

Software analysis and design includes all activities, which help the transformation of requirement specification into implementation. Requirement specifications specify all functional and non-functional expectations from the software. These requirement specifications come in the shape of human readable and understandable documents, to which a computer has nothing to do.

Software analysis and design is the intermediate stage, which helps human-readable requirements to be transformed into actual code.

Data Flow Diagram

Data flow diagram is graphical representation of flow of data in an information system. It is capable of depicting incoming data flow, outgoing data flow and stored data. The DFD does not mention anything about how data flows through the system.

There is a prominent difference between DFD and Flowchart. The flowchart depicts flow of control in program. DFDs depict flow of data in the system at various levels. DFD does not contain any control or branch elements.

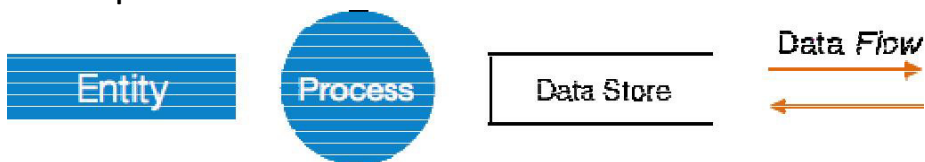
Types of DFD

Data Flow Diagrams are either Logical or Physical.

Logical DFD – They specify various logical process performed on data. It does not specify who done the operation whether it is done manually or with a system and also where it is done

Physical DFD – In physical dfd shows , how the various functions are performed. Who does them. Are they done by computer or manually, who does them.

DFD Components



DFD can represent Source, destination, storage and flow of data using the following set of components -

Entities - Entities are source and destination of information data. Entities are represented by a rectangles with their respective names.

Process - Activities and action taken on the data are represented by Circle or Round-edged rectangles.

Data Storage - There are two variants of data storage - it can either be represented as a rectangle with absence of both smaller sides or as an open-sided rectangle with only one side missing.

Data Flow - Movement of data is shown by pointed arrows. Data movement is shown from the base of arrow as its source towards head of the arrow as destination.

Level of DFD

O level / Context Diagram/ Top Level DFD

A context diagram is a top level (also known as "Level 0") data flow diagram. It only contains one process ("Process 0") that generalizes the function of the entire system in relationship to external entities and does not show data store.

1 level DFD

The O level DFD shows the main processes within the system. Each of these processes can be broken into further processes until you solve problem. These sub process is known as 1 level then 2 level n level.

Note:

In DFD flow data between

Entity to Process

Process to Entity

File to Process

Process to File

In DFD cannot flow data between

File to Entity

Entity to File

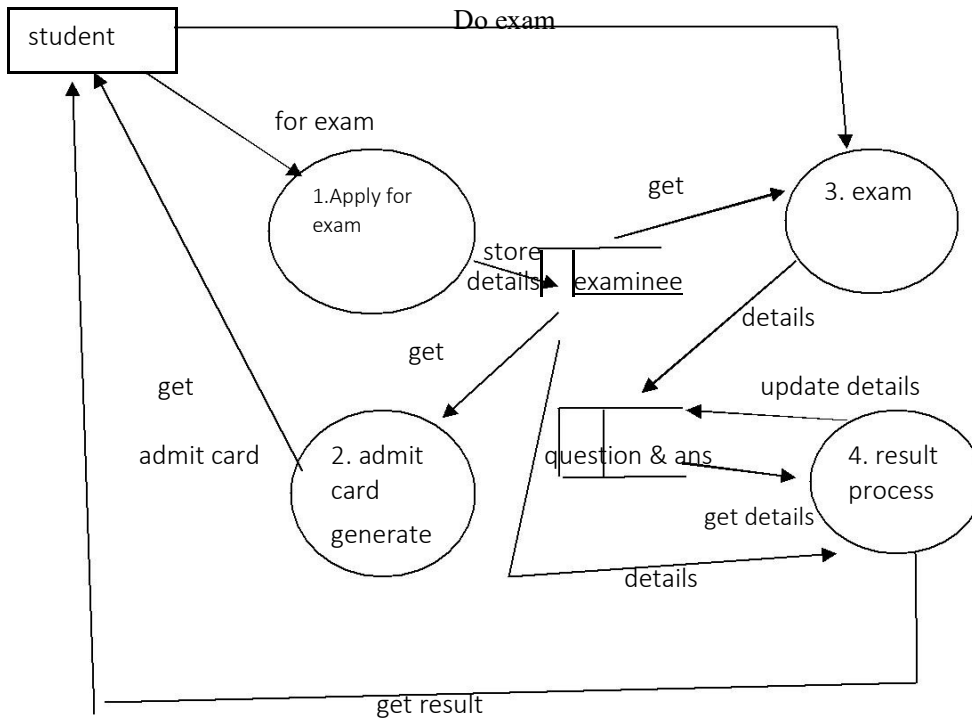
Example.

Examination system

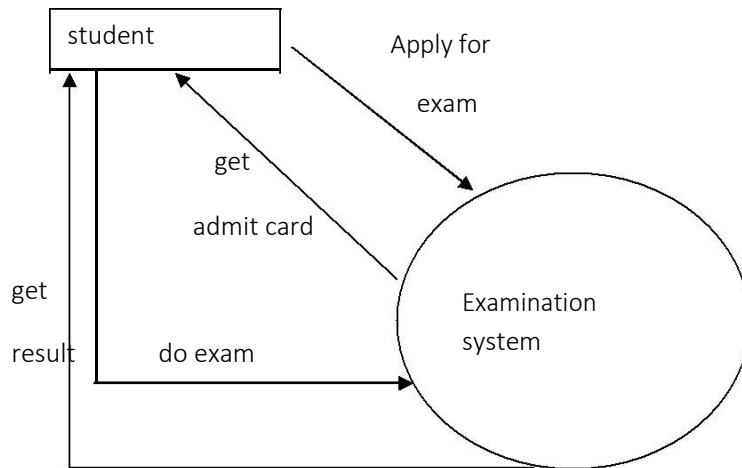
Process: .

- Firstly examinee apply for examination
- Admit card generate
- Examination process
- Result processing

DFD (Level 1) of Examination system



DFD (level 0) Examination System



PROPERTIES OF GOOD DFD DIAGRAM

DFD should be self explanatory, complete and unambiguous. To achieve these, DFD should have following properties.

1. The absence of flow chart structures.
 2. Conservation of data
 3. Good naming convention
1. DFD vis-à-vis program flowchart

Program flow chart describes computations, decisions, interactions and loops. In contrast a DFD shows only flow of data in the system.

2. Conservation of Data

A process in the data flow diagram accepts input data flows and produces output data flows. The property of conservation of data says that a process cannot create new data. It can only take data and either output it again transforms it into a new form of data. A process should not lose any data, while transformation.

3. Good naming conventions.

This properties of DFD is related to the naming convention to be followed for processes, data stores and data flow. A process name should consist of one single phrase and it should be possible to describe it in one sentence. Also the process name should define a specific action rather than a general process.

PROCESS LOGIC

In the previous unit we had described DFD is a graphical tool for representing information system. DFD constitutes many processes, which are identified by process names and they are given a unique number. Each process describe some operation of the system and has certain well defined function to performed. Describing and documenting such functions is also part of preparing systems model.

Hence while analysis and design of information system draw DFD is not enough. We also describe detailed description of each process it contains. Writing such description is process logic.

METHODS USED TO DETAILED PROCESSEES DESCRIPTION

1. Structured English
2. Decision Tree
3. Decision Table

1. Structured English

In order to remove ambiguity that arises, when one describes processes in English, the method of structured English was conceptualized. Obviously, it will containing English words and sentences , but to eliminate the ambiguity, instead of writing verbal description processes are described in a more structured way. Certain rules are followed, while writing description in English. These rules set the syntax of the process description which has well defined syntactical rules. The syntactical rules writing description tells, what are the keywords (reserved words such as If, THEN) to be used and what constructs such as If THEN ELSE construct, DO WHILE constructs are used.

KEYWORDS

BEGIN	REPEAT	IF
END	UNTIL	THEN
CASE	WHILE	ELSE
DO	FOR	

CONSTRUCTS

DO	FOR	IF	CASE
		THEN	

WHILE

ELSE

(Loop)

(Decision)

(Multiple decision)

LOGICAL , ARITHMETIC & RELATIONAL OPERATORS

Arithmetic Operators

*, +, /, -, %

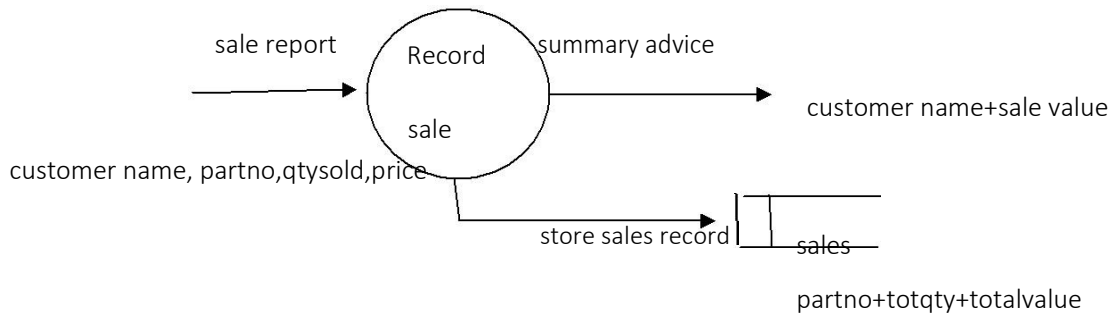
Logical Operators

And, Or, Not

Relational Operators

<, >, <=, >=, =

Example



Process Logic

BEGIN

receive sale report

totqty=totqty+qtysold

salevalue=qtysold*price

totalvalue=totalvalue+salevalue

write sales record

send summary advice

END

DECISION TREE

Decision tree is a graphical representation that presents conditions and actions sequentially. It is a method of showing the relationship of each condition & its permissible actions. The route of the tree is starting point and it proceeds towards the various possible nodes.

The size of the tree will depend upon the number of conditions & actions. Each condition is expressed in two ways True/False or Yes/No.

Advantages of Decision tree

1. It expresses the logic of if then else in pictorial form.
2. It is useful to express the logic when a value is variable or an action is dependent on nested decision i.e. the outcome of another decision.
3. It helps the analyst to identify the actual decision to be made.
4. It is used to verify logic and problems that involve a few complex decision and limited number of actions.

Disadvantages of Decision tree

1. The lack of decision tree is that there is absence of information in its format to take what other combinations of conditions to test.
2. A large number of branches with many paths will confuse rather than help in analysis.

Steps to draw Decision Tree

The first step is to analyze the statement so that the following items can be identified.

1. Condition
2. Action
3. However and but structures
4. Greater than/ less than ambiguities
5. And / Or ambiguities
6. Undefined activities

Keeping all the above points in mind, we can draw the decision tree

Example:

Specifies two tests (one for type of account and one for type of transactions and different combination of same actions (addition to a subtraction from account) depending on the tests.

BEGIN

IF A trade account THEN

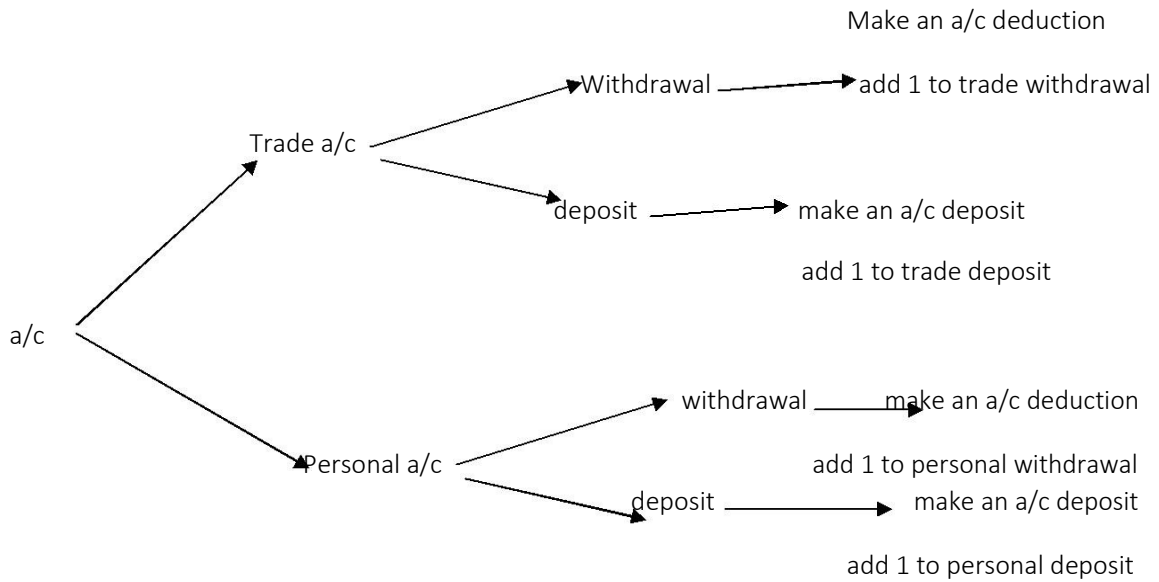
 IF WITHDRAWAL THEN

 Make an account deduction

```
        add 1 to trade withdrawal
    ELSE
        Make an account addition
        add 1 to trade deposit
    ENDIF
ELSE ( a personal account)
    IF WITHDRAWAL THEN
        Make an account dedection
        add 1 to personal withdrawal
    ELSE
        Make an account addition
        add 1 to personal withdrawal
    ENDIF
ENDIF
END
```

In the above example readers will find that certain decisions like 'make an account deduction' appear at many places. Certain tests like 'IF withdrawal' also appear at many places. In all such cases a better way is possible to describe the processes. The better way is through drawing a decision tree or decision table.

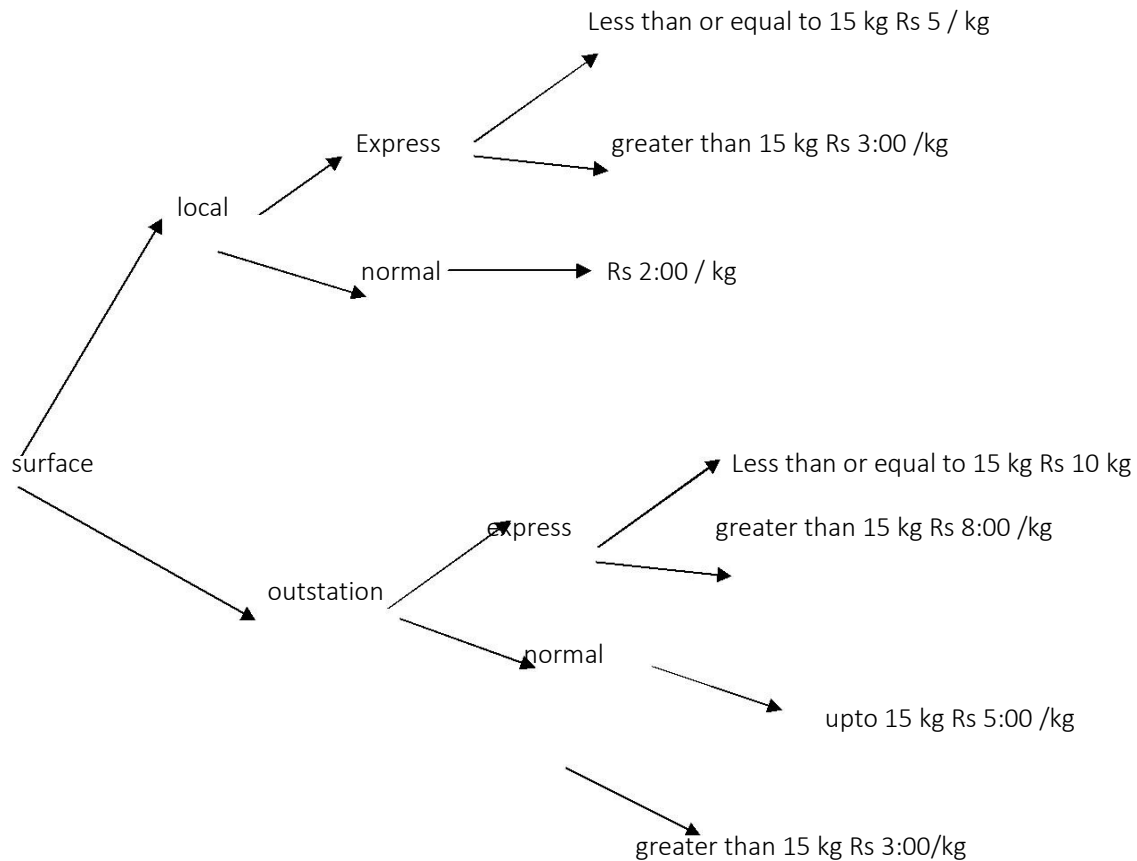
Example of decision tree



Decision tree is much better way of representing the process logic as compared to structured English which is clumsier than decision tree.

Example:

Air charges are set depending on the weight of parcel. The express rate is Rs. 5.00 per kg which is further reduced to Rs 3:00 a kg for access over 15 kg. Surface fright of normal is Rs 2:00 per kg. However, this rate is for local deliveries only. If the parcel is out-station and weight over 15 kg or express delivery is wanted, the surface fright is Rs 10:00 per kg upto 15 kg otherwise Rs 5 per kg. Normal delivery of packages upto Rs 15 kg is Rs 5:00 per kg which is further reduced to Rs 3:00 a kg if surface fright is over 15 kg,



DECISION TABLE

A decision table is a graphical method for explaining the logic of making decision in tabular format. It is a set of conditions + set of actions and different combinations of decisions.

“It is a matrix representation of logic of decisions which specify the possible conditions for decision and resulting actions.”

The decision table is divided into two parts:

1. Condition
2. Action

1. Condition:

The condition part specifies all conditions that are applied to the inputted data. It is also divided into two parts à

- i. Condition Stub:-

It describes the conditions that exist in the program logic.

- ii. Condition Entry:-

It provides answers to questions asked in the condition asked in the condition term.

2. Action:

The action part is subdivided into two parts à

i. Action Stub:-

It describes the action to be taken to meet each condition.

ii. Action Entry:-

It describes the appropriate action resulting from the answer to the condition in condition entry.

Format:

<u>Table Heading</u>	<u>Decision Rule</u>
Condition Stub	Condition Entry
Action Stub	Action Entry

Advantages of Decision Table

1. It provides compact representation of decision making process.
2. It is easier to understand particular path.
3. It can be changed according to situation.
4. These are best suited for calculating discounts, commissions or inventory control procedures.
5. The structure of decision table promotes a logically complete and consistent problem definition.

Disadvantages of Decision Table

1. It cannot express the complete sequence of operations to solve a problem therefore it may be difficult for the programmer to translate decision table into program.
2. If there are too many alternatives, it is difficult to list in decision table.
3. It does not show the flow of logic for the solution to a given problem.

Q: An insurance company uses the following rule to determine the eligibility of a driver for insurance.

The driver will be insured if:-

1. The driver lives in the city with population less than 5000 and he is married man.
2. The driver lives in the city with population less than 5000 and he is married and age is over 30 years old.
3. The driver lives in the city with population is 5000 or more and it is married female.
4. The driver is male over 30.
5. The driver is married and under 30.

Condition	Rule1	Rule2	Rule3	Rule4	Rule5	Else
Lives in city population<5000	Y	Y	N	-	-	-
Male	Y	-	N	Y	-	-
Married	Y	Y	Y	-	Y	-
Age>30	-	Y	-	Y	N	-
Action	Rule1	Rule2	Rule3	Rule4	Rule5	Else
Insured	Y	Y	Y	Y	Y	
Uninsured						Y

