**UNIT III SYSTEM ANALYSIS**

**Information Gathering Techniques**

The main aim of fact finding techniques is to determine the information requirements of an organization used by analysts to prepare a precise SRS understood by user.

Ideal SRS Document should −

* be complete, Unambiguous, and Jargon-free.
* specify operational, tactical, and strategic information requirements.
* solve possible disputes between users and analyst.
* use graphical aids which simplify understanding and design.

There are various information gathering techniques −

**Interviewing**

Systems analyst collects information from individuals or groups by interviewing. The analyst can be formal, legalistic, play politics, or be informal; as the success of an interview depends on the skill of analyst as interviewer.

It can be done in two ways −

* Unstructured Interview − The system analyst conducts question-answer session to acquire basic information of the system.
* Structured Interview − It has standard questions which user need to respond in either close (objective) or open (descriptive) format.

Advantages of Interviewing

* This method is frequently the best source of gathering qualitative information.
* It is useful for them, who do not communicate effectively in writing or who may not have the time to complete questionnaire.
* Information can easily be validated and cross checked immediately.
* It can handle the complex subjects.
* It is easy to discover key problem by seeking opinions.
* It bridges the gaps in the areas of misunderstandings and minimizes future problems.

**Questionnaires**

This method is used by analyst to gather information about various issues of system from large number of persons.

There are two types of questionnaires −

* Open-ended Questionnaires − It consists of questions that can be easily and correctly interpreted. They can explore a problem and lead to a specific direction of answer.
* Closed-ended Questionnaires − It consists of questions that are used when the systems analyst effectively lists all possible responses, which are mutually exclusive.

Advantages of questionnaires

* It is very effective in surveying interests, attitudes, feelings, and beliefs of users which are not co-located.
* It is useful in situation to know what proportion of a given group approves or disapproves of a particular feature of the proposed system.
* It is useful to determine the overall opinion before giving any specific direction to the system project.
* It is more reliable and provides high confidentiality of honest responses.
* It is appropriate for electing factual information and for statistical data collection which can be emailed and sent by post.

**Review of Records, Procedures, and Forms**

Review of existing records, procedures, and forms helps to seek insight into a system which describes the current system capabilities, its operations, or activities.

Advantages

* It helps user to gain some knowledge about the organization or operations by themselves before they impose upon others.
* It helps in documenting current operations within short span of time as the procedure manuals and forms describe the format and functions of present system.
* It can provide a clear understanding about the transactions that are handled in the organization, identifying input for processing, and evaluating performance.
* It can help an analyst to understand the system in terms of the operations that must be supported.
* It describes the problem, its affected parts, and the proposed solution.

**Observation**

This is a method of gathering information by noticing and observing the people, events, and objects. The analyst visits the organization to observe the working of current system and understands the requirements of the system.

Advantages

* It is a direct method for gleaning information.
* It is useful in situation where authenticity of data collected is in question or when complexity of certain aspects of system prevents clear explanation by end-users.
* It produces more accurate and reliable data.
* It produces all the aspect of documentation that are incomplete and outdated.

**Joint Application Development (JAD)**

It is a new technique developed by IBM which brings owners, users, analysts, designers, and builders to define and design the system using organized and intensive workshops. JAD trained analyst act as facilitator for workshop who has some specialized skills.

Advantages of JAD

* It saves time and cost by replacing months of traditional interviews and follow-up meetings.
* It is useful in organizational culture which supports joint problem solving.
* Fosters formal relationships among multiple levels of employees.
* It can lead to development of design creatively.
* It Allows rapid development and improves ownership of information system.

**Secondary Research or Background Reading**

This method is widely used for information gathering by accessing the gleaned information. It includes any previously gathered information used by the marketer from any internal or external source.

Advantages

* It is more openly accessed with the availability of internet.
* It provides valuable information with low cost and time.
* It act as forerunner to primary research and aligns the focus of primary research.
* It is used by the researcher to conclude if the research is worth it as it is available with procedures used and issues in collecting them.

**Feasibility Study**

Feasibility Study can be considered as preliminary investigation that helps the management to take decision about whether study of system should be feasible for development or not.

* It identifies the possibility of improving an existing system, developing a new system, and produce refined estimates for further development of system.
* It is used to obtain the outline of the problem and decide whether feasible or appropriate solution exists or not.
* The main objective of a feasibility study is to acquire problem scope instead of solving the problem.
* The output of a feasibility study is a formal system proposal act as decision document which includes the complete nature and scope of the proposed system.

Steps Involved in Feasibility Analysis

The following steps are to be followed while performing feasibility analysis −

* Form a project team and appoint a project leader.
* Develop system flowcharts.
* Identify the deficiencies of current system and set goals.
* Enumerate the alternative solution or potential candidate system to meet goals.
* Determine the feasibility of each alternative such as technical feasibility, operational feasibility, etc.
* Weight the performance and cost effectiveness of each candidate system.
* Rank the other alternatives and select the best candidate system.
* Prepare a system proposal of final project directive to management for approval.

Ezoic

**Types of Feasibilities**

**Economic Feasibility**

* It is evaluating the effectiveness of candidate system by using cost/benefit analysis method.
* It demonstrates the net benefit from the candidate system in terms of benefits and costs to the organization.
* The main aim of Economic Feasibility Analysis (EFS) is to estimate the economic requirements of candidate system before investments funds are committed to proposal.
* It prefers the alternative which will maximize the net worth of organization by earliest and highest return of funds along with lowest level of risk involved in developing the candidate system.

**Technical Feasibility**

* It investigates the technical feasibility of each implementation alternative.
* It analyzes and determines whether the solution can be supported by existing technology or not.
* The analyst determines whether current technical resources be upgraded or added it that fulfill the new requirements.
* It ensures that the candidate system provides appropriate responses to what extent it can support the technical enhancement.

**Operational Feasibility**

* It determines whether the system is operating effectively once it is developed and implemented.
* It ensures that the management should support the proposed system and its working feasible in the current organizational environment.
* It analyzes whether the users will be affected and they accept the modified or new business methods that affect the possible system benefits.
* It also ensures that the computer resources and network architecture of candidate system are workable.

**Behavioral Feasibility**

* It evaluates and estimates the user attitude or behavior towards the development of new system.
* It helps in determining if the system requires special effort to educate, retrain, transfer, and changes in employee’s job status on new ways of conducting business.

**Schedule Feasibility**

* It ensures that the project should be completed within given time constraint or schedule.
* It also verifies and validates whether the deadlines of project are reasonable or not.

**What is Structured Analysis?**

Structured Analysis is a development method that allows the analyst to understand the system and its activities in a logical way.

It is a systematic approach, which uses graphical tools that analyze and refine the objectives of an existing system and develop a new system specification which can be easily understandable by user.

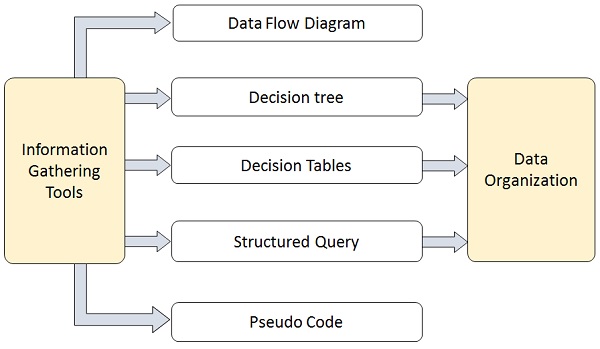
It has following attributes −

* It is graphic which specifies the presentation of application.
* It divides the processes so that it gives a clear picture of system flow.
* It is logical rather than physical i.e., the elements of system do not depend on vendor or hardware.
* It is an approach that works from high-level overviews to lower-level details.

**Structured Analysis Tools**

During Structured Analysis, various tools and techniques are used for system development. They are −

* Data Flow Diagrams
* Data Dictionary
* Decision Trees
* Decision Tables
* Structured English
* Pseudocode



**Data Flow Diagrams (DFD) or Bubble Chart**

It is a technique developed by Larry Constantine to express the requirements of system in a graphical form.

* It shows the flow of data between various functions of system and specifies how the current system is implemented.
* It is an initial stage of design phase that functionally divides the requirement specifications down to the lowest level of detail.
* Its graphical nature makes it a good communication tool between user and analyst or analyst and system designer.
* It gives an overview of what data a system processes, what transformations are performed, what data are stored, what results are produced and where they flow.

Basic Elements of DFD

DFD is easy to understand and quite effective when the required design is not clear and the user wants a notational language for communication. However, it requires a large number of iterations for obtaining the most accurate and complete solution.

The following table shows the symbols used in designing a DFD and their significance −

|  |  |  |
| --- | --- | --- |
| Symbol Name | Symbol | Meaning |
| Square | Square | Source or Destination of Data |
| Arrow | Arrow | Data flow |
| Circle | Circle | Process transforming data flow |
| Open Rectangle | Rectangle | Data Store |

**Types of DFD**

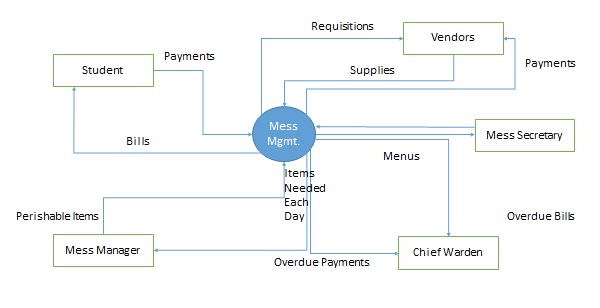
DFDs are of two types: Physical DFD and Logical DFD. The following table lists the points that differentiate a physical DFD from a logical DFD.

|  |  |
| --- | --- |
| Physical DFD | Logical DFD |
| It is implementation dependent. It shows which functions are performed. | It is implementation independent. It focuses only on the flow of data between processes. |
| It provides low level details of hardware, software, files, and people. | It explains events of systems and data required by each event. |
| It depicts how the current system operates and how a system will be implemented. | It shows how business operates; not how the system can be implemented. |

Context Diagram

A context diagram helps in understanding the entire system by one DFD which gives the overview of a system. It starts with mentioning major processes with little details and then goes onto giving more details of the processes with the top-down approach.

The context diagram of mess management is shown below.



* **Data Dictionary**

A data dictionary is a structured repository of data elements in the system. It stores the descriptions of all DFD data elements that is, details and definitions of data flows, data stores, data stored in data stores, and the processes.

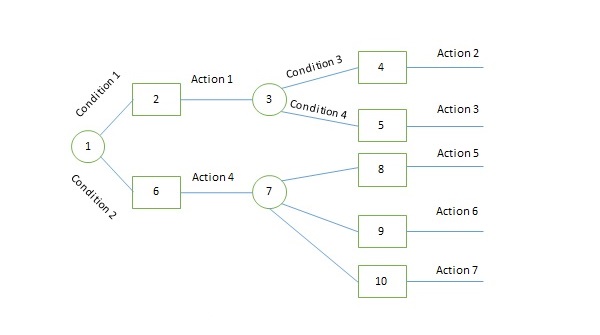
A data dictionary improves the communication between the analyst and the user. It plays an important role in building a database. Most DBMSs have a data dictionary as a standard feature. For example, refer the following table −

|  |  |  |  |
| --- | --- | --- | --- |
| Sr.No. | Data Name | Description | No. of Characters |
| 1 | ISBN | ISBN Number | 10 |
| 2 | TITLE | title | 60 |
| 3 | SUB | Book Subjects | 80 |
| 4 | ANAME | Author Name | 15 |

**Decision Trees**

Decision trees are a method for defining complex relationships by describing decisions and avoiding the problems in communication. A decision tree is a diagram that shows alternative actions and conditions within horizontal tree framework. Thus, it depicts which conditions to consider first, second, and so on.

Decision trees depict the relationship of each condition and their permissible actions. A square node indicates an action and a circle indicates a condition. It forces analysts to consider the sequence of decisions and identifies the actual decision that must be made.



The major limitation of a decision tree is that it lacks information in its format to describe what other combinations of conditions you can take for testing. It is a single representation of the relationships between conditions and actions.

**Decision Tables**

Decision tables are a method of describing the complex logical relationship in a precise manner which is easily understandable.

* It is useful in situations where the resulting actions depend on the occurrence of one or several combinations of independent conditions.
* It is a matrix containing row or columns for defining a problem and the actions.

Components of a Decision Table

* Condition Stub − It is in the upper left quadrant which lists all the condition to be checked.
* Action Stub − It is in the lower left quadrant which outlines all the action to be carried out to meet such condition.
* Condition Entry − It is in upper right quadrant which provides answers to questions asked in condition stub quadrant.
* Action Entry − It is in lower right quadrant which indicates the appropriate action resulting from the answers to the conditions in the condition entry quadrant.

The entries in decision table are given by Decision Rules which define the relationships between combinations of conditions and courses of action. In rules section,

* Y shows the existence of a condition.
* N represents the condition, which is not satisfied.
* A blank - against action states it is to be ignored.
* X (or a check mark will do) against action states it is to be carried out.

For example, refer the following table −

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CONDITIONS | Rule 1 | Rule 2 | Rule 3 | Rule 4 |
| Advance payment made | Y | N | N | N |
| Purchase amount = Rs 10,000/- | - | Y | Y | N |
| Regular Customer | - | Y | N | - |
| ACTIONS |  |  |  |  |
| Give 5% discount | X | X | - | - |
| Give no discount | - | - | X | X |

**Structured English**

Structure English is derived from structured programming language which gives more understandable and precise description of process. It is based on procedural logic that uses construction and imperative sentences designed to perform operation for action.

* It is best used when sequences and loops in a program must be considered and the problem needs sequences of actions with decisions.
* It does not have strict syntax rule. It expresses all logic in terms of sequential decision structures and iterations.

For example, see the following sequence of actions −

if customer pays advance

then

Give 5% Discount

else

if purchase amount >=10,000

then

if the customer is a regular customer

then Give 5% Discount

else No Discount

end if

else No Discount

end if

end if

**Pseudocode**

A pseudocode does not conform to any programming language and expresses logic in plain English.

* It may specify the physical programming logic without actual coding during and after the physical design.
* It is used in conjunction with structured programming.
* It replaces the flowcharts of a program.

**Guidelines for Selecting Appropriate Tools**

Use the following guidelines for selecting the most appropriate tool that would suit your requirements −

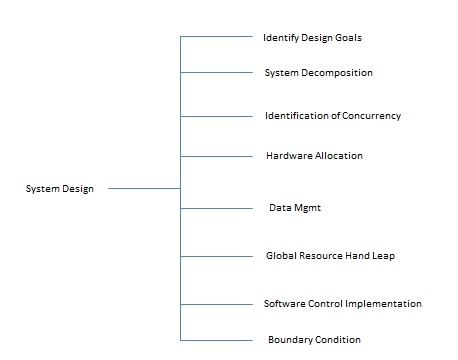
* Use DFD at high or low level analysis for providing good system documentations.
* Use data dictionary to simplify the structure for meeting the data requirement of the system.
* Use structured English if there are many loops and actions are complex.
* Use decision tables when there are a large number of conditions to check and logic is complex.
* Use decision trees when sequencing of conditions is important and if there are few conditions to be tested.

**THE PROCESS OF DESIGN**

System design is the phase that bridges the gap between problem domain and the existing system in a manageable way. This phase focuses on the solution domain, i.e. *“how to implement?”*

It is the phase where the SRS document is converted into a format that can be implemented and decides how the system will operate.

In this phase, the complex activity of system development is divided into several smaller sub-activities, which coordinate with each other to achieve the main objective of system development.



**Inputs to System Design**

System design takes the following inputs −

* Statement of work
* Requirement determination plan
* Current situation analysis
* Proposed system requirements including a conceptual data model, modified DFDs, and Metadata (data about data).

**Outputs for System Design**

System design gives the following outputs −

* Infrastructure and organizational changes for the proposed system.
* A data schema, often a relational schema.
* Metadata to define the tables/files and columns/data-items.
* A function hierarchy diagram or web page map that graphically describes the program structure.
* Actual or pseudocode for each module in the program.
* A prototype for the proposed system.

**Types of System Design**

**Logical Design**

Logical design pertains to an abstract representation of the data flow, inputs, and outputs of the system. It describes the inputs (sources), outputs (destinations), databases (data stores), procedures (data flows) all in a format that meets the user requirements.

While preparing the logical design of a system, the system analyst specifies the user needs at level of detail that virtually determines the information flow into and out of the system and the required data sources. Data flow diagram, E-R diagram modeling are used.

**Physical Design**

Physical design relates to the actual input and output processes of the system. It focuses on how data is entered into a system, verified, processed, and displayed as output.

It produces the working system by defining the design specification that specifies exactly what the candidate system does. It is concerned with user interface design, process design, and data design.

It consists of the following steps −

* Specifying the input/output media, designing the database, and specifying backup procedures.
* Planning system implementation.
* Devising a test and implementation plan, and specifying any new hardware and software.
* Updating costs, benefits, conversion dates, and system constraints.

File Organisation

A file is organized to ensure that records are available for processing. There are four methods of organizing files:

**1) Sequential organization:**

            Sequential organization means storing and sorting in physical, contiguous blocks within files on tape or disk. Records are also in sequence within each block. To access a record, previous records within the block are scanned. In a sequential organization, records can be added only at the end of the file. It is not possible to insert a record in the middle of the file without rewriting the file.

       In a sequential file update, transaction records are in the same sequence as in the master file. Records from both the files are matched, one record at a time, resulting in an updated master file. In a personal computer with two disk drives, the master file is loaded on a diskette into drive A , while the transaction file is loaded on another diskette into drive B.  Updating the master file transfers data from drive B to A controlled by the software in memory.

**Advantages:**

1.      Simple to design

2.      Easy to program

3.      Variable length and blocked records available

4.      Best use of storage space

**Disadvantages:**

1.      Records cannot be added at the middle of the file.

**2)  Indexed sequential organization:**

Like sequential organization, keyed sequential organization stores data in physically contiguous blocks. The difference is in the use of indexes to locate records. There are  three areas in disk storage: prime area, overflow area and index area.

The prime area contains file records stored by key or id numbers. All records are initially stored in the prime area.

The overflow area contains records added to the file that cannot be placed in logical sequence in the  prime area

The index area is more like a data dictionary. It contains keys of records and their locations on the disk. A pointer associated with each key is an address that tells the system where to find a record.

**Advantages:**

1.     Indexed sequential organization reduces the magnitude of the sequential search and provides quick access for sequential and direct processing.

2.      Records can be inserted in the middle of the file.

**Disadvantages:**

1.      It takes longer to search the index for data access or retrieval.

2.      Unique keys are required

3.   Periodic reorganization is required.

3)  Inverted list organization:

Like the indexed- sequential storage method the inverted list organization maintains an index. The two methods differ, however, in the index level and record storage. The indexed sequential method has a multiple index for a given key, where as the inverted list method has a single index for each key type. In an inverted list, records are not necessarily stored in a particular sequence. They are placed in the data storage area, but indexes are updated for the record key and location. The inverted keys are best for applications that request specific data on multiple keys. They are ideal for static files because additions and deletions cause expensive pointer updating.

**Advantages:**

1.      Used in applications requesting specific data on multiple keys.

**Database Design**

Before the database concept became operational, users had programs that handled their own data independent of other users. It was a conventional file environment with no data integration or sharing of common data across applications. In a data base environment, common data are available and used by several users. Instead of each program managing its own data, data across applications are shared by authorized  users with the data base software managing the data as an entity. A program now requests data through the data base management system(DBMS), which determines data sharing.

Objectives of data base:

The general theme behind a data base is to handle information as an integrated whole. There is none of the artificiality that is normally embedded in separate files or applications. A data base is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make information access easy, quick, inexpensive and flexible for the user. In data base design several specific objectives are considered:

1.Controlled redundancy: Redundant data occupies space and therefore is wasteful. If versions of the same data are in different phases of updating, the system often gives conflicting information. A unique aspect of database design is storing data only once -which controls redundancy and improves system performance

2.Ease of learning and use:A major feature of a user-friendly data base package is how easy it is to learn and use. Related to this point is that a database can be modified without interfering with established ways of using the data.

3.Data independence: An important data base objective is changing hardware and storage procedure or adding new data without having to rewrite application programs. The database should be tunable to improve performance without  rewriting programs.

4.More information at low cost: Using, storing, and modifying data at low cost are important. Although hardware prices are falling, software and programming costs are on the rise. This means that programming and software enhancements should be kept simple and easy to update.

5.Accuracy and integrity: The accuracy of a database ensures that data quality and content remain constant. Integrity controls detect data inaccuracies where they occur.

6. Recovery from failure: With multiple users accessing a database, the system must recover quickly after it is down with no loss of transactions. This objective also helps maintain data accuracy and integrity.

7.Privacy and security: For data to remain private, security measures must be taken to prevent unauthorized access. Data base security means that data are protected from various forms of destruction; users must be positively identified and their actions monitored.

8.performance: This objective emphasizes response time to inquiries suitable to the use of the data. How satisfactory the response time is depends on the nature of the user- database dialogue.

The terms that are normally used in DBMS are

1.    User’s view is a profile that the user expects to see on the report.

2.    Processing refers to the changes made.

3.    Data model is a framework of the user’s view.

4.    Data file is the area where the file is stored.

In a data base environment, the DBMS is the software that provides the interface between the data file on disk and the program that requests processing. DBMS stores and manages data. The procedure is as follows:

1. The user requests a sales report through the application program. The application program uses a data manipulation language (DML) to tell the DBMS what is required.

2. The DBMS  refers to the data model ,which  describes the view in a language called the data definition language (DDL). The DBMS uses DDL to determine how data must be structured to produce the user’s view.

3. The DBMS requests the input/output control system(IOCS) to retrieve the information from physical storage as specified by the application program. The output is the sales report

**To summarize,**

1. DML manipulates data: it specifies what is required.

2. DDL describes how data are structured.

3. DBMS manages data according to DML requests and DDL descriptions..

**Functions performed by the DBMS:**

1. Storing ,retrieveing , and updating data.

2.Creating program and data independence. Either one can be altered independently of the other.

3. Enforcing  procedures for data integrity.

4. Reducing data redundancy. Data are stored and maintained only once.

5. Proving security facilities for defining users and enforcing authorizations. Access is limited to authorized users by passwords or similar schemes.

6. Reducing physical storage requirements by separating the logical and physical aspects of the database.

Logical And Physical View Of Data

In data base design, several views of data must be considered along with the persons who use them. In addition to data structuring, where relationships are reflected between and within entities, we need to identify the application program’s logical views of data within an overall logical data structure. The logical view is what the data look like regardless of how they are stored. The physical view is the way data exists in physical storage. It deals with how data are stored, accessed or related to other data in storage.

Input / Output & Forms Design

**Input Design**

In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc.

Therefore, the quality of system input determines the quality of system output. Welldesigned input forms and screens have following properties −

* It should serve specific purpose effectively such as storing, recording, and retrieving the information.
* It ensures proper completion with accuracy.
* It should be easy to fill and straightforward.
* It should focus on user’s attention, consistency, and simplicity.
* All these objectives are obtained using the knowledge of basic design principles regarding −
* What are the inputs needed for the system?
* How end users respond to different elements of forms and screens.

**Objectives for Input Design**

The objectives of input design are −

* To design data entry and input procedures
* To reduce input volume
* To design source documents for data capture or devise other data capture methods
* To design input data records, data entry screens, user interface screens, etc.
* To use validation checks and develop effective input controls.

**Data Input Methods**

It is important to design appropriate data input methods to prevent errors while entering data. These methods depend on whether the data is entered by customers in forms manually and later entered by data entry operators, or data is directly entered by users on the PCs.

A system should prevent user from making mistakes by −

* Clear form design by leaving enough space for writing legibly.
* Clear instructions to fill form.
* Clear form design.
* Reducing key strokes.
* Immediate error feedback.
* Some of the popular data input methods are −
* Batch input method (Offline data input method)
* Online data input method
* Computer readable forms
* Interactive data input

Output Design

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

Objectives of Output Design

* To develop output design that serves the intended purpose and eliminates the production of unwanted output.
* To develop the output design that meets the end users requirements.
* To deliver the appropriate quantity of output.
* To form the output in appropriate format and direct it to the right person.
* To make the output available on time for making good decisions.

**Forms Design**

Both forms and reports are the product of input and output design and are business document consisting of specified data. The main difference is that forms provide fields for data input but reports are purely used for reading. For example, order forms, employment and credit application, etc.

* During form designing, the designers should know −
  + who will use them
  + where would they be delivered
  + the purpose of the form or report
* During form design, automated design tools enhance the developer’s ability to prototype forms and reports and present them to end users for evaluation.

Objectives of Good Form Design

* To keep the screen simple by giving proper sequence, information, and clear captions.
* To meet the intended purpose by using appropriate forms.
* To ensure the completion of form with accuracy.
* To keep the forms attractive by using icons, inverse video, or blinking cursors etc.
* To facilitate navigation.

**Methodologies**

**Structure Chart**

A hierarchical diagram showing the relationships between the modules of a computer

program. A module is the basic component of a structure chart and is used to identify a

function. Modules are relatively simple and independent components. Higher-level

modules are “control” modules that control the flow of execution. Lower level modules

are “worker bee” modules and contain the program logic to actually perform the

functions.

The vertical lines connecting the modules indicate the calling structure from the highlevel

modules to the lower-level modules. The little arrows next to the lines show the data

that is passed between modules and represent the inputs and outputs of each module. At

the structure chart level, we are not concerned with what is happening inside the module

yet. We only want to know that somehow it does the function indicated by its name using

the input data and producing the output data. A program call is when one module invokes

a lower-level module to perform a needed service or calculation. Program call: The

transfer of control from a module to a subordinate module to perform a requested service.

The arrows with the open circle, called data couples, represent data being passed into and

out of the module. A data couple can be an individual data item (e.g., a flag or a customer

account number) or a higher-level data structure (e.g., an array, record, or other data

structure). The arrow with the darkened circle is a “flag.” A flag is purely internal

information that is used between modules to indicate some result. Data couples: The

individual data items that are passed between modules in a program call.

A basic idea of structured programming is that each module only has to do a very specific

function. The module at the very top of the tree is the “boss” module. Its functions will be

to call the modules on the next tier, pass information to them, and receive information

back. The function of each middle-level module is to control the processing of the

modules below it. Each has control logic and any error-handling logic that is not handled

by the lower-level module. The modules at the extremities, or the leaves, contain the

actual algorithms to carry out the functions of the program.

Structure charts are developed to design a hierarchy of modules for a program. A

structure chart is in the form of a tree with a root module and branches. A subtree is

simply a branch that has been separated from the overall tree. When the subtree is placed

back in the larger tree, the root of the subtree becomes just another branch in the overall

tree.

**Overview:**

A structure chart is a top-down modular design tool, constructed of squares representing

the different modules in the system, and lines that connect them. The lines represent the

connection and or ownership between activities and sub activities as they are used in

organization charts.

In structured analysis structure charts, according to Wolber (2009), are used to specify

the high-level design, or architecture, of a computer program. As a design tool, they said

the programmer in dividing and conquering a large software problem, that is, recursively

breaking a problem down into parts that are small enough to be understood by a human

brain. The process is called top-down design, or functional decomposition. Programmers

use a structure chart to build a program in a manner similar to how an architect uses a

blueprint to build a house. In the design stage, the chart is drawn and used as a way for

the client and the various software designers to communicate. During the actual building

of the program (implementation), the chart is continually referred to as the master-plan".

A structure chart is also used to diagram associated elements that comprise a run stream

or thread. It is often developed as a hierarchical diagram, but other representations are

allowable. The representation must describe the breakdown of the configuration system

into subsystems and the lowest manageable level. An accurate and complete structure

chart is the key to the determination of the configuration items, and a visual

representation of the configuration system and the internal interfaces among its CIs.

During the configuration control process, the structure chart is used to identify CIs and

their associated artifacts that a proposed change may impact.

 **Applications of Structure Chart**

Use a Structure Chart to illustrate the high level overview of software structure. Structure

Charts do not show module internals. Use a method, such as Pseudo code or Structured

English, to show the detailed internals of modules.

Advantages of structure chart

o Representing Sequence, Repetition, and Condition on a Structure Chart

o Modules on a Structure Chart

o Interrelationships among Modules

o Information Transfers

o Reducing Clutter on a Structure Chart.

Example: The example Structure Chart illustrates the structure of the modules to process a

customer order.

**Forms Driven Methodology – The IPO Charts**

HIPO charts show relationships between modules. It describes the data input and output from the processes and defines the data flow. It provides a structure by which the functions of a system can be understood. It also provides a visual description of input to be used and output to be produced for each level of the diagram. It makes the transformation from input to output data visible.

There are two parts to a HIPO chart, a hierarchy chart and an IPO chart.

The hierarchy chart is useful for showing hierarchy of procedures within a program. Hierarchy

charts are also called structure charts, top-down charts, or VTOC (Visual Table of Contents)

charts. All these names refer to planning diagrams that are similar to a company's organization

chart. Hierarchy charts depict the organization of a program but omit the specific processing

logic. They describe what each part, or module, of the program does and how the modules relate to each other.

The IPO chart describes the system in terms of its inputs, outputs and the processes that are

performed on the inputs to transform them into outputs. It provides the following:

 The Input section that contains the data items used by the process steps.

 The Output section that contains the data items created by the process steps.

 Process section that contains numbered steps that describes the functions to be

performed. Arrows connect them to the output steps and the input/output data items.

The IPO chart is in the form of a table with three columns, one for each of Input, Output and

Process. The flow between screens is indicated by the use of arrows.

Use of HIPO chart?

The HIPO (Hierarchy plus Input-Process-Output) technique is a tool for planning and/or

documenting a computer program. A HIPO model consists of a hierarchy chart that

graphically represents the program’s control structure and a set of IPO (Input- Process-

Output) charts that describe the inputs to, the outputs from, and the functions (or

processes) performed by each module on the hierarchy chart.

**Advantages of HIPO Chart:**

Using the HIPO technique, designers can evaluate and refine a program’s design, and

correct flaws prior to implementation. Given the graphic nature of HIPO, users and

managers can easily follow a program’s structure. The hierarchy chart serves as a useful

planning and visualization document for managing the program development process.

The IPO charts define for the programmer each module’s inputs, outputs, and algorithms.

**Limitation of HIPO Chart:**

HIPO provides valuable long-term documentation. However, the “text plus

flowchart” nature of the IPO charts makes them difficult to maintain, so the

documentation often does not represent the current state of the program.

By its very nature, the HIPO technique is best used to plan and/or document a hierarchically structured program.