Data Flow Diagrams:

The most common tool used for deigning database systems is Data Flow Diagram. It is used to design systems graphically and expresses different system detail in different DFD levels.

DFDs show the flow of data between different processes o a specific system.

DFDs are simple, and hide complexities.

DFDs are Descriptive and links between processes describe the information flow.

- **Limitation of DFDs**

They do not provide us a way of expressing decision points.

DFDs are focused on flow of information only.

- **Symbols used in DFD:**

There are a limited number of symbols which are used for design process in DFDs.

- **DATAFLOW:**

The purpose of the dataflow in a DFD is to express the flow of information from one entity to another entity in the system

Data flows are pipelines through which packets of information flow.

Arrows are labeled with name of the data that moves through them. Figure-4 below show the Dataflow diagram

Fig: 4. Dataflow Symbol

- **DATA STORE:**

Data store is a repository for the storage of the data. When in a system the data is to be permanently stored somewhere for future reference or use the DATASTORE is used for this purpose. It is express with a rectangle open on right width and left width of the rectangle drawn with double lines.

Data in the DATASTORE is held sometimes for processing purposes also i-e it may not be a permanent data store.. Name of the DATASTORE is a noun which tells the storing location in the system. Or identifies the entity for which data is stored. Figure-5 shows a data store.

Fig: 5. Data store

- **Processes:**

Processes are expressed with ovals or rounded rectangles. Processes are used to express the transformation of incoming dataflow into outgoing dataflow. Process symbols are used for whatever is the action taking place and whatever is the magnitude or complexity of the action. Simply stating when data is transformed from one form into another the process symbol is used. Figure-6a and Figure-6b show two different shapes used for presenting process in DFD.

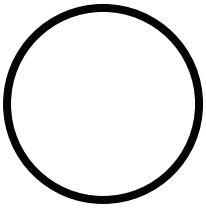


Fig: 6a Process

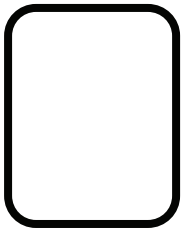
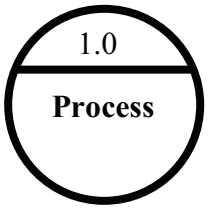


Fig-6b

- **DFD-Process:**

In DFD processes are numbered for expressing their existence at a certain level in the system.



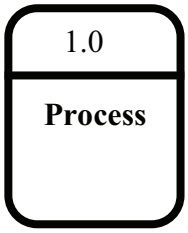


Fig: 7. Numbered DFD Processes

- **External Entities:**

These are the entities interacting with the system in any of two different ways. They may be either receiving the data from the system, or may be producing the data for the system to consume.

Shape used to express external entities is rectangle. The shape for external entity is shown in Figure-8.

Fig: 8. External Entity

- **Collector:**

This DFD shape is used to express several dataflow connections terminating at a single location. Collector is used to show the convergence of data to a single point. Fig 9a shows the Collector symbol and Fig 9b show a collector symbol acting as a sink for multiple data flows.

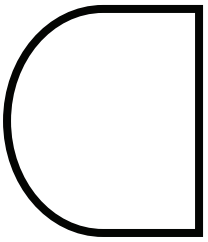


Fig: 9a Collector

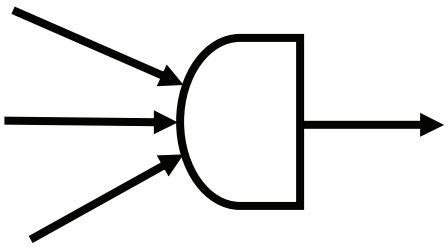


Fig 9b. Collector with Multiple Dataflow

- **Separator:**

The dataflow symbol which is used for separating data from a single source to multiple sinks is known as a separator.

Figure 10a show the presentation of separator and the figure 10b shows the separator as it may appear in a DFD.

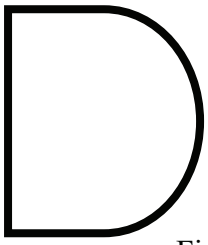


Fig: 10a Separator

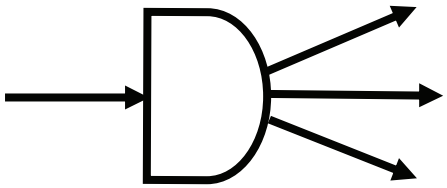


Fig 10b. Separator with Multiple Dataflow

- **Ring Sum Operator:**

This operator is used when data from a source process can flow to one of the mentioned sinks. For this purpose the symbol used is displayed in Figure: 11a and its presentation in a DFD is expressed in Figure-11b.

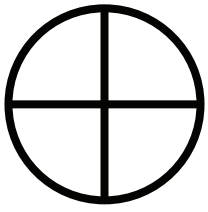


Fig: 11a Ring sum operator

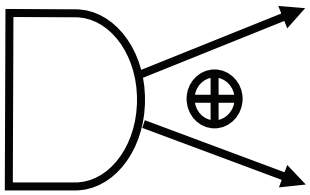


Fig 11b. Separator with Ring sum operator

- **AND Operator:**

This operator is used when data from a source process must flow to all the connected sinks. For this purpose the symbol used is displayed in Figure: 12a and its presentation in a DFD is expressed in Figure-12b.

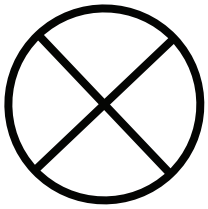


Fig: 12a AND operator

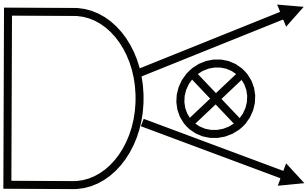


Fig 12b. Separator with AND operator

Types of DFD

- Context diagram
- Level 0 diagram
- Detailed diagram

- **Context Diagram:**

This is the level of DFD which provides the least amount of details about the working of the system. Context DFDs have the following properties:

They always consist of single process and describe the single system. The only process displayed in the CDFDs is the process/system being analyzed. Name of the CDFDs is generally a Noun Phrase.

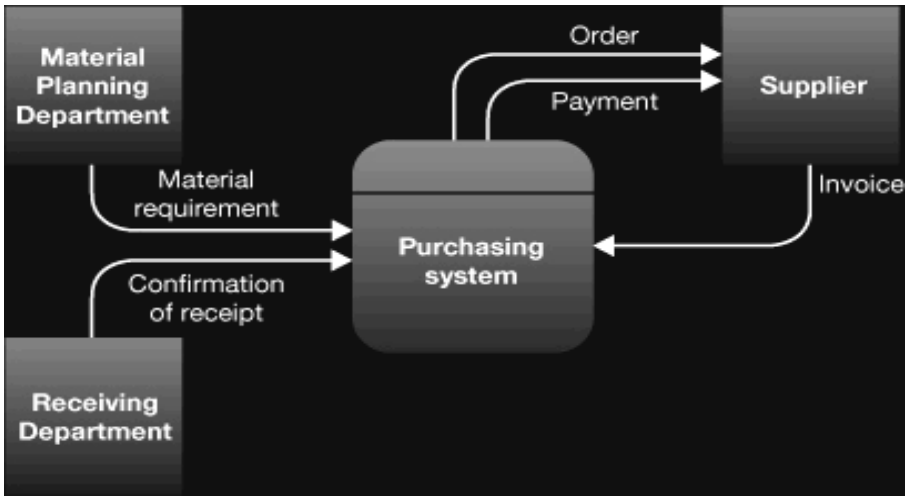


Fig: 13a. Example Context DFD Diagram

No System details are shown in the Contexts DFDs just context is shown. Input and output from and to the process are shown and interactions are shown only with the external entities. An example DFD at context level is shown in Figure: 13a and 13b.

In the context level DFDs no data stores are created. Ant dataflow from external entities are only directed toward the purported system and vice versa, no communication is show between external entities themselves.

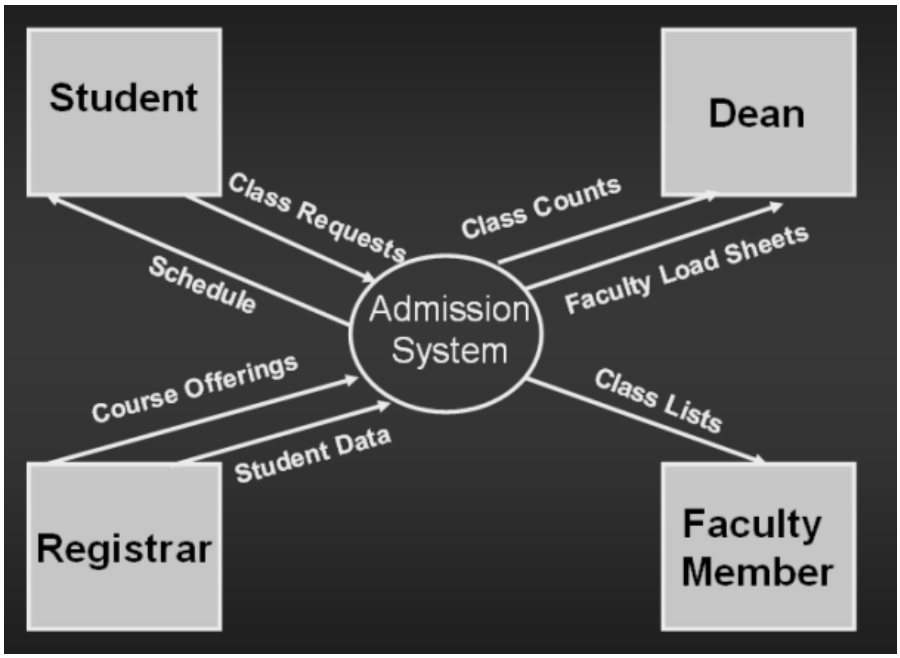


Fig: 13b. Example Context DFD Diagram

- **Level 0 Data Flow Diagrams:**

The level 0 Diagram in the DFD is used to describe the working of the whole system. Once a context DFD has been created the level zero diagram or level 'not' diagram is created. The level zero diagram contains all the apparent details of the system. It shows the interaction between a numbers of processes and may include a large number of external entities. At this level it is the duty of the designer to keep a balance in describing the system using the level 0 diagram. Balance means that he should give proper depth to the level 0 diagram processes. Because placing too much details and showing all of the miniature processes in the level 0 diagrams makes it too much complex. On the other hand it is also not recommended to just ignore even larger processes of the system, because in such a case although the level 0 DFD will become simple but now we will have to create large number of detail DFDs. So a balance in describing the system should be kept so that the depth of the Level 0 DFD is manageable.

- **Steps in creating the level 0 DFD**

1. Identify distinct modules of the system for which to create the DFD

2. Create DFDs for all the modules one by one to show the internal functionality of the system.
3. Once DFD for the distinct modules of the system have been created, establish link between different DFDs where required by either connecting the entities of the system, processes of the system or the data stores in different DFDs.
4. Now comes to the stage of placing the numbers on processes.
   As we know that the level 0 diagram encompasses a large number of smaller systems, ant is a combination of a number of context DFDs. In level 0 diagram a process when it has a lot of details, it is not explained further in the level 0, and rather it is postponed for the detailed diagram.
   In the detailed Data Flow and is given a number. Numbering processes is based on a specific notation, in the level 0 diagrams only left half or the portion before the decimal point is valid but in the detailed diagram when a complex process is expressed further its sub processes are number like 1.0, 1.1, and 1.2 and so on.

Lecture No. 06

Reading Material

"Database Systems Principles, Design and Implementation" written by Catherine Ricardo, Maxwell Macmillan.	Section 2.4

Overview of Lecture

- Detailed DFD Diagrams:
- Database Design Phase
- Data Models
- Types of Data Models
- Types of Database Designs

Detailed Data Flow Diagram:

This Type of the Data flow diagrams is used when we have to further explain the functionality of the processes that we showed briefly in the Level 0 Diagram. It means that generally detailed DFDS are expressed as the successive details of those processes for which we do not or could not provide enough details.

The symbols and other rules regarding the detailed DFD are same as are in other types of DFDs. The special features associated with this diagram are that, one, it is optional, that is, it is created for only those processes from the level 0 diagram for which we want to show the details. For a small sized system we may not need to develop even a single detailed DFD, since the level 0 diagram might be covering it sufficiently. Second specific characteristic of the detailed DFD is its processes' numbering. Numbering of processes in the detailed DFD is done on the basis of numbering of the particular process in level 0 diagrams whose sub-processes are being included in the detailed DFD. For example, a specific process which was numbered in the level 0 diagram as 1.0 or 1 may have a number of sub-processes since we did not represent the process 1.0 in detail in level 0 diagrams. So in the detailed dataflow diagram we create sub-processes of that process and then number all the sub processes of that specific process as the sublets of the process.

Numbering of such sub processes is done as 1.1, 1.2, and 1.3… for first second and third sub-processes of the process 1.0 respectively. The phenomenon of creating sub-processes does not end at creating a few sub-processes for a specific process shown at level 0 diagrams. Rather it may continue deeper if there is requirement for further explanation of the any process or sub-processes. In such a case when we create sub-process of a sub-process 1.2 then the numbering is done in further extension of that specific sub processes number and example of such a numbering process is 1.2.1, 1.2.2, 1.2.3,…

Another point that is worth mentioning here is that we call processes in the detailed DFDs as sub-processes, but they are sub-processes only in reference to the process whose details they are explaining otherwise they are just like processes; transforming some input data into some form of output. The sub-processes may be performing relatively small amount of operations, still they are processes.

Maximum Number of Process in a DFD should not be very huge. Having a moderate number for a detailed DFD is also recommended because it adds clarity to our detailed data flow diagram. For clarity propose it is good to have a maximum of 7 or 9 processes in one detailed DFD. Moreover all the processes, sub processes, data stores, entities data flows and all other components of the DFD must be named properly, so that anyone who is using this DFD should be able to understand the DFD easily.

In all the levels of DFD it must be considered that all the processes have data inputs as well as data outputs. Data being sent to one process should be processed so that it changes its form and transforms from one form to another.

When creating a detailed diagram the data inputs and data outputs must be in coincidence, mean in both the diagrams the data input to a process and data output in the form of data flows must be same.

Data Dictionary

A database that containing data about all the databases in the database system. Data dictionaries store all the various schema and file specifications and their locations. They also contain information about which programs use which data and which users are interested in which reports.

Types of Data Dictionaries:

- **Integrated**

There are basically two types of data dictionaries which are available for use by a DBMS, with respect to their existence.