A Study on Structured Analysis and Design Tools

Research · February 2019

DOI: 10.13140/RG.2.2.36022.42566

Raj Sinha
Jayoti Vidyapeeth Women's University

International Journal of Management, IT & Engineering
Vol. 9 Issue 2(1), February 2019,
ISSN: 2249-0558 Impact Factor: 7.119
Journal Homepage: http://www.ijmra.us, Email: editorijmie@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

A Study on Structured Analysis and Design Tools

Raj Sinha*

Abstract

Structured Analysis is a set of techniques and graphical tools such as ER Model, Data Flow Diagrams, Flowchart, Data Dictionary, Decision Trees, Decision Tables, Structured English and Pseudocode that allow the analyst to develop a new kind of system specification that are easily understandable to the developer. From the DFD, the next step is the definition of the modules and their relationships to one another in a form called a structure chart, using a data dictionary and other structured tools.

Keywords: *Data Flow Diagram, Data Dictionary, Decision Tree, Decision table.*

Introduction: Structured Analysis is a logical tool to understand and describe the information system in a logical way which uses graphical tools in an organized manner that makes use of graphical diagrams to develop and present system specifications to users in a way that makes them clear and easy to understand. The diagrams explain the steps that need to occur and the data that is needed to meet the design requirements of the system. Some of the important steps involved in structured analysis are as below:

- Studying the current system and evaluating all of its issues
- Modelling this system
- Modelling the new system around these issues, in order to fix them
- Modelling the new physical environment
- Evaluating any alternatives
- Choosing the best system approach
- Creating the graphical specifications

It defines

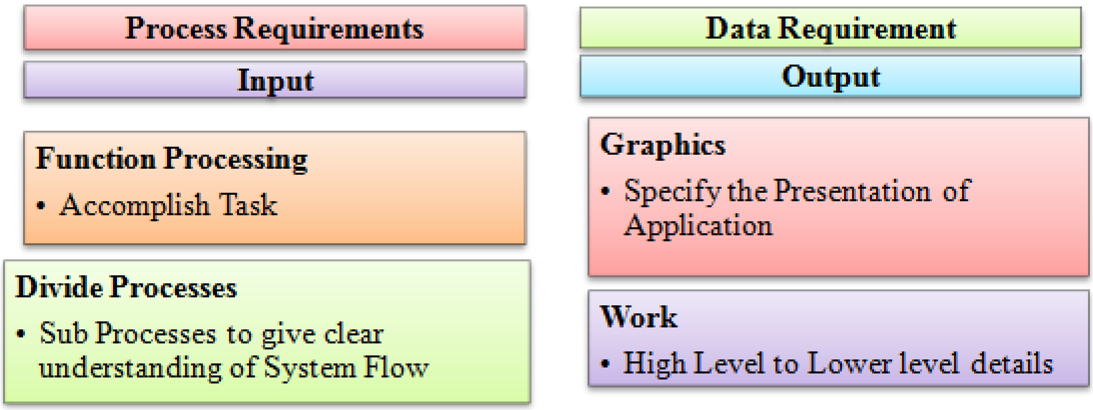

Process Requirements
Input
Data Requirement
Output
Function Processing
• Accomplish Task
Graphics
• Specify the Presentation of Application
Divide Processes
• Sub Processes to give clear understanding of System Flow
Work
• High Level to Lower level details

Fig 1.0 Structured Analysis

Objectives

- To determine if the current System is in trouble.
- To determine appropriate alternatives
- To make recommendation

[*] Guest Faculty, LNMIEDSC, Patna

Tools of Structured Analysis: Following are the various tools and techniques which are used for system development. They are –

- ER Model
- Flowchart
- Data Flow Diagrams
- Decision Trees
- Decision Tables
- Structured English
- Data Dictionary

E-R Model: Entity-relationship model is database analysis and design tool which lists real-life application entities and defines the relationship between real life entities that are to be mapped to database. E-R model forms basis for database design.

Example:

Entity	Primary Key	Attribute
Student	Student_ID	StudentName
Professor	Employee_ID	ProfessorName
Course	Course_ID	CourseName
Entity	Primary Key	Attribute

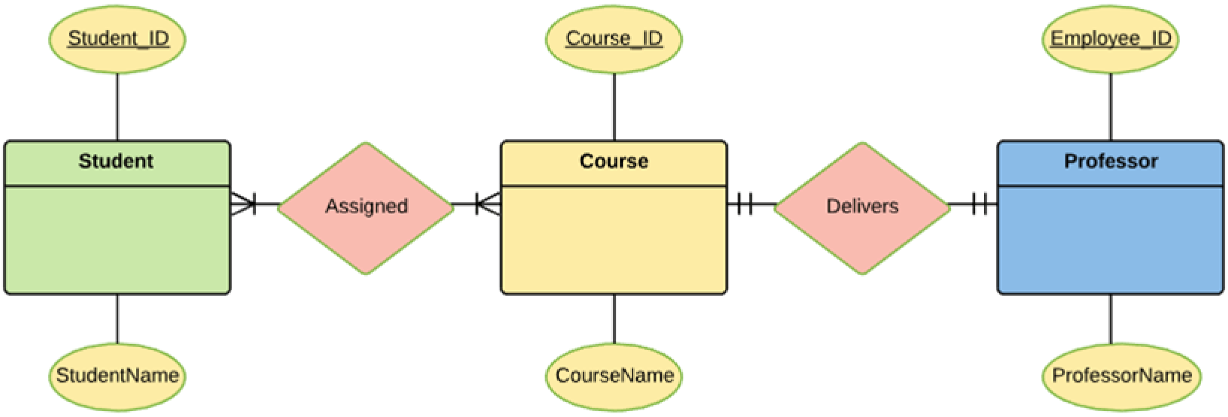
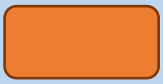
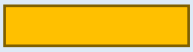
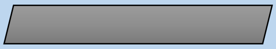
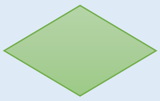
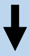
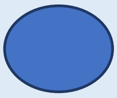
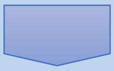


Fig 1.1 Example of ER Model

Flowchart: A flow chart which was introduced by Frank Gilberth in 1921, is a graphical or symbolic representation of a process. Earlier they were called "Process Flow Charts". Here, each step in the process is represented by a different symbol and contains a short description of the process step. The flow chart symbols are linked together with arrows showing the process flow direction.

Notation of Flowchart

Symbol Name	Purpose	Symbol
Start/Stop	Used at the beginning and end of the algorithm to show start and end of the program.	
Process	Indicates processes like mathematical operations.	
Input/ Output	Used for denoting program inputs and outputs.	
Decision	Stands for decision statements in a program, where answer is usually Yes or No	
Arrow	Shows relationships between different shapes.	
On-page Connector	Connects two or more parts of a flowchart, which are on the same page.	
Off-page Connector	Connects two parts of a flowchart which are spread over different pages.	

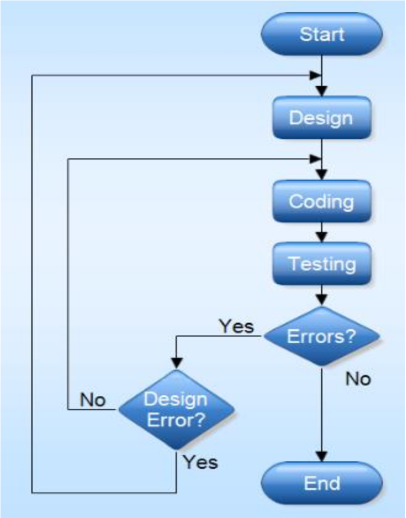


Fig 1.2 Example of Flowchart of SDLC Process

Data Flow Diagram: DFD are graphical representation to depict the flow of data and the transformation that takes place. DFD don't supply detail description of modules but graphically describe a system data and how the data interacts with the system.

Purpose of DFD

- Provides a graphical tool which can be used by the analyst to explain his understanding of the system to the user
- Can be easily converted into a structure chart which is used in design.

Context diagram: An overview of an organizational system that shows the system boundaries, external entities that interact with the system and the major information flows between the entities and the system. In this diagram, a single process represents the whole system. When we explore the context Analysis diagram then only it is called as DFD.

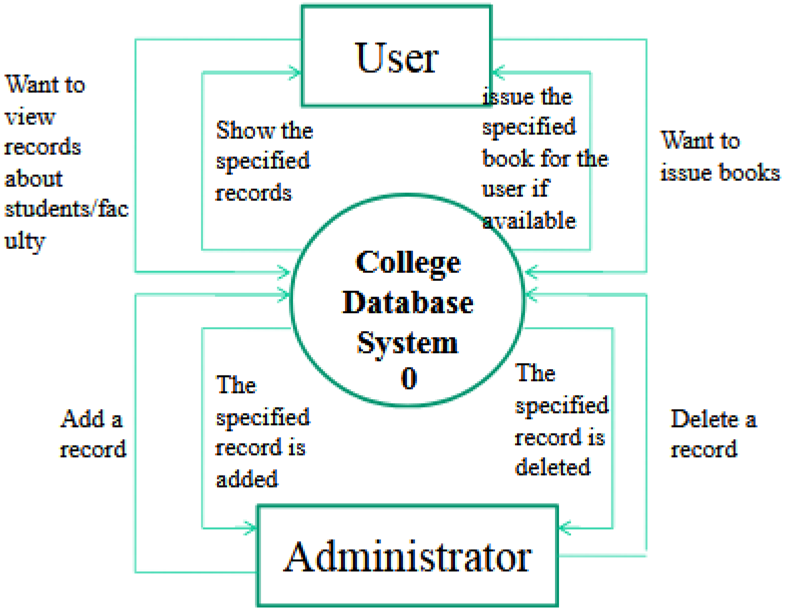

User
Want to view records about students/faculty
Show the specified records
issue the specified book for the user if available
Want to issue books
College Database System 0
Add a record
The specified record is added
The specified record is deleted
Delete a record
Administrator

Fig 1.3 Example of Context Diagram College Management System

First level DFD: A data flow diagram that represents a system's major processes, data flows, and data stores at a high level of detail.

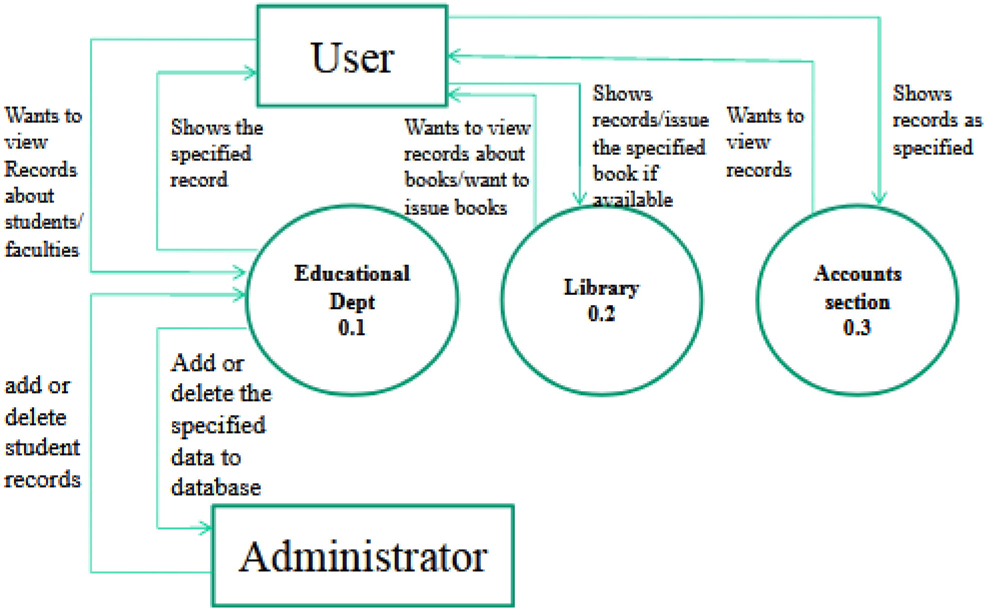

User
Wants to view Records about students/ faculties
Shows the specified record
Wants to view records about books/want to issue books
Shows records/issue the specified book if available
Wants to view records
Shows records as specified
Educational Dept 0.1
Library 0.2
Accounts section 0.3
add or delete student records
Add or delete the specified data to database
Administrator

Fig 1.4 Example of First Level DFD of College Management System

Second Level DFD: Certain elements of any data-flow diagram can be decomposed("exploded") into a more detailed model a level lower in the hierarchy.

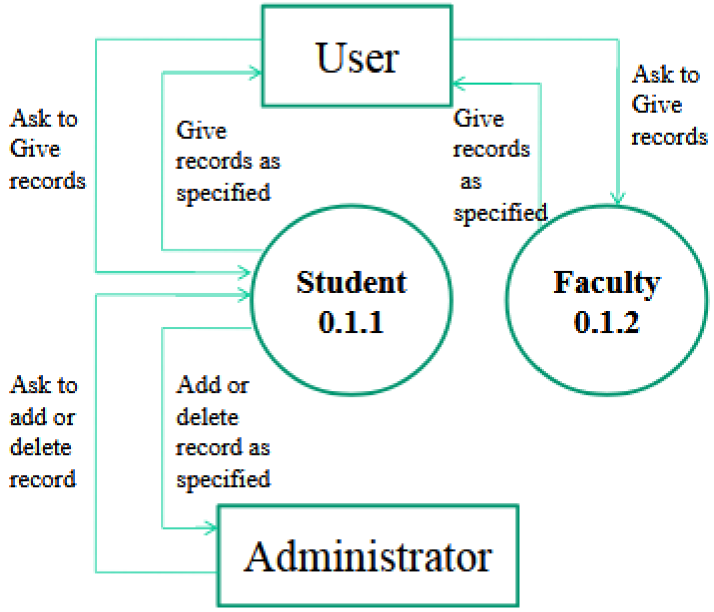
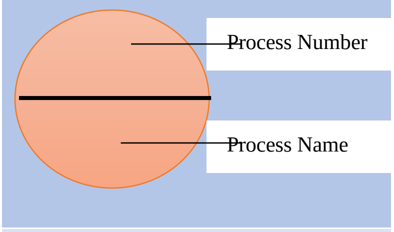
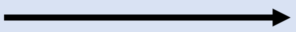
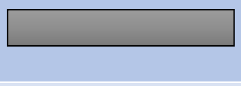
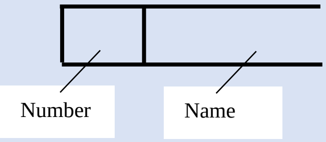

User
Ask to Give records
Give records as specified
Give records as specified
Ask to Give records
Student 0.1.1
Faculty 0.1.2
Ask to add or delete record
Add or delete record as specified
Administrator

Fig 1.5 Example of Second Level DFD of College Management System

Various components of a Data Flow Diagram

Components	Description	Symbol
Process	Any transformation of data from one form to another. People, procedures or devices can be used as processes that use or produce (transform) data.	
Data Flow	Path of data as it flows through a system. Data moves in a specific direction from a point of origin to point of destination in the form of a document, letter, telephone call or virtually any other medium	
Source or sink of data or External Entity	origin and destination of data. Lies outside the content of the system.	
Data store	Data at rest. Repository of data.	

Rules for drawing a data flow diagram:

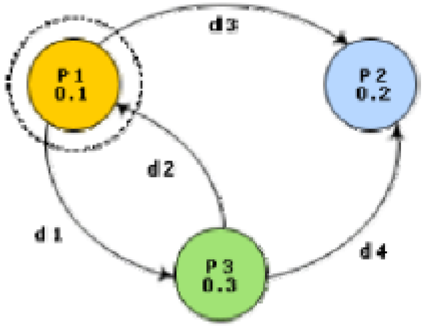
- For process:
  - No process can have only outputs and inputs.
  - A process has a verb phrase label.
- For Data Store:
  - Data cannot move directly from one data store to another data store. Data must be moved through a process.
  - Data cannot move directly from an outside source to data store. Data must be
    moved through a process that receives data from the source and places it into
  - the data store.
  - Data cannot move directly to an outside sink from a data store. Data must be
    moved through a process.
  - A data store has a noun phrase label.
- External Entity:
  - Fix the scope of system and identify all external entity.
  - It has a noun phrase label
- For data flow:
  - It cannot connect:
    - Two External Entities
    - An external Entity and a data store.
  - Can Connect:
    - Two Processes
    - An External entity and a process
    - A process and a data store.
  - Label all data flows
  - A data flow to a data store means update (delete or change).
  - A data flow from a data store means retrieve or use.
  - A data flow has a noun phrase label.

Leveling of DFD

- A Context diagram gives an overview.
- It should be split into major processes which gives greater details
- Each major process is further split to give more details

Balancing DFD: The data that flow into or out of a bubble on a parent diagram are equivalent to net inputs and output to and from a child diagram. This equivalence is called balancing.

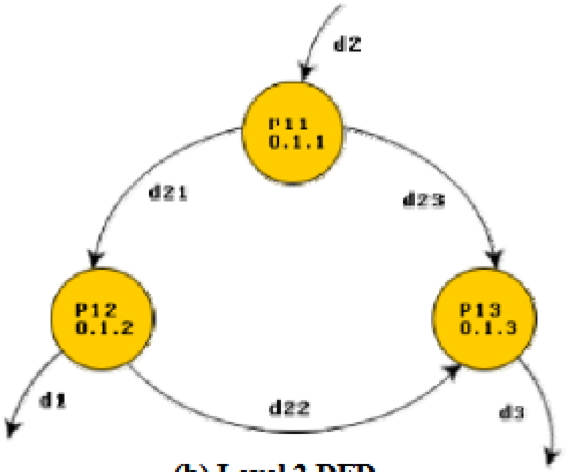
An example showing balanced decomposition



In the next level, bubble 0.1 is decomposed.

(a) Level 1 DFD

In the level 1, data item (d1, d3) flow out of the bubble 0.1. and the data item(d2) flow into the bubble 0.1.



The decomposition is balanced. As (d1, d3) flow out of the level2 diagram and d2 flows in.

(b) Level 2 DFD

Fig 1.6 Example of Balanced Decomposition

Types of DFD

Design Feature	Logical	Physical
What the model depicts?	How the business operates?	How the System will be implemented? Or, How the current system operates?
What the processes represent?	Business activities	Program, Program modules and manual procedures.
What the data stores represent?	Collection of data regardless of how the data are stored?	Physical Files and databases, manual files
Type of data store	Show data store representing permanent data collection.	Master files, Transaction Files. Any processes that operate at two different times must be connected by a data store.

System Controls	Show Business Controls.	Show control for validating input data. For obtaining a record, ensuring successful completion of a process and for system security.

Difference between Flowchart and DFD

Criteria	Flowchart	DFD
Flow of	Control	Data
Process Execute	one at a time	Parallel
Flow of data through	an Information Processing System	Business Processes
Action	Physical	Logical
View of the System	High Level	Lower Level
Input form or output to external Source	Does not have any	Describes the path of data from external source to internal store or vice versa
Timing and Sequence	Aptly Shown by flow chart	Particular order or simultaneously is not described
Show	How to make a System Function	Define the functionality of the system
Used	Designing a Process	Describes the path of data that will complete that process
Types	System Flow chart, Data Flow chart, Document Flow chart and Program Flow Chart	Physical DFD Logical DFD

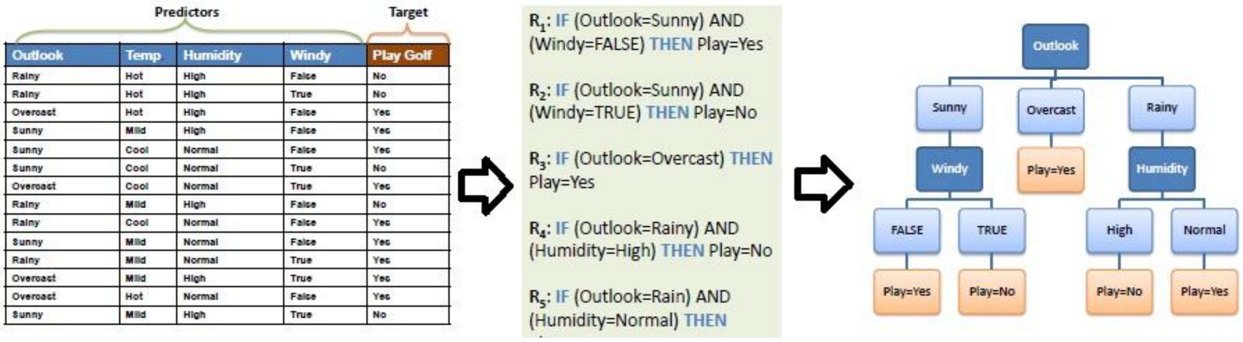
With the help of a **Question** let us understand different Structured Analysis tool …..

If Customer is individual and purchases 10 or more than 10 books, discount of 20% is given.
If Customer is individual and purchases less than 10 books but more than 5 books, discount of 15% is given.
If Customer is individual and purchases 5 or less than 5 books, discount of 10% is given
If Customer is librarian and purchases 25 or more books, discount of 15% is given
If Customer is librarian and purchases more than 10 but less than 25 books, discount of 10% is given

Decision Tree: Provides a graphical representation of decision logic that non-technical people find easy to understand. It's an easy mapping of a computer design which shows branch points but not the details of the user dialogues. It helps to show the path that is possible in a design following an action or decision by the user. Thus, it helps the designer to visualize how the user will move through the design to reach a desired location.

In Simple words, a decision tree provides an over view of the flow of control to be built into the computer program.

International Journal of Management, IT & Engineering
Vol. 9 Issue 2(1), February 2019,
ISSN: 2249-0558 Impact Factor: 7.119
Journal Homepage: http://www.ijmra.us, Email: editorijmie@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A



Predictors | Target

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

R_1: IF (Outlook=Sunny) AND (Windy=FALSE) THEN Play=Yes

R_2: IF (Outlook=Sunny) AND (Windy=TRUE) THEN Play=No

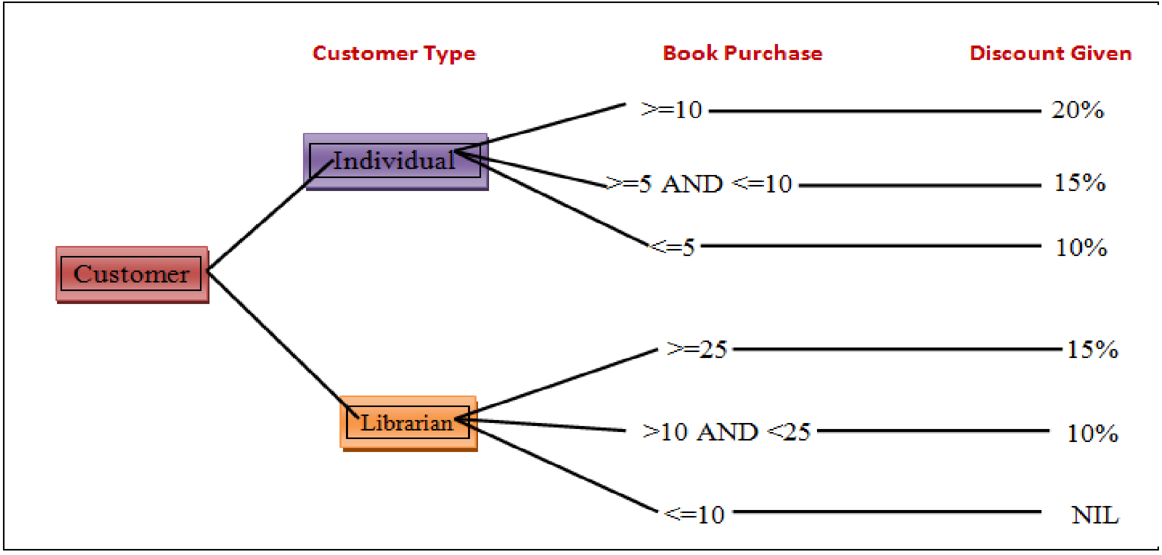
R_3: IF (Outlook=Overcast) THEN Play=Yes

R_4: IF (Outlook=Rainy) AND (Humidity=High) THEN Play=No

R_5: IF (Outlook=Rain) AND (Humidity=Normal) THEN

Fig 1.7 Example of Decision Tree

Solution for the above mentioned Question

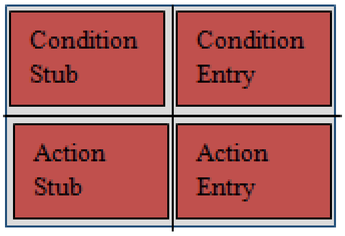


Decision Table: They are non-procedural specification of decision rules. They have been used extensively in the systematic design of information system. They have also been used for communication and documenting complex decision procedures.

A decision table defines a logical procedure by means of a set of condition and related actions. Thus, it is a tool to represent complex processing decision in a compact form.

Components of decision table:

- Condition Stub – lists all the condition to be checked
- Action Stub – outlines all the action to be carried out to meet such condition.
- Condition Entry – provides answers to questions asked in condition stub quadrant.
- Action Entry – appropriate action resulting from the answers to the conditions in the condition entry quadrant.



Condition Stub	Condition Entry
Action Stub	Action Entry

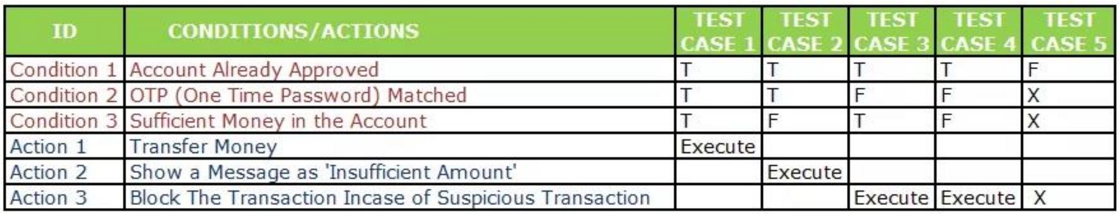
Fig 1.8 Decision Table

International Journal of Management, IT & Engineering
Vol. 9 Issue 2(1), February 2019,
ISSN: 2249-0558 Impact Factor: 7.119
Journal Homepage: http://www.ijmra.us, Email: editorijmie@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

Rules

- (Y)- existence of a condition.
- (N)– condition which is not satisfied.
- (blank or a hyphen) - states it is to be ignored.
- X (or a check mark will do) - action states it is to be carried out.

INPUT CONDITION	**Condition Entry (Rule 1)**	**Condition Entry(Rule 2)**	**Condition Entry (Rule 3)**
ACTION	**SubsequentAction Entry**	**SubsequentAction Entry**	**SubsequentAction Entry**



ID	CONDITIONS/ACTIONS	TEST CASE 1	TEST CASE 2	TEST CASE 3	TEST CASE 4	TEST CASE 5
Condition 1	Account Already Approved	T	T	T	T	F
Condition 2	OTP (One Time Password) Matched	T	T	F	F	X
Condition 3	Sufficient Money in the Account	T	F	T	F	X
Action 1	Transfer Money	Execute				
Action 2	Show a Message as 'Insufficient Amount'		Execute			
Action 3	Block The Transaction Incase of Suspicious Transaction			Execute	Execute	X

Fig 1.9 Example of Decision Table

Decision table is of 3 types

- LEDT- Limited Entry Decision Table
  - The question asked have only 'YES' or 'NO' as answer and the action can be denotes only by 'X' (Cross) or '-' (Hyphen). No other symbol or code is allowed. The numbers of condition are thus more in this type of decision table.
  - Here, the Question is written in the condition stub and their answer in the condition entry part of the decision table.

Solution for the above mentioned Question

Conditions	**Rule 1**	**Rule 2**	**Rule 3**	**Rule 4**	**Rule 5**
Customer is an Individual	Y	Y	Y		
Customer is a Librarian				Y	Y
Purchased 5 or less Books			Y		
Purchased more than 5 Books		Y			
Purchased 10 or more Books	Y	N			Y
Purchased 25 or more Books				Y	N
Actions					
10% Discount given			X		Y
15% Discount given		X		X	
20% Discount given	X				

- EEDT- Extended Entry Decision Table

○ It can have multiple answer based on the various type of Question asked which is extended into the condition entry part of the decision table. It should be observed that each question is formulated by combining the statement in the condition stub with that in the condition entry of the decision table which is commonly known as EEDT.

Solution for the above mentioned Question:

Conditions	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
C1: Customer?	Individual	Individual	Individual	Librarian	Librarian
C2: Number of Books	>=10	>5 and <10	<=5	>10 &<25	>=25
Actions					
A1: Discount =	20%	15%	10%	10%	15%

- MEDT - Mixed Entry Decision Table
  - The Question dealt in LEDT and EEDT can also be shown in another form with same question extending to the entry part and other being limited to the condition stub. This table can have 'YES' or 'NO' in the condition entry as well as different answer of the question in condition stub.
  - It is compromise between LEDT and EEDT. Thus the choice of the form in which a decision table is formulated depends on the problem, available software system, etc.

Solution for the above mentioned Question

Conditions	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
C1: Individual Customer?	Y	Y	Y	N	N
C2: Number of Books	>=10	>5 and <10	<=5	>10 &<25	>=25
Actions					
A1: 10% Discount given			X	X	
A2: 15% Discount given		X			Y
A3: 20% Discount given	X				

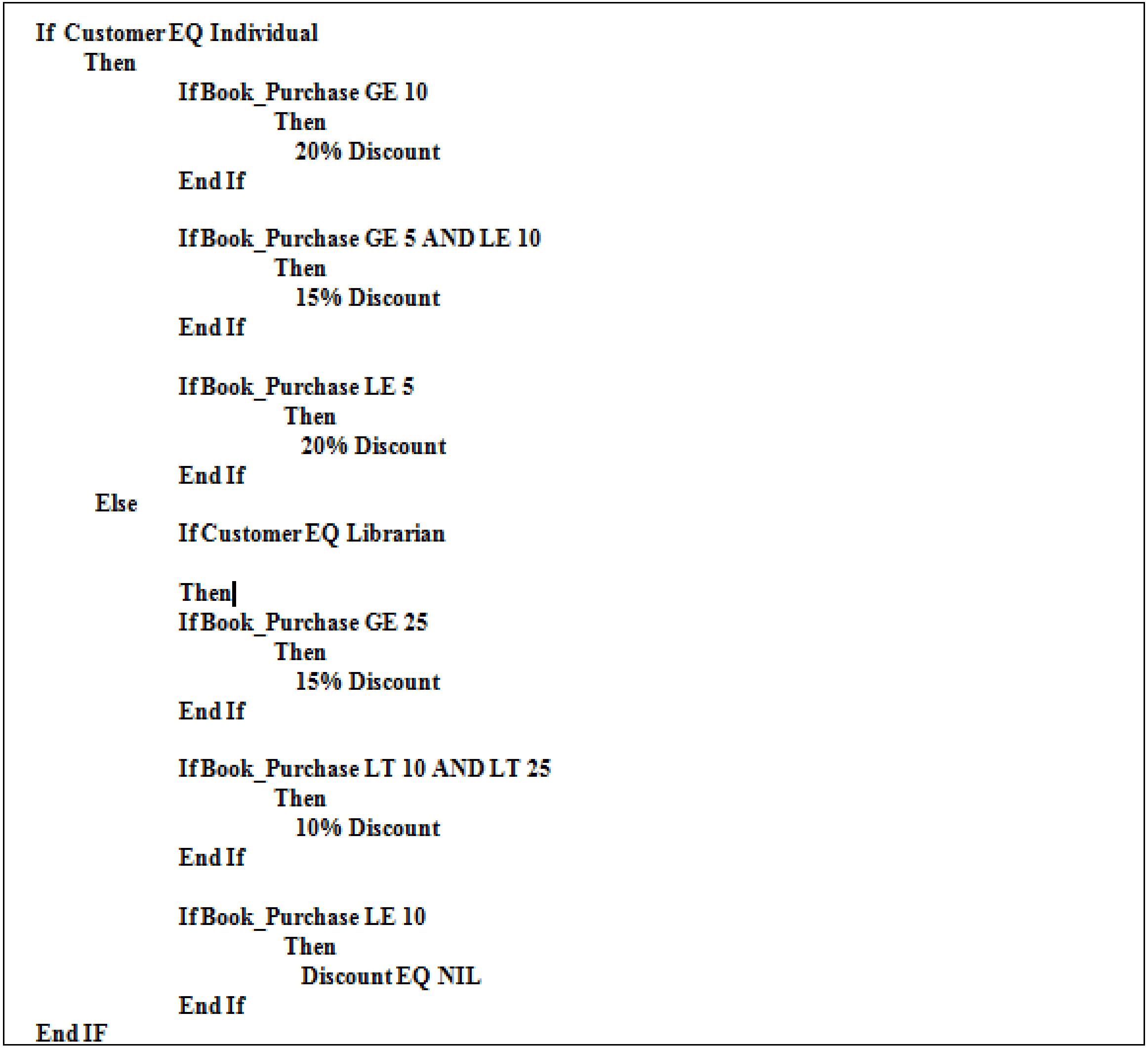
Structured English: Structured English is similar to a programming language such as Pascal, which does not have strict syntax rules in comparison to other programming languages. The intention is only to give precise description of a process in a simple English language which should be understandable to the user. In case of structured English, the following conventions are used in writing structured English process description:

- Imperative Statements: Keywords
- Arithmetic and relational operation: (+, *, -, /)(LT means less than, LE means less than equal To, GT means Greater than, GE means Greater than equal To, EQ means Equal To ,NE means Not Equal To)
- Decision Structures: (If..then..Else) (Case) Statements
- Repetition: Looping (For, While, Do while)

  The point to remember is to make individual procedures simple, so that a non-computer specialist can easily read and understand the procedure.

Notation	Meaning
LT	Less than
LE	Less than Equal to
GT	Greater than
GE	Greater than Equal to
EQ	Equal To
NE	Not Equal To

Solution for the above mentioned Question



```
If Customer EQ Individual
     Then
               If Book_Purchase GE 10
                         Then
                           20% Discount
               End If

               If Book_Purchase GE 5 AND LE 10
                         Then
                           15% Discount
               End If

               If Book_Purchase LE 5
                          Then
                            20% Discount
               End If
      Else
               If Customer EQ Librarian

               Then
               If Book_Purchase GE 25
                         Then
                           15% Discount
               End If

               If Book_Purchase LT 10 AND LT 25
                         Then
                           10% Discount
               End If

               If Book_Purchase LE 10
                          Then
                            Discount EQ NIL
               End If
End IF
```

Data Dictionary: Defines each term (element) encountered during the analysis and design of a new system. It is a special kind of dictionary which contains all the information about the data of a system. It forms an integral part of structured specification. It is only a documentation of data. The important components of data dictionary are:

- Data Element: Smallest unit of data that is meaningful or pieces of data which can't be meaningfully decomposed are known as data elements. For each data element the data dictionary should hold the following description:
    - Name
    - Description
    - Name of the relative data element
    - Length and type of data element
    - Codification Structure
    - Range of value and their meaning
- Data Structure: They are made up data element or a combination of both data element and data structure. They should contain the following information:
    - Name

- ○ Description
- ○ Included data elements and data structure with brief description

- Data Flow: Contains the volume of flow (Start, End) point with the description of data.
- Data Store: Additional information on the volume and data flows to and from it is also usually recorded.
- Process: Documenting the processes include documenting name and brief description of the process followed by a process description. The entry for each and every process forms a part of data dictionary.

Apart from a name and number the Entry contains the process description
What the process has to do?
Process Specification should be specified in such a way that they are both logically correct and easy to understand.

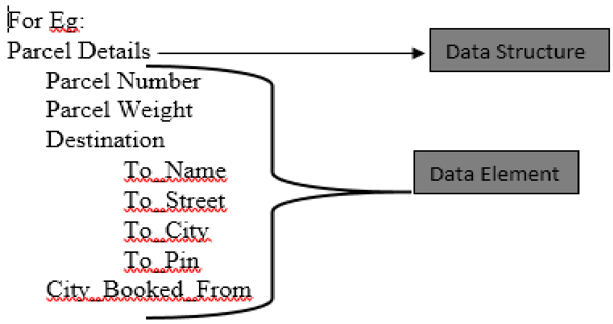
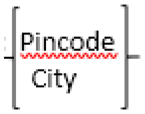


Fig 1.10 Example of Data Dictionary

Data Structure Notation: They are expressed in terms of a related set of components. Convention used in depicting data structures are:
{} Any one option can be selected. For Eg:



(n-m) Specifies number of occurrence from n to m times. Lower limit is must whereas upper limit is optional. For Eg: Item-details(1-3)
[] Enclosed Component is optional. For Eg: If the purchase order need not be present then it will be represented as [Person Sign]
For Eg: A table that contains employee details

Field Name	Data Type	Field Size for display	Description	Example
Employee Number	Integer	10	Unique ID of each employee	1645000001
Name	Text	20	Name of the employee	David Heston
Date of Birth	Date/Time	10	DOB of Employee	08/03/1995
Phone Number	Integer	10	Phone number of employee	6583648648

Structured Design: It is a data-flow based methodology which approach begins with a system specification that identifies I/O and describes the functional aspects of the system. Structured design partitions a program into small, independent modules. It's an attempt to minimize complexity and make a problem manageable by subdividing it into smaller segments., which is called modularization or decomposition which is discussed in next section in detail.

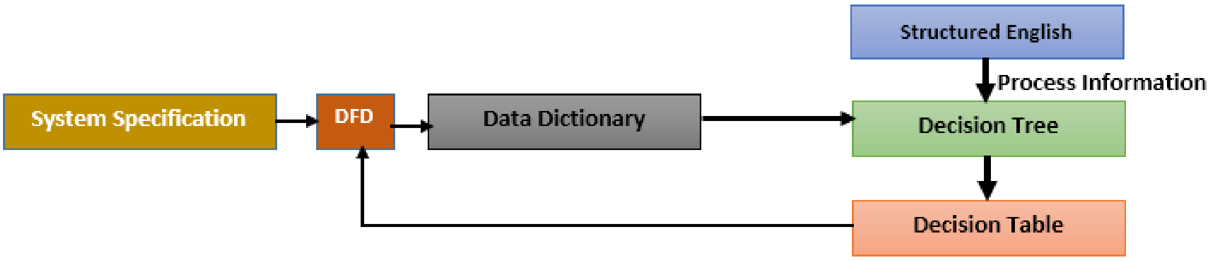


Fig 1.11 Structured Design Method

Structured Chart: A top-down modular design tool which shows the breakdown of a system to its lowest manageable levels with the help of graphical representation of:

- Squares represent different modules in the system
- lines that connect modules which shows the relationship between modules.

The documentation tool for structured design is the hierarchy or structure chart. The use of only two graphical elements forces the chart designer to provide only the essential information. A structure chart is a tree of sub-routines in a program which indicates the interconnections among the sub-routines. The sub-routines should be labeled with the same name used in the pseudo code.

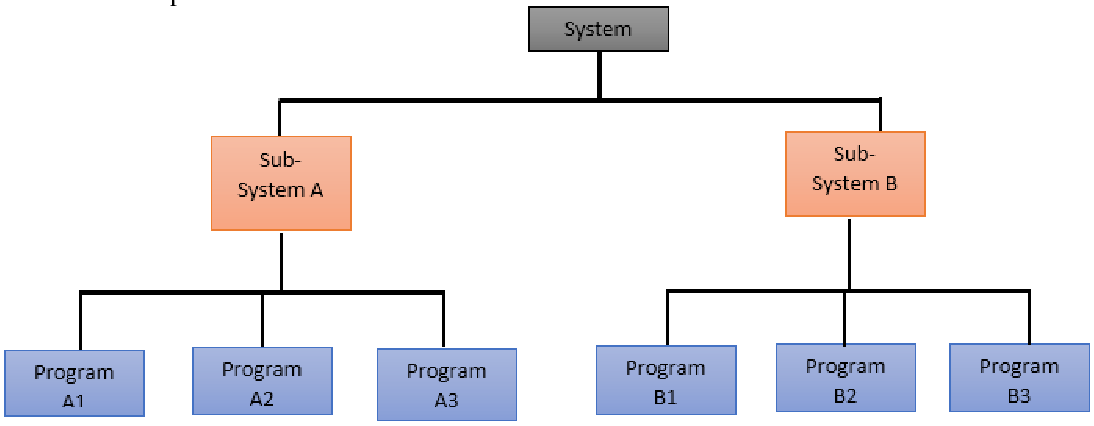
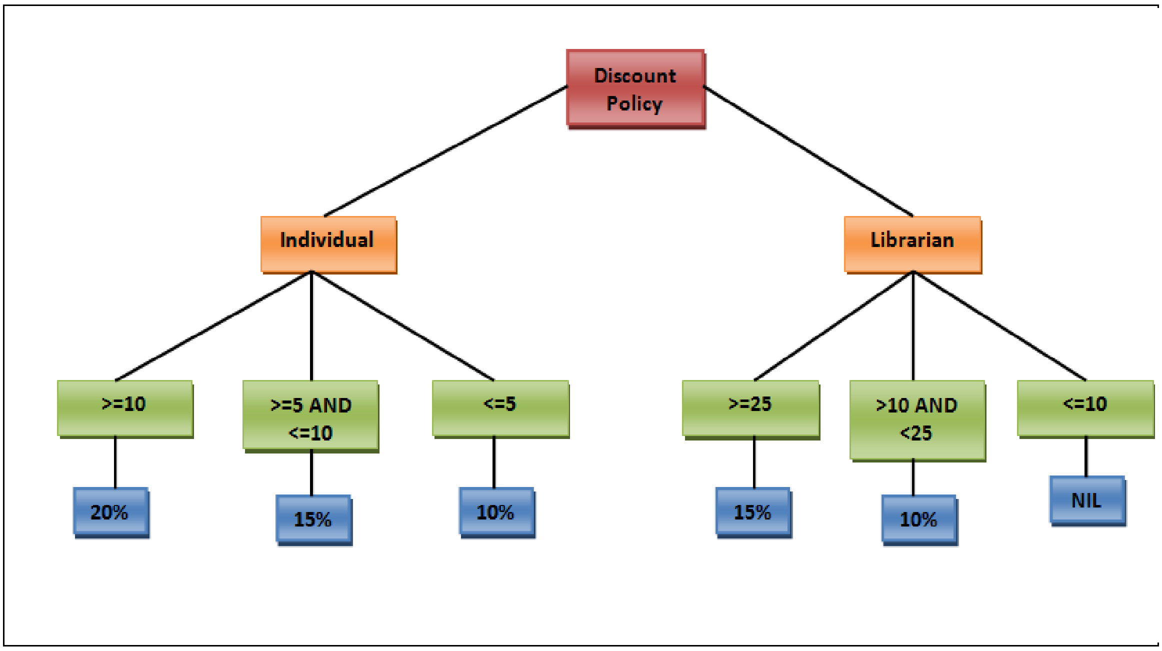


Fig 1.12 Example of Structured Chart in Programming

A structure chart depicts:

- the size and complexity of the system, and
- number of readily identifiable functions and modules within each function and
- Whether each identifiable function is a manageable entity or should be broken down into smaller components.

Solution for the above mentioned Question



Module Specification: Any system in its entirely (as a whole) performs a number of functions. A system has to be divided into programs containing one or more modules. A great care has to be taken while dividing the system into programs or module. A software system cannot be modular by simply breaking it into a set of module, for modularity each module needs to support a well-defined abstraction and have a clear interface through which it can interact with other modules.

To segment the system, the designer examines the entire system to decide the program and module boundaries. The main concepts here are the:

Coupling is between the modules and Cohesion within the module

Coupling is the measure of or degree of interdependence between modules. Two modules with high coupling are strongly inter-connected and thus depend on each other. Modules should be **loosely coupled** which means that modules should have little dependence on other modules in a system.

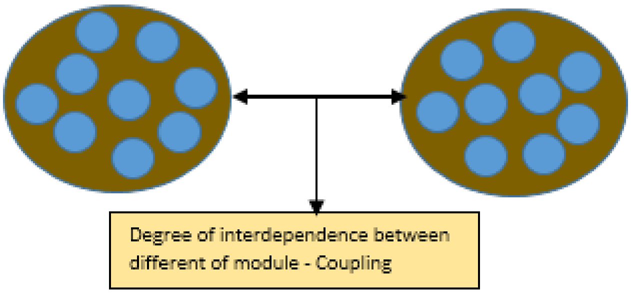

Degree of interdependence between different of module - Coupling

Fig 1.13 Coupling

Types of Coupling	Description	Example
Data	communicate by passing only data components are independent to each other and communicating through data. Therefore, it doesn't contain tramp data.	customer billing system parameter passing
Stamp	complete data structure is passed from one module to another module. Therefore, it involves tramp data.	.
Control	modules communicate by passing control information	Sort() that takes comparison function as an argument
External	the modules depend on other modules	protocol, external file, device format, etc
Common	modules have shared data	global data structures
Content	one module can modify the data of another module or control flow is passed from one module to the other module. Worst form of module	

Cohesion is the property that specifies how tightly bound the elements of a module are. Thus, it is a measure of the degree to which elements of the module are functionally related. A cohesive module performs a single task within software procedure requiring little interaction with procedure being performed in other part of the program. Modules should be **highly cohesive** which means that modules should carry out a single processing function.

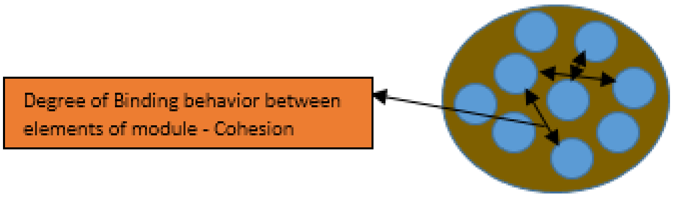

Degree of Binding behavior between elements of module - Cohesion

Fig 1.14 Cohesion

Types of Cohesion	Description	Example
Functional	Ideal Situation. Degree of cohesion- HIGHEST The elements of a module are functionally grouped into a logical unit and they work together as a logical unit.	Sort an array..sort() Search()
Sequential	An output of some data becomes the input for other element. Occurs naturally in programming language.	Search array for a certain name. Place the name in another array and sort the array. Change the name from last-first to first-last. Print first-last name.
Communicational	Two elements operate on the same input data or contribute towards the same output data	update record in the database and send it to the printer Transaction File
Procedural	Ensure the order of execution	calculate student GPA, print student record,
Temporal	Statements are grouped into a procedure and executed together during the same time-frame	Init() Component that is used to initialize a set of objects.
Logical	elements are logically related and not functionally..AVOIDED	
Coincidental	The elements are not related. breaking a module into smaller modules Accidental and the worst form of cohesion.	print next line reverse the characters of a string in a single component

The designer aims at minimizing coupling and maximizing cohesion to get a modular design. Such a design ensures that each module perform a specific function and can be developed relatively independently by programmers. Thus, a good design is characterized by low coupling that is low interdependence between the modules and high cohesion that is high inter- relationship within the elements of a single module.

Conclusion: System analysis is the detailed evaluation of a particular system to identify areas for improvements and make any further changes if necessary which includes: gathering the company requirements and to research the path that is to be taken to effect these requirements. The ultimate target is to have a fully operational system in place which provides efficiency and reliability to the company. When a system analysis is properly performed, it ensures that the correct path is taken with regards to applications which help to minimize errors thereby reducing future IT requirements for fixing problems. System analysis allows for better management policy through changing the software to suit any business changes which indicate that the final product will be totally controllable. The quality of the systems is ensured through the checking of the system on regular basis through system analysis. A risk assessment is carried out to evaluate all the negative effects on the processes. It can also be used to improve procedures in handling accounts receivable, in preparing and implementing a budget and in scheduling regular or one-time projects. The benefits of an integrated system include higher levels of quality control and lower production costs by streamlining data processing and production processes. The main objectives of business are to make money. An obvious advantage of using system analysis and design is to improve business quality thereby increasing profits.

Although System analysis offers a wide range of benefits it might also have some limitation. One of the main drawbacks which are mostly overlooked is the risk of too much analyzing which may be costly and time consuming. It is therefore part of the analyst's job to find the right balance.

Reference

- Amyuni, T.: PDF Vol. 2010. Amyuni Technologies Inc., Montreal, Canada (2010)
- De Marco, T., Structured Analysis and Systems Specification on, Prent ice Hall, 1978.
- Definition", IEEE Transactions on Software Engineering 3(1), January 1977.
- Definition," IEEE Transactions on Software Engineering 3(1), Special Issue on Requirements Analysis, January 1977, 86-95.
- Dixit, J. B. and Kumar, R. (2008). Structured System Analysis and Design. Paperback ed. New Delhi, India: Laxmi Publisher.
- Fleming, C. and Von Hall, B., (1990). An Overview of Logical Data Modelling, Data Resource Management, Vol.1, No.1, pp 5-15
- Froelich, J., Ananyan, S.: Decision Support via Text Mining. In: Burstein, F., Holsapple, C.W. (eds.): Handbook on Decision Support Systems 1. Springer Berlin Heidelberg (2008) 609-635
- Gane, C. and Sarson, T., Structured Systems Analysis, Prentice Hall, 1979.
- Gao Xiaolei, Miao Huaikou and Liu Ling. (2004). Functionality Semantics of Predicated Data Flow Diagram. Journal of Shanghai University (English Edition), Vol 8, No 3, pp. 309 – 316.
- IEEE Transacti ons on Software Engineering 3(1), Special Issue on Requirements Analysis, Ja nuary 1977, 16-34.
- Klink, S., Dengel, A., Kieninger, T.: Document Structure Analysis Based on Layout and Textual Features. International Workshop on Document Analysis Systems, Rio de Janeiro, Brasil (2000) 41-52
- Lejk, M and Deeks, D., (2002). Systems Analysis Techniques, 2nd ed., Pearson-Education, UK
- Mittra, Sitansu S., (1988). Structured techniques of System Analysis, Design and Implementation, John Wiley & Sons, New York
- Ros s, D. and Scho man, K., "Structured Analysis for Requirements
- Ross, D., "Structured Analysis: A Language for Communicating Ideas,"
- Ross, D., and Schoman, K., "Structured Analysis for Requirements

- Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorensen, W., (1991). Object Oriented Modeling and Design, Prentice Hall, Englewood Cliffs, NJ
- Satzinger, J. , Jackson, R. , Burd, S, (2004). Systems analysis and Design in a changing world, Thomson Learning Inc
- Schmoekel, I.: PDF-Analyzer Pro 4.0. Vol. 1. Software-Development and Distribution, Achim-Uesen, Germany (2010) 1-11
- Tao, Y.L. and Kung, C.H. (1991). Formal Definition and Verification of Data Flow Diagrams. Journal of Systems and Software, 16(1), pp. 29-36.
- Wirth, N., " Program Development by Stepwise Refinement," Communi cations of the ACM 14 (4), 221-227, 1971.
- Yourdon, E., (1989). Managing the Structured Techniques, Englewood Cliffs, NJ, Prentice-Hall
- Yourdpn, E., Modern St ructured Analysis, Yourdan Press, 1989.