UNIVERSITY OF AGRICULTURE, ABEOKUTA
COLLEGE OF NATURAL SCIENCES
DEPARTMENT OF COMPUTER SCIENCE

SYSTEM ANALYSIS AND DESIGN
CSC 326- 3 UNITS

A MATERIAL DEVELOPED

BY

DR. O. R. VINCENT

COURSE CODE:	**CSC 326**
COURSE TITLE:	**SYSTEM ANALYSIS AND DESIGN**
NUMBER OF UNITS:	**3 UNITS**
COURSE DURATION:	**THREE HOURS PER WEEK**

COURSE DETAILS:

Course Coordinator:	**Dr. (Mrs.) O. R. Vincent** **B. Sc., M. Sc., PhD**
Email:	**Vincent.rebecca@gmail.com**
Office Location	**Room B201, COLNAS**
Other Lecturers	**xxxxx**

COURSE CONTENT:

Introduction to the Concept of Systems – Types of systems, the need for system Analysis and design, Qualities of a System Analyst, roles of a system Analyst, System Development Life Cycle - Phase of System Development Life Cycle

System Analysis Approaches- Model-Driven Analysis Approach, Unified Modeling Language (UML), Accelerated System Analysis Approaches

Modeling System Requirements with Use Case - System Concepts for Use-Case Modeling, Relationship, Association and Extends
System Description Techniques ---Types of Data, Flowchart and Data flow diagram (DFD), Decision Tables and decision Trees
System Development Methodologies - Data Processing Systems, Centralized and Decentralized, Information System, Multi-user environment, Networking and file server system

COURSE REQUIREMENTS:

This is a compulsory course for all students in Computer Science and Statistics Departments.

The students are expected to be able to write a good proposal at the end of the course and present a seminar to defend their proposal.

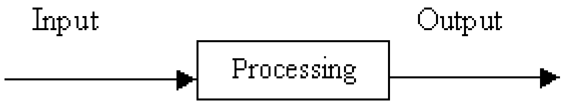
READING LIST:

1.

SYSTEM ANALYSIS AND DESIGN

1.0 INTRODUCTION

Systems are created to solve problems. One can think of the systems approach as an organized way of dealing with a problem. System Analysis and design mainly deals with the software development activities. A system is a collection of components that work together to realize some objectives. Basically, there are three major components in every system, namely input, process and output.



In a system the different components are connected with each other and they are interdependent. For example, human body represents a complete natural system. We are also bound by many national systems such as political system, economic system, educational system etc. The objective of the system demand that some output produced as a result of processing the suitable inputs.

Types of Systems

Information systems are developed for different purposes, depending on the needs of the business. Transaction Processing Systems (TPS) function at the operational level of the organization: Office Automation Systems (OAS) and Knowledge Work System (KWS) support work at the knowledge level. Higher level systems include Management Information

Systems (MIS) and Decision Support Systems (DSS). Expert system and apply the expertise of decision makers to solve specific, structured problems. On the strategic level of management, we found the executive support systems (ESS). Group Decision Support Systems (GDSS) and the more generally described computer supported Collaborate Work Systems (CSSWS) aid group-level decision making a of a semi-structured or unstructured varieties. The variety, information system that analysts may develop is shown in figure 1.

1.2 **Need for Systems Analysis and Design**

System analysis and design, as performed by a system analyst, seeks to analyze data input or data flow systematically, processing or transforming data, data storage, and information output within the context it a particular business.

Systems analysis and design is used to analyzed, design and implement improvements in the functioning of businesses that can be accomplished through the use of computerized information systems.

Installing a system without proper planning leads to great dissatisfaction and frequently causes the system to fall into disuse. System analysis and design leads structure to the analysis and design of information systems, a costly endeavour that might otherwise have been done in a haphazard system analysis and design can be thought of as a series of process systematically undertaken to improve a business through the use of computerized information systems. Anyone who interacts with an information system in the context of his or her work in the organization can be called and end user.

1.3 Roles of the System Analyst

The systems analyst systematically assesses how business function by examining the input and processing of data in the output of information with the intent of improving organizational processes. Many improvements involve better support it business functions through the use of computerized information systems. This definition emphasizes a systematic, methodical approach to analyzing and potentially improving – what is occurring in the specific context created by a business. The analyst must be able to work with people of all descriptions and be experienced in working with computers. The analyst plays many roles, sometimes balancing several at the same time. The three primary roles of the systems analysts are consultant, supporting expert, and agent of change.

i) Systems Analyst as a Consultant

The systems analyst frequently acts as a systems consultant to a business and thus may be hired specifically to address information systems issues within a business. Such hiring can be an advantage because outside consultants can bring with them a fresh perspective that other members of an organization do not possess. Though, this may be a disadvantage because the true organizational culture can never be known to an outsider.

As a consultant, you will rely heavily on the systematic methods to analyze and design appropriate information systems for a 1ar business.

ii) System Analyst As Supporting Expert: Another role that a system analyst may be required to play is that of supporting expert within a business where he/she is regularly employed in some system capacity. In this role the analyst draws on professional expertise concerning computer hardware and software and their uses in the business. This work is often not a full-blown systems project, but rather it entails a small modification or decision affecting a single department. As the support expert, the system analysts are not managing the project; he is merely serving as a resource person for those who are. If you are a systems analyst employed by a manufacturing company or organization, many of your daily activities may be encompassed by the role.

iii) **Systems Analyst as Agent of change:** The most comprehensive and responsible role that the analyst takes on is that of agent change, whether internal or external to the business. As an analyst, you are an agent of change whenever you perform any of the activities in the systems development life cycle and are present in the business for an extended period. An agent of change can be defined as a person who serve as a catalyst for change, develops a plan for change, and works with others in facilitating that change.

Your presence in the business changes it. As a systems analyst, this fact must be recognized and should be used as a starting point for your analysis. Hence, you must interact with the users and management from the very beginning of the project: without their help one cannot understand what is happening in an organization and real change cannot take place.

If change (that is, improvement to the business that can be realized through information systems) seems warranted after analysis, the next step is to develop a plan for the change along with the people who must enact the changes. You facilitate change by using your expertise with humans as well as with computers to bring about their integration in a human machine information system.

As an analyst acting as an agent the change, you advocate a 1ar avenue of change involving the use of information systems.

In addition, you teach users the process of change because of the awareness that changes in the information system do not occur independently but can also change the other aspect of the organization as well.

1.4 Qualities of the Systems Analyst

Successful systems analyst must possess a wide range of qualities:

The system analyst is a problem solver, he or she is a person who views the analysis of problem as challenge and who enjoys devising workable solutions. When necessary, the analyst must be able to systematically tackle the situation at hand through skillful application of tools and technique. The analyst must also be a communicator, capable of relating meaningful to other people over extended periods of time. Systems analyst needs enough computer experience to program, understand the capabilities of computers, information requirements from users, and communicate what is needed to programmes.

It is appropriate at this time for you to reflect on the personal and professional ethics you bring to a consulting relationship. Clarify the values that you embrace as you build relationship with users and team members. Self-knowledge can help you to become a better analyst, and code of conduct of professional groups such as the Association for Computing Machinery (ACM) provides a reasonable context for examining your ethical beliefs.

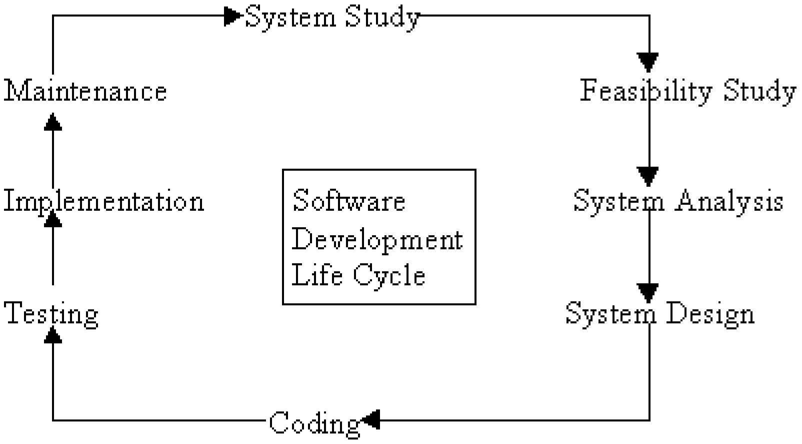
The systems analyst must be self-disciplined, self-motivated individual who is able to manage and coordinate innumerable project resources, including other of people. System analysis is a demanding career but, in compensation, an ever-changing and always challenging one.

2.0 SYSTEM DEVELOPMENT LIFE CYCLE

System development life cycle means combination of various activities. In other words we can say that various activities put together are referred as system development life cycle. In the System Analysis and Design terminology, the system development life cycle means software development life cycle.

Following are the different phases of software development cycle:

- System study
- Feasibility study
- System analysis
- System design
- Coding
- Testing
- Implementation
- Maintenance
- The different phases of software development life cycle is shown in Fig.1.1



- Fig. 2.1 Different phases of Software development Life Cycle

2.1 PHASES OF SYSTEM DEVELOPMENT LIFE CYCLE

Let us now describe the different phases and the related activities of system development life cycle in detail.

(a) System Study

System study is the first stage of system development life cycle. This gives a clear picture of what actually the physical system is? In practice, the system study is done in two phases. In the first phase, the preliminary survey of the system is done which helps in identifying the scope of the system. The second phase of the system study is more detailed and in-depth study in which the identification of user’s requirement and the limitations and problems of the present system are studied. After completing the system study, a system proposal is prepared by the System Analyst (who studies the system) and placed before the user. The proposed system contains the findings of the present system and recommendations to overcome the limitations and problems of the present system in the light of the user’s requirements.

To describe the system study phase more analytically, we would say that system study phase passes through the following steps:

- problem identification and project initiation
- background analysis
- inference or findings

(b) Feasibility Study

On the basis of result of the initial study, feasibility study takes place. The feasibility study is basically the test of the proposed system in the light of its workability, meeting user’s requirements, effective use of resources and .of course, the cost effectiveness. The main goal of feasibility study is not to solve the problem but to achieve the scope. In the process of feasibility study, the cost and benefits are estimated with greater accuracy.

(c) System Analysis

Assuming that a new system is to be developed, the next phase is system **analysis**. Analysis involved a detailed study of the current system, leading to specifications of a new system. Analysis is a detailed study of various operations performed by a system and their relationships within and outside the system. During analysis, data are collected on the available files, decision points and transactions handled by the present system. Interviews, on-site observation and questionnaire are the tools used for system analysis. Using the following steps it becomes easy to draw the exact boundary of the new system under consideration:

- Keeping in view the problems and new requirements
- Workout the pros and cons including new areas of the system

All procedures, requirements must be analysed and documented in the form of detailed data flow diagrams (DFDs), data dictionary, logical data structures and miniature specifications. System Analysis also includes sub-dividing of complex process involving the entire system, identification of data store and manual processes.

The main points to be discussed in system analysis are:

- Specification of what the new system is to accomplish based on the user requirements.
- Functional hierarchy showing the functions to be performed by the new system and their relationship with each other.
- Function network which are similar to function hierarchy but they highlight the those functions which are common to more than one procedure.
- List of attributes of the entities - these are the data items which need to be held about each entity (record)

(d) System Design

Based on the user requirements and the detailed analysis of a new system, the new system must be designed. This is the phase of **system designing**. It is a most crucial phase in the development of a system. Normally, the design proceeds in two stages :

- preliminary or general design
- Structure or detailed design

Preliminary or general design: In the preliminary or general design, the features of the new system are specified. The costs of implementing these features and the benefits to be derived are estimated. If the project is still considered to be feasible, we move to the detailed design stage.

Structure or Detailed design: In the detailed design stage, computer oriented work begins in earnest. At this stage, the design of the system becomes more structured. Structure design is a blue print of a computer system solution to a given problem having the same components and inter-relationship among the same components as the original problem. Input, output and processing specifications are drawn up in detail. In the design stage, the programming language and the platform in which the new system will run are also decided.

There are several tools and techniques used for designing. These tools and techniques are:

- Flowchart
- Data flow diagram (DFDs)
- Data dictionary
- Structured English
- Decision table
- Decision tree

Each of the above tools for designing will be discussed in detailed in the next lesson.

(e) Coding

After designing the new system, the whole system is required to be converted into computer understanding language. **Coding** the new system into computer programming language does this. It is an important stage where the defined procedure are transformed into control specifications by the help of a computer language. This is also called the programming phase in which the programmer converts the program specifications into computer instructions, which we refer as **programs**. The programs coordinate the data movements and control the entire process in a system.

It is generally felt that the programs must be modular in nature. This helps in fast development, maintenance and future change, if required.

(f) Testing

Before actually implementing the new system into operations, a test run of the system is done removing all the bugs, if any. It is an important phase of a successful system. After codifying the whole programs of the system, a test plan should be developed and run on a given set of test data. The output of the test run should match the expected results.

Using the test data following test run are carried out:

- Unit test
- System test

Unit test: When the programs have been coded and compiled and brought to working conditions, they must be individually tested with the prepared test data. Any undesirable happening must be noted and debugged (error corrections).

System Test: After carrying out the unit test for each of the programs of the system and when errors are removed, then system test is done. At this stage the test is done on actual data. The complete system is executed on the actual data. At each stage of the execution, the results or output of the system is analysed. During the result analysis, it may be found that the outputs are not matching the expected out of the system. In such case, the errors in the particular programs are identified and are fixed and further tested for the expected output.

When it is ensured that the system is running error-free, the users are called with their own actual data so that the system could be shown running as per their requirements.

(g) Implementation

After having the user acceptance of the new system developed, the implementation phase begins. Implementation is the stage of a project during which theory is turned into practice. During this phase, all the programs of the system are loaded onto the user's computer. After loading the system, training of the users starts. Main topics of such type of training are:

- How to execute the package
- How to enter the data
- How to process the data (processing details)
- How to take out the reports

After the users are trained about the computerized system, manual working has to shift from manual to computerized working. The following two strategies are followed for running the system:

i. **Parallel run:** In such run for a certain defined period, both the systems i.e. computerized and manual are executed in parallel. This strategy is helpful because of the following:

   - Manual results can be compared with the results of the computerized system.
   - Failure of the computerized system at the early stage, does not affect the working of the organisation, because the manual system continues to work, as it used to do.

i. **Pilot run:** In this type of run, the new system is installed in parts. Some part of the new system is installed first and executed successfully for considerable time period. When the results are found satisfactory then only other parts are implemented. This strategy builds the confidence and the errors are traced easily.

(h) Maintenance

Maintenance is necessary to eliminate errors in the system during its working life and to tune the system to any variations in its working environment. It has been seen that there are always some errors found in the system that must be noted and corrected. It also means the review of the system from time to time. The review of the system is done for:

- knowing the full capabilities of the system
- knowing the required changes or the additional requirements
- studying the performance

If a major change to a system is needed, a new project may have to be set up to carry out the change. The new project will then proceed through all the above life cycle phases.

3.0 SYSTEM ANALYSIS APPROACHES

Fundamentally, system analysis is about problem solving. There are many approaches to problem solving. Some of these approaches include structured analysis, Information Engineering (IE) discovery prototyping and object oriented analysis.

Three out of these approaches are referred to as model-driven analysis approach. These are:

- Structured analysis
- Information Engineering
- Object Oriented

3.1 Model-Driven Analysis Approach

It also pictures to communicate business problems, requirements and solutions. Examples of familiar models are flow charts, hierarchy charts and organizational charts.

Model-Driven Approaches are enhanced by the use of automated tools such as Visio professional. We also have Computer System Engineering (CASE) tools such as system architecture.

3.1.1 Structured Analysis

It's a model-driven, process centered technique used to either analyze an existing system, define business requirements for a new system or both. The models are pictures that illustrate the system's component pieces, processes and their associated inputs, output and files.

Structured Analysis focuses on the flow of data through business and software processes. By process-centered, we mean that the emphasis in this technique is on the process building blocks in the information system framework.

Over the years, the technique has evolved to also model the knowledge (data) and communication building blocks to a secondary emphasis.

Data flow diagrams are used to depict the existing or proposed processes in a system along with their inputs, outputs and files.

3.1.2 Information Engineering (IE)

It's model-driven and data-centered but it is process-sensitive. It's technique for planning, analyzing and designing information systems.

IE models are pictures that illustrate and synchronize the system's data and process. IE focuses on the structured of stored data in a system. The data models in IE are called entity relationship diagrams.

It is said to be data-centered paradigm because it emphasizes the studying and requirement analysis of knowledge (data) before those of the process and communication requirements. Both IE and Structured-Analysis attempt to synchronize data and process models. These two approaches differ only in the choice of model it draws first.

IE draws data models first while Structured-Analysis draws process models first.

3.1.3 Object Oriented Analysis

Object is the encapsulation of both data and the methods. The data are called properties that describe a discrete person, object, place, events or things with all of the processes (called methods) that are allowed to use or update the data properties. The only way to access or update the object data is to use the object predefined processes (methods).

In the past, most system development approaches deliberately separated the issues of data from those of processes. Though, object technologies eliminate this artificial separation of data and processes. Instead of specifying the data and processes, they are integrated into constructs called objects. The only way to create, read, update or delete an object data (properties) is through one of its embedded processes (methods).

Object-Oriented Analysis is a model-driven technique that integrates data and processes into constructs called objects. OOA models are pictures that differentiate the system's objects from the objects. The modeling standard for OOA is the Unifier Modeling Language (UML). The UML defines several different types of diagrams that collectively model an information system or application in terms of objects.

3.2 Unified Modeling Language (UML)

UML provides a useful tool set for systems analysis and design. As with any product created with the help of tools, the value of the UML variables in a project depends on the expertise with which the systems analyst wields the tools. The analyst will initially use the UML toolset to break down the system requirements into a use case model and an object model. The use case model describes the use cases and actors. The object model describes the objects and object associations and the responsibilities the collaborators, and attributes of the objects.

The put UML to work, the following are important

1. you define the use case model
2. you define the object model
3. use UML to model the system
4. Refine UML diagram by deriving their classes and properties.

3.2.1 Importance of using UML for modeling

UML is a powerful tool that can greatly improve the quality of your systems analysis and design, and it is hoped that the improved practices will eventually translate into higher-quality systems. By using UML in an iterative cycle of systems analysis, you can achieve a greater understanding between the business team and the IT team regarding the system requirements and the processes that need to occur within the system to meet those requirements. The first iteration of analysis should be at a very high level to identify the overall system objectives and validate the requirements through use case analysis. Identify the actors and defined the initial use case model are part of this first iteration. Subsequent iterations of analysis further refine the system requirements through the development of use case scenarios, class diagrams, sequence diagrams, state chart diagrams, and so on. Each iteration takes a successively more detailed look at the design of the system are clearly and precisely defined within the UML documents.

Unified Modeling Language (UML) provides a standardized set of tools to document the analysis and design of a software system. UML is fundamentally based on an 0 – 0 technique known as use case modeling. A use case model describes what a system does without describing how the system does it. The primary components of UML are called *“things”* structural things are most common; they include classes; interfaces, use cases, and many other elements that provide a way to create models. Structural things allow the user to describe relationships. There is also behavioural ‘things’ which describes how things work, the group things are used to define boundaries while   things permit the analyst to add notes to the diagrams.

A use case model partitions system functionality into behaviours (called use cases) that are significant to the users of the system (called actors).

Relationships are the glue that holds the things together. Structural relationships include dependencies, aggregates, associations and generalization. Behavioural relationships are used in the behavioural diagrams to show communication. The UML diagrams are of two types: structural diagram and behavioural diagram.

3.3 ACCELERATED SYSTEM ANALYSIS APPROACHES

Discovery prototyping is an example of ASAA that emphasize the construction of prototypes that identify the business, the users and the managers. Prototypes are incomplete samples of a desired system. By incomplete, we mean that a prototype will not include the error checking, the input data validation, security and processing completeness of a finished application.

DISCOVERY PROTOTYPING

It uses rapid development technology to help users discover their business requirements. System Analysts can rapidly create an information system using simpler tools like MICROSOFT ACCESS. The aim is to develop the final new system in a more sophisticated application development tool. While tools like Microsoft Access can indeed accelerate system development, their use in discovery prototyping is fast only because we omit some details in database and application programming required for a complete and secure application.

Requirement Discovery Method

Requirement Discovery is the process used by system analysis in identifying or extracting system problems with solution requirements from the user community. Both model-driven and accelerated prototype system analysis approaches attempt to express user requirements for the new system either as models or as prototypes. These approaches are dependent on the need to actually identify and manage those requirements.

Furthermore, the requirements for systems are dependent on the analyst ability to discover or the problems and opportunities that exist in the current system. There are different approaches to requirement discovery

1. Facts finding techniques
2. Joint Requirement Planning

- **Facts finding** is the process of collecting information about system problems, opportunities, solution requirements and priorities. Facts-finding is an essential skill for all systems' analyst. The facts-finding techniques include

  a) Sampling of existing documentation, reports, forms, files, data bases and memos.

  b) Research on relevant literature, bench marking of other solutions

  c) Observation of the current system in action and the work environment

d) Questionnaires and surveys of the management and user community.

e) Interviews of appropriate managers, users and technical staff

- **Joint Requirement Planning (JRP)**

JRP is the use of facilitated workshops to bring together all of the system owners, users and analysts and some system designers and builders to jointly perform system analysis.

3.4 Decision-Analysis Phase

Given the business requirements for an improved information system, we can address how the new system might be implemented technology. The purpose of this phase is to identify candidate solutions and recommend a target system that will be designed constructed and implemented.

The basic tasks to be performed are

1. Identity candidate solution
2. Analyze candidate solution
3. Compare these candidate solutions
4. Update the project plan
5. Recommend a system solution

I. Facts-Finding Techniques for Requirements Discovery

Effective fact-finding techniques are crucial to the report of system projects because to develop such systems, we first must be able to correctly identify, analyze, and understand what the user's requirements are or what the users want the system to do. The process techniques that a systems analysts uses to identify, analyze and understand system requirements are referred to as requirement discovery.

One of the best ways to get requirement discovery done is to talk to the people who are directly or indirectly involved in the different parts of the organization affected by the possible system changes: users, managers, funders etc. Another way to find out about the current system is to gather copies of documentation relevant to current systems and business processes.

Interviewing and Listening

Interviewing is one of the primary ways analysts gather information about an information system project. Early in a project, an analyst may spend a large amount of time interviewing people about their work, the information they use to carry out their work and the types of information processing that might supplement their work. During interviewing, you will gather facts, opinions and speculations and observe body language, emotions and other signs of what people want and how they assess current system. Some of the points to be kept in mind while interviewing are:

1. What data to collect?
2. On what to gain agreement
3. What areas to explore

In order to achieve this, we must have the following guidelines

1. Plan the interview- Prepare the interviewee: appointment, priming questions

- Prepare checklist, agenda and questions.

2. Listen carefully and take notes (take record of permitted)
3. Review notes within 48hrs of interview
4. Be neutral.

II. Crossing Interview Questions

Open-ended and Closed ended questions to be used must be decided upon. Open-ended questions are usually used to probe for information for which you do not know the precise question to ask.

An example is this: what will you say is the best thing about the info-system you currently use to do your job or list the three most frequently needed data? One advantage of open-ended questions main interview is that previously unknown information can surface. You can then continue exploring questions along unexpected lines of inquiry to reveal more new information.

Open-ended questions also often put the interviews (a) ease because they are able to respond in their own words using their own structure. Open-ended questions give interview e more of a sense of involvement and control in the interview. A major disadvantage is the length of times it can be take for questions to be answered.

Closed-Ended Questions

Closed-ended questions provide a range of answers from which the interviewee may choose. The interviewee is to choose only one option. They are (the range)

a. Having easy access to all of the data you need

b. The system's response time

c. The ability to access the system from remote locations.

Interview Guidelines

1. Do not phrase a question in a way that implies a right or wrong answer either with open or closed ended question.
2. Listen very carefully to what is being said.

3. Take careful note or if possible record the interview on a tape recorded (Be sure to ask for permission first. The answers may contain extremely important information for the project. Also, this may be the only chance you have to get information from this particular person.
4. Once the interview is over, go back to your office and type up your notes within 48hrs, your memory of the interview will fade quickly. As you organize your note, write down any additional questions that might arise from lapses in your notes or from ambiguous information. Separate facts from your opinion. Make a list of unclear points that need clarification. Finally, make sure you thank the person for his/her time.
5. Be careful during the interview not to set expectations about the new or replaced system. Let the respondent know that their ideas will be carefully considered along with what is technically possible but that due to the native nature of the system development process, it is premature to say what the ultimate system will or will not do.
6. Seek a variety of perspectives from the interviews. Find out what potential users of the system, users of other systems that might be affected by changes, managers, supervisors, information system staff who have experience with the current system. You want to understand all possible perspectives so that in a lather approval step, you will have information on which to base a recommendation or design decision that all stakeholders can accept.

III. SAMPLING OF EXISTING DOCUMENTARY FORMS AND FILES

The first document the analysts may wish to seek for is the organization chart. An organization chart serves to identify key individual owners and users for a project of their reporting relationships. The analyst may also want to trace the history that led to the project. To accomplish this, the analyst should collect and review documents that describe the problem. Also, there are documents that describe the business function being stored or designed. This may include

1. The companies mission statement
2. Formal objectives for the organization submits
3. Samples of manual and computerized database.

IV. RESEARCH AND SITE VISIT

This technique has to do with researching the problem domain thoroughly, most problems are not completely unique i.e. other people have solved them before us. Computer journals and reference books are a good source of information. Exploring the internet and intranet can provide immeasurable amount of information. It also involves visiting other companies or departments that have addressed similar problems.

V. OBSERVATORY OF THE WORK ENVIRONMENT

This is an effective fact finding technique where in the systems Analyst either participates in or watches a person perform activities to learn about the system. This technique is often used when the validity of data collected through other methods is in question or when the complexity of certain aspects of the system prevents a clear explanation by the end-users.

Observation can be a very useful and beneficial that finding technique provided that you have the ability to observe all aspects of the work being performed by the users and that the work is being performed in the usual manner.

Advantages

1. Data gathered by this techniques can be very reliable
2. The systems Analysts is able to see exactly what’s being done
3. Observation is relatively inexpensive compared with other fact finding techniques

Other techniques usually require substantially more employee release time.

Disadvantages

1. Because usually people feel uncomfortable when being watched, they may unwittingly perform differently when being observed
2. The task being observed are subject to various types of interruption
3. Some tasks may not always be performed in the manner in which they are observed by systems Analyst.
4. People may let analyst see what they want him to see

Guidelines for Observation

An analyst should plan to observe a site when there is "typical workload”. The following guidelines are keys to observation skills.

1. Determine the who, what, where, when, why and how of the observation
2. Obtain permission to observe from appropriate supervisors
3. Keep a low profile
4. Take notes during observations
5. Don’t interrupt individual at work
6. Don’t make assumptions.

VI. QUESTIONNAIRES

These allow the analyst to collect facts from a large number of people while maintaining uniform responses. When dealing with a large audience, no other technique can tabulate the same facts as efficiently.

4.0 MODELING SYSTEM REQUIREMENTS WITH USE CASE

Capturing and documentary system requirements have proved to be a critical outcome of a successful information system development project. Documentary the requirement from the perspective of the users in a manner that they can understand, promote users' involvements which greatly enhances the probability for the success of the project. One of the primary challenges of vital importance to any information system development team is the ability to elicit the correct and necessary system requirement from the stakeholders and specify them in a manner that's understandable to the stakeholders in other for those requirements to be verified and validated.

The difficulty of specifying requirements, especially functional requirements has played information technology community for long. In the past, we had tools like data models, process models, prototypes and requirement specification that we used but they were hard to understand for any user who wasn't educated in software development practices. USE case modeling has its roots in object oriented modeling. USE case modeling has proved to be a valuable aid in meeting the challenges of determining what a system is required to do from a user and stakeholder's perspective and it's now widely recognized as the best practice for the defining, documenting and understanding information systems functional requirement using USE are modeling facilitates and encourages user involvement which is one of the primary critical success factors for ensuring project success.

4.1 System Concepts for Use-Case Modeling

There are two primary artifacts involved when performing USE case modeling. The first is the USE CASE diagram which graphically depicts the system to the collection of USE cases, actors (users) and their relationships. This diagram communicates at a high level the scope of the system event that must be processes by the system.

The second artifact is the USE case narrative which describes the details of each business event and how the users interact with the system.

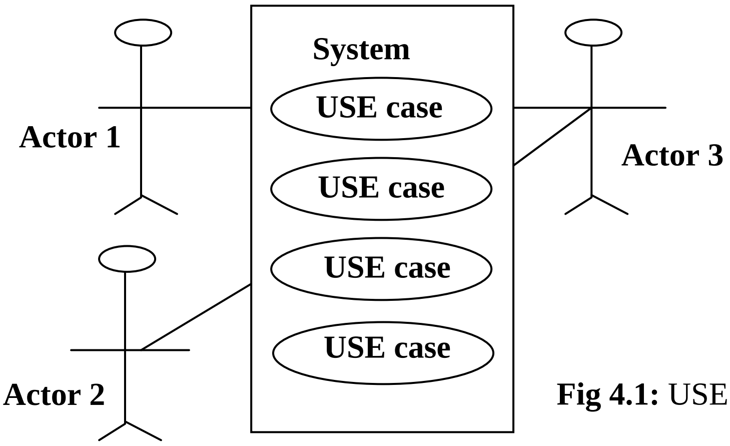


Fig 4.1: USE case diagram

USE case modeling identifies and describes the system function by using a tool called **USE** cases. USE cases describe the system functions from the perspective of external users and in a manner and terminology they understand. In accurately and thoroughly accomplish demands a high level of user involvement and a subject expert who is knowledgeable about the business or system event. USE cases are the results of decomposing the scope o f system functionality into many smaller or statements of system functionality. They are represented graphically by the horizontal ellipse with the name of the USE case appearing above, below or the ellipse. A USE case represents a single goal of the system and describes the sequence of activities and user interactions in trying to accomplish the goal. USE cases are initially defined during the requirement stapes of the life cycle and will be additionally refined throughout the life cycle.

ACTORS

Actor is anything that needs to interact with the system to exchange information. Actors are external users that initiate or trigger USE cases. An actor initiates system activity for the purpose of completing some business tasks that produces something of measurable value.

There are primarily four types of actors. They are

- **Primary Business Actor:** This is the stakeholder that primarily benefits from the execution of the USE care by receiving something of measurable value.

The PBA may or may not initiate the business event for example, the system involving the payment of an employee using a pay slip for the online registration; the student is both the Primary Business Actor and Primary System Actor (PSA).

- **Primary System Actor:** This is the stakeholder that directly interfaces with the system to initiate or trigger the system event. PSAs may interact with PBAs for the purpose of hire the actual system e.g. banking system.

  They facilitate the event than the direct use of the system for the benefit of a PBA. Examples are clerk in a grocery store, telephone operator. The PBA and PSA may be the same person for events where the business actor interfaces with the system directly e.g. cash
- **External Server Actor:** This is the stakeholder that responds to a request from the USE case.
- **External receiver Actor:** This is the stakeholder that's not the primary actor but receives something of measurable value from the USE case.

RELATIONSHIPS

A relationship is deputed as a line between the symbols on the USE case diagram. The meaning of the relationships may differ depending on how the lines are drawn and what type of symbol they connect.

ASSOCIATIONS

This is a kind of relationship between an actor and a USE case whenever the USE case describes an interaction between them. An association is modeled as a solid line connecting the actor and the USE case. An association that contains an arrowhead on the end touching the USE case indicates:

i) That the USE case was initiated by the actor on the other end of the line

Association without arrowheads

ii) An interaction between the USE case and an external server or receiver actor.

When an actor is associated with a USE case, we say the actor communicates with tshe USE case. Associations may also be bi-directional or uni-directional

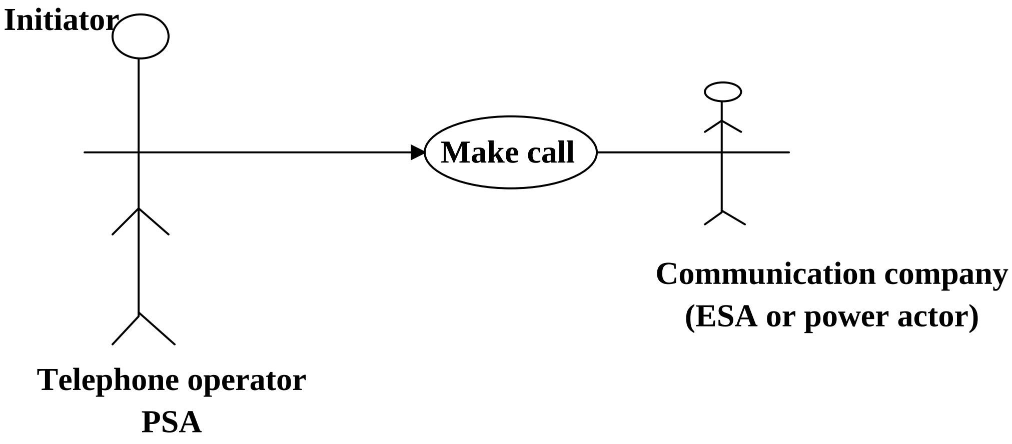


Fig. 4.2: Bi-directional Association

Extends

A USE case may contain complex functionalities consisting of several steps making the USE case logic difficult to understand. For the purpose of sampling the USE case and making it more easily understood, we can extract the more complex steps into their own USE case. The resulting USE case is called an USE case in that it extends the functionality of the original USE case. The relationship between the extension USE case and the USE case it's extending is called an extends relationship. A USE case may have many extends relationship but an extension USE case can be invoked only by the USE case it's extending.

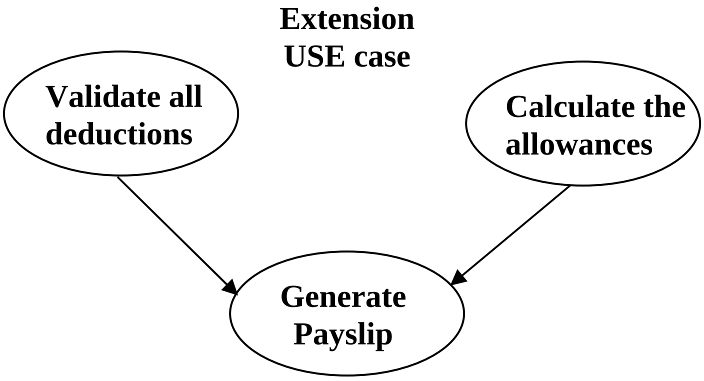


Figure 4.3: Extension of Use CASE

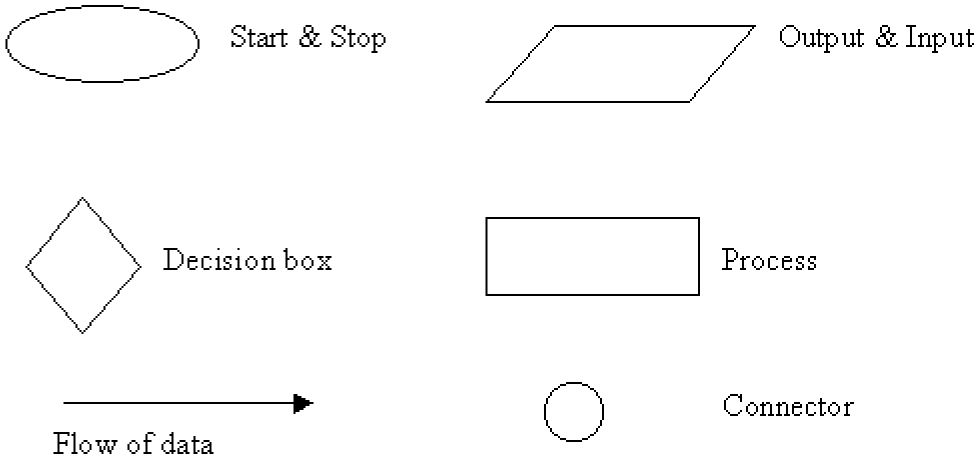
5.0 SYSTEM DESCRIPTION TECHNIQUES

5.1 INTRODUCTION

Graphical representation of any process is always better and more meaningful than its representation in words. Moreover, it is very difficult to arrange and organise the large amount of data into meaningful interpretation of the whole. System Analysis and Design makes use of the various tools for representing and facilitating comprehension of the complex processes and procedure involved. In this lesson, we present some details about Flowcharts, data flow diagram (DFD), Decision Tables and Decision Trees.

5.2 FLOWCHARTS

The pictorial representation of the programs or the algorithm is known as flowcharts. It is nothing but a diagrammatic representation of the various steps involved in designing a system. Some of the boxes which are used in flowcharts are:



A flowchart consists of a set of 'flowchart symbols' connected by arrows. Each symbol contains information about what must be done at that point & the arrow shows the 'flow of execution' of the algorithm i.e. they show the order in which the instructions must be executed. The purpose of using flowcharts is to graphically present the logical flow of data in the system and defining major phases of processing along with the various media to be used.

Flowcharts are of three types:

- System flowcharts
- Run flowcharts
- Program flowcharts

(a) System Flowcharts

System flowchart describes the data flow for a data processing system. It provides a logical diagram of how the system operates. It represents the flow of documents, the operations performed in data processing system. It also reflects the relationship between inputs, processing and outputs. Following are the features of system flowcharts:

- the sources from which data is generated and device used for this purpose
- various processing steps involved
- the intermediate and final output prepared and the devices used for their storage

Figure 5.1 is a sample of system flowchart for the following algorithm:

1. Prompt the user for the centigrade temperature.
2. Store the value in C
3. Set F to 32+(9× C/5)
4. Print the value of C , F
5. Stop

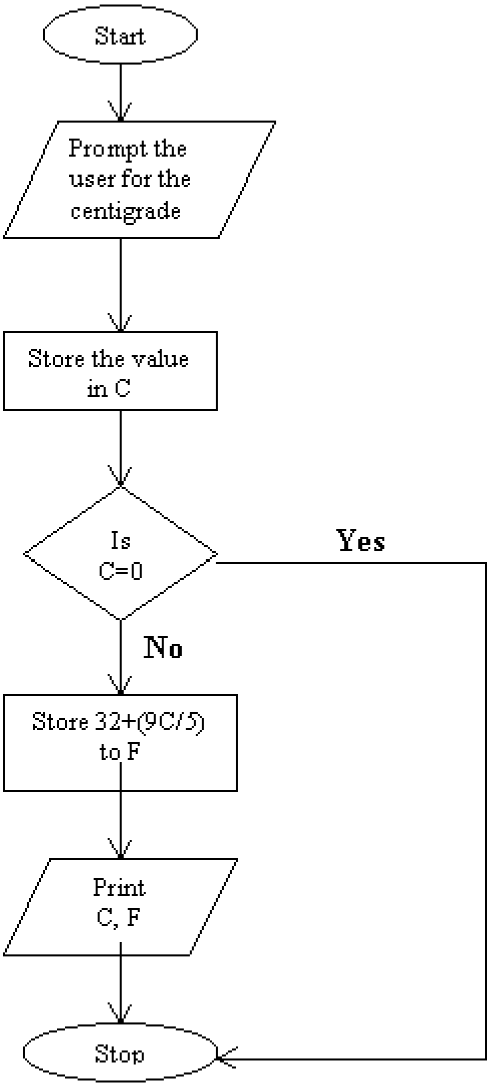


Figure: 5.1 A System Flowchart

(b) Run flowcharts

Run flowcharts are used to represent the logical relationship of computer routines along with inputs, master files, transaction files and outputs. Figure 5. 2 illustrates a run flowchart.

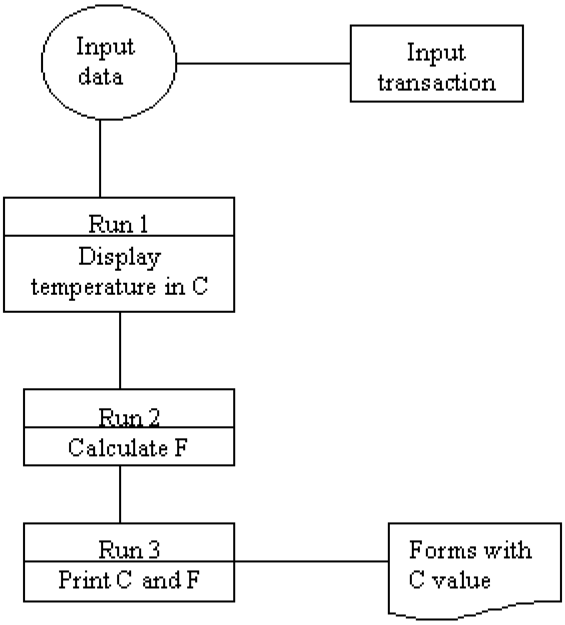


Figure: 5.2 Running a Flowchart

(c) Program flowcharts

A program flowchart represents, in detail, the various steps to be performed within the system for transforming the input into output. The various steps are logical/ arithmetic operations, algorithms etc. It serves as the basis for discussions and communication between the system analysts and the programmers. Program flowcharts are quite helpful to programmers in organising their programming efforts. These flowcharts constitute an important component of documentation for an application.

Figure 5.3 represents a program flowchart for finding the sum of first five natural numbers (i.e. 1,2,3,4,5).

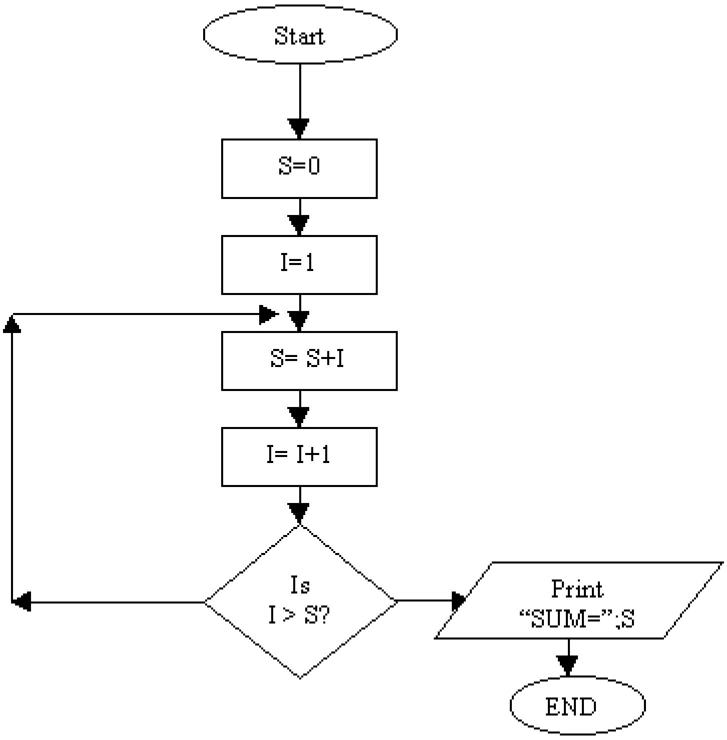


Fig 5.3 Program Flowchart

(d) Data flow diagram

Data flow diagrams are the most commonly used way of documenting the process of current & required systems. As their name suggests they are a pictorial way of showing the flow of data into, around & out of a system.

(e) Defining DFD

Graphical representation of a system's data and how the processes transform the data is known as Data Flow Diagram (or DFD). Unlike, flowcharts, DFDs do not give detailed descriptions of modules but graphically describe a system's data and how the data interact with the system.

(f) Components of DFD

DFDs are constructed using four major components

- external entries
- data stores
- processes and
- data flows

(i) External Entities

External entities represent the source of data as input to the system. They are also the destination of system data. External entities can be called data stores outside the system. These are represented by squares.

(ii) Data Stores

Data stores represent stores of data within the system. Examples are computer files or databases. An open-ended box represents a data/store – data at rest or a temporary repository of data.

(iii) Process

Process represents activities in which data is manipulated by being stored or retrieved or transferred in some way. In other words we can say that process transforms the input data into output data. Circles stand for a process that converts data into information.

(iv) Data Flows

Data flows represent the movement of data from one component to the other. An arrow identifies data flow – data in motion. It is a pipeline through which information flows. Data flows are generally shown as one-way only. Data Flows between external entities are shown as dotted lines.

(g) Physical & Logical DFD

Consider the figure30.4. It is clear from the figure that orders are placed, orders are received, the location of ordered parts is determined and delivery notes are dispatched along with the order.

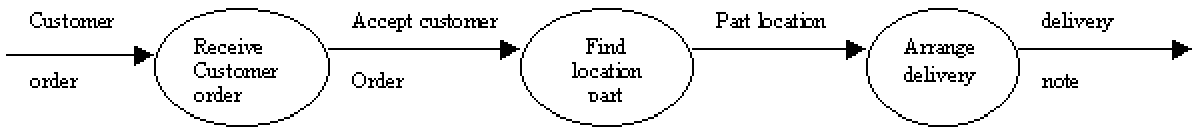


Fig 5.4

It does not however tell us how these things are done or who does them. Are they done by computers or manually and if manually who does them ? A logical DFD of any information system is one that models what occurs without showing how it occurs.

A physical DFD shows, how the various functions are performed? Who does them? Consider the following figure:

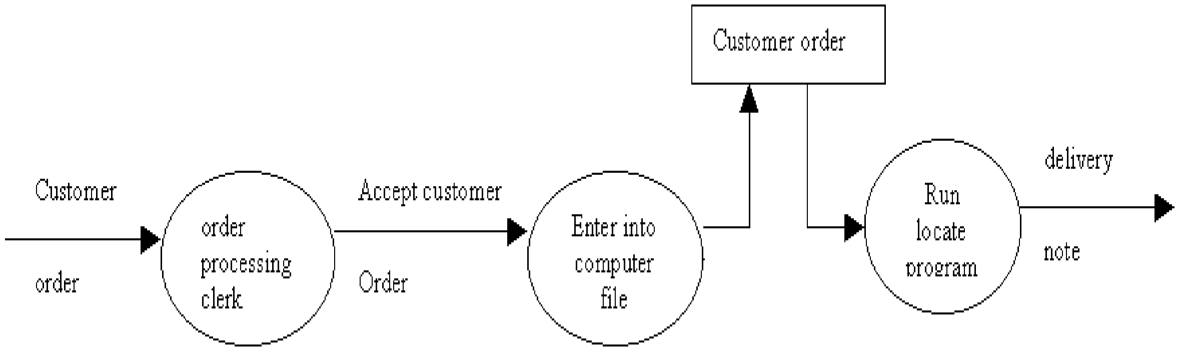


Fig 5.5

The figure 30.5 is opposite, it shows the actual devices that perform the functions. Thus there is an "order processing clerk", an "entry into computer file" process and a "run locate program" process to locate the parts ordered. DFD that shows how things happen or the physical components are called physical DFD(s).

Typical processes that appear in physical DFDs are methods of data entry, specific data transfer or processing methods.

(h) Difference between flowcharts & DFD

The program flowchart describes boxes that describe computations, decisions, interactions & loops. It is an important to keep in mind that data flow diagrams are not program flowcharts and should not include control elements. A good DFD should

- have no data flows that split up into a number of other data flows
- have no crossing lines
- not include flowchart loops of control elements
- not include data flows that act as signals to activate processes.

5.4 DECISION TABLES AND DECISION TREES

Decision tables and trees were developed long before the widespread use of computers. They not only isolate many conditions and possible actions but they help ensure that nothing has been overlooked.

(a) Decision Tables

The decision table is a chart with four sections listing all the logical conditions and actions. In addition the top section allows space for title, date, author, system and comment as shown in the fig.30.6

Five sections of a decision table:

TITLE : Author : Comments :	DATE : System :
Condition Stub	Condition Entry
Action Stub	Action Entry

Table 5.1: Decision Table

The **condition stub** contains a list of all the necessary tests in a decision table. In the lower left-hand corner of the decision table we find the action stub where one may note all the processes desired in a given module. Thus **Action Stub** contains a list of all the processes involved in a decision table.

The upper right corner provides the space for the **condition entry** - all possible permutations of yes and no responses related to the condition stub. The yes and no possibilities are arranged as a vertical column called rules. Rules are numbered 1,2,3 and so on. We can determine the rules in a decision table by the formula:

Number of rules = 2^N = 2^N where N represents the number of condition and ^ means exponentiate. Thus a decision table with four conditions has 16 (2^4 = 2 x 2 x 2 x 2 = 16) rules one with six conditions has 64 rules and eight conditions yield 256 rules.

The **Condition entry** contains a list of all the yes/no permutations in a decision table. The lower right corner holds the action entry. X's or dots indicate whether an action should occur as a consequence of the yes/no entries under condition entry. X's indicate action; dots indicate no action.

Thus Action entry indicates via dot or X whether something should happen in a decision table. Let us consider the following example of book order illustrated by figure 30.7

If order is from book store

And if order is for 6 copies

Then discount is 25%

Else (if order is for less than 6 copies)

No discount is allowed

Else (if order is from libraries)

If order is for 50 copies or more

Then discount is 15%

Else if order is for 20 to 49 copies

Then discount is 10%

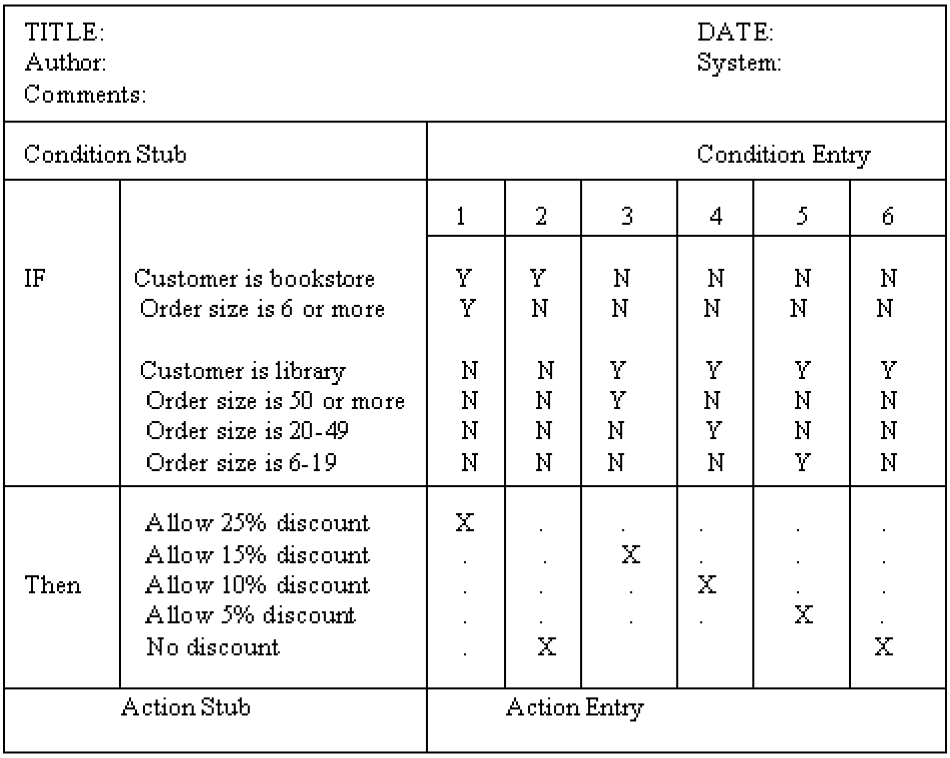
Else if order is for 6 to 19 copies

Then discount is 5%

Else (order is for less than 6 copies)

No discount is allowed

A decision table for the above process is illustrated below



TITLE: Author: Comments:					DATE: System:		
Condition Stub				Condition Entry			
		1	2	3	4	5	6
IF	Customer is bookstore	Y	Y	N	N	N	N
	Order size is 6 or more	Y	N	N	N	N	N
	Customer is library	N	N	Y	Y	Y	Y
	Order size is 50 or more	N	N	Y	N	N	N
	Order size is 20-49	N	N	N	Y	N	N
	Order size is 6-19	N	N	N	N	Y	N
Then	Allow 25% discount	X	.	.	.	.	.
	Allow 15% discount	.	.	X	.	.	.
	Allow 10% discount	.	.	.	X	.	.
	Allow 5% discount	.	.	.	.	X	.
	No discount	.	X				X
Action Stub		Action Entry					

Table 5.2: Decision Table

(b) Decision Tree

The decision tree defines the conditions as a sequence of left to right tests. A decision tree helps to show the paths that are possible in a design following an action or decision by the user. Figure 30.8 illustrates the concept of decision tree.

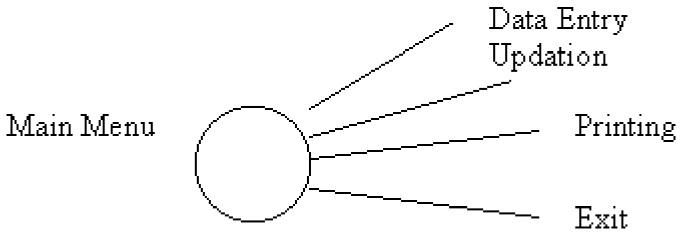


Figure 5.6: Decision Tree

Decision tree turns a decision table into a diagram. This tool is read from left to right, decision results in a fork, and all branches end with an outcome. Figure 6 illustrates the decision tree for the book order decision table we saw earlier.

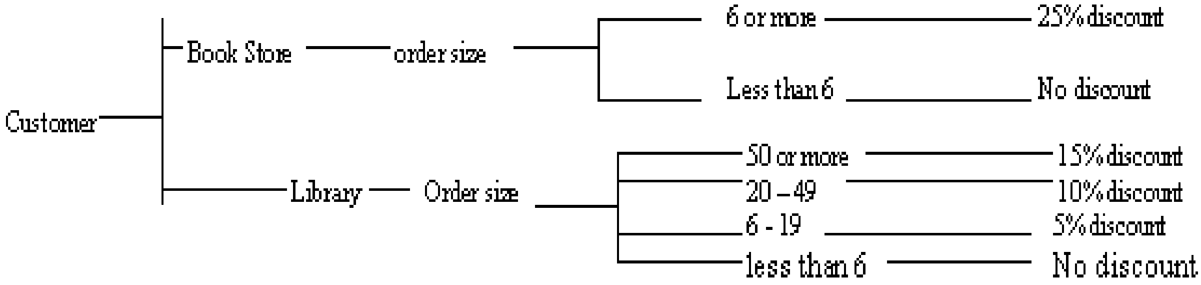


Figure 5.7: Decision Tree for Book Order

6.0 SYSTEM DEVELOPMENT METHODOLOGIES

6.1 INTRODUCTION

Different types of system development methodologies are used in designing information system. Depending upon the actual requirement of the system, different approaches for data processing are adopted. However, some system groups recommend the Centralised data processing system while others may go in for distributed data processing system. In a Centralised data processing, one or more centralized computers are used for processing and the retrieval of information is done from them. The distributed processing systems involve number of computers located remotely in the branches/departments of the organisation. The client/server technologies are also gaining popularity these days.

6.2 DATA PROCESSING SYSTEM

Data processing techniques are very much dependent on the kind of applications and the working environment. The activities involved in the data processing are along departmental lines and are application based such as Store Management, Production Planning & Control, Sales Accounting, Financial accounting, Student Information System, and so forth. The basic input data are the real resource of the data processing. With the increase of the technologies the concept of the integrated data processing also came into being where the output data of one application can be used as the input of another application. Depending upon the application area, working environment and the needs of the management there are basically two approaches of data processing:

- Centralised data processing
- Decentralised data processing

6.3 CENTRALISED DATA PROCESSING SYSTEM

With the increasing use of computer based data processing, there has been a growing tendency in the minds of management to centralise the data processing activities. A separate department EDP (Electronic Data Processing) department is established to carry out the data processing work of different department in the organisation. Many a times the data processing is also done by hiring the services of the outside agencies and with the passage of time and experience in-house set is developed for data-processing.

The centralised data processing system provides the following benefits:

- The emergence of data takes place only at one place.
- The loss of data is minimised.
- The methods and machines can be standardised.
- Services of more competent and technical personnel can be taken.
- It is also very cost-effective particularly in the case of large operations.
- Duplication of work can be avoided.

The disadvantages, however, are:

- Lack of cooperation from managers, who do not like to be under control of centralised Data Processing department.
- Resistance from managers for mechanising the data processing activities relating to their various functions.
- It is difficult to provide equitable services to various departments.
- The data security is also questioned.

6.4 DECENTRALISED DATA PROCESSING SYSTEM

In the decentralized data processing system, there is really a divisional breakdown of computing services. Each division, unit or department handles its own computer needs and does not like to interact with any other division, unit or department. It is well suited to a decentralized management scheme in which organizational autonomy is important. For example, research divisions of large organisation may adopt the decentralized data processing approach to provide data security of their work

Arguments in the support of decentralized data processing include the following:

- Familiarity with local problems.
- Rapid response to local processing needs
- Profit-and-loss responsibility can be easily fixed

The drawbacks of the decentralized data processing system are:

- There is duplication of activities and redundancy in the maintenance of files.
- It is difficult to maintain uniformity in the procedures throughout the organisation.
- The overall cost of the data processing for the organisation is more.

6.5 INFORMATION SYSTEM

The information system aims at providing detailed information on a timely basis throughout the organisation so that the top management can take proper and effective decisions. The information system cuts across departmental lines and help achieving overall optimization for the organisation.

The organisation is viewed as a network of inter-related sub-systems rather than as a hierarchy of manager-subordinate relationship. The information system can be of two types:

- Integrated information system
- Distributed information system

(a) Integrated Information System

The integrated information system is based on the presumption that the data and information are used by more than one system in the organisation and accordingly, the data and information are channeled into a reservoir or database. All the data processing and provision of information is derived and taken from this common database. The development of an integrated information system requires a long-term overall plan, commitment from management at all levels, highly technical personnel, availability of sufficient fund, and sophisticated technology. It also requires adequate standby facilities, without which the system is doomed to failure. Because of its integrated component, the modification to the system is quite difficult and the system development takes a fairly long time.

(b) Distributed Information System

There are opinion that development of an integrated information system is embodied with several practical problems and therefore, not feasible. This view has been reinforced by the failure of integrated systems in various large organisations. The concept of a distributed information system has emerged as an alternative to the integrated information system. In the distributed information system, there are information sub-systems that form islands of information systems. The distributed information system aims at establishing relatively independent sub-systems, which are, however, connected through communication interfaces.

Following are the advantages of the distributed information system:

- The processing equipment as well as database were dispersed, bringing them closer to the users.
- It does not involve huge initial investment as is required in an integrated system.
- It is more flexible and changes can be easily taken care of as per user's requirements.
- The problem of data security and control can be handled more easily than in an integrated system.
- There is no need of standby facilities because equipment breakdowns are not as calamitous as in an integrated system.

The drawbacks of the distributed system are:

- It does not eliminate duplication of activities and redundancy in maintaining files.
- Coordination of activities becomes a problem.
- It needs more channels of communication than in an integrated system.

It is possible to consider several alternative approaches, which fall between the two extremes - a completely integrated information system and a totally independent sub-system. It is to be studied carefully what degree of integration is required for developing an information system. It depends on how the management wants to manage the organisation, and the level of diversity within the organisation.

6.6 MULTI-USER ENVIRONMENT

The necessity of sharing of data and information gave rise to multi-user environment. In a multi-user environment, there is a concept of file server and user nodes or user terminals connected to the file server. There are various ways of developing a multi-user environment depending upon the connectivity. There is local area network (LAN) where nodes are connected with the file server with cables through which the data and information are transferred from file server to the different nodes connected to the file server and vice-versa. In a Wide Area Network, the nodes are connected through MODEM or through satellite.

6.7 **NETWORK/FILESERVER SYSTEM**

In a Local Area Network, all the data and programme files are stored in a file server. A file server is the central node in the network. All the users connected to the file server through different nodes can access the data and information stored in the fileserver simultaneously. The file server in a LAN acts as a central hub for sharing peripherals like, printers, modems, etc. In a LAN, an application running on a workstation reads and writes files on the file server. In many cases the entire files are pumped across the network on behalf of the operations taking place on LAN PCs. A file server does not involve in processing of an application. It simply stores files for applications that run on LAN PCs. For example, you might have a personal database manager and then request information in a file on the on the file server. The file server sends all or part of the data file across the network to your workstation. As you work with your personal database manager and the database on your workstation, the file server does not take part at all when you save the file back to the file server across the network.

Two flaws limit a file server system for multi-user applications. First, the file server model does not deliver the data concurrency (simultaneous access to a single data set by more than one user), that is required frequently by multi-user applications. The reason behind it is that the file server operates in files, which are set of large number of data records and prevent a user from sharing a file when another user has it locked out. Second, if many workstations request and send many files in a LAN, the network can quickly become saturated with traffic, creating a bottleneck that degrades overall system performance.

6.9 CLIENT /SERVER SYSTEM

The limitations of the network/file server system have led to the genesis of the client/server system. It delivers the benefits of the network-computing model along with the stored data access. Any local area network could be considered as client/server system, since workstations (clients) request services such as data, program file or printing from server.

A client/server has three distinct components, each focusing on a specific job: a database server, a client application and a network.

6.10 Database Server

A server (or "back end") manages the resources such as database, efficiently and optimally among various clients that simultaneously request the server for the same resource. Database server mainly concentrates on the following tasks:

- Managing a single database of information among many concurrent users.
- Controlling database access and other security requirements.
- Protecting database of information with backup and recovery features.
- Centrally enforcing global data integrity rules across all client applications.

6.11 Client Application

A client application (the "front end") is the part of the system that users apply to interact with data. The client application in a client/server model focus on the following job:

- Presenting an interface between the user and the resource to complete the job.
- Managing presentation logic.
- Performing application logic
- Validating data entry
- Managing the request traffic of receiving and sending information from a database server.

6.**12** **Network**

The third component of a client/server system is network. The communication software are the vehicles that transmit data between the clients and the server in client server system. Both the client and the server run communication software that allows them to talk across the network.