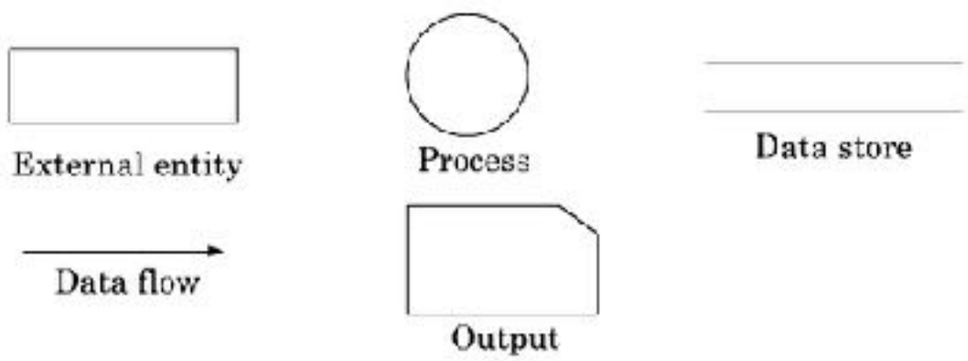
SE Assignment III

1. **List and explain primitive symbols used for constructing DFDs. Illustrate the use of these symbols with the help of an example.**

There are essentially 5 different types of symbols used to construct DFDs.



Function symbol

A function is represented by a circle. Symbol is called a process or a bubble.

External entity symbol

An external entity such as a librarian, library member, etc is represented by a rectangle.

Data flow symbol

A directed arc (arrow) is used as a data flow symbol. It represents the data flow occurring between two processes or between an external entity and a process in the direction of the data flow arrow.

Data store symbol

A data store is represented using two parallel lines. It represents a logical file i.e. a data store symbol can represent either a data structure or a physical file on disk.

Output symbol

Output symbol is used when an hard copy is produced.

2. **With the help of an example explain how you transform a DFD model into Structure chart.**

Given a specific DFD model, one examines the data input to the diagram. If all the data flow in the diagram are processed in similar wars, then the transform analysis is applicable.

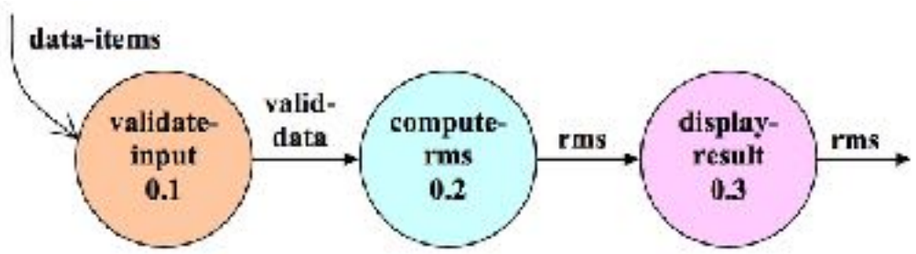
The first step in transform analysis is to divide the DFD into three parts:

Input *(processes that transform input data from physical to logical form)*
Processing *(processes that perform operations on logical data)*
Output *(processes that transform logical data to physical form)*

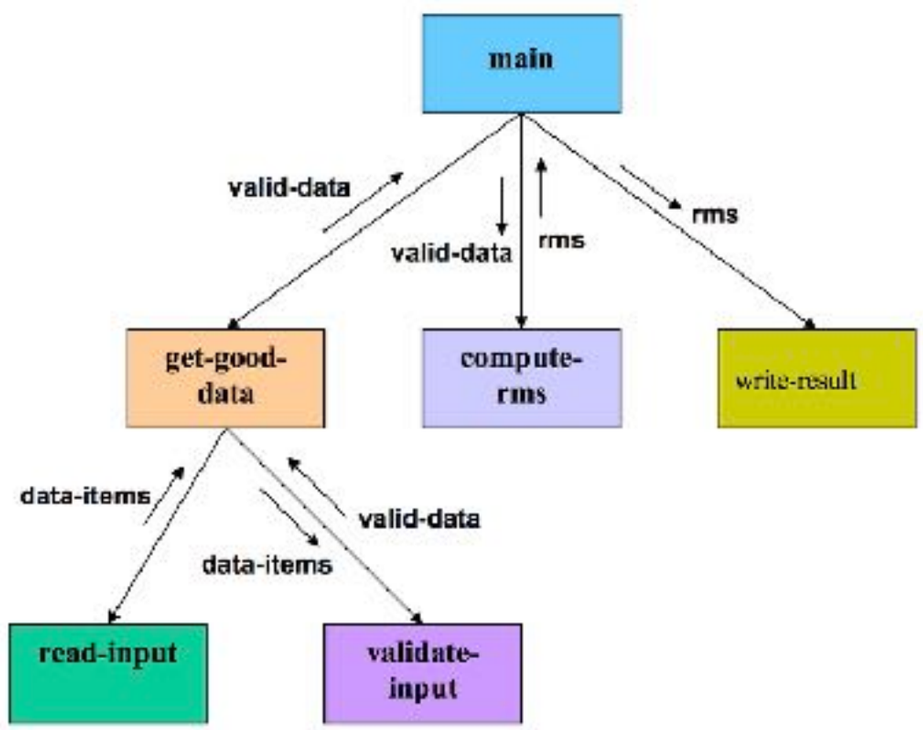
In the next step of transform analysis, the structure chart is derived by drawing one functional component for the central transform, and the afferent and efferent branches. These are drawn below a root module, which would invoke these modules.

In the third step of transform analysis, the structure chart is refined by adding sub-functions required by each of the high-level functional components. Many levels of functional components may be added. This process of breaking functional components into subcomponents is called factoring.

For example, a DFD for a RMS software,



can be converted into a structure chart as follows:



3. **What views of the system is captured by UML diagram. Elaborate your answer.**

UML diagrams can capture the following views of a system:

User's View

The users' view captures the view of the system in terms of the functionalities offered by the system to its users.

It's a black box of the system where the internal structure, the dynamic behavior of different system components, the implementation, etc are not captured.

Structural View

The structural view defines the structure of the problem in terms of the loads of objects and (classes)

It captures the relationships among the classes (objects).

Behavioral View

It captures how objects interact with each other in time to realize the system behavior. The system behavior captures the dynamic behavior.

Implementation View

This view captures the important components of the system and their interdependencies. For example, the implementation view might show the GUI part, the middleware and the database part as separate modules and interconnections between them.

Environmental View

This view models how the different components that are implemented on different pieces of hardware.

4. **Write a note on Activity diagram, use case diagram and class diagram.**

Activity diagram describes the flow of control in a system. So it consists of activities and links. The flow can be sequential, concurrent or branched.

Activities are nothing but the functions of a system. Numbers of activity diagrams are prepared to capture the entire flow in a system.

Activity diagrams are used to visualize the flow of controls in a system. This is prepared to have an idea of how the system will work when executed.

Use case diagrams are a set of use cases, actors and their relationships. They represent the use case view of a system.

A use case represents a particular functionality of a system.

So use case diagram is used to describe the relationships among the functionalities and their internal/external controllers. These controllers are known as actors.

Class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. So a collection of class diagrams represent the whole system.

Class diagram is a static diagram and it is used to model static view of a system. The static view describes the vocabulary of the system.

Class diagram is also considered as the foundation for component and deployment diagrams. Class diagrams are not only used to visualize the static view of the system but they are also used to construct the executable code for forward and reverse engineering of any system.

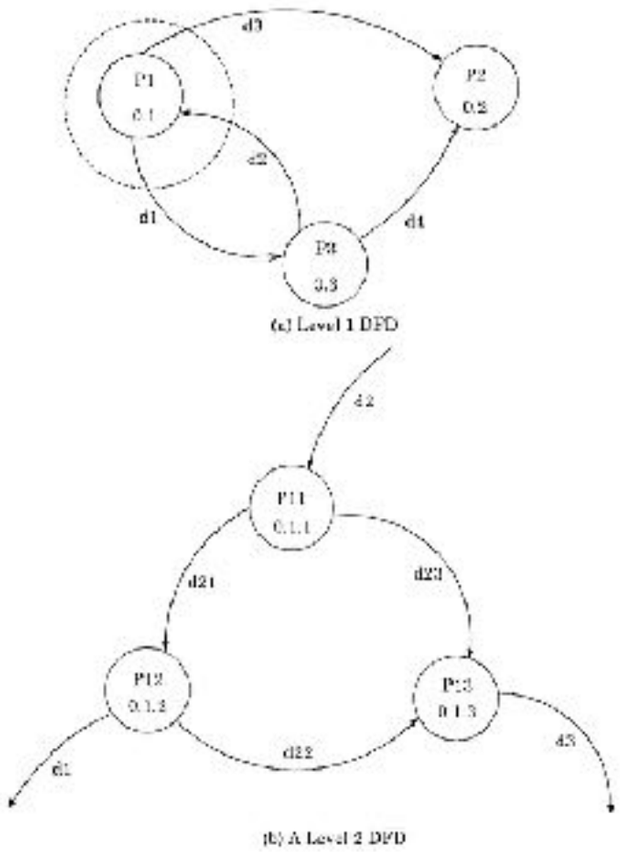
5.

A. **What do you mean by balancing a DFD? Illustrate your answer with a suitable example.**

The DFD model of a system usually consists of many DFDs that are organized in a hierarchy. In this context, a DFD is required to be balanced with respect to the corresponding bubble of the parent DFD.

The data that flows into or out of a bubble must match the data flow at the next level of DFD. This is known as balancing a DFD.

For example, In level 1 DFD, the data items d1 and d3 flow out of bubble 0.1, in the level 2 DFD, bubble 0.1 is decomposed into three DFDs, 0.1.0, 0.1.2, and 0.1.3. This decomposition is balanced as d1 and d3 flow out of the system and d2 flows in.



B. Which aspects are checked during design review? Briefly explain.

The design review team checks the design documents for the following aspects:

Traceability

Whether each bubble of the DFD can be traced to each module in the structure chart and vice versa.

Correctness

Whether all algorithms and data structures of detailed design are correct.

Maintainability

Whether the design can be easily maintained in the future.

Implementation

Whether the design can be easily and efficiently be implemented