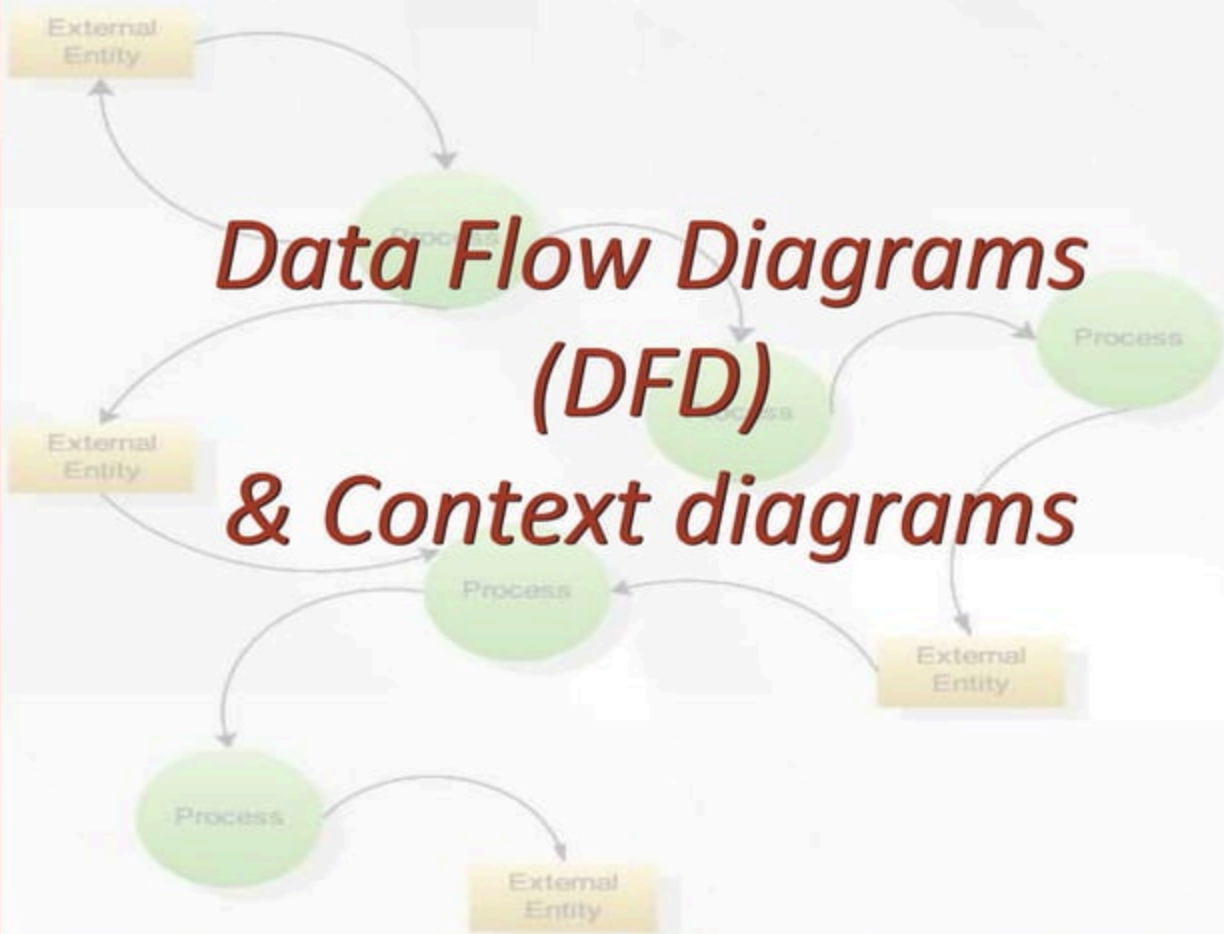


Data Flow Diagrams (DFD) & Context diagrams



Data Flow Diagrams

- A *graphical tool*, useful for communicating with users, managers, and other personnel.
- Used to perform *structured analysis* to determine logical requirements.
- Useful for analyzing existing as well as proposed systems.
- Focus on the *movement* of data between external entities and processes, and between processes and data stores.
- A relatively simple technique to learn and use.

Why DFD ?




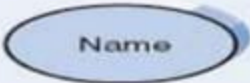
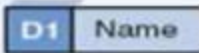
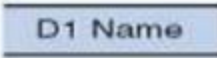
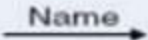
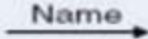
- Provides an overview of-
 - What data a system processes
 - What transformations are performed
 - What data are stored
 - What results are produced and where they flow
- Graphical nature makes it a good communication tool between-
 - User and analyst
 - Analyst and System designer

DFD elements

- Source/Sinks (External entity)
- Processes
- Data Stores
- Data flows



Symbols Used:

Symbol	Gane & Sarson Symbol	DeMarco & Yourdan Symbol
External Entity		
Process		
Data store		
Data flow		

Descriptions :

- **External Entity** - people or organisations that send data into the system or receive data from the system.
- **Process** - models what happens to the data
i.e. transforms incoming data into outgoing data.
- **Data Store** - represents permanent data that is used by the system.
- **Data Flow** - models the actual flow of the data between the other elements.

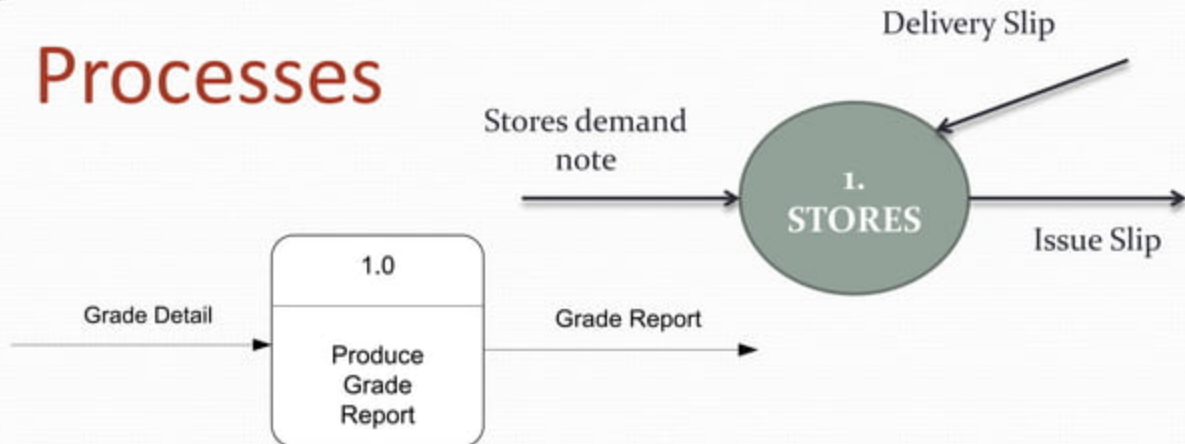
Symbol naming

- External Entity → Noun
- Data Flow → Names of data
- Process → verb phrase
- Data Store → Noun

External Entities

- They either supply or receive data
 - **Source** – Entity that supplies data to the system.
 - **Sink** – Entity that receives data from the system.
- They do not process data

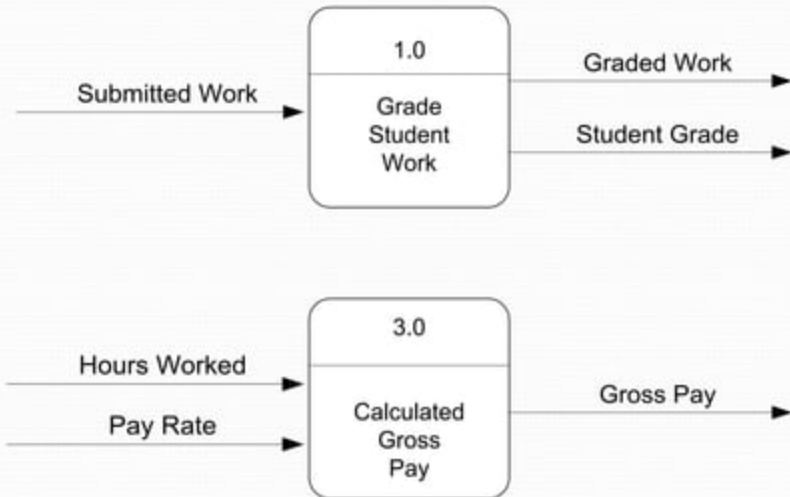
Processes



- Work or actions performed on data (inside the system)
- Straight line with incoming arrows are input data flows
- Straight lines with outgoing arrows are output data flows
- Labels are assigned to Data flow. These aid documentation

Processes

- Can have more than one outgoing data flow or more than one incoming data flow

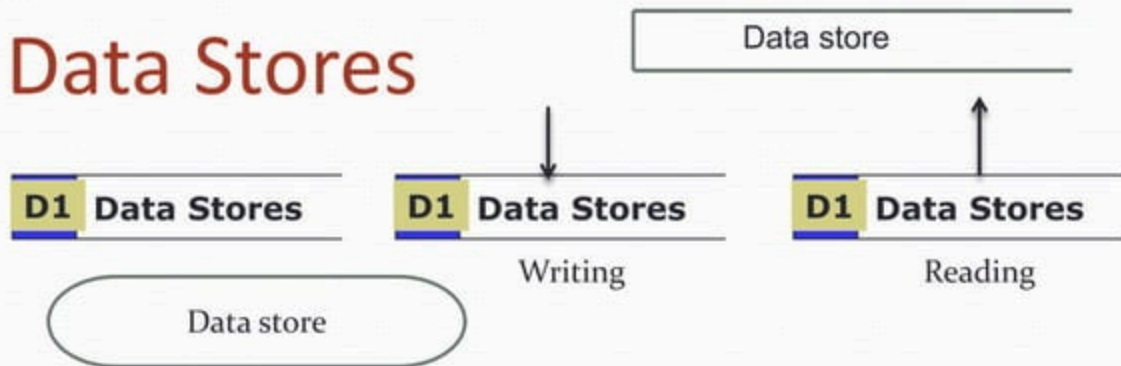


Processes

- Can connect to any other symbol (including another process symbol)
- Contain the business logic, also called business rules
- Referred to as a black box



Data Stores



- A Data Store is a repository of data
- Data can be written into the data store. This is depicted by an incoming arrow
- Data can be read from a data store. This is depicted by an outgoing arrow

Data Flows

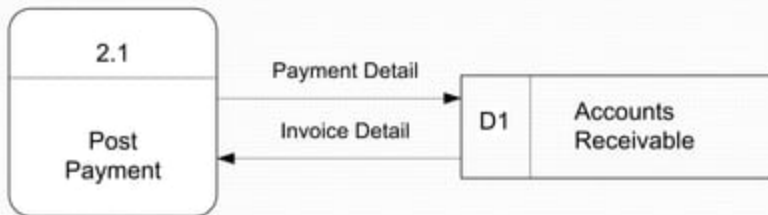
Data Flow



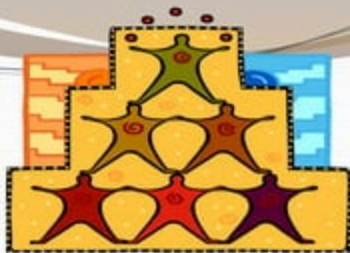
- Data in motion
- Marks movement of data through the system
 - a pipeline to carry data.
- Connects the processes, external entities and data stores.

Data Flow

- Generally unidirectional, If same data flows in both directions, double-headed arrow can be used.
- Can represent flow between process and data store by two separate arrows



Decomposition Of DFD

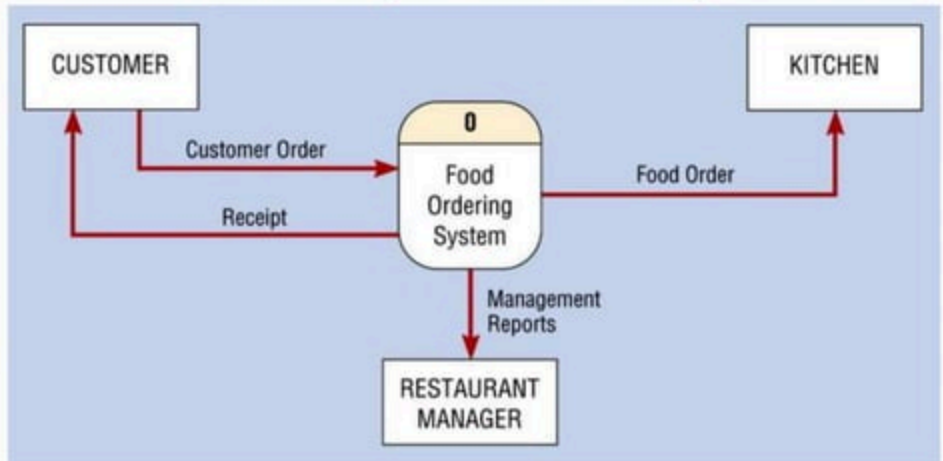


Levels	Description	Explanation
Level 0	Context diagram	Contains only one process
Level 1	Overview diagram	Utilizes all four elements
Level 2	Detailed diagram	A breakdown of a level 2 process

There is **no rule** as to **how many levels** of DFD that can be used.

Rules for Level 0 Diagram :

- 1 process represents the entire system.
- Data arrows show input and output.
- Data Stores NOT shown. They are within the system.



Layers of DFD Abstraction for Course Registration System

Context Diagram

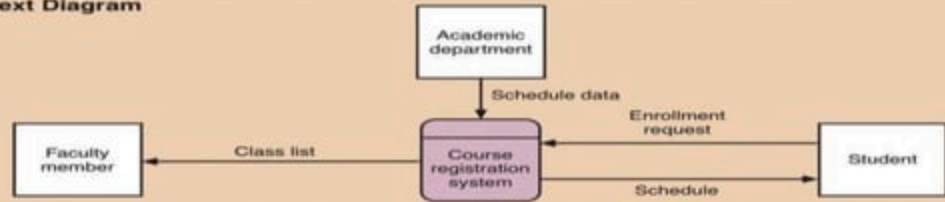


Diagram 0

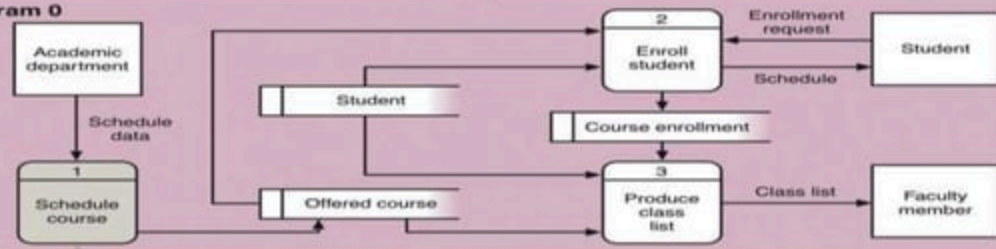
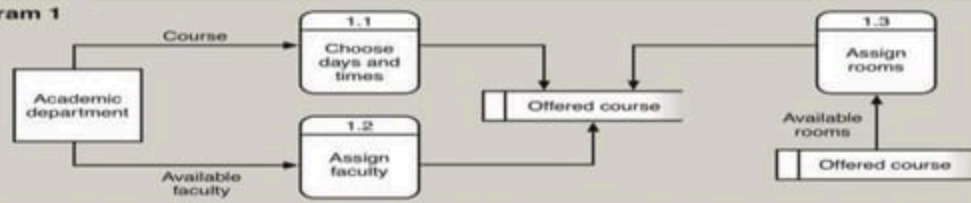


Diagram 1



A Context Diagram (Level 0)

- The major information flows between the entities and the system.
- A Context Diagram addresses only *one process*.

Rules for Level 1 Diagram :

- Level 1 DFD, must balance with the context diagram it describes.
- Input going into a process are different from outputs leaving the process.
- Data stores are first shown at this level.

Rules for Level 2 Diagram :

- Level 2 DFD **must balance with the Level 1** it describes.
- **Input going** into a process are **different** from **outputs leaving** the process.
- **Continue to show data stores.**

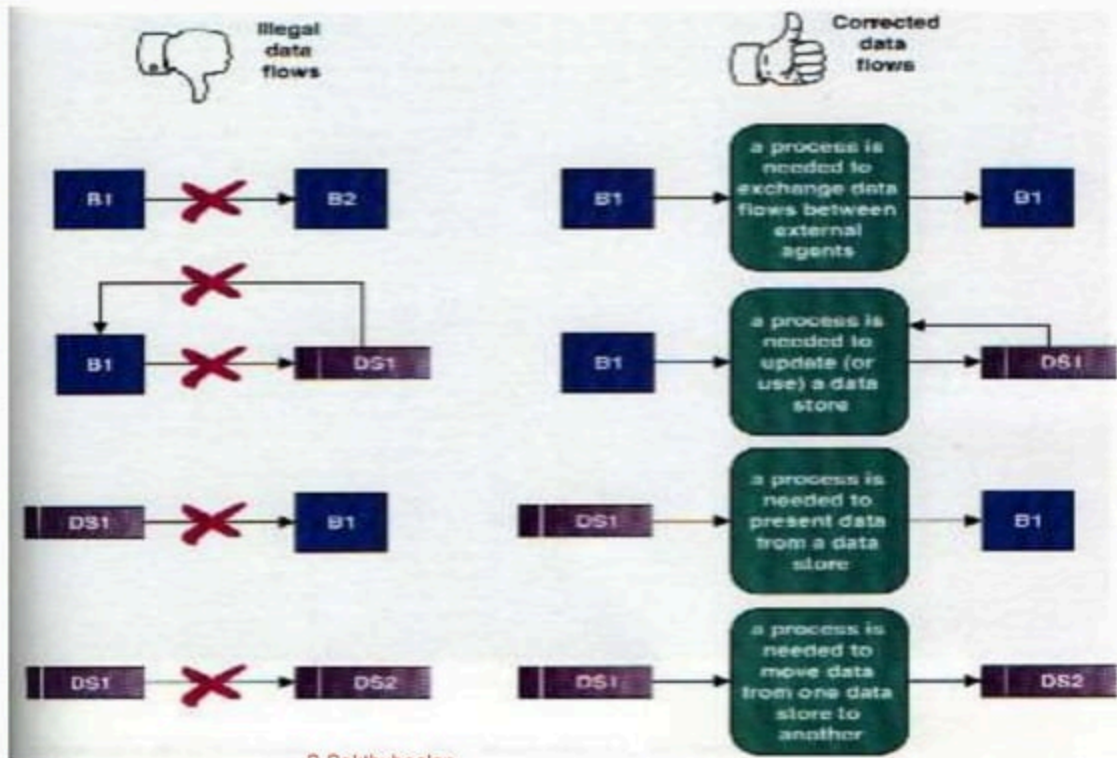
Numbering

- On level 1 processes are numbered 1,2,3...
- On level 2 processes are numbered x.1, x.2, x.3... where x is the number of the parent level 1 process.
- Number is used to uniquely identify process **not** to represent any order of processing
- Data store numbers usually D1, D2, D3...

Rules of Data Flow

- Data can flow from
 - ✓ External entity to process
 - ✓ Process to external entity
 - ✓ Process to store and back
 - ✓ Process to process
- Data cannot flow from
 - External entity to external entity
 - External entity to store
 - Store to external entity
 - Store to store

Common errors in DFD



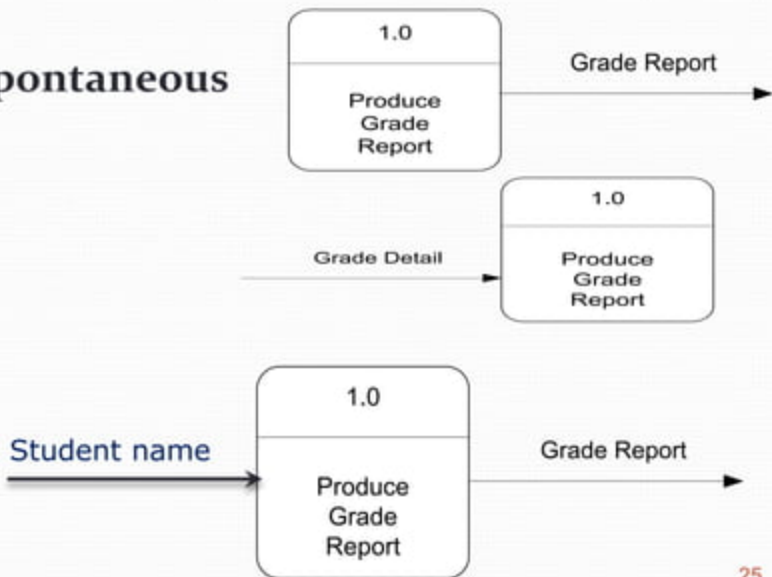
Three INCORRECT Data Flow

- Miracle generation)

(Spontaneous

- Black Hole

- Gray Hole



Good Style in Drawing DFD

- Use meaningful names for data flows, processes and data stores.
- Use top down development starting from context diagram and successively levelling DFD
- Only previously stored data can be read
- A process can only transfer input to output. It cannot create new data
- Data stores cannot create new data

Creating DFDs

- Create a preliminary Context Diagram.
- Identify Use Cases, i.e. the ways in which users most commonly use the system.
- Create DFD fragments for each use case.
- Create a Level 0 diagram from fragments.
- Decompose to Level 1,2,...
- Validate DFDs with users.



Creating the Context Diagram

- Draw one process representing the entire system (process 0)
- Find all inputs and outputs that come from or go to external entities; draw as data flows.
- Draw in external entities as the source or destination of the data flows.



Creating Level 0 Diagram

- Combine the set of DFD fragments into one diagram.
- Generally move from top to bottom, left to right.
- Minimize crossed lines.



Creating Level 1 Diagram

- Each use case is turned into its own DFD.
- Take the steps listed on the use case and depict each as a process on the level 1 DFD.
- Inputs and outputs listed on use case become data flows on DFD.
- Include sources and destinations of data flows to processes and stores within the DFD.
- May also include external entities for clarity.

When to stop decomposing DFDs?

Ideally, a DFD has at least three levels.

When the system becomes primitive i.e. lowest level is reached and further decomposition is useless.



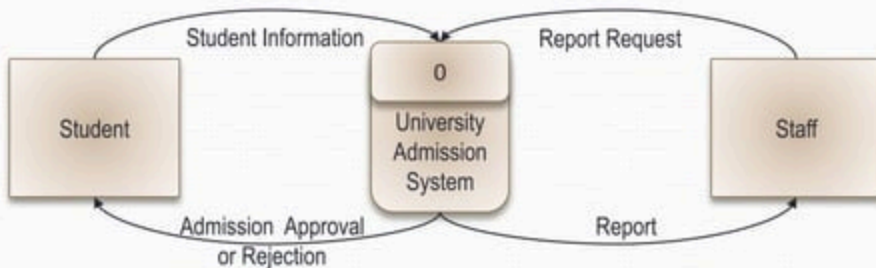
Validating DFD

- Check for *syntax errors* to assure correct DFD structure.
- Check for *semantics errors* to assure accuracy of DFD relative to actual/desired system.

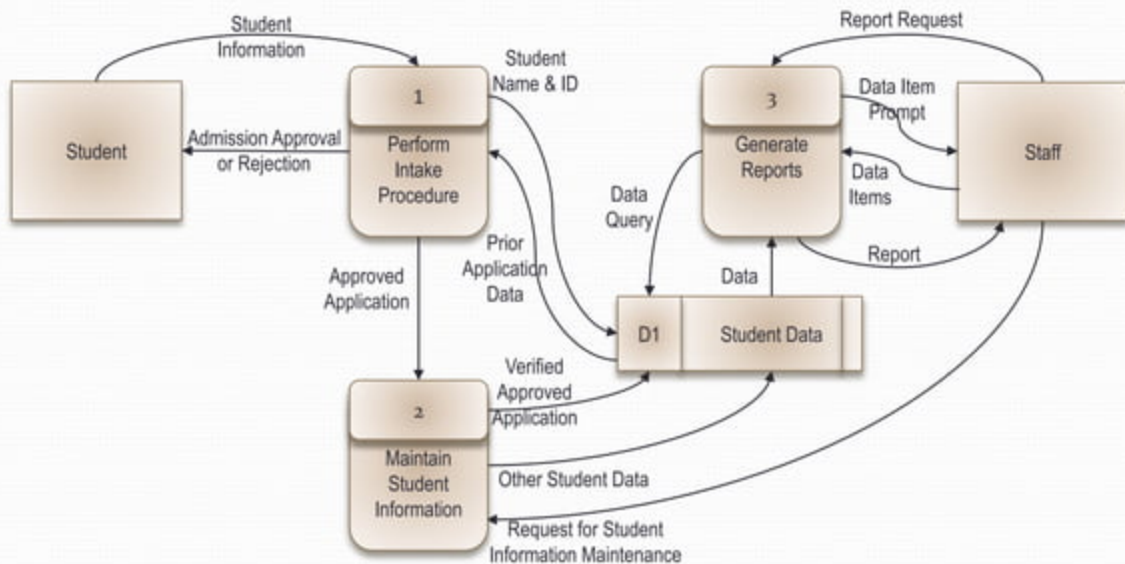


DFD for University Admission System

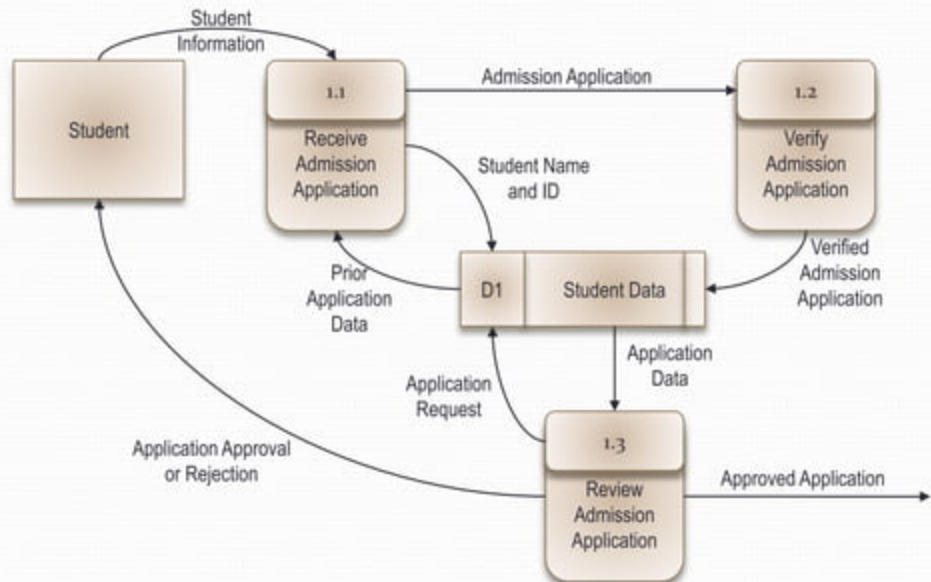
Context Diagram



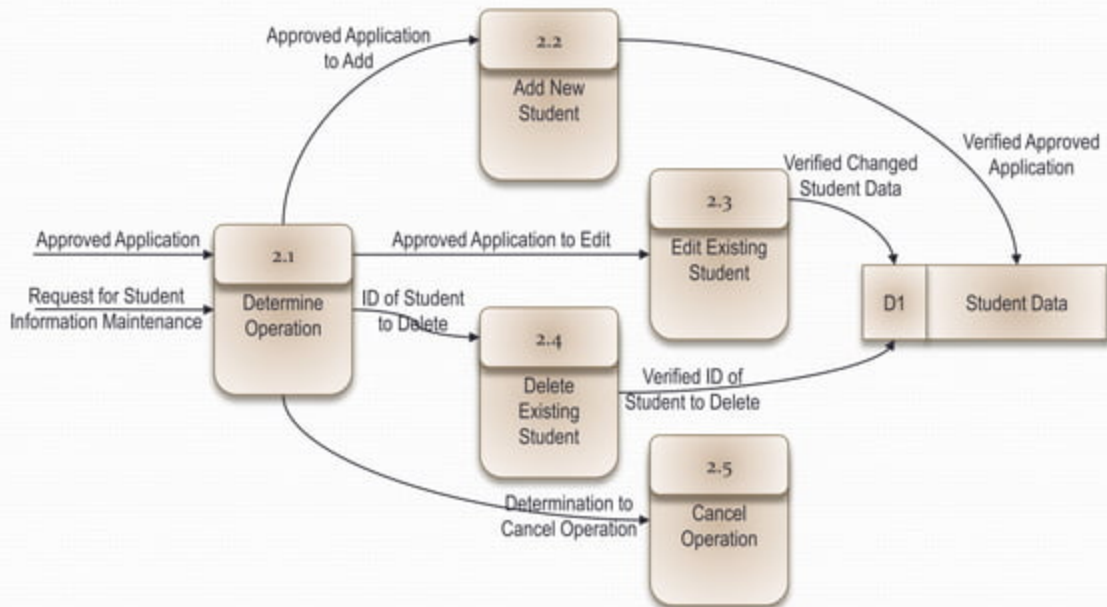
Level 0



Level 1 Process 1, Perform Intake Procedure

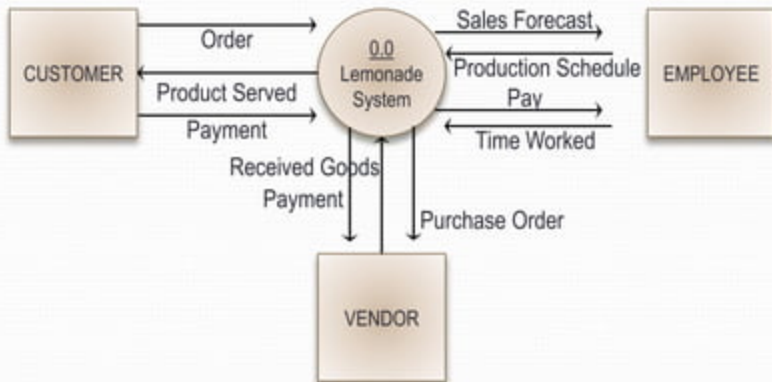


Level 1 Process 2, Maintain Student Information

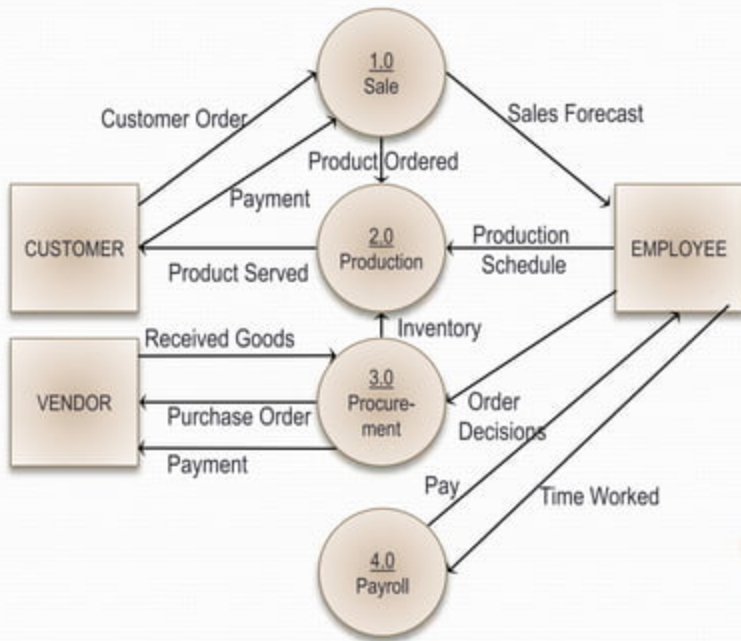


DFD for Lemonade Stand

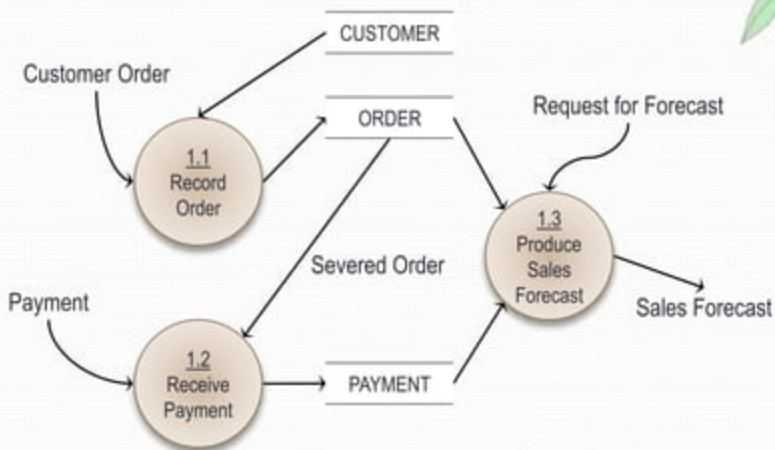
Context Diagram



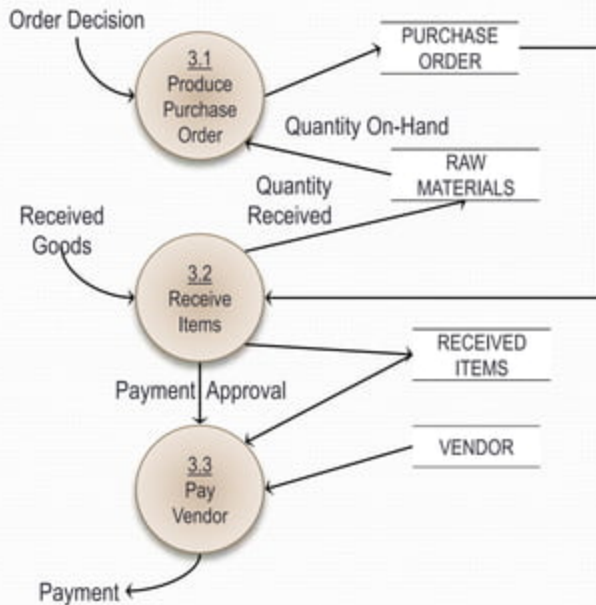
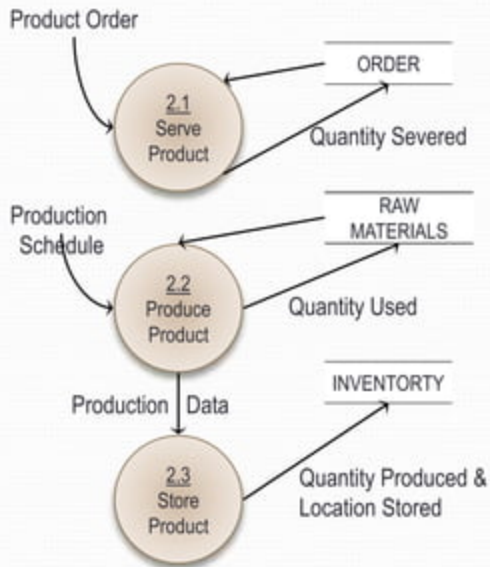
Level 0



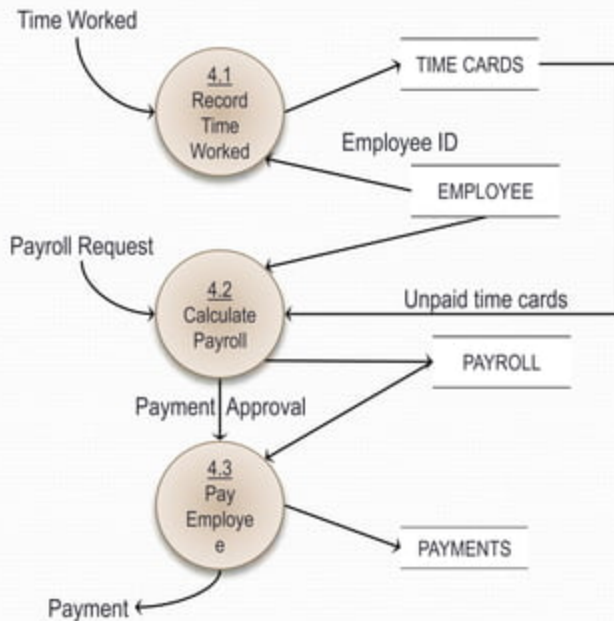
Level 1, Process 1



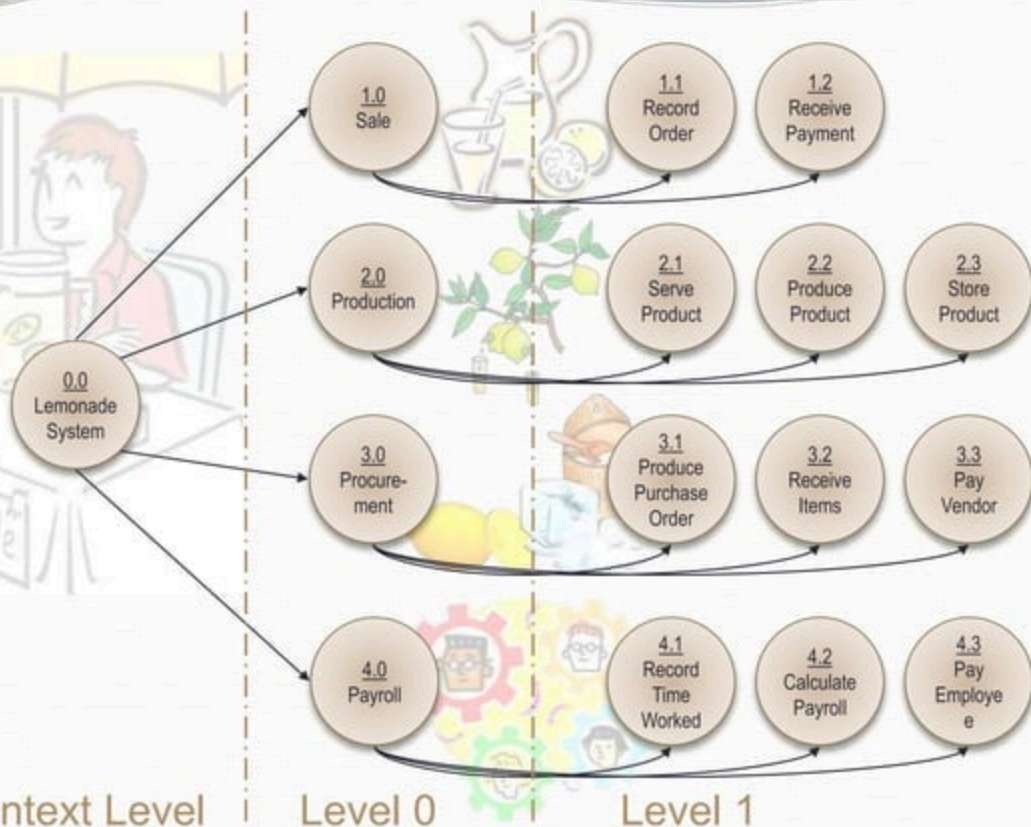
Level 1, Process 2 and Process 3



Level 1, Process 4



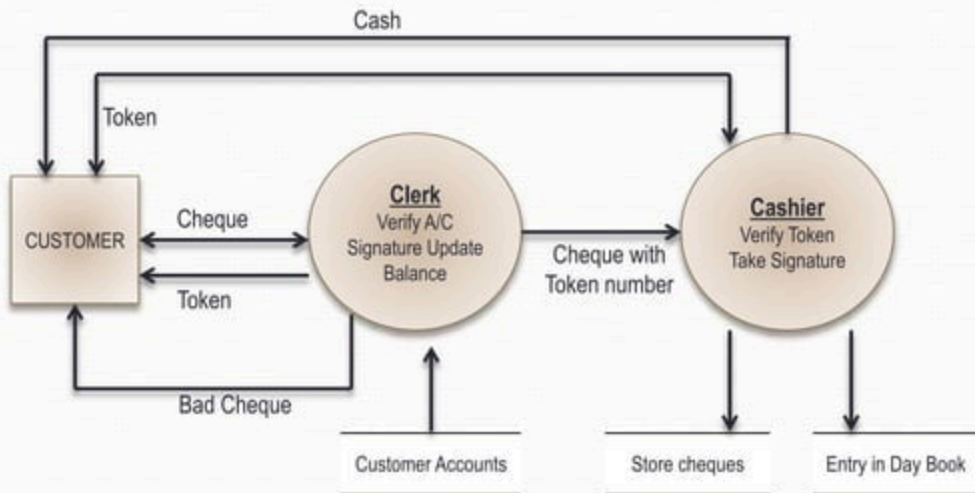
Process Decomposition



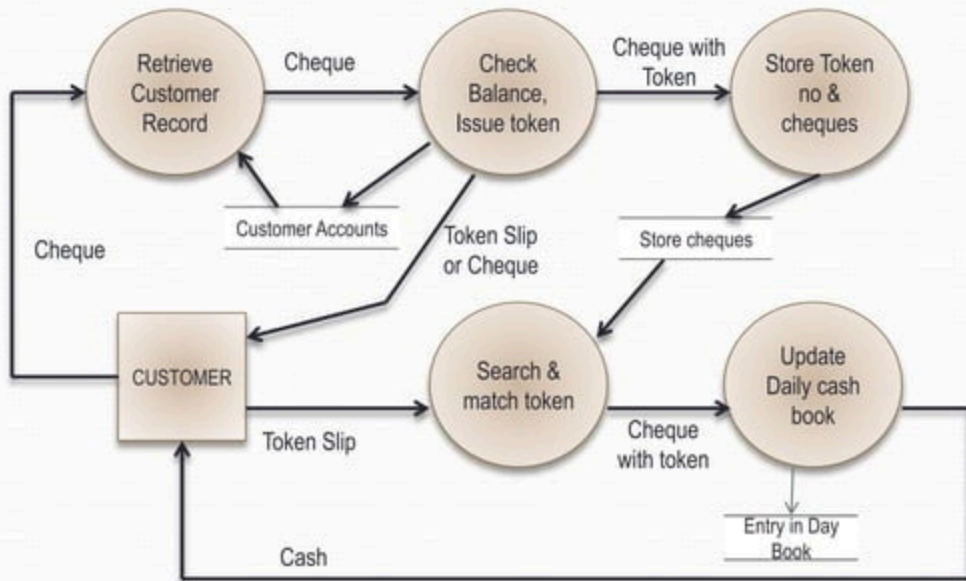
Logical and Physical DFD

- DFDs considered so far are called logical DFDs
- A physical DFD is similar to a document flow diagram
- It specifies who does the operations specified by the logical DFD
- Physical DFD may depict physical movements of the goods
- Physical DFDs can be drawn during fact gathering phase of a life cycle

Physical DFD for Cheque Encashment



Logical DFD for Cheque Encashment



Questions

???



- In a DFD external entities are represented by a
 - a. Rectangle
 - b. Ellipse
 - c. Diamond shaped box
 - d. Circle
- External Entities may be a
 - a. Source of input data only
 - b. Source of input data or destination of results
 - c. Destination of results only
 - d. Repository of data
- A data store in a DFD represents
 - a. A sequential file
 - b. A disk store
 - c. A repository of data
 - d. A random access memory

- By an external entity we mean a
 - a. Unit outside the system being designed which can be controlled by an analyst
 - b. Unit outside the system whose behaviour is independent of the system being designed
 - c. A unit external to the system being designed
 - d. A unit which is not part of DFD
- A data flow can
 - a. Only enter a data store
 - b. Only leave a data store
 - c. Enter or leave a data store
 - d. Either enter or leave a data store but not both
- A circle in a DFD represents
 - a. A data store
 - b. A an external entity
 - c. A process
 - d. An input unit

**Thanks for
your
Cooperation**

