactions.

❖ **Objectives of report design :**

- Designing reports to serve a specific purpose
- Making meaningful reports to users
- Delivering the appropriate output distribution
- Providing output on time
- Providing appropriate output distribution
- Choosing the most effective output method

❖ **Designing Reports :**

The report's designing is performed in the logical design phase and is related to DFD and E–R diagrams.

Report and DFD Relationship

The data input indicates forms while data output indicates reports. Thus, prototypes are required for designing reports.

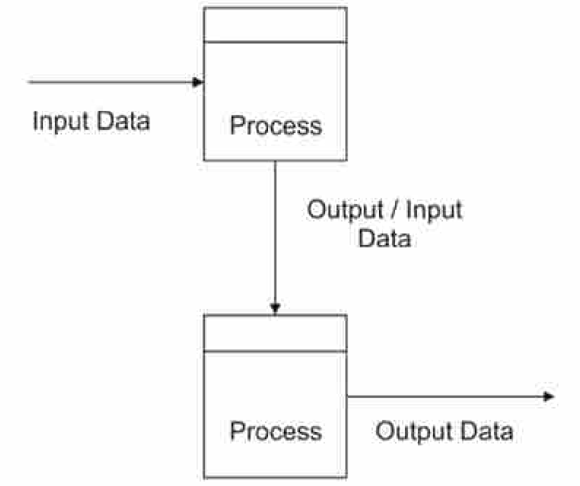


Fig. 7.4 : Report Designing Process

❖ **Report Designing Process**

The report designing process includes several steps, which are explained below :

- Collect and analyse the needed data
- Determine requirements
  - Who will use the report ?
  - What will be the need of such report ?
  - What is the time when you require such report ?
  - What place is required for such report ?
  - Probable number of people who use or want to see such report ?
- Generalisation of the report background
- Apply the prototype methodology

❖ **Prototyping :**

In this, the first prototype can be framed depending on the data and requirements. Here the user will go through such prototype designs that will acknowledge it or asked for any alteration. If the alterations needs to be done, then under such case the construction evaluation refinement cycle gets continuously in the loop till the design gets acknowledged.

❖ **Rules for Designing Reports :**

- Clear Titles
- Included desired information
- Provide effective alignment and balanced structure
- Keep it easy to use

❖ **Guidelines for Formatting Reports :**

❖ **Highlighting**

- Highlight the contents sparingly to draw users to or away from certain information.

- The blinking and audible tones should only be used to highlight critical information requiring user's immediate attention.
- The methods should be consistently selected and used based upon level of importance of emphasized information.

❖ **Benefits of Using Colour :**

- It gives an appealing effect.
- Comment on type of display
- Feels good under complicated displays
- Stress on actual arrangement of data or information
- It warns the user
- It shows for extra touching reactions

❖ **Problems from using Colour :**

- The group of colours will result in problems for certain users.
- If the colour displays is different then there will be less of resolution
- The lower of colour fidelity occurs if colour displays are different
- The conversion to selected media cannot be easily converted

❖ **Displaying Text :**

- The text should always see in upper and lower case using standard punctuation.
- Between the paragraphs, apply double spacing if there occurs a space otherwise put a blank line.
- The text should always be left justified leaving the right margin
- The words should not be with hyphen when placed in between lines.
- The application of abbreviations and acronyms permits only when they are clearly followed by users.

❖ **Designing Tables and Lists :**

- There should me respectable labels in columns and rows.
- All information should be distinguished by using labels which are highlighted.
- Use the re–display label, if the data is too big to be fitted on single screen or page.

❖ **Formatting, Columns, Rows and Text :**

- You can sort columns, rows and text in the required order
- When there is long column, the put an empty line between every five rows
- Each columns should contains at least two blank spaces between them
- The blank space on the printed report is mend for user to write certain notes
- To make columns, rows and text attractive, apply single typeface
- Use simple fonts

❖ **Formatting Numeric, Textual and Alphanume Ric Data :**

- To format, use right justified with decimal points or sequence of characters
- To left justify your textual data, you have to apply the line which is usually having 30–40 characters per line

❖ **Data Entry Process :**

The input process should be efficient and logical. The system analysts by applying business process engineering techniques while studying transactions and business operations to determine how and when the data should be entered into the system. The method for the same is batch or online input method, each method has its own advantage and disadvantages.

Batch Input : With the help of batch input, the data is normally done on particular time schedule that can be every day, every week, every month or can be even every quarterly. The presence of batch input normally takes place when a payroll department gathers time cards which are collected during the closing of week in which the data enters in form of batch.

Online Input : Normally, the batch input method is applied in case of certain activities that involve with online data entry operations. The idea of online method is based on certain features of quick data validation as well as presence of required data. It is analysed that one of the pioneer online input methods results in source data automation where online data entry as well as automated data capture combines together with the used of certain input devices. It is found that the source data automation is quick and correct way that results in less human interference in case of process to be translated.

Nowadays, many organisations make use of combined source data automation as well as strong network to handle worldwide operations quickly. Few examples of source data automation include :

- Automatic Teller Machines (ATMs) that can read data strips issued by the bank in shape of cards.
- Automation in case of point–of–sale (POS) terminals that carries bar code scanners as well as magnetic swipe scanners that carries credit card data.
- Use of ID cards designed for factory workers to clock on and off for certain particular job type.

❖ **Disadvantages :**

If not a data automation is applied, the manual data entry results in slower working and is more expensive as compared to batch input since it involves transition at particular time when computer requirement is maximum.

The usability of batch or online input depends as per the business and customer needs. To do hotel reservations, you need to perform online input while the accounts working inside the hotel can be done by using batch input.

❖ **Data Entry Methods :**

You can enter the data by using certain data entry methods such as :

- Keyboards
- Optical Character Recognition (OCR)
- Magnetic Ink Character Recognition (MICR)
- Mark–Sense Forms

- Punch–Out Forms
- Bar Codes
- Intelligent Terminals

In order to confirm a data, you have to do the following test :

- missing data test
- correct field length test
- class or composition test
- range or reasonableness test
- invalid values test

❖ Input Design and Data Collection :

With input design concept, you can convert a user oriented input to computer based format. It is a part of system design where collection of input data serves as an expensive feature in a system. The idea about input design is shown as :

- The source document is different from turnaround document as the earlier carries data which can adjust the presence of resources as latter is machine readable format data.
- The operations all through involve much error less transactions that can be feed in at a particular time.
- The document should be short and contains less data in order to avoid errors.
- The character data should carry reserve number coding whereas mnemonic coding shows data form representation that is simple for user to understand and remember.
- There should be fast error detection in order to catch the desired person who generated it hence forth the error gets easily corrected.
- With the help of multiple levels of messages, the user will be able to find out with the detailed explanations of error with help option without taking into account the lengthy messages.
- With the idea of error suspense record, the fields covered in these could be locating data–entry operator, date and time of transaction entered with transaction type and image, error fields and codes along with successful date transaction status.
- The data input particulars gives detailed idea about particular fields located on an input document along with its features.

❑ Check Your Progress – 5 :

1. While designing a report, which rule is not applicable ?

a. having clear titles   b. no information

c. effective alignment   d. balanced structure

2. Which is not used as data entry method ?

a. mouse   b. keyboards

c. Optical Character Recognition (OCR)

d. Magnetic Ink Character Recognition (MICR)

7.7 Let Us Sum Up :

In this unit we have learnt that prototyping involves starting of development process and acts as per customer evaluation and feedback report.

The generation languages such as 3GLs are made from small set of possibly simple and low level building blocks, while 4GLs serves as an important prototyping tools.

It is studied that the system design phase is an important and challenging domain of system life cycle which is normally used for new designs where designers select the equipment required to perform the system.

The designs can be input as well as output. The output design phase is convenient for user where as an input design phase will decide about the types of inputs for system.

7.8 Answers for Check Your Progress :

❑ **Check Your Progress 1 :**

1 : d

❑ **Check Your Progress 2 :**

2 : c

❑ **Check Your Progress 3 :**

1 : c

❑ **Check Your Progress 4 :**

1 : a

❑ **Check Your Progress 5 :**

1 : b **2 :** a

7.9 Glossary :

1. **3GL –** It is a third generation programming language which makes the languages more programmer friendly with special features.
2. **4GL –** It is a fourth generation programming language which is designed to create commercial business software.
3. **Prototype –** Initial sample or model of particular product that was created to check a particular idea or process that can be replicated or learned from.
4. **Event Diagram –** It is a sequence diagram contains vertical parallel lines with certain processes simultaneously with horizontal arrows that exchange messages.

7.10 Assignment :

1. Collect information on Object oriented analysis and design and give it in your own words.

7.11 Activities :

- Collect information on software development in 4GL and prepare a report on the same.

7.12 Case Study :

- Collect information on system design and explain the same with system development template.

7.13 Further Readings :

1. Analysis and Design and Information System, Rajaraman.
2. Analysis and Design of Information System 2nd Ed. Senn.
3. Introducing Systems Analysis and Design, Lee.
4. System Analysis and Design, Edwards, Tata McGraw–Hill.
5. Systems Analysis and Design, Elias Awad.
6. System Analysis and Design, J.F. Gerald, Tats McGraw–Hill.

BLOCK SUMMARY :

While studying this block, the user will achieve knowledge and understanding about system design and its features with more of generation programming languages. The block explains about Object Oriented Analysis, Prototyping, Entity Integrity constraints with certain examples that will help the user to grab the concept.

The block explains about basic understanding of various Prototyping and concepts related to interviews, questionnaires, record reviews and case tools. The students or programmers will get benefit while reading this block as it gives shortcuts and related examples that will clear all doubts.

BLOCK ASSIGNMENT :

❖ **Short Questions :**

1. What are the types of interviews ?
2. Discuss 3GL in your own words ?
3. What is a typical OO system ?
4. What does a model represent ?
5. List the advantages of 4GL in prototyping stage ?

❖ **Long Questions :**

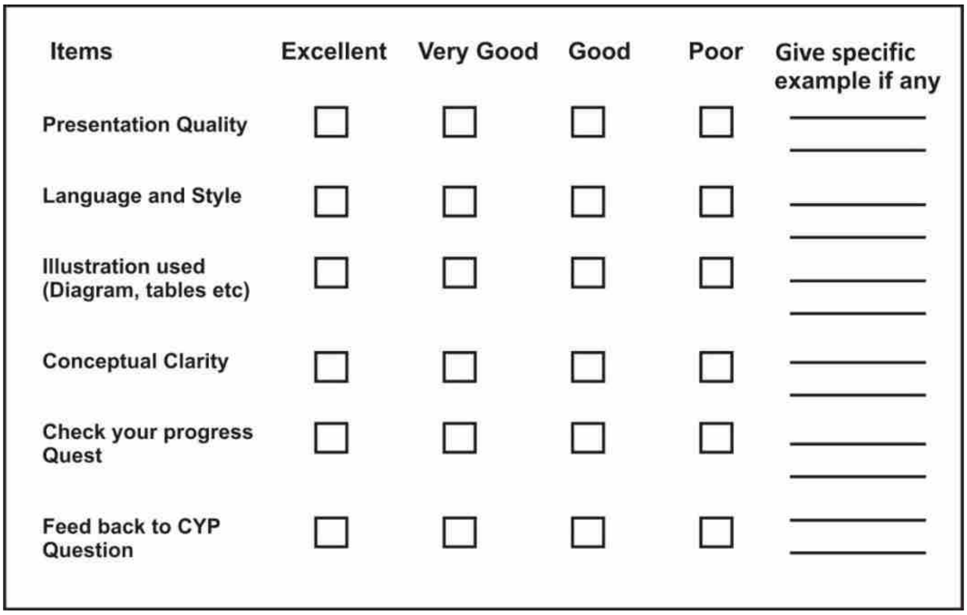
1. What are the guidelines for output design ?
2. What do you mean by initial investigation and feasibility study ?
3. What do traditional 3GLs use ?

❖ **Enrolment No. :** □

1. How many hours did you need for studying the units ?

Unit No.	4	5	6	7
No. of Hrs.				

2. Please give your reactions to the following items based on your reading of the block :



Items	Excellent	Very Good	Good	Poor	Give specific example if any
Presentation Quality	□	□	□	□	________ ________
Language and Style	□	□	□	□	________ ________
Illustration used (Diagram, tables etc)	□	□	□	□	________ ________
Conceptual Clarity	□	□	□	□	________ ________
Check your progress Quest	□	□	□	□	________ ________
Feed back to CYP Question	□	□	□	□	________ ________

3. Any other Comments

....

....

....

....

....

....

....

....

Dr. Babasaheb Ambedkar Open University Ahmedabad

BCAR-205/ DCAR-205

System Analysis and Design

BLOCK 3 : FILE DESIGNING AND TESTING

UNIT 8 FILE DESIGN

UNIT 9 TESTING AND MAINTENANCE

UNIT 10 SYSTEM ADMINISTRATION AND TRAINING

UNIT 11 SYSTEM SECURITY, AUDIT AND QUALITY ASSURANCE

FILE DESIGNING AND TESTING

Block Introduction :

A file is a group of records that shows information regarding the operations, planning, management and decision–making for a software program. They are used for keeping data for longer period and can be stored in primary as well as secondary storage devices.

In this block, we will learn and study about basic of file and its design specifications. The block will cover details on preparing of files for various programmes and method of storing data. The student will be given with the knowledge about different rules and criteria's about data files.

After studying this block students will be able to handle different software file and software programmes. The will be trained with software testing procedures and trainings. The student will be given with adequate knowledge about system maintenance.

Block Objectives :

After learning this block, you will be able to understand :

- The basic of File Design
- Idea about Database Design
- Overview of Design Implementation
- Concept of Scheduling and Assigning tasks
- Knowledge about Testing
- Understanding about Training
- Basic of System Maintenance

Block Structure :

Unit 8 : File Design

Unit 9 : Testing and Maintenance

Unit 10 : System Administration and Training

Unit 11 : System Security, Audit and Quality Assurance

Unit

08

FILE DESIGN

UNIT STRUCTURE

8.0 Learning Objectives

8.1 Introduction

8.2 File Design

8.3 Database Design

8.4 Overview of Implementation

8.5 Scheduling and Assigning Tasks

8.6 Let Us Sum Up

8.7 Answers for Check Your Progress

8.8 Glossary

8.9 Assignment

8.10 Activities

8.11 Case Study

8.12 Further Readings

8.0 Learning Objectives :

After learning this unit, you will be able to understand :

- The file design
- Outline database design
- Describe concept of implementation

8.1 Introduction :

This unit will deal with designing of files and databases. The file carries bunch of records that carries required user information which can be utilised for operations, planning, and management and in carrying related decisions. The data in the files can be stored individually or in folders which can be kept for indefinite period that can be utilised for specific purpose. Once any material or data is created, it will generate a master file which has all original records. A transaction file contains the detail records representing individual transactions. Transactions records may be used to update a master file.

8.2 File Design :

In designing a particular file we have to first realise its objectives that are applied in case of designing of data storage organisation which can be :

- Made available to user as and when it requires.
- Correct and reliable.
- Efficient and latest to keep necessary records
- Beneficial and can be used for certain purposes.

The idea regarding the stored data is that it should contained overall form that can be fruitful in order to manage, plan, control and can be applied to take any form of decisions.

In case of computer based systems, there exist two data storage mechanism :

- Storing of data individually in files for individual application, where file concerns with lower the data availability and security. Also, it is up to system analyst to select the required file structure as per desired application according to its criteria.
- Designing a database that explains and controls data storage which can be utilized for several applications.

Keys are data items in a record that are used to identify the record. Key fields are used for record retrieval (lookup). There are different types of keys :

1. **Primary Key –** Uniquely identifies a record
2. **Secondary Key –** May not be unique
3. **Concatenated Key –** The key consists of a combination of two or more fields
4. **Foreign Key –** A field is one record that is the key of another record. Used for linking a record in one file or table with a record in another file or table.

Design of files will cover decisions related to nature as well as content of file itself that will explain whether files are to be used for storing transaction details, data or reference information. While designing a file, following decisions takes place :

- What data items to cover record format inside the file ?
- Size of record as per data item features
- Series of records inside the file

In case of physical file database design, the design is not visible to users and applications. It happens due to the fact that the physical file design carries optimise presentation. If the software is dependent on physical data structures, then it will bring about difficulty in designing which results in difficult maintenance. Hence it is confirmed that database systems will make software design decisions to act independently of physical data organisation and storage.

In physical designing of file, there exist certain features such as physical layout, addressing techniques, compaction techniques as well as storage devices.

❖ **Physical Layout :**

The data physical structure is related to :

- Arrangement of files and data elements
- Data division
- Data collection
- High performance and lowering of space

❖ **Addressing Techniques :**

The addressing technique is mostly applied for flat files as well as relational database tables, which will cover :

- Sequential
- Indices
- Inverted Lists
- Hashed

In case of network and hierarchical databases, the content of addressing techniques will cover :

- Immediate Lists
- Indices
- Different Pointing Methods
- Bitmaps Images

❖ **Compaction Techniques :**

The idea behind factors related to compaction techniques are :

- Usability of packed fields to lower the need of required space
- Improves the response time
- Images can be compressed in various versions such as JPEG, MPEG as well as DVI
- Squeezing of text and files

If the data is compressed, it will become smaller in size which uses small space that can be utilised as :

- Excess data which can be present in main memory immediately
- Lowering of big data that can travel through communication channels

❖ **Storage Devices :**

To show storage hierarchy, there are certain storage devices that can be used like :

- Primary memory
- Solid state disk
- Magnetic and Optical disk
- Tape drives

For storing of data in particular storage device, it should be visualised that a proper response time should matched which can effect buffering or caching, that can be further result in good performance.

❑ **Check Your Progress – 1 :**

1. Which among the following is the combination of two or more keys ?

   a. Primary Key   b. Secondary Key

   c. Concatenated Key   d. Foreign Key

8.3 Database Design :

The idea of database design is to control information which can be fully or partly related to database. It is seen that a database is gathering of mixed data which is kept having least laying off that can give out several users at the same time by taking not much time. The system analyst after framing the input and output, will started focusing on designing of concrete database or should plan to adjust data as required by the user. The only aim of analyst at time is to see that the designed information should be accessible easily and quickly and should not be too expensive for different users. While designing a database, the system analyst has to keep the mentioned objectives in his mind :

- Controlled laying–off
- No limitations of data
- Accuracy and jointly
- Cost effective
- Failure recovery
- Privacy and security
- Achieve good performance
- Easy to design
- Easy to apply

Database design relates to arrangement of collecting data needs for an establishment, business process or sanctioned information system which will further be converted into set of specifications which will produce database.

The main inputs to database design based on user's views in finding information which is explained while defining a problem and in analysing different stages of system development life cycle. The idea of database design is not restricted to simply collection of data or information but also to arrange or organise such information in an effective and correct format. For doing this, a great knowledge related to particular information is gathered and will further study about how such thing to be utilised in computing environment where database is done.

❖ **Database Structures :**

One of the easy and simple database structures is known as flat–file database structure. In this database structure, all data is kept in a particular spreadsheet like table which is independent and is not connected with any other files. It is easy and simple to apply. It is not exactly a database, but is used by computer systems as storing of data.

In case of a branched database structure, the files are connected one above another; for example, the name and address file is the parent and the other files are its children. To work in database, the top file is used and flows downward, so it is possible to access a student's academic file starting with his/her name and address. We see that a parent can have many children but a child can have only single parent.

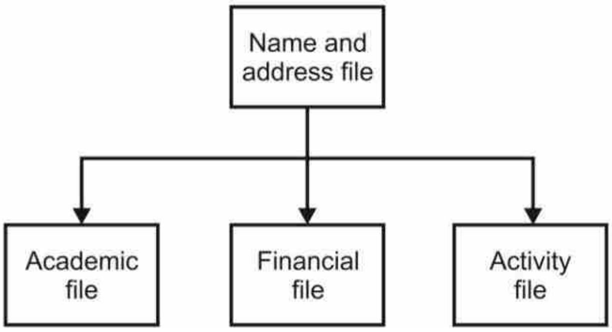


Fig 8.1 : A Hierarchical Database

If you are considering network database, links or pointers shows coordination among certain files in any direction, hence a child will result in many parents. If we see an example where name and address of a particular file carries a link among academic file and further such link will link to name and address file, then we see that a student's mark sheet gets prepared by considering any of the files. Since such links are simple to add with data structure, so the difference among hierarchical database and network databases practically does not result.

It is found that the files which form a relational database can be seen as two dimensional tables which are similar to the spread sheet program. Here the file carries column which will carry out the values of particular field and every row carry out a single record. It is seen that files are connected by pointers or specifically relationships.

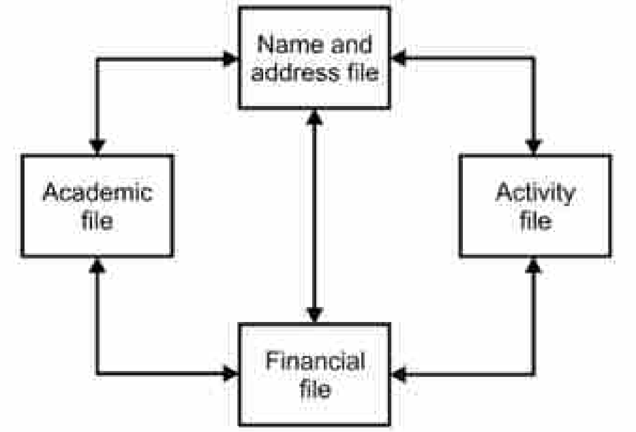


Fig 8.2 : A Network Database

❖ **Database Design Methodologies :**

Several organisations retain a conventional approach for designing practice databases. On account of database design, there exists an essential constituent of information system design, the database concept methodology continues commonly connected into the consideration phase of the organisation's acknowledgement development methodology. Furthermore the definite steps as well as utilities differ consequentially from organisation to organisation, the approach is alienated into two aboriginal stages : logical (or imaginary) database design as well as corporal database design.

Alterations in the corporal database concept can influence the logical design. Consequently, these two criteria are rigidly composed, with countless feedback cycles. We see that, although, that the approach design as well as the logical database design necessity be accomplished preceding the physical database design can be finalised.

❖ **Logical Database Design :**

Logical or conceptual database design is concerned with defining and documenting the database in user terms. The objective is to define the user's understanding of data and how various data elements and composites are interrelated. This phase is geared to a non–technical audience. From the user's standpoint, it is the database design.

The process starts with a study of the user's data views and data uses in the context of the application. Following the organisation's standards, data elements and composites are defined, named and documented in the data dictionary.

Additionally, the relationships between the entities are studied and documented; often using such tools as entity relationship models and/or data normalisation. The output of the logical design process is a set of detailed documentation for all the data that will be stored in the database and for the interrelationships between data.

❖ **Physical Database Design :**

The idea about physical database construction is to prepare a master layout for physical workability of database. Additionally, logical database design, which takes inside of design, will cover key elements of applications. The logical database design defines the data requirements.

On the other hand, application design describe, how data is applied in database. The idea of such design is to apply physical database that correctly show all correct use of data.

Initially, the physical database design process works with transaction analysis where they know about expected usage levels is defined along with several other functions. The research users are engaged in such process as they know with which idea they have to work. The purpose of application design is to find out the points where application applies with such data and hence provide a distinguished transaction analysis.

❑ **Check Your Progress – 2 :**

1. Selection of data type uses ________.

   a. maximum storage area   b. shows value

   c. good data integrity   d. none of these

8.4 Overview of Implementation :

System implementation depends on the successful completion of system testing. If any error occurs in system testing phase then system is completely rolled back to its original phase (first phase).

During system implementation, the components built during development are put into operational use. Following are the activities, which take place during system implementation phase.

- Writing, testing, debugging and documenting programs.
- Converting data from the old to the new system.
- Giving training to the users about how to operate the system.
- Ordering and installing any new hardware.
- Developing operating procedures for the computer centre staff.
- Establishing a maintenance procedure to repair and enhance the system.
- Completing system documentation.
- Evaluating the final system to make sure that it is fulfilling original needs and that it began operation on time and within budget.

❑ Check Your Progress – 3 :

1. System design is done

   a. after determination of system requirement

   b. on request of system analyst

   c. on approval of final specifications

   d. none of these

8.5 Scheduling and Assigning Tasks :

While in an organization, the management or committee should decide and arrange for important major project phases and should describe the related tasks that are to be performed in every phase. Because the project scheduling is not fixed and decided, then in such situation, the management should develop certain flexibility procedures that should be implemented in project schedules. Moreover, it is seen that the flexibility level developed gets lower as and when the project progress with more clear need of project.

The identification and assignment of roles and responsibilities of all affected parties including the functional and technical managers throughout the system development life cycle is done.

❑ Check Your Progress – 4 :

1. Project scheduling means :

   a. completing project in required time frame

   b. completing project as per the project manager request

   c. completing the project as per staff request

   d. all of these

2. _________ should decide and arrange for important major project phases and should describe the related tasks that are to be performed in every phase.

   a. Analysis b. Designer

   c. Programmer d. Management or Committee

8.6 Let Us Sum Up :

It is learnt in this unit that there exist two approaches while storing of data in computer–based systems. These approaches will store data either on an individual files or contains a database that is defined and control data for different applications.

We see that a database is a compilation of interrelated data stored with minimum redundancy to provide for several users swiftly and effectively with the idea that the information can be accessed easily and quickly with flexibility and in low cost for several other users.

It is seen that in case of scheduling and assigning particular work, the management should locate and arrange important project phases and distributes the tasks to be completed in respective phases.

8.7 Answers for Check Your Progress :

❑ **Check Your Progress 1 :**

1 : c

❑ **Check Your Progress 2 :**

1 : c

❑ **Check Your Progress 3 :**

1 : c

❑ **Check Your Progress 4 :**

1 : a **2 :** d

8.8 Glossary :

1. **Database Design –** It is a process of producing detailed data model of a database.
2. **System Life Cycle –** It is an examination of a system which shows all phases of its existence which covers system design, development, production, construction, distribution, operation, maintenance and support.
3. **System Analyst –** It is person who involved in research problems, plans solutions, recommends software and systems, and coordinates with development for business requirements.
4. **Scheduling –** It is a process of deciding how to commit resources between a variety of possible task.

8.9 Assignment :

1. Explain as to why there arises a need for scheduling.

8.10 Activities :

- Discuss file design in your own words.

8.11 Case Study :

- Prepare a detailed report on the overview of system implementation.

8.12 Further Readings :

1. Analysis and Design and Information System, Rajaraman.
2. Analysis and Design of Information System 2nd Ed. Senn.
3. Introducing Systems Analysis and Design, Lee.
4. System Analysis and Design, Edwards, Tata McGraw–Hill.
5. Systems Analysis and Design, Elias Awad.

Unit 09

TESTING AND MAINTENANCE

UNIT STRUCTURE

9.0 Learning Objectives

9.1 Introduction

9.2 Testing

9.3 Training

9.4 System Maintenance

9.5 Management Issues

9.6 Let Us Sum Up

9.7 Answers for Check Your Progress

9.8 Glossary

9.9 Assignment

9.10 Activities

9.11 Case Study

9.12 Further Readings

9.0 Learning Objectives :

After learning this unit, you will be able to understand :

- About testing
- Outline of software testing
- Describe training
- Explain System Maintenance
- Familiarize with Management issues

9.1 Introduction :

This unit deals with various testing methods that are explained using SDLC. Here we will give the description about testing and maintenance in software. The method of various types of system maintenance along with management issues is also explained.

9.2 Testing :

This is most important phase of the system development life cycle. After completion of system coding phase, system–testing phase is carried out.

Software testing is a mechanism which will confirm and verify the following parameters that should be present in a program/application/product:

- The design and development should handle and takes care of all business and technical measures
- The work to be carried off using certain specifications or features

Software analysing, depending on the checking approach exercised can be applied at any time in the execution manner. Although, maximum of the test action exists later the conditions have been explained along with the coding mechanism has been accomplished. As alike, the methodology relevantly the test is directed by the software production methodology selected.

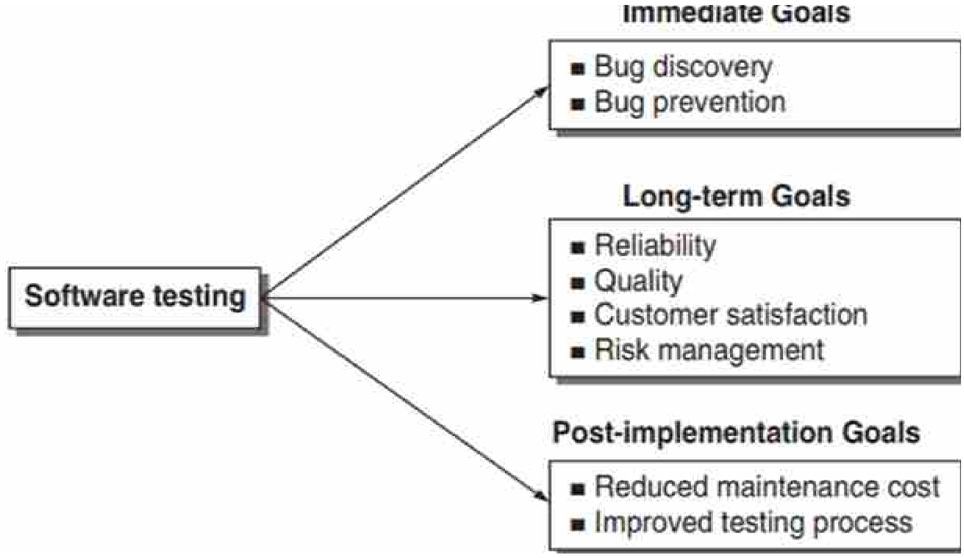
Testing can be defined as –

Testing is the process of demonstrating that there are no errors.

OR

Testing is the process of executing a program with the intent of finding errors.

❖ **Goal of Software Testing :**

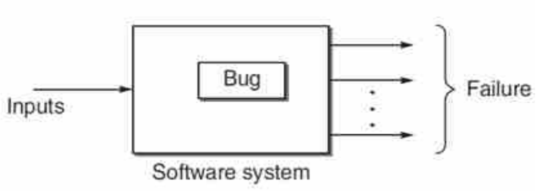


The Immediate goals of Software testing are Bug discovery and Bug prevention. In a Long–term we can achieve Reliability in the software product, we can produce a high Quality of software product and if the product is reliable and quality oriented then it will give satisfaction to the customer. We can easily manage risk in the development process of the software. Maintenance of the software is the last and very long phase of SDLC (after implementation of software) and that will increase the overall cost of the software product. If the software product is properly tested then it will reduce the maintenance cost of the product as its post-implementation goal.

Now let us understand some basic terms like – Failure, Defect, Bug and Error which are related to the Software Testing:

- **Failure :** When the software is tested, *failure is the first term being used*. It means the inability of a system or component to perform a required function according to its specification. In other words, when results or behavior of the system under test are different as compared to specified expectations, then failure exists.
- **Fault/Defect/Bug :** Failure is the term which is used to describe the problems in a system on the output side, as shown in following Figure, *fault is a condition that in* actual causes a system to produce failure. Fault is synonymous with the words *defect or bug. Therefore, fault is the reason embedded in any phase of SDLC* and results in failures. It can be said that failures are manifestation of bugs. One failure may be due to one or more bugs and one bug may cause one or more failures. Thus, when a bug is executed, then failures are generated. But this is

not always true. Some bugs are hidden in the sense that these are not executed, as they do not get the required conditions in the system. So, hidden bugs may not always produce failures. They may execute only in certain rare conditions.



- **Error :** Whenever a development team member makes a mistake in any phase of SDLC, errors are produced. It might be a typographical error, a misleading of a specification, a misunderstanding of what a subroutine does, and so on. Error is a very general term used for human mistakes. Thus, an error causes a bug and the bug in turn causes failures.

Error
Bug
Failures

❖ **Testing Principles :**

To test the system properly and to build a quality product, which gives satisfaction to the customer, the following principles should be follow by a test engineer.

1. **Testing should be effective and not exhaustive.**

Consider we accept a password from the user of six-character with certain rules. The first character of the password should be digit and rest of the five characters should be alpha-numeric. Using this rule 10 * 625 = 9,16,13,28,320 different passwords can be generated. Can we test all the passwords exhaustively? It will take 2,905 years if we consider only 10 second of time to test one password. You need to test only few passwords (5 to 10) in such a way that all possible combinations have to be tested.

2. **Testing is not a single phase performed in SDLC.**

Testing should not be performed as a single phase of SDLC, but it should be performed in each phase of the SDLC.

3. **Destructive approach for constructive testing.**

Try to use destructive approach to test a particular software. Find all different types of invalid inputs and test the software. Your destructive approach of testing will help you a lot to find all possible errors and to do constructive testing.

4. **Early testing is the best policy.**

Start software testing as early as possible. "Prevention is always better than Cure".

5. **Probability of the existence of an error in a particular section of a program is proportional to the number of errors already found in that section.**

If two modules X and Y are coming for testing after rectification of the errors found in the previous tests. Suppose, in previous test 50 errors found in module X and only 3 errors found in module Y. The chance of the error in module X is higher, and you need to give higher probability to retest it again compare to module Y.

6. **Testing strategy has to be start at the smallest module level, and needs to be expand towards the whole program.**

7. **Testing should also be performed by an independent team.**

To test the software properly, there has to be an independent team than a development team.

❖ Types of Testing :

Software testing process can be roughly divided into two categories : [1] Static Testing and [2] Dynamic Testing.

[1] STATIC TESTING :

It is a technique for assessing the structural characteristics of source code, design specifications or any notational representation that conforms to well-defined syntactic rules. It is called as static because we never execute the code in this technique. For example, the structure of code is examined by the teams but the code is not executed.

[2] DYNAMIC TESTING :

All the methods that execute the code to test a software are known as dynamic testing techniques. In this technique, the code is run on a number of inputs provided by the user and the corresponding results are checked. This type of testing is further divided into two parts : (A) Black-box testing and (B) White–box testing.

[A] **Black–box testing :** This technique takes care of the inputs given to a system and the output is received after processing in the system. What is being processed in the system? How does the system perform these operations ? Black–box testing is not concerned with these questions. It checks the functionality of the system only. That is why the term black–box is used. It is also known as functional testing. It is used for system testing under validation.

[B] **White–box testing :** This technique complements black-box testing. Here, the system is not a black box. Every design feature and its corresponding code is checked logically with every possible path execution. So, it takes care of the structural paths instead of just outputs. It is also known as structural testing and is used for unit testing under verification.

❖ Static Vs Dynamic Testing :

There are plenty applications to software checking. Views, walkthroughs, or examinations are assessed static checking, whereas exactly exercising programmed code with an allowed assign of test cases is acknowledged to as dynamic analysing. Static analysing can be evacuated. Dynamic analysing acquires spot when the programme itself is exercised for the initial time. Dynamic analysing may commence before the programme occurs 100% comprehensive in array to analysis particular areas of code.

Unconditional approaches for this are matching employing stubs/drivers or execution from a debugger arrangement. For simulation, Spreadsheet approaches are nearby very peculiar that can be analysed comprehensively or mutually with outcomes that appeared currently following each conclusion or text manipulation.

Static Testing	Static Testing
Code is not executed during Static testing.	Dynamic testing is done by executing the code.
Static testing means to review and to examine the software	Dynamic testing means running and then testing the software.
Static testing is cheaper.	Dynamic testing is costly.
It is done as phase verification.	It is done as phase validation.
Static testing techniques includes formal technique review, inspection, walkthrough.	In dynamic testing various techniques like white box and black box are used.
It does not take time as its purpose is to check the item.	It takes time because it executes the software code and we need to run the test cases.

❖ **Black–box Vs White–box Testing :**

Black-box Testing	White–box Testing
It is also known as internal structure testing technique.	It is called as glass-box testing technique
It is a testing technique in which everything like code, internal structure of the program being tested is unknown from the tester	It is a testing technique in which the code and internal structure of the program being tested must be known to the tester.
It is performed by the independent team of Software Testers	It is performed by the Software Developers
No programming knowledge is required for Black–box testing	White–box testing requires programming knowledge

❖ **Level of Testing :**

Neither the System can be designed in a single phase not its testing can be done in a single phase. The analyst must perform different types of testing in each phase of the SDLC.

[1] **Unit Testing :** Unit testing is the smallest building block of the testing. Before we validate the entire System, small units of the software components like functions, classes etc. must be tested individually. Testing small unit component of the software is called Unit testing. Usually, Unit testing has to be performed by the developer of that particular unit. To test the various units' developer can use Drivers or Stubs.

[2] **Integration Testing :** Once the individual units are developed and tested, the units are combined with each other. With each integration of different Modules, testing has to be performed. It is possible that the unit individually perform well, but when we integrate it with some other module(s), it generates some errors. To test, whether modules are working perfectly with each other Integration testing is essential. Different methods are there to perform Integration testing these are : [1] Top–Down Integration, [2] Bottom–Up Integration, [3] Call Graph–based method, and [4] Path–based Integration.

[3] **Function Testing :** After performing Integration testing successfully, the whole software tested for each functional requirement, which the software should perform. All the functional requirements mentioned in SRS document is performed and entire software is tested for finding bugs.

[4] **System Testing :** Once the software passed from the Functional testing, then System testing is conducted. During the System testing all non–functional parameters like performance, security, maximum load etc. factors are tested.

[5] **Acceptance Testing :** After conducting System testing successfully, Acceptance testing is performed. It is most important level of the testing. Acceptance testing is the formal testing conducted to determine whether a software satisfies all criteria of the buyer of that software or not. Acceptance testing can be divided in two parts :

[A] **Alpha Testing :** Alpha testing is conducted in the presence of the users (customers) of that software at the developer's site. It is conducted in controlled environment, in the presence of its developer. Developer keeps record of the Errors and Bugs detected during Alpha testing. It includes both Black–box and White–box testing.

[B] **Beta Testing :** Beta testing is conducted by the users (customers) at user's side. Beta testing version will be prepared by the developer and given it to the users. It is not performed in controlled environment as developer in not present during testing. Here users will test the Beta version product at their end and report the errors or bugs to the developer. It includes only Black–box testing.

❑ **Check Your Progress – 1 :**

1. Alpha and Beta testing are type of ________ testing.

   a. Unit b. System c. Function d. Acceptance

2. Security, Performance and Load of the system is tested during ________ testing.

   a. Unit b. System c. Function d. Acceptance

3. In ________ testing different modules of software are combined and tested.

   a. Unit b. System c. Integration d. Acceptance

4. Software Testing means ________.

   a. Validating software program.

   b. Verifying software application.

   c. Validating and Verifying product.

   d. All of the above.

9.3 Training :

Once the user gives a green signal for development of fresh system, then initially, the implementation phase will begin. The process of implementation is basically introduction of practical ideas based on theoretical write off. During this phase, particularly all programs get loaded on user's computer for designing. Once the designs are loaded on user's computer, then the actual process of training to users starts which will cover following topics :

- How to carry out the design
- When and how to put the data
- How to make the data available for processing
- How to generate an output and takes the reports

When the user gets informed with above process, then the training related to computer–based system starts where manual work started applying on computers.

❑ **Check Your Progress – 2 :**

1. Which among the topic is not related to training ?

   a. How to execute the package b. How to enter data

   c. How to process application d. How to take the reports

9.4 System Maintenance :

The system is considered to be working when phase six is completed. However, there are still a number of activities that take place after a system is completed. The main activities are :

❖ **Post Implementation :**

The post implementation activity takes place after a year to determine whether the system is perfectly working or not, whether the system is satisfying the user needs or not, if not, changes are made to make the system to make it perfect.

❖ **Maintenance :**

It is a part of SDLC. It includes following tasks :

- Correcting errors
- Resolving necessary changes
- Enhances or modifies the system
- Assigns staff to perform maintenance activities
- Provides for scheduled maintenance

For example, if the Corporation changes octroi rate from one value to another, corresponding changes will have to be done in the programs, which uses this octroi rate.

❑ **Check Your Progress – 3 :**

1. Post implementation of system means :

   a. repairing of system b. saving the data

   c. satisfying customer demand d. all

9.5 Management Issues :

There are certain discussion and if and but position arises sometimes while performing the design process. At the time of software maintenance there exists certain problems or issues which can be managerial, technical or both. Some of the common management issues are related to discussion based on customer priorities, staffing and often cost factor that arises. Some of the technical problems or issues that exist during maintenance period are less of subject knowledge; understand of program, impact analysis and testing and maintainability measurement.

❑ **Check Your Progress – 4 :**

1. Which is not concerned with management issues ?

   a. alignment with customer priorities

   b. purchase of machinery

   c. staffing

   d. change of office

9.6 Let Us Sum Up :

In this unit, we have learnt that data file management, test is performed on software, which can be done by several testing methods employed during development process.

We see for designing a process or system, we have to take the consent from the user. Once the user accepts all conditions and layout, then the initial step involved will be an implementation phase, where theory is turned into practice.

After the implementation phase, the post implementation work takes place after a year to determine that whether the system is perfectly working or not, whether the system is satisfying the user needs or not, if not, changes are made to make the system to make it perfect.

While performing development or maintenance of software programs, there are various key issues that arise that could be both managerial as well as technical. The management issues relate to aligning with customer priorities, staffing, cost while technical issues are restricted to knowledge, understanding, impact analysis, testing and maintainability measurement.

9.7 Answers for Check Your Progress :

❑ **Check Your Progress 1 :**

1 : d　　**2 :** b　　**3 :** c　　**4 :** d

❑ **Check Your Progress 2 :**

1 : c

❑ **Check Your Progress 3 :**

1 : c

❑ **Check Your Progress 4 :**

1 : d

9.8 Glossary :

1. **Testing** – It is the process to investigate the process and to show the stakeholders with relevant information related to quality and durability of product or services.

2. **Traits** – It is a model for structuring object–oriented programs.

9.9 Assignment :

1. Study in detail system testing and prepare a report on the same in your own words.

9.10 Activities :

- Discuss system maintenance and management issues and give it in your own words.

9.11 Case Study :

- Discuss the importance of testing in Software Development Life Cycle (SDLC). Explain the various methods of testing.

9.12 Further Readings :

1. Software Testing Principles and Practice (Second Edition), By Naresh Chauhan, From Oxford publication.
2. Software Testing Principles and Practice, By Srinivasan Desikan and Gopalaswamy Ramesh, From Pearson publication.
3. Analysis and Design and Information System, Rajaraman.

Unit 10

SYSTEM ADMINISTRATION AND TRAINING

UNIT STRUCTURE

10.0 Learning Objectives

10.1 Introduction

10.2 Training

- **a. Training Systems Operators**
- **b. User Training**
- **c. Training methods**
  - **i. Vendor and In–Service Training**
  - **ii. In–house Training**

10.3 Conversion

- **a. Conversion Methods**

10.4 Conversion Plan

- **a. Operating Plan**

10.5 Let Us Sum Up

10.6 Answer for Check Your Progress

10.7 Glossary

10.8 Assignment

10.9 Activities

10.10 Case Study

10.11 Further Readings

10.0 Learning Objectives :

After learning this unit, you will be able to understand :

- objectives of system administration
- the purpose of giving training to the user of system
- different types of training
- how conversion takes place from existing system to new system
- conversion plan
- phases of conversion
- operating plan

10.1 Introduction :

Putting a new system into operation is usually complicated by the fact that there is an older system already in operation. The analyst has to deal with changing from something familiar to something new and different, while also attending to the mechanics of implementation. Since the concern for simultaneous conversion and implementation is usual. New system brings in new equipment.

It may represent a change from manual to automated operation or a change in the level of available machine capacity. During implementation, planning plays a decisive factor in determining the ultimate success or failure of system. Due attention should be paid to :

1. Assigning system personnel.
2. Structuring user relationship.
3. Preparing for new equipment.
4. Training user personnel.

1. **Assignment of Systems Personnel :** Assign people to the implementation who demonstrates the ability in dealing with the unique problem situations associated with the process.
2. **Structuring user Relationships :** Plan for periodic meeting between user and system personnel for the duration of the implementation, to discuss problems being faced. Also, there should be provisions to meet when the need arises. Certainly, waiting for the meeting in critical problems is not a reasonable approach. Also, if people meet only during crisis, they cannot expect a very positive encounter.
3. **Preparing for New Equipment :** New equipment means more complexity. For new equipment additional areas of concern are to be taken care of :
   a. Structuring a relationship with the equipment vendor.
   b. Preparing a physical site for installation and use of new equipment.
   c. Installation of new equipment and removing old equipment.
   d. Training personnel to use the new equipment.
4. **Training of user Personnel :** Planning for the formal training of user personnel in the operation of the new system is important. A new method may drastically affect people's lives by changing their work methods, work style and relationships with other employees. One of the most effective ways of dealing with the potential impact of these changes is to provide a well–designed program of training. The training program should :

10.2 Training :

Even well designed and technically elegant systems can succeed or fail because of the way they are operated and used. Therefore, the quality of training received by the personnel involved with the system in various capacities helps or hinders, and may even prevent, the successful implementation of an information system. Those whose will be associated with or affected by the system must know in detail what their roles will be, how they can use the system, and what the system will or will not do. Both systems operators and users need training.

a. Training Systems Operators :

Many systems depend on the computer – center personnel, who are responsible for keeping the equipment running as well as for providing the necessary support service. Their training must ensure that they are able to handle all possible operations, both routine and extraordinary. Operator training must also involve the data entry personnel.

If the system calls for the installation of new equipment, such as a new computer system, special terminals, or different data entry equipment, the operators training should include such fundamentals as how to turn the equipment on and use it, how to power it down, and a knowledge of what constitutes normal operation and use. The operators should also be instructed in what common malfunctions may occur, how to recognize them, and what steps to take when they arise. As part of their training, operators should be given both a troubleshooting lists that identifies possible problems and remedies for them, as well as the names and telephone numbers of individuals to contact when unexpected or unusual problems arise.

Training also involves familiarization with run procedures, which involves working through the sequence of activities needed to use a new system on an ongoing basis. These procedures allow the computer operators to become familiar with the actions they need to take (such as mounting magnetic disks or tapes, copying files, changing printer forms, or turning on communication systems), and when these actions must occur. In addition, they find out how long applications will run under normal conditions. This information is important both to enable users to plan work activities and to identify systems that run longer or shorter than expected – a sign that typically indicates problems with the run.

b. User Training :

User training may involve equipment use, particularly in the case where, say, a microcomputer is in use and the individual involved is both operator and user. In these cases, user must be instructed first in how to operate the equipment. Questions that seem trivial to the analyst, such as how to turn on a terminal, how to insert a diskette into a microcomputer, or when it is safe to turn off equipment without danger of data loss, are significant problems to new users who are not familiar with computers.

User training must also instruct individuals in troubleshooting the system, determining whether a problem that arise is caused by the equipment or software or by something they have done in using the system. Including a troubleshooting guide in systems documentation will provide a useful reference long after the training period is over. There is nothing more frustrating than working with a system, encountering a problem, and not being able to determine whether it is the user's fault or a problem with the system itself. The place to prevent this frustration is during training.

Most user training deals with the operation of the system itself. Training in data coding emphasizes the methods to be followed in capturing data form transactions or preparing data needed for decision support activities. For example, in an accounting system, it may be important to translate customer names into customer account numbers that are input as part of the accounting transaction. Users must be trained so that they know how to determine the customer account number, that it is four digits in length, and that there are no alphabetic characters in it.

Data – handling activities receiving the most attention in user training are adding data (how to store new transactions), editing data (how to change previously stored data), formulating inquiries (finding specific records or getting responses to questions) and deleting records of data. The bulk of systems use involves this set of activities, so it follows that most training time will be devoted to this area.

From time to time, users will have to prepare disks, load paper into printers, or change ribbons on printers. No training program is complete without some time devoted to systems maintenance activities. If a microcomputer or data entry system will use disks, users should be instructed in formatting and testing disks. They should also actually perform ribbon changes, equipment cleaning and other routine maintenance. It is not enough to simply include this information in a manual, even though that is essential for later reference.

As the above discussion demonstrates, there are two aspects to user training : familiarization with the processing system itself (that is, the equipment used for data entry or processing) and training in using the application (that is, the software that accepts the data, processes it, and produces the results). Weaknesses in either aspect of training are likely to lead to awkward situation that produce user frustration, errors, or both. Good documentation, although essential, does not replace training. There is no substitute for hands – on – operation of the system while learning its use.

c. Training Methods :

The training of operators and users can be achieved in several different ways. Training activities may take place at vendor locations; at rented facilities, for example, in hotels or on university campuses; or in house at the employee's organizations. The methods and content of the training often vary, depending on the source and location of the training.

i. Vendor and In–Service Training :

Often the best source of training on equipment is the vendor supplying the equipment. Most vendors offer extensive educational programs as part of their services, in some cases, there is a charge, but in many instances training is free. For example, IBM offers complimentary two and three – day courses to purchasers of many of their minicomputers and mainframe computers. The courses, offered by experienced trainers and sales personnel, cover all aspects of using the equipment, from how to turn it on and off, to the storage and removal of data, to handling malfunctions. This training is hands–on, so the participants actually use the system in the presence of the trainers. If questions arise, they can quickly be answered. Since the system is intended for training, there is generally no rush to get training out of the way so that the productive use of the system can start. Training conducted at the organization's location might be rushed, a danger that installation personnel must guard against.

If special software such as a teleprocessing package or database management system is being installed, sending personnel to off – site short courses providing in – depth training is preferable to in – service training. These courses, which are generally provided for a fee, are presented to personnel from many organizations that are acquiring or using the same system. The benefit of sharing questions, problems, and experiences with persons from other companies is substantial. The personal contacts made during the sessions frequently last for years, with the continual sharing of information benefiting both parties. Short courses often involve additional time and costs for travel to other cities.

ii. In–house Training :

The advantage of offering training for the system on site is that the instruction can be tailored to the organization where it is being offered and focused on special procedures used in that setting, the organization's plans for growth, and any problems that have arisen. Often, the vendors or training companies negotiate fees and charges that are more economical and that enable the organization to involve more personnel in the training program than is possible when travel is required.

There are also disadvantages. The mere fact that employees are in their own surroundings is a distraction, since telephone calls and emergencies can disrupt training sessions. Moreover, when outside firms' come on – site, they many present courses that emphasize general concepts but that lack sufficient hands – on training. The training coordinator must recognize this possibility and deal with it an advance to ensure that the course content will meet operating needs.

In – house training can also be offered through special purchased instructional materials. A variety of professional training programs on special topics can be rented or purchased from computer training firms such as Edutronics (McGraw – Hill, Inc.); Deltak, Inc.; Professional Development, Inc.; and Learning Corporation of America. Other vendors offer printed and audiovisual programmed instruction materials that are either self – instructional or that supplement other training activities.

However, there is no substitute for hands –on experience. Training manuals are acceptable for familiarization, but the experiences of actually using the equipment, making and correcting mistakes, and encountering unexpected situations are the best and most lasting way of learning.

Training manuals generally take one of two approaches. Some have the user work through different activities step by step. For example, the checklist is provided to list the steps necessary to implement a system. Each individual step is listed in the proper order.

The other common approach is to create a case – study example that include all frequently encountered situations that the system is able to handle and that the users should be able to handle. Then the users must use the system to handle the actual situations; that is enter data as required, process the data, and prepare reports. If the system is inquiry – oriented, the case study should require the user to pose and receive responses to inquiries. If the results produced do not match those provided in the training guide, the users will know that mistakes were made.

During training, systems personnel should be alert to comments made by users or to problems that users may encounter. Although human factors testing, performed earlier, is intended to detect difficulties, some problems may not occur until inexperienced users are directly interacting with the system. Despite testing, awkward keying requirements to enter data, unexpected transactions, or unusual ways of preparing transactions may still arise during training. The trainer must be certain to involve systems personnel when problems in the design are found, while assisting users who are reluctant to change from their old ways to the new methods required to use the system. Of course, the trainer must first be certain that the new methods are necessary and do represent an improvement over current method.

❑ **Check Your Progress – 1 :**

1. Training can be ________.

   a. Vendor and in–service b. In–house

   c. Both [a] and [b] d. Out-door

2. Training must include ________.

   a. Use of equipment b. Handling data

   c. Troubleshooting d. All of the above

10.3 Conversion :

Conversion is the process of changing form the old system to the new one.

a. Conversion Methods :

There are four methods of handling a systems conversion (Table 9.1). Each method should be considered in light of the opportunities that it offers and problems that it may cause. However, some situations dictate the use of one method over others, even though other methods may be more beneficial. In general, systems conversion should be accomplished as quickly as possible. Long conversion periods increase the possible frustration and difficulty of the task for all persons involved, including both analysts and users.

i. Parallel Systems :

The most secure method of converting from an old to new system is to run both systems in parallel. Under this approach, users continue to operate the old system in the accustomed manner but they also begin using the new system. This method is the safest conversion approach, since it guarantees that, should problems such as errors in processing or inability to handle certain types of transactions arise in using the new system, the organization can still fall back to the old system without loss of time, revenue, or service.

The disadvantages of the parallel systems approach are significant. First of all, the system costs double, since there are two sets of systems costs. In some instances, it is necessary to hire temporary personnel to assist in operating both systems in parallel. Second, the fact that users know they can fall back to the old ways may be a disadvantage if there is potential resistance to the change or if users prefer the old system. In other words, the new system may not get a fair trial.

All in all, the parallel method of systems conversion offers the most secure implementation plan if things go wrong, but the costs and risks to a fair trial cannot be overlooked.

ii. Direct Cutover :

The direct cutover method converts from the old to the new system abruptly, sometimes over a weekend or even overnight. The old system is used until a planned conversion day, when it is replaced by the new system. There are no parallel activities. If the analyst must make the change and wants to ensure that the new system fully replaces the old one so that users do not rely on the previous methods, direct cutover will accomplish this goal. Psychologically, it forces all users to make the new system work; they do not have any other method to fall back on.

The advantage of not having a fallback system can turn into a disadvantage if serious problems arise with the new system. In some instances, organizations even stop operations when problems arise so that difficulties can be corrected.

One organization allocated its entire accounting staff to entering data to start a new automated system. The task took approximately three weeks, during which time none of the regular accounting operations that were to be converted to the new system were performed. Consequently, a three – week backlog of work developed. However, such backlog was expected and management had planned to authorize overtime work and the hiring of temporary assistance to catch up after conversion. Approximately two days before the direct cutover was to take place, a senior manager realized that the accounting department was not planning to preserve the data for accounts receivable aging. The manager stopped the conversion. As a result, the accounting staff had to catch up on three weeks work and reschedule the conversion to a date one–month later, when many of the previous steps had to be restarted. The system was finally implemented three months later, after much extra work, overtime, and staff frustration because of the way the cutover was handled.

Stopping conversion was a particularly drastic measure. It would have been doubly bad had the steps been taken because of technical problems needing correction. If users know that a system was once halted because of difficulties, they may not be fully confident that the system will be reliable, even if analysts tell them that the problems have been corrected. The time it takes to redo work that was stopped because of the conversion can be both lengthy and costly, and time lost can never be recaptured.

Direct cutover requires careful planning. Training sessions must be scheduled and maintained. The installation of all equipment must be on time, with ample days allowed in the schedule to correct any difficulties that occur. Any site preparation must be complete before the conversion can be done.

Direct conversions are quite common, particularly with purchased or turnkey systems. For example, a hotel operation decided to install an automated reservation system. The entire system was implemented during a one – week period, when the computer system was set up, the software loaded, and the system tested. During that week, a separate training crew worked with all the accounting and front desk personnel to familiarize them with the operation and use of the system. These activities occurred Monday through Saturdays. On Sunday, all personnel were brought in to enter reservations, guest charges, and accounting information into the new system so that it coincided with the current system. On Sunday, evening, after the close of business for the day, the new system was started and used permanently. The old paper reservation file was removed, and the cash registers and bookkeeping machines were replaced with the terminals. The new system became live at midnight on Sunday. There was no old system to fall back on.

iii. Pilot Approach :

When new systems also involve new techniques or drastic changes in organization performance, the pilot approach is often preferred. In this method, a working version of the system is implemented in one part of the organization, such as a single work area or department. The users in this area typically know that they are piloting a new system and that changes can be made to improve the system.

When the system is deemed complete, it is installed throughout the organization, either all at once (direct cutover method) or gradually (phase – in method).

This approach has the advantage of providing a sound proving ground before full implementation. However, if the implementation is not properly handled, users may develop the impression that the system continues to have problems and that it cannot be relied on. For example, they may feel that the difficulties they experienced for two or three weeks may in fact not be gone just because the analysts claim they are.

iv. Phase – In Method :

The phase– in method is used when it is not possible to install a new system throughout an organization all at once. The conversion of files, training of personnel, or arrival of equipment may force the staging of the implementation over a period of time. Ranging from weeks to months. Some users will begin to take advantage of the new system before others.

For example, a medical system aimed at linking 10 or 15 different clinics to hospital may phase in over a year. The work required to convert patient and insurance records on paper to files stored on magnetic disks requires 2 to 3 weeks for each clinic. A week of user training is also required for each clinic. Therefore, the analysts may phase this system in one clinic at a time, allowing 3 to 4 weeks for each conversion. It is conceivable in this system that the full conversion will be phased over one year.

Long phase – in periods create difficulties for analysts, whether the conversions go well or not. If the system is working well, early users will communicate their enthusiasm to other personnel who are waiting for implementation. In fact, enthusiasm may reach such a high level that when a group of users does finally receive the system, there is a letdown. In the clinic example, for instance, the medical staff may exaggerate the time savings that accrue from not having to search for medical records or manually prepare insurance claims, activities that will be handled by the new system. Later, when conversion occurs, the staff finds out that the system does not do the processing instantly. The disappointment is understandable.

On the other hand, if there are problems in the early phases of implementation, word of difficulties will spread also. Then the users may expect difficulties at the time of conversion and react negatively to the smallest mistakes. When systems are phased in, they must work well on the first conversion and all that follow.

10.4 Conversion Plan :

The conversion plan includes a description of all the activities that must occur to implement the new system and put it into operation. It identifies the persons responsible for each activity and includes a timetable indicating when each activity will occur.

During the pre–implementation stages, when the conversion is being planned, analysts should assemble a list of all tasks, including the following :

i. List all files for conversion.

ii. Identify all data required to build new files during conversion.

iii. List all new documents and procedures that go into use during conversion.

iv. Identify all controls to be used during conversion. Establish procedures for cross–checking the old and new systems. Determine how team members will know if something has not been completed properly.

v. Assign responsibility for each activity.

vi. Verify conversion schedules.

The conversion plan should anticipate possible problems and ways to deal with them. Among the most frequently occurring problems are missing documents, mixed data formats between current and new files, errors in data translation, missing data or lost files, and situations that were overlooked during systems development. The conversion manager must guard against the omission of steps in the conversion. A checklist will prevent missed steps. Personnel absences must also be expected and adequate fallback plans specified.

Conversion timing is challenging, since there are so many aspects of the conversion, ranging from the installation of equipment to the ordering of forms and supplies.

a. Operating Plan :

The operating plan is checking of all arrangements. It includes reviewing conversion plans, verifying the delivery of equipment, software, forms, preparing the site and preparing the data and files.

i. Site Preparation : Analysts often work with vendor personnel to outline site – preparation guidelines. Due importance should be paid to electrical using, air conditioning needs, humidity controls, space requirements, etc.

ii. Data and File Preparation : For a new system to begin master files and system files need to be entered into the system before the normal functioning of the system. Master files are generally created manually. The number of records in older system master file should tally with the number of records in new master file.

In case of financial software, the balance brought forward should be checked for validation before implementation of the new system.

❑ **Check Your Progress – 2 :**

1. ________ is the process of changing form the old system to the new one.

a. Conversion b. Transform

c. Interchange d. All of the above.

2. ________ system is a most secure method for conversion from the old system to new one.

a. Direct cutover b. Parallel

c. Pilot approach c. Phase In–Method

3. The ________ method is used when it is not possible to install a new system throughout an organization all at once.

a. Direct cutover b. Parallel

c. Pilot approach d. Phase In-Method

4. When new systems also involve new techniques or drastic changes in organization performance, the ________ is often preferred.

   a. Direct cutover b. Parallel

   c. Pilot approach d. Phase In–Method

10.5 Let Us Sum Up :

For administration putting a new system into operation is usually complicated by the fact that there is an older system already in operation. The analyst has to deal with changing from something familiar to something new and different, while also attending to the mechanics of implementation. Since the concern for simultaneous conversion and implementation is usual. Planning for the formal training of user personnel in the operation of the new system is important. A new method may drastically affect people's lives by changing their work methods, work style and relationships with other employees. One of the most effective ways of dealing with the potential impact of these changes is to provide a well-designed program of training. The training program should:

i. Inform the user about the system in general

ii. Inform the user about specific operation of the system

iii. Give the user some practice in operating the system

iv. Provide opportunity for user feedback.

v. Provide ample opportunity to adjust to the new system.

vi. Provide answers to the queries raised by the employees. Generate a feeling among employees that the new system is "their" system. The quality of training received by the personnel involved with the system in various capacities helps or hinders, and may even prevent, the successful implementation of an information system. Conversion is the process of changing form the old system to the new one. The conversion plan includes a description of all the activities that must occur to implement the new system and put it into operation. It identifies the persons responsible for each activity and includes a timetable indicating when each activity will occur. The operating plan is checking of all arrangements. It includes reviewing conversion plans, verifying the delivery of equipment, software, forms, preparing the site and preparing the data and files.

10.6 Answers for Check Your Progress :

❑ **Check Your Progress 1 :**

1 : c **2 :** d

❑ **Check Your Progress 2 :**

1 : a **2 :** b **3 :** d **4 :** c

10.7 Glossary :

1. **Conversion :** Conversion is the process of changing form the old system to the new one.

2. **Phase In :** The phase- in method is used when it is not possible to install a new system throughout an organization all at once.

3. **Direct Cutover :** The direct cutover method converts from the old to the new system abruptly, sometimes over a weekend or even overnight.

The old system is used until a planned conversion day, when it is replaced by the new system.

4. **Pilot Approach :** When new systems also involve new techniques or drastic changes in organization performance, the pilot approach is often preferred. In this method, a working version of the system is implemented in one part of the organization, such as a single work area or department.

10.8 Assignment :

1. Compare the different conversion methods with each other.
2. Explain the operating plan.
3. To what extent training is important.

10.9 Activities :

- "Conversion is simple". Do you agree? Justify your answer.

10.10 Case Study :

- Suppose you are implementing a "Payroll System" for some organization. How you will train the staff of the organization? Briefly explain your training strategy.

10.11 Further Readings :

1. Analysis and Design and Information System, Rajaraman.
2. Analysis and Design of Information System 2nd Ed. Senn.
3. Introducing Systems Analysis and Design, Lee.

Unit 11

SYSTEM SECURITY, AUDIT AND QUALITY ASSURANCE

UNIT STRUCTURE

11.0 Learning Objectives

11.1 Introduction

11.2 System Audit

11.3 Quality Assurance in SDLC

11.3.1 Specifications for Quality Factors

11.3.2 Software Requirement Specification

11.3.3 Software Design Specification

11.3.4 Software Testing and Its Implementation

11.3.5 Software Support and Maintenance

11.4 Levels of Quality Assurance

11.5 Let Us Sum Up

11.6 Answer for Check Your Progress

11.7 Glossary

11.8 Assignment

11.9 Activity

11.10 Case Study

11.11 Further Readings

11.0 Learning Objectives :

After learning this unit, you will be able to understand :

- Know, what is System Audit ?
- Understand, basic tools of the System Audit.
- Understand, the importance of Quality Assurance
- Know, the levels of Quality Assurance.

11.1 Introduction :

The advancement of information systems and technology offers a vital benefit for businesses. However, it also brings ever–increasing challenges due to the existence of hackers, malware, viruses, cybercrimes, etc. Therefore, frequent and strong follow–up is required via regular information systems security audits. Nevertheless, the scarcity of professionals and the lack of well–suited frameworks in this domain are frequently cited as main barriers to success. The main objective of this paper is to propose a simple and applicable information system security auditing framework to support practitioners in order to minimize the professionals' requirements and simplify managers' involvement in the follow–up.

An information systems security audit (ISSA) is an independent review and examination of system records, activities and related documents. These audits are intended to improve the level of information security, avoid improper information security designs, and optimize the efficiency of the security safeguards and security processes.

11.2 System Audit :

It is an investigation to review the performance of an operational system. The objectives of conducting a system audit are as follows ?

- To compare actual and planned performance.
- To verify that the stated objectives of system are still valid in current environment.
- To evaluate the achievement of stated objectives.
- To ensure the reliability of computer based financial and other information.
- To ensure all records included while processing.
- To ensure protection from frauds.

❖ Audit of Computer System Usage :

Data processing auditors audits the usage of computer system in order to control it. The auditor needs control data which is obtained by computer system itself.

❖ The System Auditor :

The role of auditor begins at the initial stage of system development so that resulting system is secure. It describes an idea of utilization of system that can be recorded which helps in load planning and deciding on hardware and software specifications. It gives an indication of wise use of the computer system and possible misuse of the system.

❖ Audit Trail :

An audit trail or audit log is a security record which is comprised of who has accessed a computer system and what operations are performed during a given period of time. Audit trials are used to do detailed tracing of how data on the system has changed.

It provides documentary evidence of various control techniques that a transaction is subject to during its processing. Audit trials do not exist independently. They are carried out as a part of accounting for recovering lost transactions.

❖ Audit Methods :

Auditing can be done in two different ways

Auditing around the Computer

- Take sample inputs and manually apply processing rules.
- Compare outputs with computer outputs.

Auditing through the Computer

- Establish audit trial which allows examining selected intermediate results.
- Control totals provide intermediate checks.

❖ **Audit Considerations :**

Audit considerations examine the results of the analysis by using both the narratives and models to identify the problems caused due to misplaced functions, split processes or functions, broken data flows, missing data, redundant or incomplete processing, and non–addressed automation opportunities.

The activities under this phase are as follows –

- Identification of the current environment problems
- Identification of problem causes
- Identification of alternative solutions
- Evaluation and feasibility analysis of each solution
- Selection and recommendation of most practical and appropriate solution
- Project cost estimation and cost benefit analysis

❖ **Security :**

System security refers to protecting the system from theft, unauthorized access and modifications, and accidental or unintentional damage. In computerized systems, security involves protecting all the parts of computer system which includes data, software, and hardware. Systems security includes system privacy and system integrity.

- **System privacy** deals with protecting individuals' systems from being accessed and used without the permission/knowledge of the concerned individuals.
- **System integrity** is concerned with the quality and reliability of raw as well as processed data in the system.

❖ **Control Measures :**

There are varieties of control measures which can be broadly classified as follows –

❖ **Backup :**

- Regular backup of databases daily/weekly depending on the time criticality and size.
- Incremental back up at shorter intervals.
- Backup copies kept in safe remote location particularly necessary for disaster recovery.
- Duplicate systems run and all transactions mirrored if it is a very critical system and cannot tolerate any disruption before storing in disk.

❖ **Physical Access Control to Facilities :**

- Physical locks and Biometric authentication. For example, finger print
- ID cards or entry passes being checked by security staff.
- Identification of all persons who read or modify data and logging it in a file.

❖ **Using Logical or Software Control :**

- Password system.
- Encrypting sensitive data/programs.

- Training employees on data care/handling and security.
- Antivirus software and Firewall protection while connected to internet.

❖ **Risk Analysis :**

A risk is the possibility of losing something of value. Risk analysis starts with planning for secure system by identifying the vulnerability of system and impact of this. The plan is then made to manage the risk and cope with disaster. It is done to accesses the probability of possible disaster and their cost.

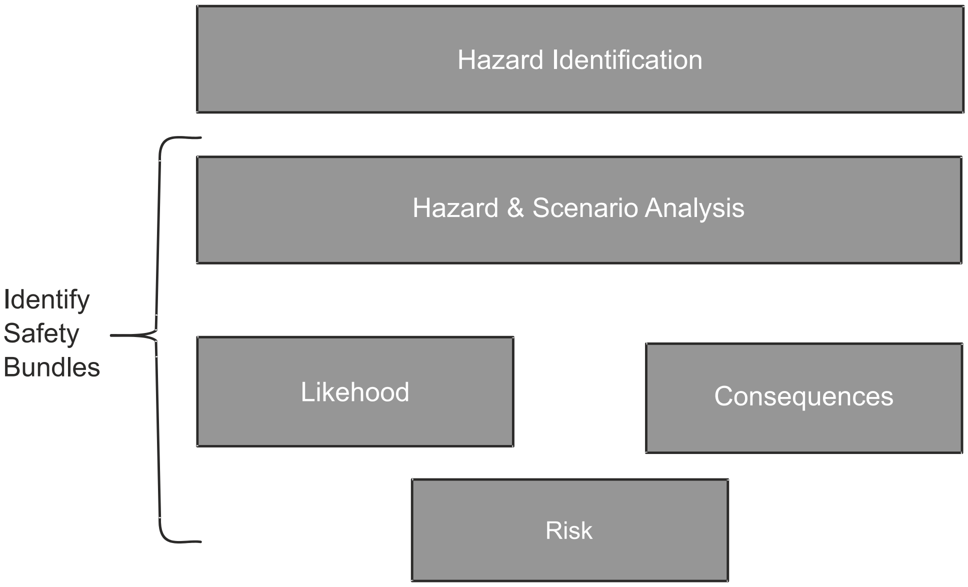
Risk analysis is teamwork of experts with different backgrounds like chemicals, human error, and process equipment.

The following steps are to be followed while conducting risk analysis –

- Identification of all the components of computer system.
- Identification of all the threats and hazards that each of the components faces.
- Quantify risks i.e. assessment of loss in the case threats become reality.

❖ **Risk Analysis – Main Steps :**

As the risks or threats are changing and the potential loss are also changing, management of risk should be performed on periodic basis by senior managers.



Risk management is a continuous process and it involves the following steps –

- Identification of security measures.
- Calculation of the cost of implementation of security measures.
- Comparison of the cost of security measures with the loss and probability of threats.
- Selection and implementation of security measures.
- Review of the implementation of security measures.

❑ Check Your Progress – 1 :

1. An ________ or audit log is a security record which is comprised of who has accessed a computer system and what operations are performed during a given period of time.

   a. Privacy b. Integrity c. Audit trail d. Audit method

2. ________ deals with protecting individuals' systems from being accessed and used without the permission/knowledge of the concerned individuals.

   a. Privacy b. Integrity c. Audit trail d. Audit method

3. From the given below ________ is/are considered to be a security control measure(s).

   a. Installing Antivirus software b. Configuring Firewall

   c. Regular backup of data d. All of the above

11.3 Quality Assurance In SDLC :

The software life cycle includes various stages of the development and its each stage has the goal of quality assurance. The following steps needs to be taken in this regard:

11.3.1 Specifications for Quality Factors :

The goal of this particular stage is to represent various factors mainly responsible for quality of the system which we are proposing. These are as follows :

1. **Reliability :** The degree to which the system performs its intended function over a time.
2. **Efficiency :** Resources of the computer required by a program to perform particular functionality of the software.
3. **Correctness :** The extends to which a software meets its specification from users' perspective.
4. **Usability :** It deals with understandability and operativity of the system.
5. **Maintainability :** It deal with, how easy to detect and correct the software program.
6. **Testability :** The efforts required to test a software program to ensure its correct performance.
7. **Accuracy :** The required precision in taking Input, editing, processing data and generating output.
8. **Portability :** The ease transportation of a software program from one hardware configuration environment to another.
9. **Tolerance of errors :** Detection and correction of errors versus avoidance of errors.
10. **Expandability :** Ease to expand the database and software product.
11. **Audit and Access control :** Control of access to the system and the extend to which that access can be audited.

11.3.2 Software Requirement Specifications :

The goal of a quality assurance in this stage is to generate the SRS (System Requirement Specification) document that helps in providing technical details and guideline to develop a software.

11.3.3 Software Design Specifications :

In this particular stage, the software product design document defines the complete architecture of the software product that provides the features and functions, which are mentioned in the SRS document.

11.3.4 Software Testing and its Implementation :

The overall goal of the quality assurance of the testing phase is to ensure that accuracy and completeness of the system and minimizing the process of retesting. During the implementation phase, the goal is to provide a logical ordering for the creation of the various modules and in turn the creation of the software.

11.3.5 Software support and Maintenance :

This quality phase of quality assurance provides the necessary development for the software system to continue and to comply with the original specifications mentioned by the user. The goal of the quality Esurance is to a procedure for correcting bugs, errors and enhancing the software product.

11.4 Levels of Quality Assurance :

Most of the time Analysts provides three levels of quality assurance. These are : [1] Testing [2] Verification and [3] Validation assurance.

11.4.1 Software Support and Maintenance :

Testing of the System is time consuming and costly process. The common view of the testing held by the users is that it is performed to prove that the software product is free from the errors. But it is quite difficult as the analyst can't prove that the software product is totally free from all types of errors.

To prove that the software product is free from most practical and useful approach, which an analyst have is Testing. As we have seen that, the testing is the process of executing the software program with the intention of finding errors and bugs which fails the software product.

11.4.2 Verification and Validation :

Similar to the testing, verification is also intended to finding of errors from the software product. Verification is done by executing a program in simulated environment. It refers of the process of using software product in a live environment to detect the errors.

When any commercial software product is developed with main intention of distributing them to the dealers for the purpose of sale, they first perform verification, which also known as alpha testing. The feedback from the phase of verification may bring certain changes and corrections in the software to deal with errors and failures that are uncover.

After the completion of the alpha testing, the software product will be tested in the user's environment (at user's site). It is known as beta testing or validation. The beta test users, install and use the software product in their day–to–day activities. The record and process live transactions and produce normal system output.

Validation may continue for few months. During the period of validation of the system, it is possible that failure may occur and the software needs to be changed.

11.4.3 Certification :

The final level of the software product quality assurance is to certify that the particular software package developed follows the standards. By looking the growing demand for purchasing ready–to–use software product, importance of the certification has increased. A software package that is certified goes via a team of the computer software specialist who test, review and determine that the software product fulfil all user's requirements and claim of the vender. Certification is if and only if the software completes (pass) all necessary tests successfully. If particular software get certification, it does not mean that it is the best product, but it means that it meets the user's and vendor's requirements.

❑ Check Your Progress – 2 :

1. Verification is also known as ________ testing.

   a. Unit   b. Integration   c. Alpha   d. Beta

2. ________ is also known as beta–testing.

   a. Verification   b. Validation

   c. Checking   d. None of the above

3. From the given below options, which option is not correct about the certificate issuance to the particular software.

   a. It is the best product

   b. It is free from errors

   c. It fulfil all requirements of the user

   d. It has passed all tests successfully

11.5 Let Us Sum Up :

In this chapter we have focused on Security and Quality concerns of the software product. We have seen that how security is an important factor, and what measures needs to be taken to protect the System from unauthorised access. We have discussed how Audit trail logs the system and different methods of Audit.

We have also discussed about how the quality of the software is important and what standards need to be followed to produce a quality software product.

11.6 Answers for Check Your Progress :

❑ Check Your Progress 1 :

1 : c   **2 :** a   **3 :** d

❑ Check Your Progress 2 :

1 : c   **2 :** b   **3 :** a

11.7 Glossary :

1. SRS : It stands for System Requirement Specification. It is a document produced during the System Analysis phase of SDLC, which includes all functional requirements and strategies to build and test the quality software product.

2. Audit Trail : An audit trail or audit log is a security record which is comprised of who has accessed a computer system and what operations are performed during a given period of time.

3. **Systems Privacy :** System privacy deals with protecting individuals' systems from being accessed and used without the permission/knowledge of the concerned individuals.

4. **System Integrity :** System integrity is concerned with the quality and reliability of raw as well as processed data in the system.

11.8 Assignment :

1. Write a short note on security control measures.
2. Write a brief note on quality assurance in SDLC ?
3. Explain the levels of the quality assurance in brief.

11.9 Activity :

- Design objectives for : Reliability and Maintenance.

11.10 Case Study :

- Design a format for Audit trail.

11.11 Further Readings :

1. Analysis and Design and Information System, Rajaraman.
2. Analysis and Design of Information System 2nd Ed. Senn.
3. Introducing Systems Analysis and Design, Lee.

BLOCK SUMMARY :

While studying this block, the user will achieve knowledge and understanding about various design specifications with idea of file management. The block explains different rules and criteria's about data files with certain examples that will help the user to grab the concept.

The block explains about different software file and software programmes. The students or programmers will get benefit while reading this block as it gives shortcuts and related examples that will clear all doubts related to software testing procedures and trainings.

BLOCK ASSIGNMENT :

❖ **Short Questions :**

1. What are addressing and compaction techniques ?
2. Explain software testing ?
3. What are the main topics of training ?
4. What are the objectives considered during database design ?
5. Explain the key software maintenance issues ?

❖ **Long Questions :**

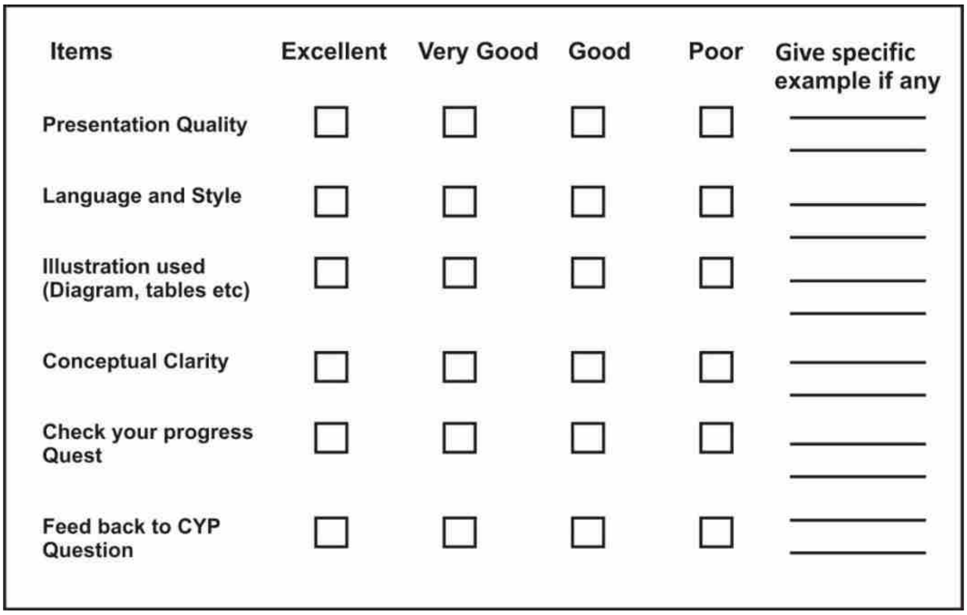
1. What are the tasks of system maintenance ?
2. Explain physical and logical database design ?
3. What are the approaches to software testing ?

❖ **Enrolment No. :** ▭

1. How many hours did you need for studying the units ?

Unit No.	8	9	10	11
No. of Hrs.				

2. Please give your reactions to the following items based on your reading of the block :



Items	Excellent	Very Good	Good	Poor	Give specific example if any
Presentation Quality	□	□	□	□	____ ____
Language and Style	□	□	□	□	____ ____
Illustration used (Diagram, tables etc)	□	□	□	□	____ ____
Conceptual Clarity	□	□	□	□	____ ____
Check your progress Quest	□	□	□	□	____ ____
Feed back to CYP Question	□	□	□	□	____ ____

3. Any other Comments

....

....

....

....

....

....

....

....

Dr. Babasaheb Ambedkar Open University Ahmedabad

BCAR-205/ DCAR-205

System Analysis and Design

BLOCK 4 : STRUCTURED SYSTEM DESIGN AND DATA ORIENTED SYSTEM DESIGN

UNIT 12 STRUCTURED SYSTEMS ANALYSIS AND DESIGN

UNIT 13 DATA ORIENTED SYSTEMS DESIGN

UNIT 14 OBJECT-ORIENTED ANALYSIS AND DESIGN

STRUCTURED SYSTEM DESIGN AND DATA ORIENTED SYSTEM DESIGN

Block Introduction :

Analysis is a detailed study of the various operations performed by a system and their relationships within and outside of the system i.e. it includes finding out in more detail what the system problems are and what the new changes the user wants.

In this block, we will learn and study about basic of structured English and its applications. The block will be detailed about decision table and its structure. The student will be given with the knowledge about different rules and criteria's of ER model.

In this block, we will learn and study about features of Entity–Relationship (ER) Model with detailed on Normalization. The block will cover topics related to conceptual modelling, mapping cardinalities, relationships among entities and necessity of normalization. The student will be given with the knowledge about different rules and criteria's of normalization.

Apart from the Structured Analysis, many Analysts prefers Object–Oriented Design as they are more convenient to program as most of the programming languages available today are Object–Oriented. To do the Object–Oriented analysis, analyst must know UML (Unified Modeling Language) diagrams. Use–Case diagram, Sequence diagram, Class diagram etc. are UML diagram. In the last unit of these block, we will learn how can we do object–oriented analysis using these diagrams.

Block Objectives :

After learning this block, you will be able to understand :

- The concept of design
- Idea about structured design
- Basic of Structured English
- Concept of Entity–Relationship Model
- Basic of entity–relationship diagram
- Knowledge about Strong and Weak Entity
- Understanding of Normalizing Relations
- Detailed about normal forms
- Idea about Relational Data Base Design
- Learn about Object–Oriented Analysis
- Knowledge about UML diagrams
- Understand how to draw Use–Case, Sequence and Class diagrams.

Block Structure :

Unit 12 : Structured Systems Analysis and Design

Unit 13 : Data Oriented Systems Design

Unit 14 : Object-oriented Analysis and Design

Unit 12

STRUCTURED SYSTEMS ANALYSIS AND DESIGN

UNIT STRUCTURE

12.0 Learning Objectives

12.1 Introduction

12.2 Procedure Specifications in Structured English

12.2.1 Examples and Cases

12.3 Decision Tables For Complex Logical Specifications

12.4 Specification Oriented Design Vs Procedure Oriented Design

12.5 Let Us Sum Up

12.6 Answers for Check Your Progress

12.7 Glossary

12.8 Assignment

12.9 Activities

12.10 Case Study

12.11 Further Readings

12.0 Learning Objectives :

After learning this unit, you will be able to understand :

- Feature of analysis
- Layout of design structure
- Study of structured English

12.1 Introduction :

System analysis and design for information system theory, which emphasizes a close look at all parts of a system. Too often, analysis focuses on only one component and overlooks other equally important components. The idea of a system becomes most practical and necessary in conceptualizing the interrelationship and integration of operations, especially when using components. Thus, a system is a way of thinking about organizations and their problems. It also involves set techniques that help in solving problems.

The details of the process are developed for ancient processes as well as for higher level processes that are located on data flow diagram known as minispecs. It is found that the process specifications are not developed for :

- Input and/or output processes.
- Simple data validation process.
- Process for code already exists.

Structured English basically depends on structured logic as well as plain statements that uses terms such as add, multiply, move etc. It is the best way for finding the system in case of complex structured decisions.

❖ **Advantages of Structured English :**

- Simplifies logic and relationships in human languages
- Simple communication tool which is easy to teach and understand

12.2 Procedure Specifications in Structured English :

Structured English is basically presentation of English with the help of various structures. Further it is termed as program design language and problem statement language. It serves as a balance between formal programming language and casual informality of English language. Consider a sentence in structured English which represents an equation :

X = (A*B)/(C+10)

It contains an imperative sentence having a verbs and objects. In this sentence, there is no semicolon which can end a programming statement in various different programming languages which can stop with ("."), as desired. We found that the sentences which show computations can be prefixed with verbs COMPUTE, ADD, SET as :

COMPUTE X = (A*B)/(C+10)

So as per structured English computations :

SET TAX–RATE TO 9

ADD3TOX

MULTIPLY UNIT – PRICE BY QUANTITY

DIVIDE CURRENT – ASSETS BY CURRENT – LIABLITIES.

It is found that the verbs should be selected from smallest set of action oriented verbs like :

GET (or ACCEPT or READ)

PUT (or DISPLAY or WRITE)

FIND (or SEARCH or LOCATE)

ADD

SUBSTRACT

MULTIPLY

DIVIDE

COMPUTE

DELETE

FIND

VALIDATE

MOVE

REPLACE

SET

SORT

It is found that many organizations contain 40 to 50 verbs which is enough to find any policy in a process. It is analyzed those objects therefore comprises of data elements which have been explained by data dictionary. Also, it is seen that local terms are terms which explains inside a process specification which are known, relevant as well as meaningful in the process. As seen the example of such term is an intermediate calculation that shows final output of process.

It is found that local terms are explained inside a process specification where they occur but are not present in data dictionary. Such types are calculated from terms which are previously present in data dictionary as of which addition of local term is done.

In structured English process details, we see that a series of deliveries is stored in DELIVERY store which will calculate the daily total as :

```
Daily total = 0
DO WHILE as there are many DELIVERIES carrying Invoice date = today's date
```

It seems that IF–THEN–ELSE statement is framed which shows alternative sentences that are taken on binary decision. So, IF–THEN–ELSE statement can take two forms :

```
IF condition – 1
Sentence – 1
ENDIF
Or
IF condition –1
Sentence – 1
ELSE
Sentence – 2
ENDIF
```

Here we see that a CASE construct explains alternative sentences that will be carried out on multivalued decision. The CASE construct takes the general form :

```
DO CASE
CASE variable = value – 1
Sentence – 1
CASE variable = value–n
Sentence–n
OTHERWISE
Sentence–n+1
ENDCASE
```

So in such case a systems analyst write this as :

```
DO CASE
CASE customer–age < 13
Set billing–rate to child–rate
CASE customer–age ≥ 13 and customer–age < 20
Set billing–rate to teenage–rate
CASE customer–age > 20 and customer–age < 65
Set billing–rate to adult–rate
```

Further, in an example, the following portion of structured English process specification can be written as :

```
DO CASE
CASE state = "NY"
Set sales tax to 0.0825
CASE state = "NJ"
Set sales tax to 0.07
CASE state = "CA"
Set sales tax to 0.05
OTHERWISE
Set sales tax to 0
END CASE
```

12.2.1 Examples and Cases :

Example 1

Number	1
Name	Add Customer Order
Description	Key and add the Customer Order

The order to be altered for required information.

Customer as well as Item master files are updated.

❖ **Input Data Flow :**

Customer Order Form from the Customer

Customer Record from data store D1, Customer Master File

Item Record from data store D2, Item Master File

❖ **Output Data Flow :**

Pending Order to data store D3, Order File

Backordered Item Record to the Inventory Control Department Updated Customer and Item records

Type of process Online

❑ **Check Your Progress – 1 :**

1. Structured English is representation of English as :

a. data b. language c. structures d. all

12.3 Decision Tables for Complex Logical Specifications :

A decision table represents the condition and action to be taken against that condition in the form of table. The table consists of two parts : stub and entry. The stub part is divided into an upper quadrant called the condition stub and a lower quadrant called the action stub.

Header

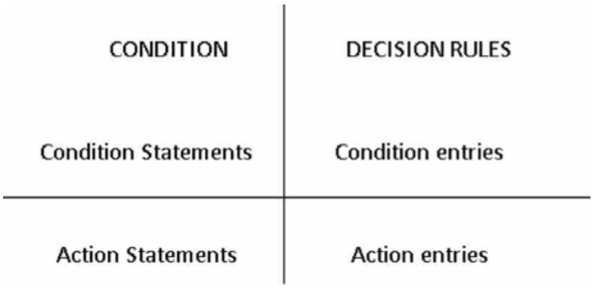
Condition stub	Condition entry
Action stub	Action entry

Characteristics of Decision Table

It is made up of four sections as follows :

a. Condition statement b. Condition entries

c. Action statement d. Action entries



CONDITION	DECISION RULES
Condition Statements	Condition entries
Action Statements	Action entries

Fig 12.1 : General Forms for Decision Tables

1. Conditional statements identify the relevant condition. Condition entities are the value applied.
2. Action statement consists of all steps when certain condition occurs
3. Right side column of the table is taken as linking conditions and action form decision rules

❖ **Primary uses of Decision Table :**

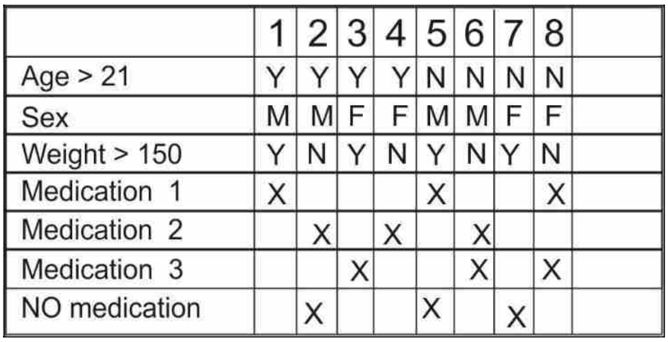
A decision table is useful when :

- Information has to be represented in the form of problem and action to be taken against it.
- All possible combinations of problem and the actions to be taken
- Information has to be presented in tabular format

The entry part is also divided into an upper quadrant called the conditions entry and a lower quadrant called the action entry. The condition stub part contains conditions in the form of questions; answer for which in form of Yes/ No are given in condition entry part.

Similarly, condition action part contains the statements of the actions to be taken, which action has to be taken against each of the condition, it is shown in 'X' mark in entry block.

Many times, it is seen that there are certain situations where neither structured English nor pre or post conditions is fit for writing certain process features and data specifications. This will happen when the process generates certain output or can act with certain actions as per complicated decisions. When certain decisions are acted on variable input data elements and when variables carry many different values, under such situation, there exists logic which is explained by structured English which can be either pre or post arrangement which becomes complicated for user to understand. Under such circumstances, the decision table will be mostly preferred.



	1	2	3	4	5	6	7	8	
Age > 21	Y	Y	Y	Y	N	N	N	N	
Sex	M	M	F	F	M	M	F	F	
Weight > 150	Y	N	Y	N	Y	N	Y	N	
Medication 1	X				X			X	
Medication 2		X		X		X			
Medication 3			X			X		X	
NO medication		X			X		X		

Fig 12.2 : Variables in Decision Tables

From the figure 12.2, a decision table is formed by showing all required variables and actions which located on left of decision table. In the table, the variables and actions are distributed by strong horizontal line where variable is logical variable.

It is found that many uses of applications makes easy to express variables as binary, whereas decision tables are being constructed from multivalued variables. It is found that on considering decision table having customer–age as variable having values less than 10, between 10 and 30 and greater than 30.

We see that for every required combination of values of variables listed in separate column called as rule. Simply the rule will show action which is carried for definite mixture of values of variables. For this, mainly single action is required for every rule or nature of system prevailed for such situation.

If there exists N variables having 0 or 1 values, so there exists 2N distinct rules; having 3 conditions, 8 rules. If there exists7 conditions, then there occur 128 rules. On assuming YES, it shows T as true which is binary 0, and on assuming NO, it means False, which is binary 1, so we see that a sequence of combination exists as :

000, 001, 010, 011, 100, 101

Till it will at 2^N combinations.

So, to have a total number of rules exists in a decision table, we need to multiply values that will contain variable 1 which can take value as variable 2. So, in an application we have :

var 1 having three values

var 2 having five values

var 3 having four values

So it will be 3 x 5 x 4 = 60 distinct rules.

❑ Check Your Progress – 2 :

1. The decision table comprises of :

   a. Condition and Action statements

   b. Condition and Action entries

   c. Both a and b

   d. Neither a nor b

12.4 Specification Oriented Design Vs Procedure Oriented Design :

The specification–oriented design contains specifications in shape of data. In procedure–oriented design we have both procedures and data to design. We discuss a category of design methods that concentrate on procedures. In procedure–oriented design, the whole system is divided into a set of procedures or modules.

A common tool for illustrating the relations between modules in procedure–oriented design is a structure chart.

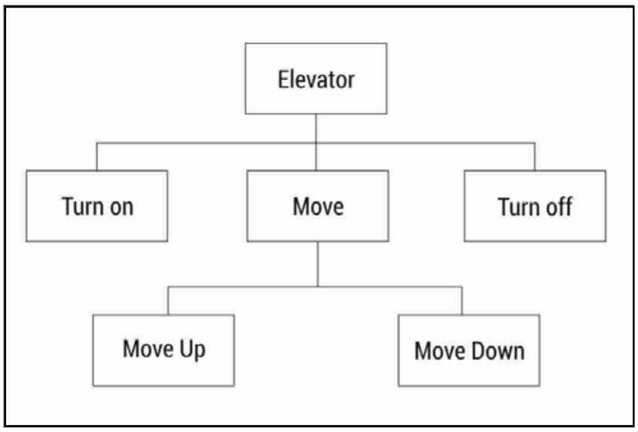


Fig. 12.3 : Structure Chart

❑ Check Your Progress – 4 :

1. In procedure-oriented design, the whole system is divided into ________.

   a. set of procedures b. set of modules

   c. set of procedures and modules d. none of these

2. If there exists N variables having 0 or 1 values, there exists ________ distinct rules in the decision table.

   a. 2*N b. 2N c. 2N+1 d. 2N–1

3. If there exist 7 variables having 0 or 1 values, there exists ________ distinct rules in the decision table.

   a. 7 b. 14 c. 49 d. 128

12.5 Let Us Sum Up :

In this unit, we have learned :

- System analysis and design is a type of information system theory that will emphasize on closer look for all parts of system.
- Structured English basically depends on structured logic as well as plain statements that uses terms such as add, multiply, move etc.
- Structured English is basically presentation of English with the help of various structures.
- A decision table represents the condition and action to be taken against that condition in the form of table.

12.6 Answers for Check Your Progress :

- ❑ **Check Your Progress 1 :**

  1 : c

- ❑ **Check Your Progress 2 :**

  1 : c

- ❑ **Check Your Progress 3 :**

  1 : c **2 :** b **3 :** d

12.7 Glossary :

1. **Field –** It is an area inside the records which is kept reserved for particular piece of data.
2. **Record –** A record is the collection of values for all the fields pertaining to one entity.
3. **Conceptual Model –** It is a type of model which is prepared from work of art of concepts.
4. **Entity–Relationship Diagram –** It is a chart which signifies correlation involving database entities.
5. **Entity Set –** It is a containing a primary key is named as Strong entity set.

12.8 Assignment :

1. Define decision table.

12.9 Activities :

- Write steps on designing of procedure–oriented design.

12.10 Case Study :

- Discuss the applications of structured English and discuss.

12.11 Further Readings :

1. Tom DeMarco, Structured Analysis and Systems Specification. Englewood Cliffs, N. J. : Prentice–Hall, 1979.
2. Chris Gane and Trish Sarson, Structured Systems Analysis : Tools and Techniques. Englewood Cliffs, N. J. : Prentice–Hall, 1978.
3. Edward Yourdon, Techniques of Program Structure and Design. Englewood Cliffs, N. J. : Prentice–Hall, 1976.
4. James Martin and Carma McClure, Diagramming Techniques for Software Engineering. Englewood Cliffs, N. J. : Prentice–Hall, 1985.

Unit 13 DATA ORIENTED SYSTEMS DESIGN

UNIT STRUCTURE

13.0 Learning Objectives

13.1 Introduction

13.2 Entity Relationship Model

13.2.1 E–R Diagrams

13.3 Relationships Cardinality and Participation

13.4 Normalizing Relations

13.4.1 Various Normal Forms and their Need

13.5 Examples of Relational Data Base Design

13.6 Let Us Sum Up

13.7 Answers for Check Your Progress

13.8 Glossary

13.9 Assignment

13.10 Activities

13.11 Case Study

13.12 Further Readings

13.0 Learning Objectives :

After learning this unit, you will be able to understand :

- Basic of data
- Idea about Entity
- Knowledge about cardinalities
- Understanding of Relationships connectivity

13.1 Introduction :

The data is prearranged in several etiquettes so that the information enclosed inside the database can be easily recovered. Some of the easy databases that you know are things such as phone books or rolodexes. At the same time, as data processing has turn out to be more complicated, so there are certain methods for collecting, storing as well as retrieving information. Databases have turn out to be the foundation stone for an irresistible amount of the computing background in survival.

The conceptual modelling is separated into three decades that is 1970ies, 1980ies as well as 1990ies. In 1970ies database design is extremely significant particularly as per Peter Chen ?s paper "Entity–Relationship Model that is Unified View of Data" which is highlighted in the area of data modelling as well as database design. Readily available discussions endeavour to expand high level data definition languages designed for defining conceptual schemas that can be Conceptual Schema Language (CSL).In 1980s; conveniently there are a number of approaches to expand Chen ?s Entity Relationship Model. During this period, is REMORA by Roland Colette ? In fact information systems as well as design of IS are attractive subjects. Next to start of 1990ies, there are many queries such as schema integration, schema transformation as well as quality measures meant for conceptual schemas in region of database design. However this moment in time is also predisposed by object–oriented modelling methods in addition to languages in software engineering.

13.2 Entity Relationship Model :

Entity–Relationship (ER) Model is a type of database model structure that is based on conception of real world things as well as relations surrounded by them. At the same time, the preparation of actual world situation hooked on the database model, ER Model generates entity set, relationship set, common characteristic as well as constraints. It is most excellent model applied for theoretical plan of a database. It is based on ?

- Entities and their attributes.
- Relationships among entities.

Data Oriented

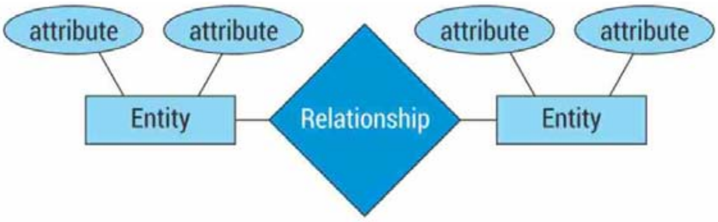


Fig. 13.1 : ER Model

❖ **Entity :**

We can say that the unit of ER Model is an actual entity example that carries properties called attribute. Every characteristic is definite by set of principles called domain. For instance, while considering school database, a student is considered as an entity having various characteristics such as name, age, class, etc.

❖ **Relationship :**

The rational relationship amongst entities is called correlation. Relationships are mapped by means of entities in a variety of behaviour. Mapping cardinalities characterize the amount of relationship among two entities. In this, the mapping cardinalities are :

- one to one
- one to many
- many to one
- many to many

13.2.1 E–R Diagrams :

An entity–relationship diagram, or ERD, is a type of chart which optically signifies the correlation involving database entities. The chart is an organizational data storage needs that carries three major mechanisms :

- Entities
- Attributes
- Relationships

This model chart will describe about the hypothetical dream of a database. This type of chart model will be comfortable in the region of real–world entities in addition to relationship surrounded by them. At view stage, ER model is well thought–out as good alternative for designing databases.

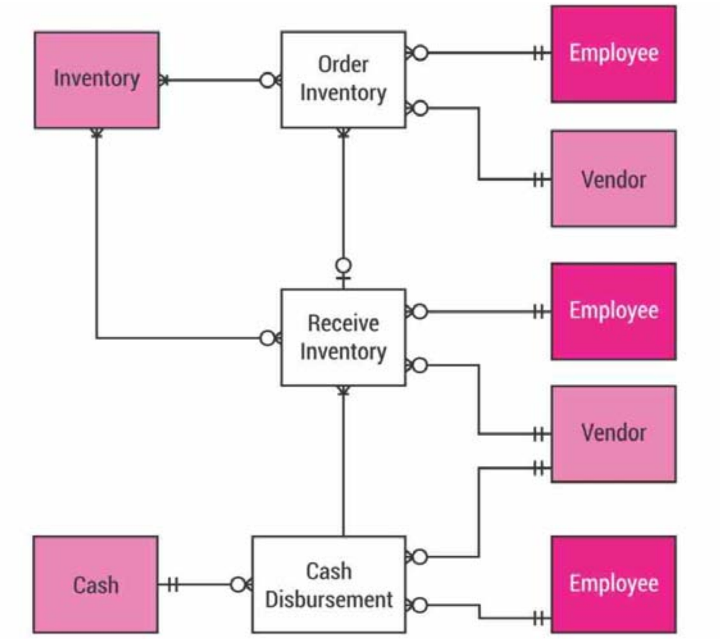


Fig. 13.2 : ERD Model Example

Entity–Relationship model works on the phenomena of actual–world entities as well as relationships that appear among them. At similar instance, preparing such actual–world condition in database model will led to formation of entity set, relationship set, general attributes as well as constraints. It was established for theoretical design of database which relies on :

- Entities and their attributes.
- Relationships among entities.

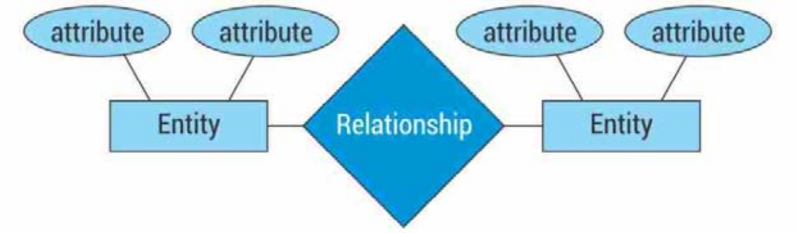


Fig. 13.3 : ER model

❖ **Symbols of Entities :**

It seems that an entity is an actual–world entity that can be either animate or inanimate, but is identifiable. For instance, if we consider a school database, we see that students, teachers, classes as well as courses serve as an entity. It is seen that every such entities carries various attributes or properties that give them their identity.

Moreover, it is found that an entity set is mixture of similar types of entities. An entity set may perhaps hold entities with characteristic sharing comparable values. For illustration, a Students set may perhaps enclose each and every students of a school, similarly as Teachers set that perhaps might carry all teachers of school with all faculties. Entity sets need not be disjoint.

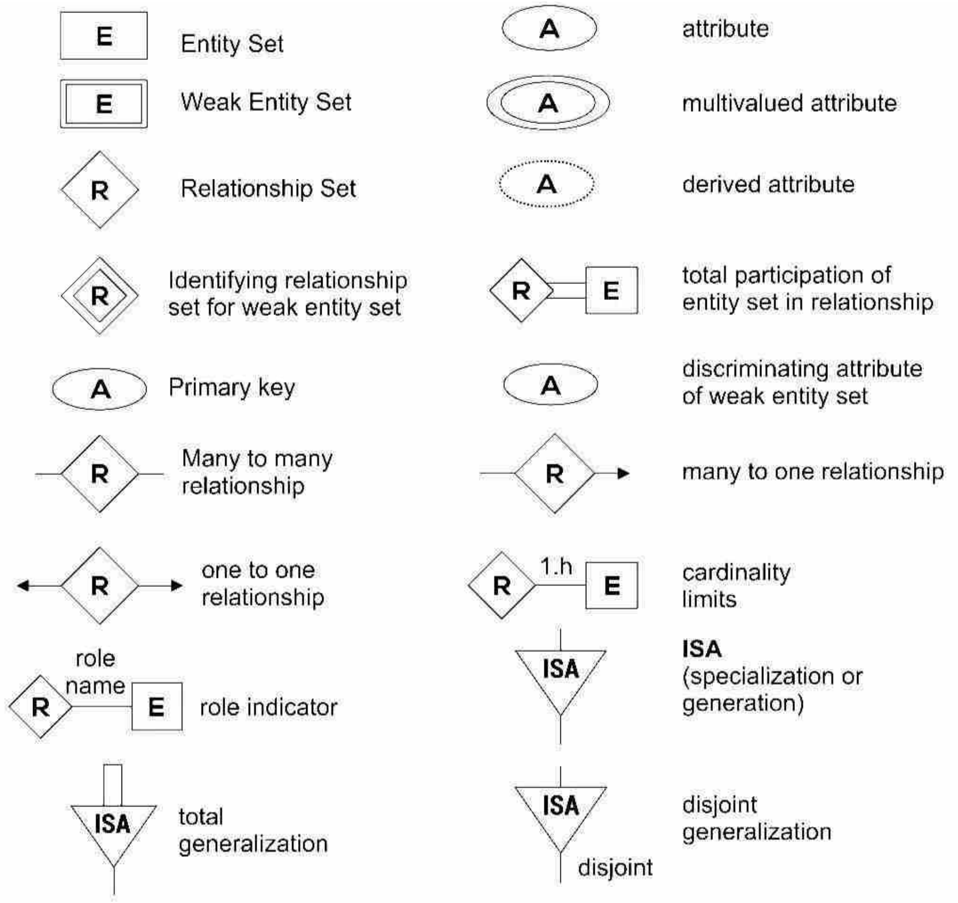


Fig. 13.4 : ER Model Symbol

❖ **Types of Entities :**

Entity Relationship (ER) model carries various types of entities.

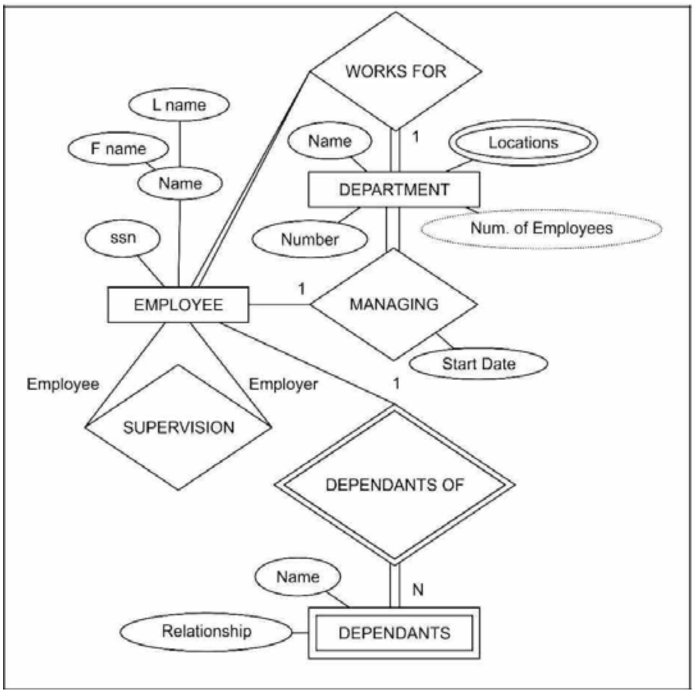


Fig 13.5 : Entity in ER Model

The continuation of an entity possibly will depend on the continuation of one or more previous entities, like existence dependent. It is that an entity whose continuation is not bound on every previous entity is called as not existence dependent entity. Entities which are as per their characteristics are marked as :

- Strong Entities
- Weak Entities

❖ Strong Entity and Weak Entity :

The entity set which does not encompass enough attributes to figure a primary key is termed as Weak entity set. The entity set which carries a primary key is named as Strong entity set.

It is seen that a weak entity is reality dependent. It has a survival of a weak entity that depends on the continuation of a classifying entity set. In this the partial key is applied to make out previous attributes of a weak entity set. The primary key of a weak entity set is created by primary key of recognizing entity set as well as the discriminator of weak entity set.

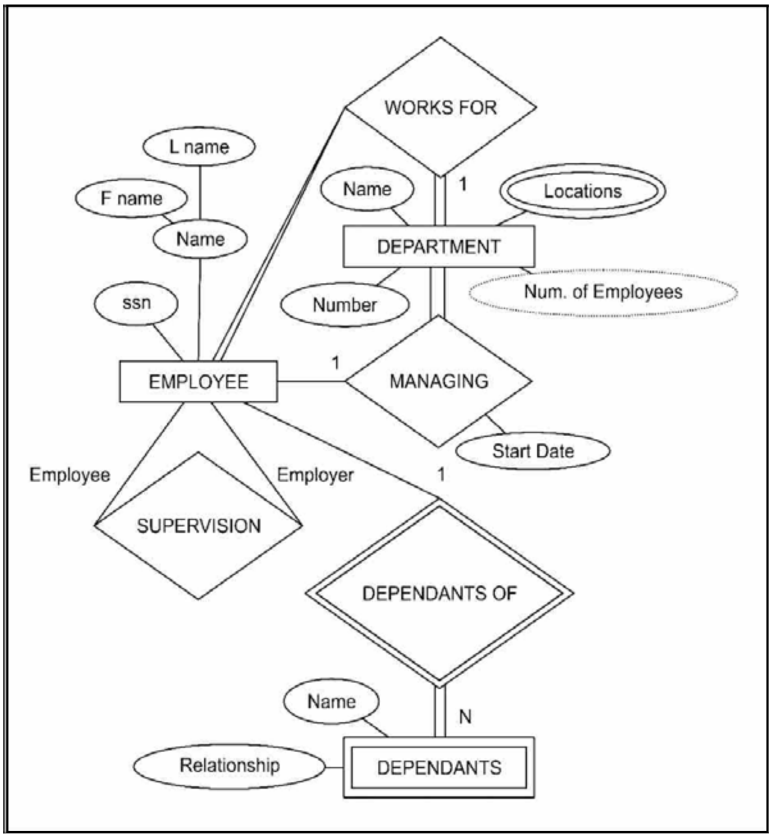


Fig. 13.6 : Strong Entity and Weak Entity

From the diagram 13.6, it is clear that the underlined is the discriminator of weak entity set that carries dashed line.

❑ Check Your Progress – 1 :

1. Which of the following are the properties of entities ?

   a. Groups   b. Table   c. Attributes   d. Switchboards

2. E–R model uses this symbol to represent weak entity set ?

   a. Dotted rectangle   b. Diamond

   c. Doubly outlined rectangle   d. None of these

3. Conceptual design

   a. is a documentation technique.

   b. needs data volume and processing frequencies to determine the size of the database.

   c. involves modelling independent of the DBMS.

   d. is designing the relational model.

13.3 Relationships Cardinality and Participation :

The cardinality can be seen as a relationship that exists for a number of occurrences of entity B so as to associate with entity A. There is least cardinality as well as highest cardinality for each relationship, through an indefinite greatest cardinality being given away as N. Cardinality limits are more often than not derived from the organisations policies or external constraints.

Consider a case at the college where every Teacher is able to teach an indefinite utmost number of subjects as lengthy as his/her weekly hours carry out that will not go beyond 24. Teachers possibly will teach 0 subjects but since they are occupied in non–teaching projects, so the cardinality limits for Teacher can be (O,N) as shown in fig 13.7.

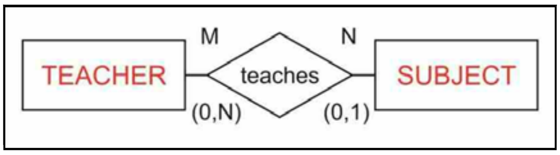


Fig. 13.7 Cardinality Limits for Teacher

As per the college policies, each Subject to be taught by only single teacher, further possibly, Subjects might not have allotted to a teacher, so in this case the cardinality limits for SUBJECT will be (0,1).

The connectivity of a relationship is its classification. It may be a one to one (1:1), one to many (1:M) or many to many (M:N) relationship. Relationships connectivity is represented by a 1, M or N next to the related entity.

❖ **One to one (1 :1)**

In this, a Principal manages one Department, where every Department is managed by single Principal Teacher as described in fig 13.8

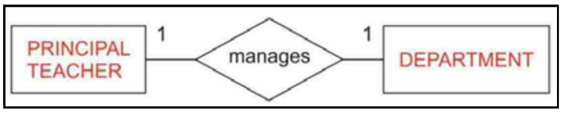


Fig. 13.8 : 1:1 Cardinality of Principal Teacher

❖ **One to Many (1:M)**

In this, a Subject can be given many times, but each contribution belongs to one Subject as shown in fig 13.9 of one to many relationships

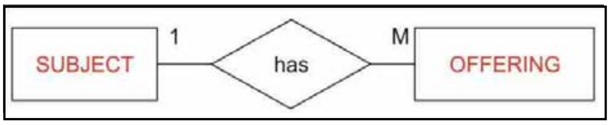


Fig 13.9 : One to Many Relationships

- ❖ **Many to Many (M:N)**

In this case, a Teacher can teach several different Subjects, but every Subject may be taught by several Teachers. This is explained in fig 13.10 in many to many relationships

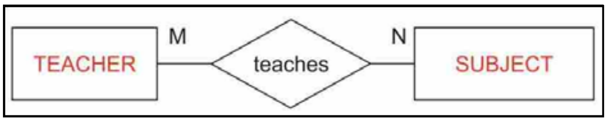


Fig 13.10 : Many to Many Relationship

- ❖ **Many to One :**

An entity in A is related to at most one entity in B, but an entity in B is related to any number of entities in A.

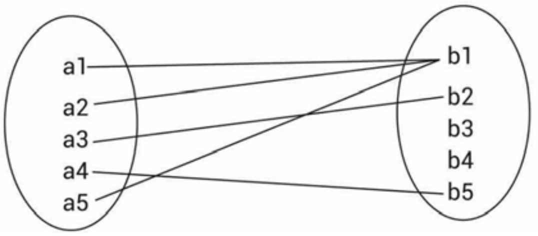


Fig 13.11 : Many to One Relationship

- ❑ **Check Your Progress – 2 :**

1. Which cardinality is explain from, in a School, a teacher can teach many different Subjects, but every subject can be taught by many teachers

a. Many to many　　b. One to many

c. Many to one　　d. One to one

13.4 Normalizing Relations :

Normalization is a process where the table gets divides into various pieces also is applied to lower the redundancy in data that results in data inconsistency. If the tables are splited into pieces, then to do manipulations on these, you will need joins as well as nested queries. Basically, database normalization is used to lower the redundancy as well as dependency. It significantly helps in dividing huge table into smaller ones meant for less redundancy as well as better relationship among data. Because of isolation of data, you can do additions, deletions as well as modifications of field in a single table plus spread throughout the rest of the database by the use of defined relationships. So normalization eliminates redundancy as well as potentially keeps informed anomalies. While normalising a database you will accomplish four goals :

- Arranging data in logical groups with each group explains about small portion of the whole
- Minimizing the quantity of reproduction data stored in a database
- Construction of database where you use as well as control the data rapidly as well as professionally with no compromise on integrity of data storage
- Arranging the data in such a way that on modification, you can do changes in simply one position.

13.4.1 Various Normal Forms and Their Need :

In normalization, there exist four different rules which are explained as below :

1. First Normal Form
2. Second Normal Form
3. Third Normal Form
4. BCNF

1. First Normal Form (1NF) :

As per First Normal Form, no two Rows of data are required to hold repeating group of information. It seems that in FNF, every set of column should possess a unique value like multiple columns which cannot used to carry out in similar row. Every table should be arranged in rows and every row should contain primary key that distinguishes it as only one of its kind.

It is analysed that a Primary key is basically is of single column as many times it contains more than one column which will co–ordinately produce a single main key. If you see the table shown below which is not in First normal form, where the details about students is given.

Student	Age	Subject
Nishit	17	Biology, Maths
Rohit	14	Maths
Amit	16	Maths

By analysing table with First Normal Form, we find that the row carries more columns where more values are assigned. Instead of keeping like this, we should separate such data by writing into multiple rows. By considering another table shown below with subsequent 1NF :

Student	Age	Subject
Nishit	15	Biology
Nishit	15	Maths
Rohit	14	Maths
Amit	17	Maths

Applying First Normal Form, we see that data redundancy rises, since there are several columns having similar data in many rows having a unique identity.

2. Second Normal Form (2NF) :

If you see the Second Normal Form, there exists no partial dependency of columns that is present on primary key. It is found that if certain column based on single part of concatenated key, it seems that the table will not be in relevance with Second normal form.

In illustration of First Normal Form, it is seen that there are 2 rows which covers many subjects that are taken by the students. At the same time, this is searchable; additionally pursue First normal form, so it is inefficient usage of space. In the Table, the First Normal Form, the candidate key is {Student, Subject}, Age of Student simply rely on Student column, which is not correct as per Second Normal Form. To get second normal form, it is advisable to break subjects into independent table as well as match them by student names with the help of foreign keys. Now the new student table of 2NF will show :

Student	Age
Nishit	15
Rohit	14
Amit	17

Consider a Student Table shown where the candidate key results in student column as all earlier such as Age depends on it. Currently the fresh Subject Table 2NF will be :

Student	Subject
Nishit	Biology
Nishit	Maths
Rohit	Maths
Amit	Maths

In the above table we find that candidate key contains information about Student as well as Subjects. If you see both the tables, you find that both are pertaining similar requirements for Second Normal Form with updation in Anomalies. Further, updated Anomalies contain latest information regarding Third Normal Form.

3. Third Normal Form (3NF) :

It is analysed that the Third Normal form is correct for every non–prime features exists in the table which depends on main key. Moreover, such type of movement with practical dependency should keep away from the table since the table is inspired to keep in Second Normal form. For this, consider the following details :

Student Detail Table :

Student_id Student_name DOB Street City State Pincode

It is found in the table that Student id is the main key, whereas street, city and state will follow pincode. As seen, there exists a dependency among the pincode as well as with other fields which is mainly called as transitive dependency. Consider the case as shown below, where you need to apply 3NF.

New Student_Detail Table :

Student_id Student_name DOB Pincode

Address Table :

Pincode Street City State

The main advantage of removing transitive dependency is :

- Lowering of data duplication.
- High data integrity.

4. Boyce and Codd Normal Form (BCNF) :

Boyce in addition to Codd are the Normal Form which is the superior description of Third Normal form. Such type of occurrence is definite with anomaly that cannot be handled by 3NF. It is seen that a 3NF table having no multiple overlapping candidate keys can act as BCNF. In case of BCNF table, following conditions should match :

Here, R should be in 3rd Normal Form where X is a super Key in the functional dependency (X → Y) expression, so the table follows :

Consider the following relationship : **R(A,B,C,D)**

and following dependencies :

A -> BCD

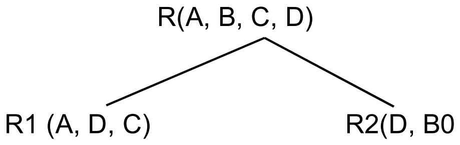
BC -> AD

D -> B

Above relationship is already in 3rd NF. Keys are **A** and **BC**.

Hence, in the functional dependency, **A -> BCD**, A is the super key.
in second relation **BC -> AD**, BC is also a key.
but in, **D -> B**, D is not a key.

Hence we can break our relationship R into two relationship **R1** and **R2**.



Breakin, table into two tables, one with A, D and C while the other with D and B.

Fig. 13.12 : Function Dependency Chart

❑ **Check Your Progress – 3 :**

1. The fields F1, F2, F3, F4, and F5 having functional dependencies as :

   F1 → F3

   F2 → F4

   (F1, F2) → F5

   Then as per normalization, the table will be in :

   a. 1NF   b. 2NF

   c. 3NF   d. None of the mentioned

2. In case of relational plan R(A,B,C,D,E,P,G), if FDs mentioned are known :

   AB → CD   DE → P   C → E   P → C   B → G

   Then the relation plan R will be :

   a. in BCNF   b. in 3NF, but not in BCNF

   c. in 2NF, but not in 3NF   d. not in 2NF

3. Select the normal form that will fit for normal relational database design ?

   a. 2NF   b. 5NF   c. 4NF   d. 3NF

13.5 Examples of Relational Data Base Design :

After designing a database, if the design is not correct, then it seems that it contains anomalies. To handle database with anomalies, it seems that it is impracticable. To do this, there exist three possibilities :

1. **Updating anomalies –** If data items are spread as well as are not connected to each one properly, after that it possibly will lead to extraordinary situations. If we try to update one data piece having its copies spread over numerous places, the small number of instances gets modernized correctly at the same time few others are left by old values. Such example will make the database in not consistent state.
2. **Deleting anomalies –** When you make an effort to delete a record, however parts of record was remained undeleted as lack of knowledge, then the data at that time gets saved somewhere else.
3. **Inserting anomalies –** When you try to place in the data in record which is not available at all.

With this method, you can eliminate all anomalies as well as fetch the database to a steady state. By Normalization process, you can thoroughly inspect relations for anomalies furthermore, after detection; you can remove such anomalies by dividing relations into two new connected relations.

Normalization is a significant measurement of the database development process as it frequently gives the appearance of real look of database and shows the working of how data are going to work together in database.

Example :

Semantic Modelling – Employee has ID, salary and reports to a manager who is also an employee. The conceptual graph (CG) is

Employee – has → ID
– has → Salary
– report–to → Manager
Manager – is–a → Employee

ER Diagram : ID, Salary and Manager are attributes of the entity Employee.

Dependency graph : ID → Salary, ID → Manager

Database schema

EMP (ID, SALARY, MANAGER)

Here MANAGER is foreign key of EMP where domain is subset of domain of ID.

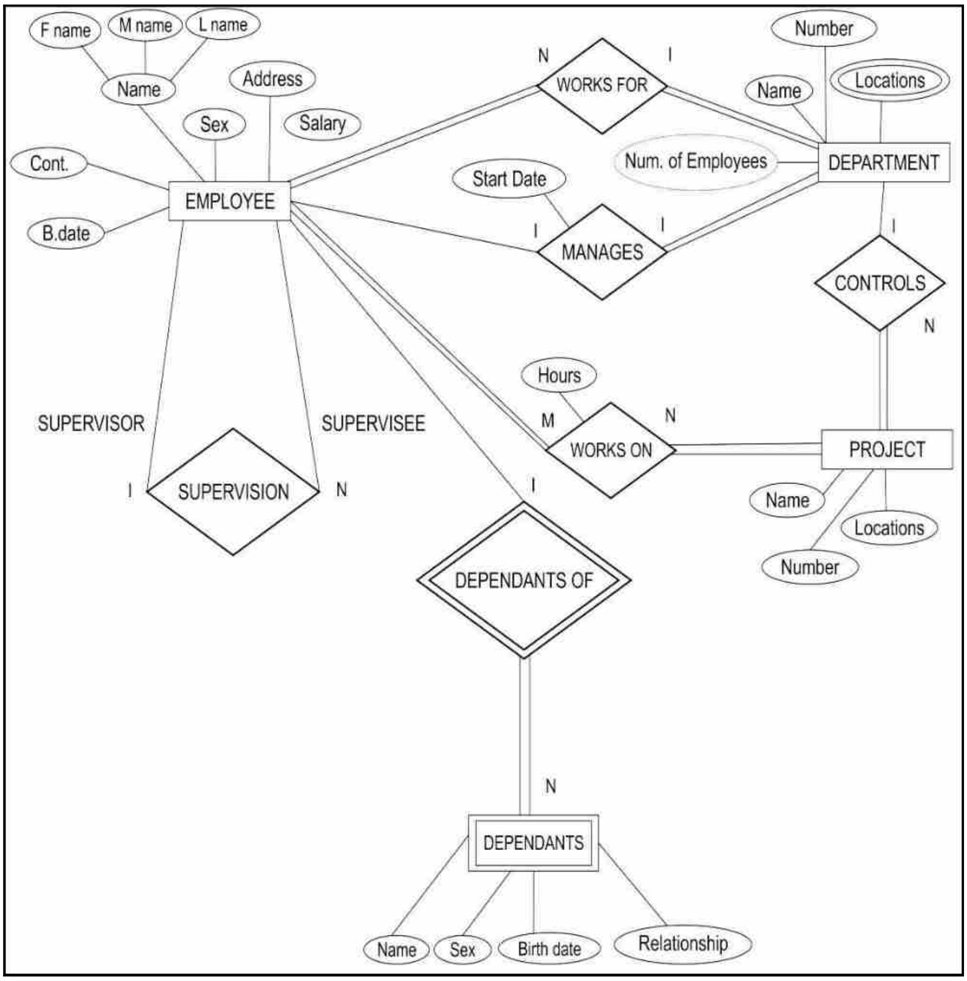


Fig. 13.13 : Semantic Modelling

- **Step 1 :** In regular entity E in ER schema, we design relation R having all attributes by selecting primary key.
- **Step 2 :** In case of weak entity W in ER schema with owner entity E, framing a relation R will include simple attributes Was attributes of R.
- **Step 3 :** In case of 1:1 relationship X, if S and T are relations for entity types having primary key T as foreign key S.
- **Step 4 :** In case of 1:N relationship if S relation corresponds to entity type N–side, and T relation corresponds to entity type at other side.
- **Step 5 :** In case of M :N relationship Z, develop relation R to show Z. Carries simple attributes Z in R.
- **Step 6 :** In case of multi–valued attribute A, develop relation R having an attribute corresponding to A.

❖ **Output :**

Entity Type → Relations

Relationship Type

1:1 or 1:N → absorbed in relations for entity types

M:N or N–ary → Relations

❑ **Check Your Progress – 4 :**

1. Normalization is used to :

a. reduce the table size	b. increase the table size
c. expand the table size	d. all of these

13.6 Let Us Sum Up :

It is learnt from this unit that a field is an area present inside which is reserved for a specific piece of data. As seen a record is the collection of values for all the fields pertaining to one entity. Further, table is a collection of related records.

Entity–Relationship Model is a Unified View of Data. In case of a conceptual model, the development is done from work of art of concepts. The Entity–relationship diagram is a chart which signifies correlation involving database entities.

Further, it is studied that an entity set is a collection of similar types of entities and an entity set is such which carries a primary key is named as Strong entity set

13.7 Answers for Check Your Progress :

❑ **Check Your Progress 1 :**

1 : c **2 :** c **3 :** c

❑ **Check Your Progress 2 :**

1 : a

❑ **Check Your Progress 3 :**

1 : a **2 :** d **3 :** d

❑ **Check Your Progress 4 :**

1 : a

13.8 Glossary :

1. **Software –** It is a program that contains operating system, utilities, files and applications programs from data stored in files.
2. **Procedures –** The instructions and rules that govern design and use of software component.
3. **Data –** The collection of facts.
4. **Normalization –** A process where data in relational construct is organized to lower redundancy and non–relational constructs.
5. **Physical Model –** It is types of data model phase where you can combine database and database management system particularly like as tables, columns, and data types.
6. **Primary Key –** This is a king of attribute which differently finds an example of an entity.
7. **Referential Integrity –** This is a kind of statement where foreign key approximation in a child entity matches with the approximation of parent entity.

13.9 Assignment :

1. Write the features of Relational Data Base Design.

13.10 Activities :

- Collect the information about the value of normalizing a table, if normalization is done from second normal form to third normal form ?

13.11 Case Study :

- In data mining and statistical data analysis, when do you need to normalize data and why is it important to do so ?

13.12 Further Readings :

1. Cardinal numbers and Formalized Mathematics by Grzegorz Bancerek.
2. The fundamental properties of natural numbers by Grzegorz Bancerek.
3. Finite sequences and tuples of elements of a non–empty sets by Czeslaw Byłiński

Unit 14

OBJECT ORIENTED ANALYSIS AND DESIGN

UNIT STRUCTURE

14.0 Learning Objectives

14.1 Introduction

14.2 Object–Oriented Analysis and Design

14.3 Basic Terms of Object–Oriented Analysis

14.4 UML Diagrams

14.5.1 Use–Case Diagram

14.5.2 Class Diagram

14.5.3 Sequence Diagram

14.5 Analysis Modeling

14.6 Let's Sum Up

14.7 Answers for Check Your Progress

14.8 Glossary

14.9 Assignment

14.10 Activity

14.11 Case Study

14.12 Further Readings

14.0 Learning Objectives :

After learning this unit, you will be able to understand :

- Know how object–oriented analysis can be done
- Understand basic terms related to object–oriented technology
- Understand object–oriented design and use of UML diagram
- How to draw different kind of UML diagrams
- Know how the quality software product can be designed

14.1 Introduction :

Analyst can do the analysis of the system using any of the method from (1) Structured analysis or (2) Object–oriented analysis. In the previous chapter we have seen that, if structured analysis is used then data flow diagram and entity relationship diagram are the basic tools. In this chapter we will focus on object–oriented analysis.

Object–oriented technology is extremely popular now a day. In this section we will focus on some basic concepts of object–oriented analysis. We will focus on UML (Unified Modeling Language) which is a standard of object–oriented systems. Before we start to do object–oriented analysis, and learn how to design UML diagrams, first we will focus on some important keywords of object–oriented system.

❑ **Check Your Progress – 1 :**

1. System Analysis can be done using ________.

   a. Structured Analysis　　b. Object–Oriented Analysis

   c. Both a and b　　d. None of these

2. ________ diagram is a useful tool for structured analysis.

   a. DFD　　b. Use–Case　　c. Class　　d. Sequence

3. UML stand for ________.

   a. User Metadata Language　　b. Unified Metadata Language

   c. Unified Modeling Language　　d. User Modeling Language

14.2 Basic Terms of Object–Oriented Analysis :

❖ **Object :**

Object is a tangible entity of the real world. Usually, object has its data is known as property of the object, and in the same way object can perform some action, which known as method of the object. Normally one object cannot access the data of another object. Object itself can use its own private data. Object can be anything, any real entity which has something (properties or attributes) and it can do something (methods). For example, customer of bank is an object. Customer Id, Name, Account No, Balance are the properties of the customer object, where the action performed by the customer such as withdraw or deposit are the methods of customer. Similarly, students, teachers, accountants are the example of objects.

❖ **Class :**

Similar objects constitute a class. Those objects possessing similar properties and having similar behaviour form a class. Class is a kind of template, which represent object. Class has variables and functions. Object is an instance of the class. When the object is initiated, from the class then variables declared in the class will become properties of the object and functions written in the class will becomes methods of the object. Classes are also known as Abstract Data Types (ADTs).

❖ **Abstraction :**

The process of identifying properties and methods of an object is called abstraction. Abstraction is the selective examination of certain aspects of a problem while ignoring the remaining aspects of the problem. In another way, abstraction is the process by which we concentrate on those aspects of the problem which are relevant and suppressed which are not relevant.

❖ **Encapsulation :**

Encapsulation is a technique in which data (properties) of the class and functions (methods) are packaged together as a single unit, and data can be accessible by the outsiders through the methods only. Encapsulation provides black box view to the class, where outsiders (non–member functions of the class) are not allowed to access the data directly. In the class variables (properties or data) are kept in the private sections, so that only those methods written in the class can access it. If any other function wants to access it, they need to call the method of the class, which verify the request. If the request is valid then and then the data will be provided by the method to the outsider function. So basically, encapsulation provides security to the data from the outside world.

- **Polymorphism :**

Polymorphism means more forms of the same thing. An operation may exhibit different behaviours in different instances. Function overloading and operator overloading features of an object–oriented programming languages are kind of Polymorphism. Two or more functions having same name, can be separated by a system at run time by looking to number of arguments passed or types of argument passed. Here we have different functions and same name, that is polymorphism. Polymorphism is increase readability of the program. For example, if we have 2 functions to do sum of integer numbers and another function to do sum of float numbers. Because of the behaviour of the function is same (doing sum) we can give same name 'sum' to both the function in object–oriented programming languages like C++, Java, C# etc., and we do not have to give different names like sum and sum1 like C–language.

- **Inheritance :**

Inheritance is the process by which we can derive a new class from the existing class. Existing class will act as base or parent class, and new class which generated from the base class is called child or derived class. Derived class can automatically have properties and methods of the base class, so we do not have to define those properties which are in the base class, again in the derived class. Inheritance provides reusability of the code (Code written in the base class can be accessible in the derive class).

❑ Check Your Progress – 2 :

1. ________ is a tangible entity of the real world.

   a. Class   b. Object

   c. Polymorphism   d. Inheritance

2. Objects possessing similar properties and having similar behaviour form a ________.

   a. Inheritance   b. Object   c. Class   d. Polymorphism

3. ________ is a technique in which data (properties) of the class and functions (methods) are packaged together as a single unit.

   a. Inheritance   b. Abstraction

   c. Polymorphism   d. Encapsulation

4. Deriving a new class from the existing class is called ________.

   a. Inheritance   b. Abstraction

   c. Polymorphism   d. Encapsulation

14.3 Object Oriented Analysis and Design :

- **Object–Oriented Analysis :**

Object–oriented analysis focus on examines at the problem domain, with the intension of producing a conceptual model of the information gathered in the preliminary investigation, feasibility study and requirement gathering phases of the SDLC, are being examined. Analysis model do not focus on implementation part, or how system is to be built. Analysis has to be done before the design.

Written problem statement or formal vision document is the source of the analysis. A system may have divided into multiple domains, representing different business, technologies, or other interest areas. The result of the object–oriented analysis is functional requirements in the conceptual model.

❖ Object–Oriented Design :

If we consider analysis to be a definition of the problem, then design is the process of defining the solution. Object–oriented design process defines the components, classes, objects, properties, methods and interfaces which satisfies functional requirement of the system. OOD transforms the conceptual level model produced by analysis into the environmental or technical model.

14.4 UML Diagrams :

Unified Modeling Language (UML) consists of different types of diagrams. Each diagram focuses on different way to define and analyse the system. These diagrams are :

1. **Class diagram :** It represents different classes and relationships among them of the system. Class consists of properties and methods.
2. **Object diagrams :** Objects are the instances of the class. Object diagram represents the relationships among the various objects of the system. One class diagram can have multiple object diagrams.
3. **Use–case diagram :** Use–case represents external behavior of the system. A use–case diagram consists of various actions and their interactions with different people.
4. **Sequence diagram :** Sequence diagram represents interactions between users over time period. A sequence diagram is detail behavior with respect to the time of each action of use–case diagram. Means for each action shown in the use–case diagram, a separate sequence diagram has to be designed.
5. **Collaboration diagram :** Collaboration diagram represents the interactions of objects of the system with respect to the relationships among the objects.
6. **State–chart diagram :** State–chart diagram represents various states of the system in response to the events triggered by the user. A state–chart diagram shows, how the state of the system changes in response to the internal or external events.
7. **Activity diagram :** Activity diagram represents elaboration of the behavior of the system. Activity diagram shows details behavior of the single function.

❑ Check Your Progress – 3 :

1. ________ diagram represents interactions between users over time period.

   a. Class　　b. Object　　c. Sequence　　d. Activity

2. ________ diagram represents external behaviour of the system

   a. Use–Case　　b. Object　　c. Class　　d. Collaboration

14.4.1 Use Case Diagram :

Use–case diagram provides a fast and simple way to describe the purpose of the project. Recently it is employed by many software engineers, to record high–level objectives of the project in its initial phase of development.

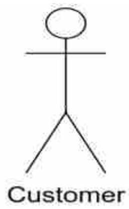
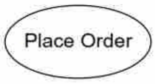
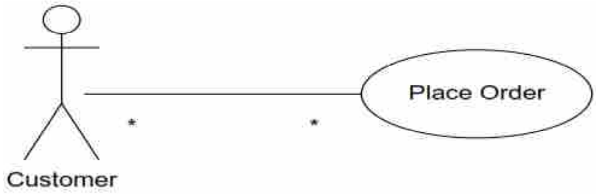
Use–case diagram is used to identifies different processes as well as primary elements of the system. The primary elements are also known as actors and processes of the system are called use–cases. Use–case diagram shows how the different actors interact with the different use–cases of the system.

Use–case diagram focuses on functional requirements of the system. It represents the graphical view of the system functionalities (use–cases) and users (actors).

❖ **Elements of Use–Case Diagram :**

It is easier to design use–case diagram, if we have proper knowledge of its different elements. The elements of the use–case diagrams are :

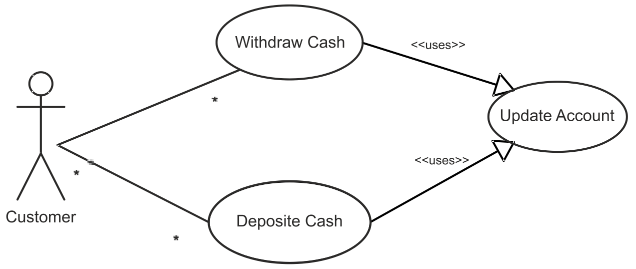
1. Actors
2. Use–Case
3. Relationship between Actor and Use–case
4. Relationship between Use–cases
5. Relationship between Actors
6. System boundary.

<table>
<tr><th>Sr. No.</th><th>Symbol</th><th>Description</th></tr>
<tr><td>1</td><td>
</td><td><b>Actors :</b> An actor can be a user or role which interact with the system by invoking different use–cases of the system. Usually Actor can be human, a hardware device, or another system which operates the functionalities of the system. Actors are external to the system. Actor may provide data or get information from the system by interacting with different use–cases.</td></tr>
<tr><td>2</td><td>
</td><td><b>Use–Cases :</b> It represents set of sequences of actions that system performs. A use–case describes what a system, sub–system, class or interface does, but not describe how it does.</td></tr>
<tr><td>3</td><td colspan="2"><b>Relationship between Actor and Use–cases :</b> Relationship between the Actor and Use–case is communication between the instance of Actor and instance of the Use–case. It is represented by a straight line between the participating Actor and requested Use–case.


Customer place order</td></tr>
</table>

4 **Relationship between Use–cases :** There are two relationships are there between Use–Cases :

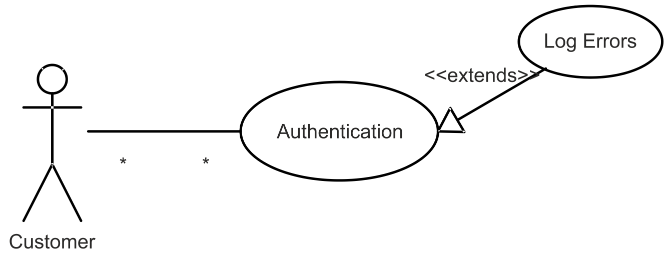
[1] Uses / Includes : An Uses / Include relationship between two use–cases means that the sequence of behaviors described in the sub (or included) use–case is included in the sequence of the base (including) use–case. For example, customer can 'Withdraw cash' or 'Deposit cash', but in both the cases Account has to be updated.



Use–cases Withdraw Case and Deposit Cash
includes Use–case Update Account

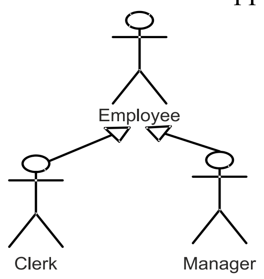
Note : Uses / Includes indicate compulsion. In the case of withdraw or deposit account updating is mandatory (Compulsory) sub Use–case.

[2] Extends : Extend is basically use to extend the functionality if one Use–case by another. For example, in the case of Authentication if any error is there then it must be logged. Make sure here the sub Use–case will perform if error occurs. If Authentication done successfully then sub Use–case 'log error' will not perform. Thus, extended Use–cases are optional.



Use–case 'Authentication' extends sub Use–case 'Log Errors'

5 **Relationships between Actors :** Some time different Actors of the system has to be generalized into one Actor. It is useful while the roles of different Actors are overlapping.

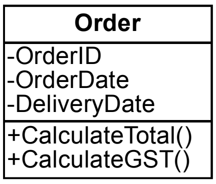
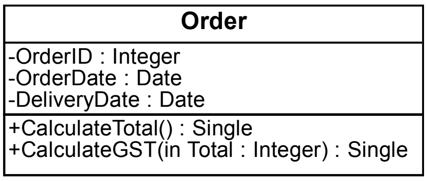


Generalization

6 **System Boundary :** System boundary is a rectangular box, which represents whole system. All the Use–cases are inside this rectangle box, to denote Use–cases are in the system. Actors are the external entities so they are placed outside of the system (rectangle).

14.4.2 Class Diagrams :

Based on the purpose, class diagram can be designed of 2 types : (1) Analysis class diagram or (2) Design class diagram. Analysis model provides just provides overview of the class, that is names of the properties and methods. Whereas, Design model represents properties with its data type, and methods with arguments and its return type. So, design class diagram is more detailed version of the class diagram.

Analysis Class Diagram	Design Class Diagram
 Order -OrderID -OrderDate -DeliveryDate +CalculateTotal() +CalculateGST()	 Order -OrderID : Integer -OrderDate : Date -DeliveryDate : Date +CalculateTotal() : Single +CalculateGST(in Total : Integer) : Single

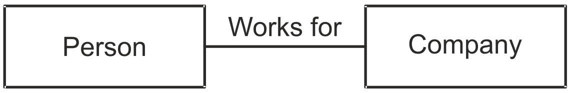
❖ **Elements of Class Diagram :**

1. **Class :** As we have seen in the above example in the class diagram class is represented as a rectangle divided in the sections. First section shows the name of the class. Second section shows the attributes of the class and third section shows methods of the class.

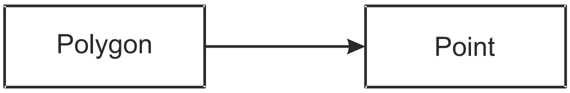
2. **Relationships :** Different types of relationships are described below :

Association :

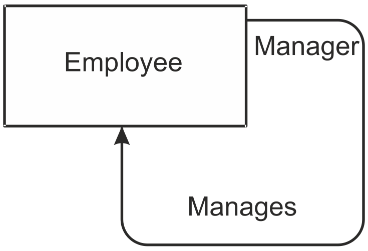
Physical or conceptual connections between the classes can be represented as association (simple line). It corresponds to a verb like teaches, work for, manages etc. Association can be Unidirectional, Bidirectional or Reflexive.



Bidirectional Association



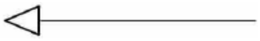
Unidirectional Association



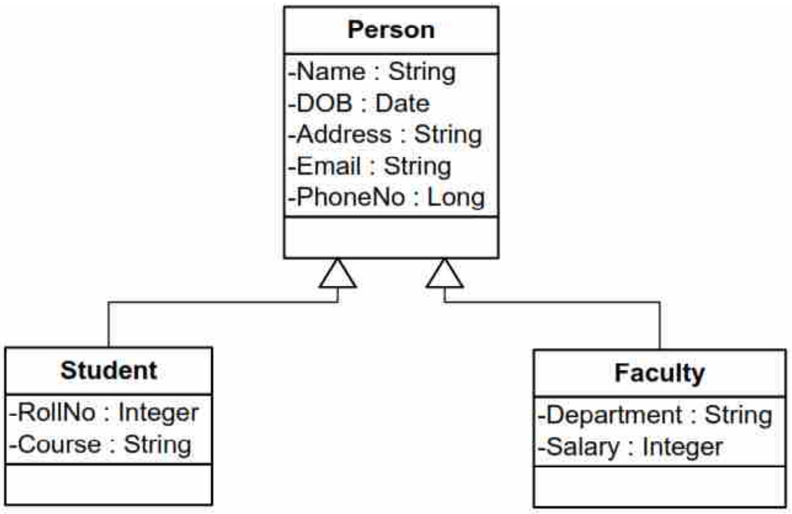
Reflexive Association

Generalization :

Generalization is process of creating base class If two or more classes have common attributes or methods. Common attributes and method are placed in the generalized class (base class). For example, student class has attributes, Name, DOB, Address, Email, Phone number, Roll No, course etc. and Faculty class has attributes Name, DOB, Address, Phone number, Department, Salary etc. Then generalized class Person can be created with common attribute. Base class (Person) and derived classes (Student and Faculty) has 'is a' relationship. For example, every student and faculty are person. Generalization can be represented by symbol :



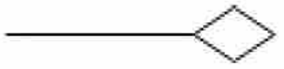
Generalization Relationship



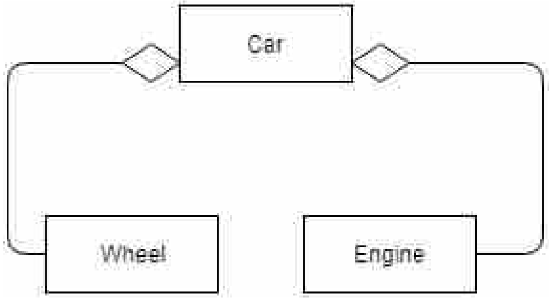
Example of Generalization

Aggregation :

The relationship between aggregated object and its components can be described as aggregation. Aggregation is a kind of association. Aggregation can be represented by symbol :



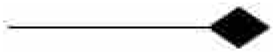
Aggregation Relationship



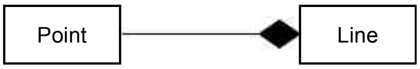
Example of Aggregation relationship (Car made by While Engine etc.)

Composition :

When multiple instances of the same class represent another class then composition is used. The difference between Aggregation and Composition is in aggregation multiple instances of different class represent another class, whereas in composition multiple instances of same class. It is represented by :



Composition Relation



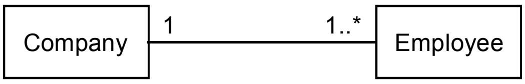
Example of Composition

Multiplicity :

Multiplicity notation is placed near ends of the relationship. It shows how the instances of one class are linked instances of another class.

Indicator	*Meaning*
0..1	Zero or one
1	One only
0..*	Zero or more
*	Zero or more
1..*	One or more
3	Three only
0..5	Zero to Five
5..15	Five to Fifteen
2,4	Two or Four

For example, one company can have one or more employees can be represented as :



14.4.3 Sequence Diagram :

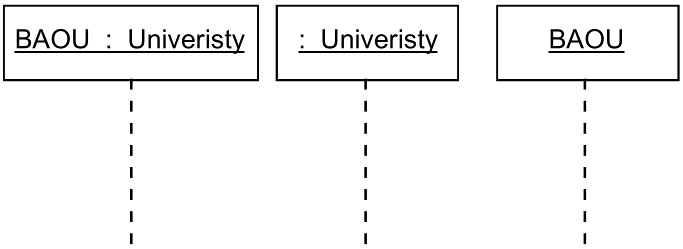
A sequence diagram is used to express each Use–case in details with respect to time. A sequence diagram represents the sequence of actions occurs in a Use–case, and order of each action with respect to time.

❖ **Elements of Sequnce Diagram :**

The following elements are used to draw the sequence diagram.

Life Lines :

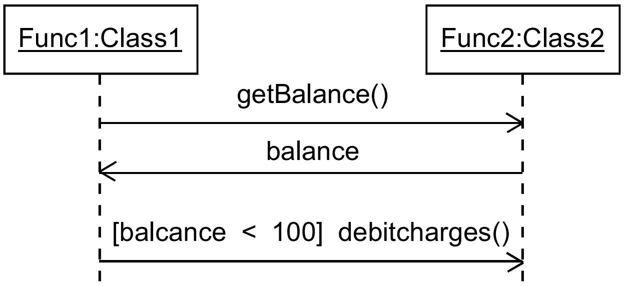
Lifeline represents role or instances which participate in the sequence of interactions. Lifelines are drawn as a rectangle with a dashed vertical line from the centre of the rectangle. Inside the rectangle name of the class, name of the instance or both can be specified.



Lifeline of the Sequence Diagram

Messages :

Message defines a kind of integration between instances (Actor or Lifeline). Communication (message passing) can invoke by mean of function calls. It is shown in the following figure :

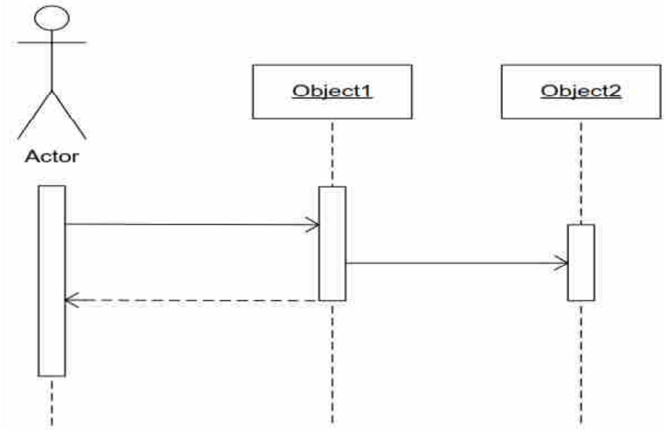


Message passing between Lifelines

The condition placed in the sequence diagram, that is the balance is less than 100 the call function 'debitcharges()' is called guard condition.

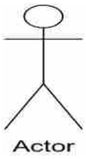
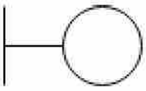
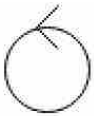
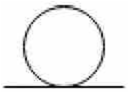
Activation :

Activation is represented as vertical thin boxes on the dotted lines of the Lifelines, which represents the time an object takes to complete the task. Following diagram shows the activation.



Objects :

There are four different types of objects are the, who interact with each other in the sequence diagram. All these objects are described below :

Actor		Actor object initiate the task. Actor is an instance of the class and it is external entity. The role of the actor is same as in Use–case diagram.
Boundary		Instance of the boundary class is used to model the communication between system and external objects like Actor.
Controller		Controller is used to control the behavior specific use–case. It represents logic. Usually, it comes between boundary and entity.
Entity		Entity object is used to store the associated behavior or model information. It represents stores of information in the system.

14.5 Analysis Modeling :

Analysis modeling can be organized by it four elements – scenario based modeling, flow oriented modeling, class based modeling and behavioural modeling.

1. **Scenario based modeling :** In the scenario–based model, we draw the use case diagram, Activity diagram and Swimlane diagram.

   Use–Case : Use case is to represent the various scenario by the user (Actor) point of view. User–case diagram is a simple and relatively easier approach to represent what is outside of the system (Actor) and what system should be performed (use–cases). We have already discussed how to draw the use case diagram in section 3.6

   Activity Diagram : In the activity diagram we focus on main tasks or function (use cases) of the use–case diagram, and represents what Actor can acquires, produces or change in the system. Detailed interaction of the different users of the system (Actors) and tasks of the system (use cases) can be represented by activity diagrams.

   Swimlane Diagram : It is nothing but the useful variation in the activity diagram and allows the modeler to represent the flow of activities described by the user–case and at the same time indicate which actor or analysis class has responsibility for the action described by an activity rectangle.

   Responsibilities are represented as parallel segments that divide the diagram vertically, like the lanes in a swimming pool.

2. **Flow–oriented modeling :** Flow–oriented modeling represents flow of the data in the system. It Represents how data objects are transformed as they move through the system. We have already discussed Data Flow Diagram (DFD), which shows the transitions of the data in the system. To draw the DFD, we need to identify Entity, Process, Data stores and transition of the data among them are represented by arrow. In the Block–13 we have already discussed the notations and rules of how to draw DFD.

3. **Class–based modeling :** Class based modeling is also known as Object Oriented Analysis. We have seen in that object–oriented analysis begins by identifying classes. Once the classes are recognized then its attributes and methods are identified. Classes are represented with their relations with the other classes. In this process basic fundamentals of the object–oriented analysis such as Abstraction, Encapsulation, Polymorphism and Inheritance is used. The elements of the class diagram are already discussed in the section 14.7.

4. **Behavioral Model :** It indicate how system will behave or respond to the event triggered by the external entities or actors of the system. To create the model, the analyst must perform the following steps :

   1. Evaluate all use–cases to fully understand the sequence of interaction within the system.
   2. Identify events that drive the interaction sequence and understand how these events relate to specific objects.
   3. Create a sequence for each use–case.

4. Build a state diagram for the system.

5. Review the behavioural model to verify accuracy and consistency.

To represents the various changing states of the system, state chart diagram is used.

The States of a System :

- State–a set of observable circumstances that characterizes the behavior of a system at a given time
- State transition–the movement from one state to another
- Event–an occurrence that causes the system to exhibit some predictable form of behavior
- Action–process that occurs as a consequence of making a transition

❑ **Check Your Progress – 4 :**

1. In Use–Case diagram, ________ describes what a system, sub–system, class or interface does, but not describe how it does.

   a. Actor b. Use–Case

   c. Relationship d. None of the above.

2. ________ relationship between two or more Use–Cases enforce compulsion.

   a. Extends b. One–to–Many

   c. Include d. None of the above.

3. In class diagram, ________ is used if two or more classes have common attributes.

   a. Association b. Aggregation

   c. Generalization d. All of the above.

14.6 Let Us Sum Up :

In this chapter we have learnt how can we do object–oriented analysis and design using UML diagram. We have seen the rules, symbols and components of Use–case diagram, Class diagram, Sequence diagram and so on. At the end we have seen the fundamental concept of designing good and quality system. We hope now student can draw UML diagrams of any system if clear and concise requirements are given.

14.7 Answers for Check Your Progress :

❑ **Check Your Progress 1 :**

1 : c **2 :** a **3** : b

❑ **Check Your Progress 2 :**

1 : b **2 :** c **3 :** d **4 :** a

❑ **Check Your Progress 3 :**

1 : c **2 :** a

❑ **Check Your Progress 4 :**

1 : b **2 :** c **3 :** c

14.8 Glossary :

1. **Class :** Similar objects constitute a class. Those objects possessing similar properties and having similar behavior form a class. Class is a kind of template, which represent object.

2. **Object :** Object is a tangible entity of the real world. Usually, object has its data is known as property of the object, and in the same way object can perform some action, which known as method of the object.

3. **Encapsulation :** Encapsulation is a technique in which data (properties) of the class and functions (methods) are packaged together as a single unit, and data can be accessible by the outsiders through the methods only.

4. **Polymorphism :** Polymorphism is the ability of an object to take on many forms. In object–oriented programming languages Function overloading and operator overloading are type of Polymorphism.

5. **Inheritance :** The ability to derive new classes from an existing class is called Inheritance. Inheritance in the programming language facilitates reusability of the code.

14.9 Assignment :

1. What is Use–Case diagram ? Explain all the components of Use–Case diagram in brief.

2. What is Sequence diagram ? Draw and explain a sequence diagram for registration process for any web–site.

3. What is class diagram ? Discuss types of relationships in the class diagram with an example of each.

14.10 Activity :

- Draw sequence diagram for the following use–cases : [1] Registration [2] Login [3] Forgot Password [4] Change Password

14.11 Case Study :

- Design all UML diagram for any E–Commerce website. Make suitable assumptions when it is needed.

14.12 Further Readings :

1. Software Engineering – A Practitioner's Approach by Roger S. Pressman (McGraw–Hill international edition).

2. Fundamentals of Software Engineering by Rajib Mall (PHI)

3. System Analysis and Design Methods by Gary B. Shelly, Thomas J. Cashman, Harry J. Rosenblatt (CENGAGE Learning)

4. Magnifying object–oriented analysis and design by Arpita Gopal and Netra Patil (PHI)

5. Object–oriented modeling and design by James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, William Lorensen (PHI)

BLOCK SUMMARY :

While studying this block, the user will achieve knowledge and understanding about Database Integrity Concepts and the idea of Normalization and Object Oriented Analysis and Design. The block–12 explains about Domain Integrity constraints, Referential Integrity constraints and Entity Integrity constraints with certain examples that will help the user to grab the concept.

The block–13 has given detail idea on features of Entity–Relationship (ER) Model along with concept of Normalization. The details related to conceptual modelling, mapping cardinalities and relationships among entities and required for students in order to gain knowledge regarding system analysis. The student will be given with the knowledge about different rules and criteria's of normalization The students or programmers will get benefit while reading this block as it gives shortcuts and related examples that will clear all doubts.

Whereas the block–14 focuses on basic concepts of object–oriented analysis. It focuses on UML (Unified Modeling Language) which is a standard of object–oriented systems and Analysis Modeling.

BLOCK ASSIGNMENT :

❖ **Short Questions :**

1. What is Entity Integrity constraint ?
2. Is database normalization important ?
3. What is a decision table ?
4. What is structured English ?
5. What is data dictionary ?
6. Explain the basic terms of Object Oriented Analysis.
7. Write a note on different types of UML Diagrams.

❖ **Long Questions :**

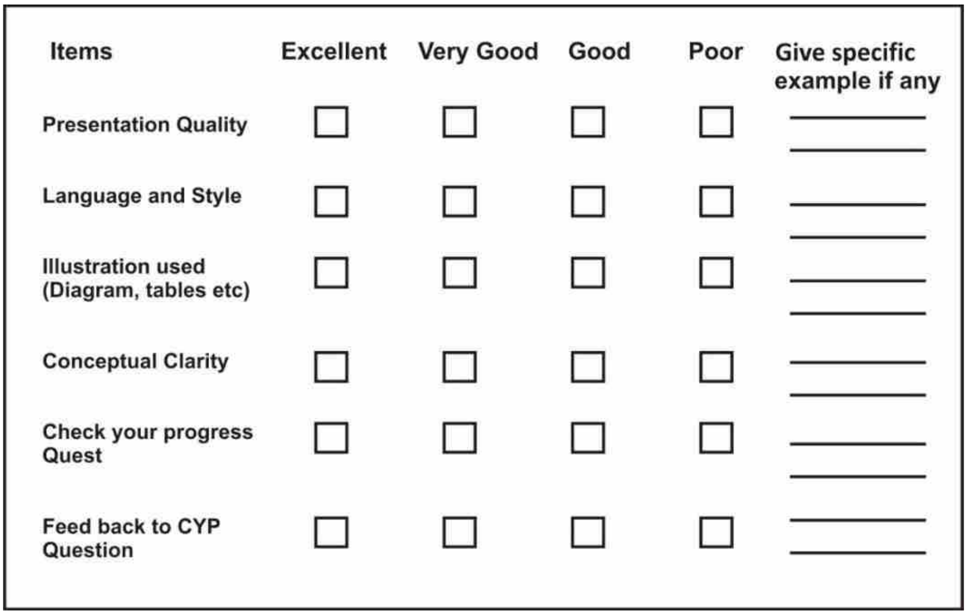
1. Explain the rules of normalization ?
2. Explain Procedure Oriented Design ?
3. State the advantages of Structured English ?

❖ **Enrolment No. :**

1. How many hours did you need for studying the units ?

Unit No.	12	13	14
No. of Hrs.			

2. Please give your reactions to the following items based on your reading of the block :



Items	Excellent	Very Good	Good	Poor	Give specific example if any
Presentation Quality	□	□	□	□	________ ________
Language and Style	□	□	□	□	________ ________
Illustration used (Diagram, tables etc)	□	□	□	□	________ ________
Conceptual Clarity	□	□	□	□	________ ________
Check your progress Quest	□	□	□	□	________ ________
Feed back to CYP Question	□	□	□	□	________ ________

3. Any other Comments

....

....

....

....

....

....

....

....

BAOU
Education
for All

DR.BABASAHEB AMBEDKAR
OPEN UNIVERSITY

'Jyotirmay' Parisar,
Sarkhej-Gandhinagar Highway, Chharodi, Ahmedabad-382 481.
Website : www.baou.edu.in