

Structured Analysis Tools

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:

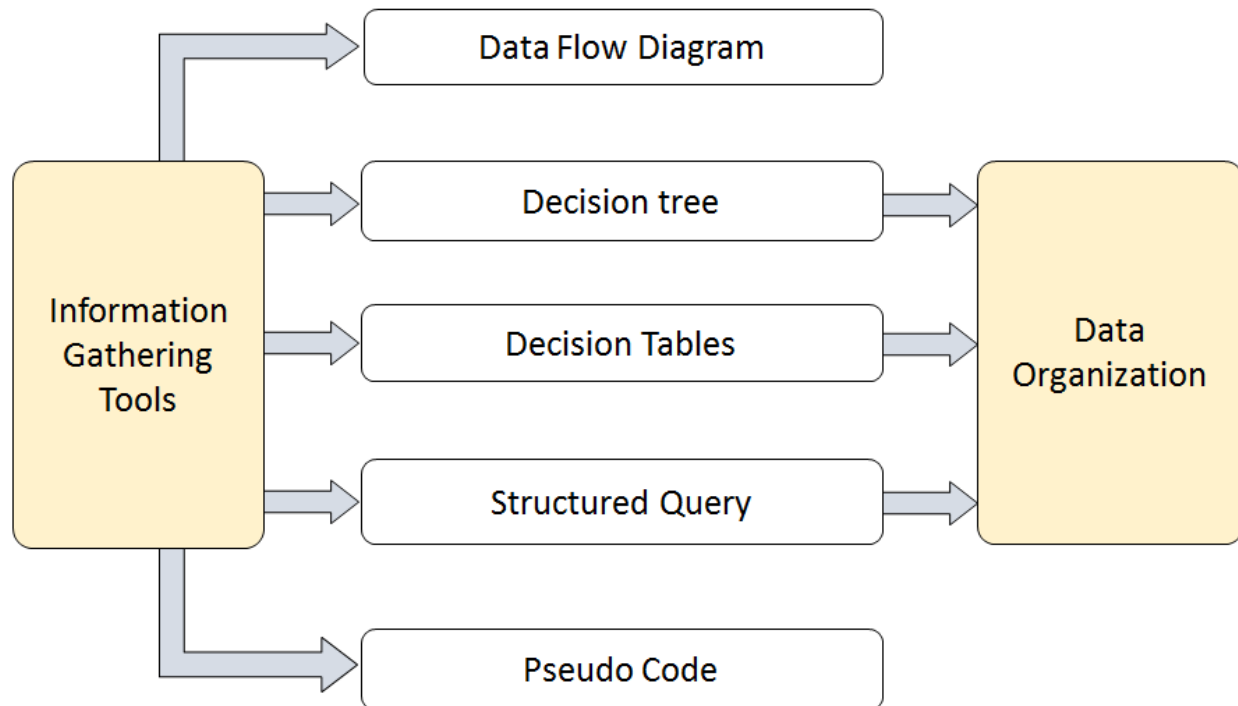
1. It is graphic which specifies the presentation of application.
2. It divides the processes so that it gives a clear picture of system flow.
3. It is logical rather than physical i.e., the elements of system do not depend on vendor or hardware.
4. It is an approach that works from high-level overviews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

Structured Analysis Tools

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

1. Data Flow Diagrams
2. Data Dictionary
3. Decision Trees
4. Decision Tables
5. Structured English

6. Pseudocode



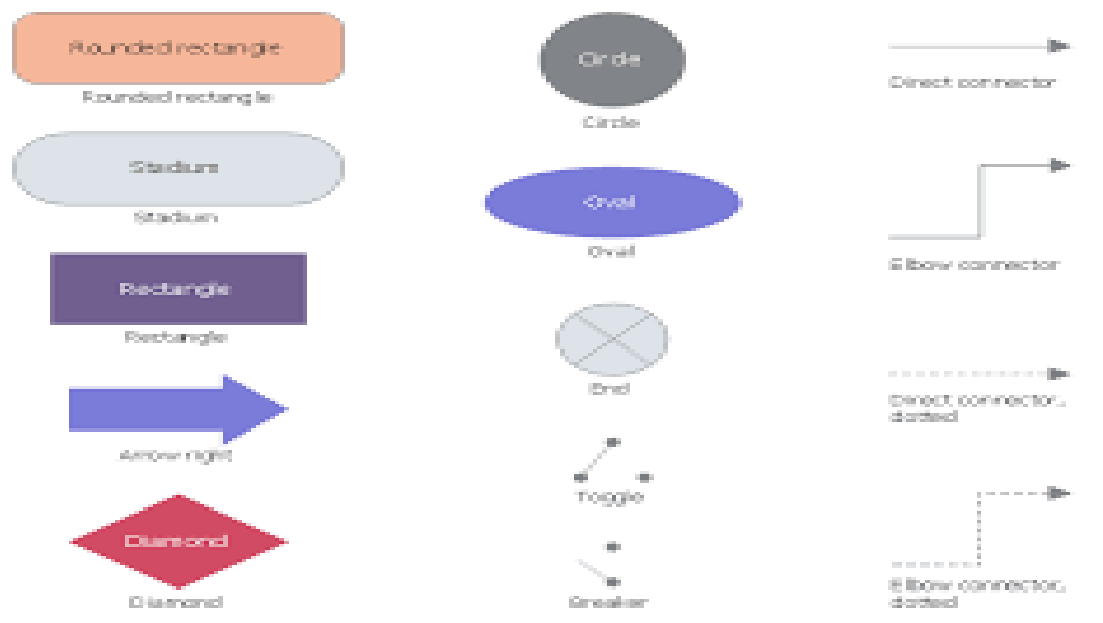
Data Flow Diagrams (DFD) or Bubble Chart

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

1. It shows the flow of data between various functions of system and specifies how the current system is implemented.
2. It is an initial stage of design phase that functionally divides the requirement specifications down to the lowest level of detail.
3. Its graphical nature makes it a good communication tool between user and analyst or analyst and system designer.
4. 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.



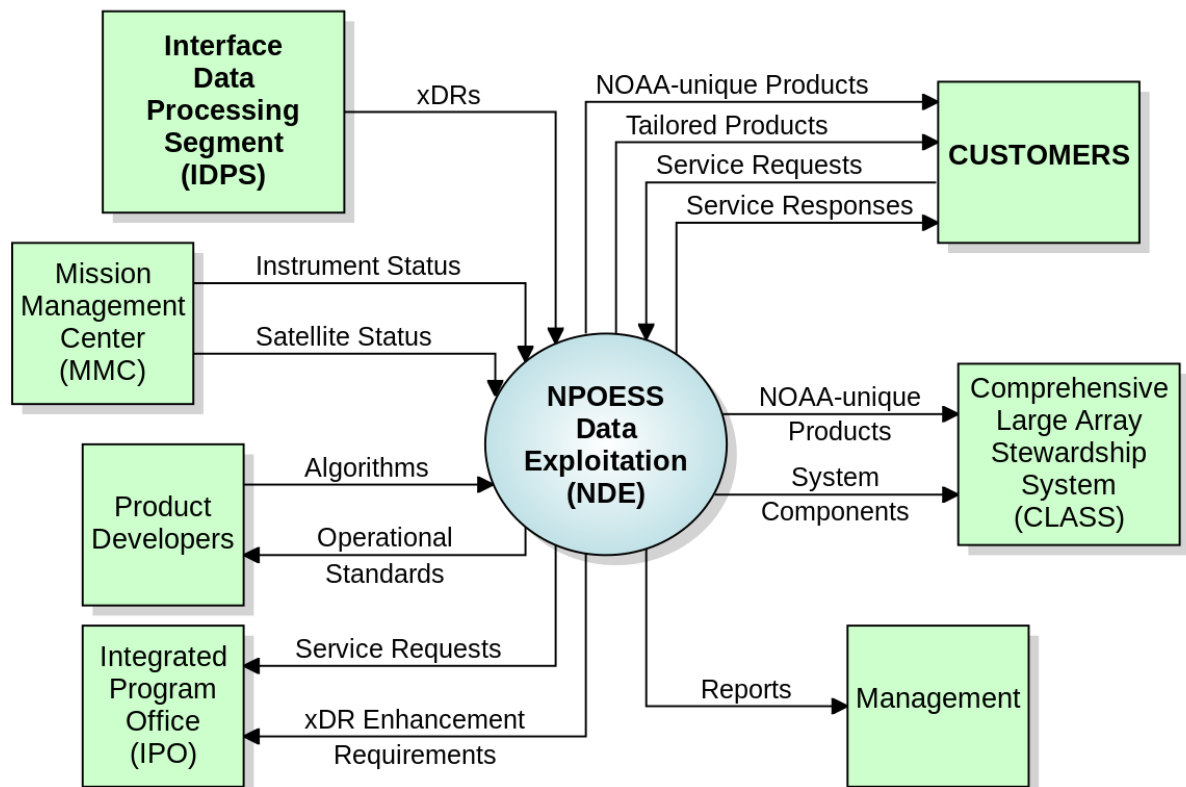
Types of DFD

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

- DFD is categorized as:
 - Physical DFD
 - Shows how the system will be implemented including software, hardware, file, people etc.
 - Logical DFD
 - Shows how business operates
 - It is concerned with how the system will be constructed.
 - Describes business events and data required and produced by the event

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.



Data Dictionary

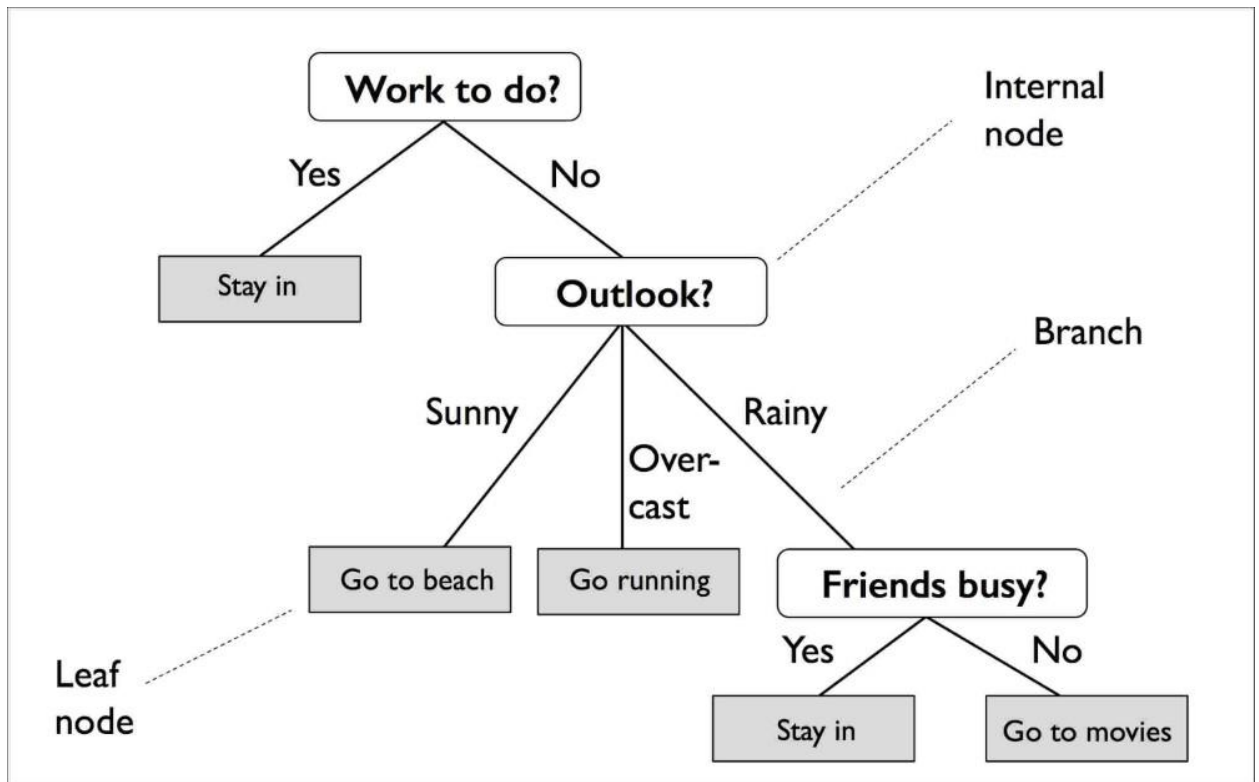
A data dictionary is a structured repository of data elements in the system. It stores the descriptions of all DFD data elements that is, details and definitions of data flows, data stores, data stored in data stores, and the processes.

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:

s.no	Data name	Description	No. of character
1	ISBN	ISBN NO	15
2	TITLE	BOOK NAME	50
3	SUB	SUBJECT	50
4	ANAME	AUTHOR NAME	50

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.



Decision Tables

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

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

Components of a Decision Table

1. **Condition Stub:** It is in the upper left quadrant which lists all the condition to be checked.
2. **Action stub:** It is in the lower left quadrant which outlines all the action to be carried out to meet such condition.

3. **Condition Entry:** It is in upper right quadrant which provides answers to questions asked in condition stub quadrant.

4. **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.

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.

Pseudocode

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

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

-----X-----