Chapter Five

SYSTEM ANALYSIS

5.1 INTRODUCTION

System analysis is an interdisciplinary branch of science that deals with the analysis of system; and prior to its automation as computer systems and the interactions within those systems. The term analysis came from classical Greek which means "***to loosen up***". Generally, analysis is defined as the procedure by which we break down a system into parts or components.

In this chapter, more shall be discussed about system analysis; which is an essential part of the systems development process. Without having a comprehensive and detailed knowledge of user requirements, it may be practically impossible to develop an effective and suitable information system. This chapter explores the process of system analysis and some of the tools and techniques of capturing user requirements in the course of developing an information system.

5.2 SYSTEM ANALYSIS

The term system analysis can be defined as a problem-solving technique that decomposes a system into its component pieces for the purpose of studying how well those components parts work and interact to accomplish their purpose (Whitten et al, 2000).

The first three phases in systems development are: the preliminary investigation, problem analysis, requirements analysis and decision analysis. The phases are collectively referred to as systems analysis.

System analysis stage describes the early phase of system development which consists of the definition of the problem, identifying the cause, specifying the solution and recognizing the information requirements that must be met by the system solutions.

Some other activities involved in system analysis are:

- **Feasibility Study**: System analysis also involves a feasibility which is to determine whether the solution is feasible or achievable given the organization resources and other constraints.

- **Information Requirement**: This is the most different aspect of the system analysis which defines the specific information requirement that must be met by the system solution selected. In most cases the basic information requirements of a new system involve identifying who needs, what information, where, when and how. Requirement analysis also states the objectives of a new or improved system, economics, technical and time constraints, as well as the organization goals, procedures and decision process.

5.3 SYSTEM ANALYSIS TECHNIQHUES

Basically, system analysis is all about solving problems. There are so many techniques of solving problem; invariably many approaches or strategies for performing system analysis.

Some of these popular techniques are:

- Model-Driven Analysis Approaches: (i). Modern Structured Analysis
  (ii). Information Engineering
  (iii). Object–Oriented Analysis (OOA)

- Accelerated Analysis Approaches:
  (i). Prototyping Acceleration Analysis

- Requirements Discovery Methods:
  (i). Joint Application Development (JAD)
  (ii). Fact-Finding Techniques

- Business Process Redesign (BPR)
  (i). FAST Systems Analysis Techniques

5.3.1 Model–Driven Analysis

This is one of the oldest and most commonly used techniques for analyzing and designing information system based on modeling. Model–driven development defines business requirements and information system designs by drawing models which becomes the blueprints for constructing the final system.

Modeling is an act of drawing one or more graphical representation or pictures of a system. It is also a communication technique which is based on the saying "**a picture is worth a thousand words**". Examples of familiar model include data flow diagram (DFD), flowchart and organizational chart.

These days' model driven techniques can be enhanced with the use of automated tools. The tools that are commonly used by some analysts for drawing system models are the general-purpose graphics software such as Visio Professional or Corel Flow. Other analysts and organizations use repository–based CASE or modeling tools such as System Architect, Visible Analyst, Rational Rose and Visio Enterprise.

(i). Modern Structure Analysis

Modern structured analysis is a model-driven, process centered technique that is used to either analyze an existing system, define a business requirement for a new system or both. It is one of the first formal techniques for systems analysis of information systems and still one of the most widely practical techniques. This strategy is very simple in concept and lays initial emphasis on the process building blocks in the information system framework; this is referred to as **process–centered**.

(ii). Information Engineering (IE)

Information Engineering (IE) is a model–driven and data–centered, but PROCESS–sensitive technique that is used to plan, analyze and design information system. It is said to be data–centered because it lays emphasis on the study and requirements analysis of data requirements before that of the process and interface requirements.

The data models in information engineering are referred to as entity relationship diagrams. The system analyst draws entity relationship diagrams (ERDs) to model the systems raw data before drawing the data flow diagrams which illustrate how data will be captured, stored, used and maintained. These days' many organizations have adopted information engineering modeling techniques.

(iii). Object–Oriented Analysis
This is a model–driven approach that incorporates data and process concerns into constructs called **objects**. The OOA are pictures that show the system's objects from various perspectives such as structure and behaviour. Object technology attempts to eliminate the separation of concerns about data and process. This technique is used to study existing objects to see if they can be reused or adapted for new uses and to define new or modified objects into a useful business computing application.

5.3.2 Accelerated Analysis
This technique lays emphasis on the concentration of prototypes which makes it very fast for identifying business and user requirement for a new system.

Prototyping technique is not limited to accelerated analysis approach only, it is also used by many techniques to perform rapid application development. **Prototype can be described as a small-scale, incomplete but working sample of a desired system**. Examples of accelerated analysis are: feasibility prototyping and discovery prototyping.

- **Feasibility Prototyping**: It is used to test the feasibility of a specific technology that might be applied to the business problem.

- **Discovery Prototyping**: This is sometimes called requirements prototyping and is used to identify the user's business requirements by having them react to a quick and-dirty implementation of those requirements. (whitten et al, 2000)

5.3.3 Requirements Discovery Methods
These are the techniques that are used by systems analysts for identifying or extracting system problems and solution requirements from users' group.

(i). Joint Application Development (JAD)
JAD describes a close partnership between users and developments. These include system owners, users, designers and builder that participate in the joint application; they jointly define and design the system. It is also referred to as Joint Application Design and Joint Requirement Planning.

(ii). Fact-Finding Techniques
This is an established set of techniques for collecting information about system problems and solution requirements from various sources. This technique includes the following which have been earlier discussed:

- Interviews of system end user, relevant manager and technical staff
- Observation of the existing system in its working condition
- Use of questionnaire
- Documentation through research of relevant literature books and system manual
- Inspection of records

5.3.4 Business Process Redesign (BPR)
This is a technique that is applied when the goal of systems development is to dramatically change and improve the fundamental business processes of an organization. BPR method is always conducted independent of information technology and is one of many types' projects set off by the trend we refer to as total quality management (TQM).

(i). FAST System Analysis Techniques
FAST is an acronym Framework for the Application of System Techniques; it is a methodology that supports different types of projects including application development,

information strategy planning, business area analysis, decision support system development and business process redesign.

5.4 MODELS

System models play an important role in system development. As a system analyst or user, you will constantly deal with structured problem and one of the ways of structuring such problems is to draw models.

Definition: **A model is a representation of reality**.
Model can be built for existing systems as a way to document business requirement or technical design.

There are two types of models, these are:

(i). Logical Model

Logical model or design involves describing a business requirements or purpose of a system to show what a system is or will do. It does not include any specific hardware or software requirements.

It is more concerned with the processes to be performed. More so the implementations of logical models are independent that is, they depict that system independent of any technical implementation. As such logical models illustrate the essence of the system. Logical model can also be referred to as essential model, conceptual model or business model. Example of a model that can be used to show and explain what a system will do is the Data Flow Diagrams (DFDs).

(ii). Physical Models

This model shows not only what a system is or does but also how system is physically and technically implemented. Physical models are implementation dependent because they reflect technology choices and the limitations of those technology choices. It also includes the technical specification for the hardware and software required.

5.5 DATA MODELLING

Data modelling is a representation of the properties of the data within an existing or proposed system. It can also be referred to as a technique for organizing and documenting a system data. Data modelling is sometimes called **database modelling** because a data model is usually implemented as a database.

The basic elements from which a model called modelling are constructed, and the complete process of constructing a data model from the scratch is called **data analysis**. Once the information concerning a proposed system has been captured, the next thing is to express or model it in a way that is useful for the system development process. And data can be modelled in several ways, among these are:

- Process Models
- Structure Models
- Event Models

5.5.1 Process Models

This model defines how data enters a system; how they are processed, stored and eventually how they bring output or transferred to another system. Process Model may be used to model business systems and to show the movement and processing actions the data is being subjected to. Examples of process models are data flow diagrams (DFD), flowcharts, decisions tables and pseudo-code. All these will be discussed extensively under programming tools.

5.5.2 Structure Models

A structure model provides an understanding of the logical data requirements of a system independent of the systems organization and process. There are several types of data model, but the commonest and successful type is the EAR model. Its basic elements are called entity attributes and relationships. Hence the name EAR model, or occasionally ER model—Entity Relationship Model (ERM).

The ERM can also be referred to as entity model or a logical data structure and shows a static view of the relationship between different entities but does not show how entities can change over time.

5.5.3 Event Models

An event is a logical unit of work that must be completed as a whole. An event is activated by a discrete input and is completed when the process has responded to appropriate outputs. According to Whitten et al (2000) events are sometimes called transactions.

There are three types of events namely:

- External events such as a customer placing an order.
- Internal events such as calculation of interest on late payments

Entity relationship model is a static diagram that shows the relationship between entities but does not demonstrate how entities can change over time. This changes that occur to data stored about entities are triggered by events.

En entity life history (EHL) is a type of event model that gives a dynamic view of data. They also display how an entity occurrence can come into existence (creation), how it can be modified and how it can be deleted from a system. It is also a diagram of the event that happens to an entity.

5.6 REPOSITORY

A repository is a location or set of locations where systems analysts, system designers and system builders keep the documentation associated with one or more systems or projects. A repository may be created for a simple project or shared by all projects and systems. It is normally implemented as some combinations of the following:

- A network directory of word processing, spreadsheet and other computer-generated files that contain project correspondence, reports and data.
- One or more CASE tool dictionaries or encyclopedia (CASE: Computer–Aided Software Engineering)
- Printed documentation such as that stored in builders and system libraries.
- An intranet website interface.

REVIEW QUESTIONS ON CHAPTER FIVE

1. (a) What do you understand by the term systems analysis?
   (b) Differentiate between feasibility study and information requirements which are some of the activities under systems analysis.

2. (a) Briefly explain the following systems analysis techniques and state the reason why you would prefer them in analyzing your system under investigation:
   (a) Modern Structured Analysis
   (b) Prototyping

(c) Joint application Development (JAD)
(d) Fact finding techniques

3. (a) Briefly explain the term repository
(b) Outline the common features and differences in the following systems analysis approaches:
(i) Information Engineering
(ii) Object-Oriented Analysis (OOA)

4. (a) What is the purpose of feasibility study in systems analysis?
(b) Briefly discuss the need for stage of systems analysis

5. (a) What is a model?
(b) Briefly explain the two types of models
(i) Logical model
(ii) Physical model

ASSIGNMENT

1. Identify and describe three fact-finding methods used by analysts to gather information about an existing system.

2. What is a feasibility report, and what key elements should it include?
3. List and explain the four major conversion strategies used during system implementation.
4. What are the three types of event models, and how do they differ from one another?
5. Describe the modern structured design technique, highlighting the importance of module cohesion and coupling.
6. Explain the prototyping approach to system design, including its advantages, disadvantages, and the iterative process.

Chapter Six

SYSTEM DESIGN

Learning Objectives:

After studying this chapter you will be able to:

- Explain System Design

- Identify and differentiate between several systems design strategies Differentiate between logical design and physical design. Understand the concept of technical system options
- Explain the tasks involved in completing systems design

6.1 INTRODUCTION

The term design or synthesis originated from classical Greek and means "**to put together**" it is a term that is often used with analysis in scientific discipline from mathematics and logic to economy and psychology to denote investigative and design activities. System design can be described as the opposite procedure to combine separate element or components in order to form a coherent whole. In this chapter you will learn more about the concept of system design. Also to be discussed are requirements specification, types and technical system options.

6.2 MEANING OF SYSTEMS DESIGN

System design also referred to as ***system synthesis*** is a complementary problem-solving technique that reassembles a system component pieces back into a complete system that is hoped to be an improved system. It also examines the design of systems from the perspective of both in-house development or build projects and software procurement or buy projects.

Systems design further details how a system will meet the information requirements as determined during the system analysis. Unlike system analysis that emphasizes business problems, systems design focuses on the technical or implementation of the system.

The design of an information system can be divided into two namely:

- Logical design
- Physical design

1. Logical Design: Logical design involves describing the purpose of a system; what the system will do. It does not include any specific hardware or software requirements. Logical lays out the system's components and their relationship to each other as they would appear to users. It also shows what the system will do as against to how it is actually implemented physically. Other activities under logical design are inputs and outputs, processing functions to be performed, business procedures, data models and controls. The models such as Data Flow Diagrams (DFD) or written descriptions can be used to show and explain what a system will do.

2. Physical Design: This is the process of translating the abstract logical model into the specific technical design for the new system. As stated by BPP professional education it is the actual; '***nuts and bolt***' of the system; it includes the technical specification that transform the abstract logical Design plan into a functioning system of people and machines.

Like houses or building, information systems may have many possible designs. They may be centralized or distributed, on-line or batch, partially manual or heavily automated. The remaining steps in the system development process translate the solution specifications established during system analysis and design into a fully operational information system.

6.3 TECHNICAL SYSTEM OPTIONS

Every organization will have to make choice concerning the specifications of the physical components of their new system. There are numbers of options available from which organization can make their choice regarding the following.

(1). Software: Use of specialist application software (bespoke software) or application packages (an off–the–shelf package). Specialist application software is designed for the specific needs of an organization. This software maybe written in–house if the organization is big enough to have its own programming team, or may be contracted to software developers; while application packages are pre-written and pre-coded suites of programs which are commercially made available for individuals and organizations to carry out their business functions, thereby eliminating the need for them to write their own software.

(2). Hardware Configuration: For eexamples mainframe, minicomputer, PC, centralized or distributed processing and Internet connections. The performance objectives for the system are then specified in details for it to be followed in the actual design of the system.

6.4 SYSTEM DESIGN TECHNIQUES

There are many techniques for performing systems design and these are viewed as competing alternatives, although in reality, certain combinations complement one another. Some of these approaches or techniques to systems design include the following:

- Modern Structured Design
- Information Engineering (IE)
- Object–Oriented Design (OOD)
- Rapid Application Development (RAD)
- Prototyping
- Joint Application Development (JAD)
- Fast System Design Strategies

Some of these techniques like modern structured design, information engineering and object–oriented design are using model–driven approaches.

Model–driven approaches to system design emphasizes on the drawing of pictorial system models in order to document the technical or implementation aspects of the system. The design models are often derived from the logical models earlier developed in model–driven analysis. The system design models acts as the blue prints for constructing and implementing the new system.

These days' model-driven approaches are enhanced by the use of automated tools. Some designers draw system models using general purpose graphics software such as Visio Professional or Corel Flow, while other designers uses repository–based CASE or modeling tools such as System Architect 2001, Visio Enterprise, Visible Analyst, or Rational ROSE; though CASE tools offers consistency and completeness as well as rule-based error checking.

6.4.1 Modern Structured Design

This is an example of model-driven approach to systems design. It is a process–oriented technique that breaks up a large program into a hierarchy of modules which result in a computer program that is much easier to implement and maintain.

The concept of modern structured design is very simple as it involves the top down hierarchy of modules. **A module is a group of instruments which may be a paragraph, block, subprogram or subroutine.** The top–down structure of these modules is then developed according to various design rules and guidelines.

Modern structured design is considered to be a process–oriented techniques because it lays emphasis on the PROCESS building blocks in our information system specifically software processes. The modern structured design factors a program into the top-down hierarchy of modules which have the following properties:

- **Modules should be highly cohesive**: Meaning that each module should accomplish only a function. This makes it to be reusable in future programs.

- **Module should be loosely coupled**: That is modules should be minimally dependent on one another, thereby minimizing the effect that future changes in one module would have on other modules.

However, modern structured design is still a popular technique that involve the design of mainframe-based application software and is used for add rising coupling and cohesion issues at the system level.

6.4.2 Information Engineering (IE)

Information engineering (IE) is a model–driven and DATA–centered, but PROCESS–sensitive technique to plan, analyze and design information system. (Whitten, J.C. et al 2001).

The primary tool used in information engineering is a data model diagram which involves conducting business area requirements analysis from which information system applications are carved out and prioritized. The applications that are identical under it become projects in which other system analysis and design methods are intended to be applied in order to develop the production systems.

6.4.3 Prototyping

Traditionally, a physical design is a paper-and-pencil process. With this, the analyst draws pictures that show the layout or structure of outputs, input and database and the flow of dialogue and procedure.

Prototyping as earlier defined in this text is a small but working sample of a desired system designed for the purpose of demonstration and evaluation. It consists of building an experimental and inexpensive system for end user to evaluate.

Nowadays many systems analysts and designers prefer prototyping which is a modern engineering based approach to design. Prototyping can be quickly developed using fourth generation languages (4GLS) and object–oriented programming languages that are available now.

Prototypes can be built for simple outputs, computer dialogues, key functions, subsystems or even a whole system; and the prototype is then reviewed by the end–users and management, who makes recommendations about requirements, methods and format. Thereafter the prototype is then corrected, enhanced or refined to reflect the new requirement. This continues until the prototype is accepted. The process of refining and trying again is referred to as **iterative process.** The approach of prototyping has a lot of advantages; while the major disadvantage is that prototyping encourages ill–advised shortcuts through the life cycle.

6.4.4 Object–Oriented Design (OOD)

This is the newest design technique and an extension of object-oriented analyst strategy discussed in chapter six. This technique is used to refine the object requirements definitions earlier identified during the analysis stage and to define design specific objects.

For instance based on a design implementation, during object–oriented design the designer may need to revise the data or process characteristics for an object that has earlier been defined during systems analysis. Similarly a design implementation decision may require the designer to define a new set of objects that will make up an interface screen which the user may interact with in the new system.