Chapter 6. Data Modeling
Data Flow Diagram

Prepared by:

Dr. Muhammad Iqbal Hossain

Assistant Professor,

Dept. of CSE, BRAC University

Models: Logical and Physical

Model – a pictorial representation of reality.

Just as a picture is worth a thousand words, most models are pictorial representations of reality.

Logical model – a nontechnical pictorial representation that depicts what a system is or does. Synonyms or *essential model, conceptual model*, and *business model*.

Physical model – a technical pictorial representation that depicts what a system is or does and how the system is implemented. Synonyms are *implementation model* and *technical model*.

What is a Data Flow Diagram?

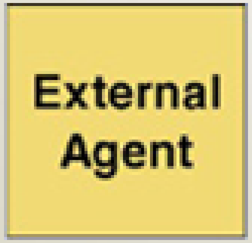
- A data flow diagram (DFD) is a graphical representation of the movement of data between external entities, processes and data stores within a system.

- Simply put, DFD’s show how data moves through an information system.

External Agents

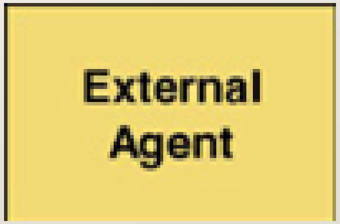
External agent – an outside person, unit, system, or organization that interacts with a system. Also called an *external entity*.

- External agents define the “boundary” or scope of a system being modeled.
- As scope changes, external agents can become processes, and vice versa.



External Agent

Gane and Sarson shape



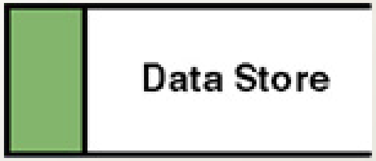
External Agent

DeMarco/Yourdon shape

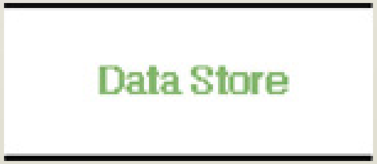
Data Stores

Data store – stored data intended for later use. Synonyms are *file* and *database*.

- Frequently implemented as a file or database.
- A data store is "data at rest" compared to a data flow that is "data in motion."
- Almost always one of the following:
  - Persons, Places, Objects etc
- Data stores depicted on a DFD store all instances of data entities (depicted on an ERD)



Gane and Sarson shape

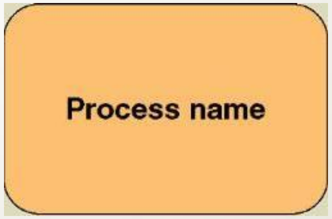


DeMarco/Yourdon shape

Process

Process – work performed by a system in response to incoming data flows or conditions. A synonym is *transform*.

- All information systems include processes - usually many of them
- Processes respond to business events and conditions and transform data into useful information
- Modeling processes helps us to understand the interactions with the system's environment, other systems, and other processes.
- Named with a strong action verb followed by object clause describing what the work is performed on/for .

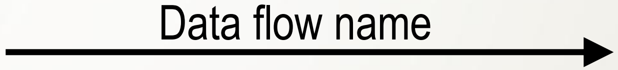


Gane and Sarson shape

Data Flows & Control Flows

Data flow – data that is input to or output from a process.

- A data flow is data in motion
- A data flow may also be used to represent the creation, reading, deletion, or updating of data in a file or database (called a data store).

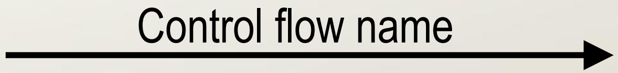


Data flow name

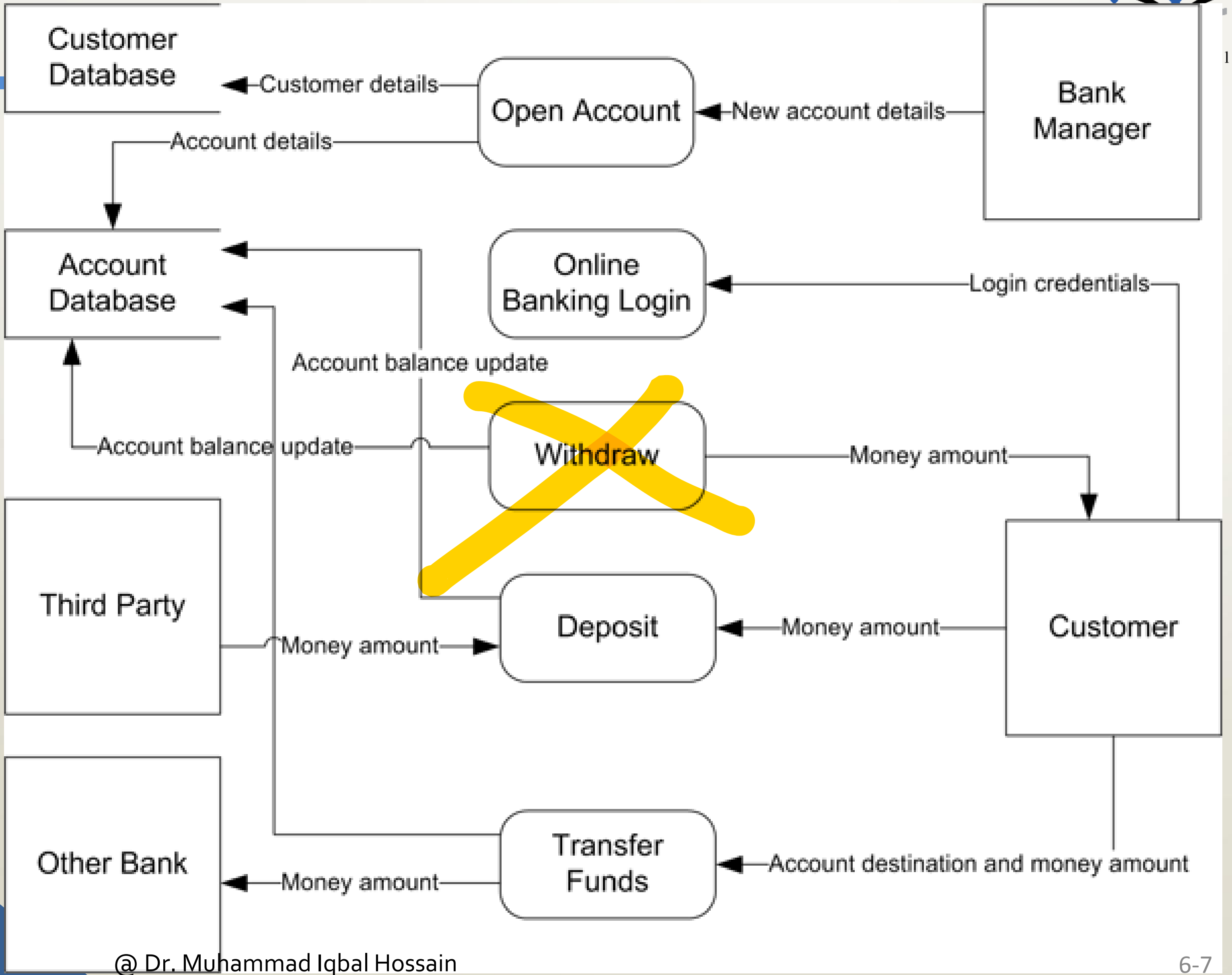
Composite data flow – a data flow that consists of other data flows.

Control flow – a condition or nondata event that triggers a process.

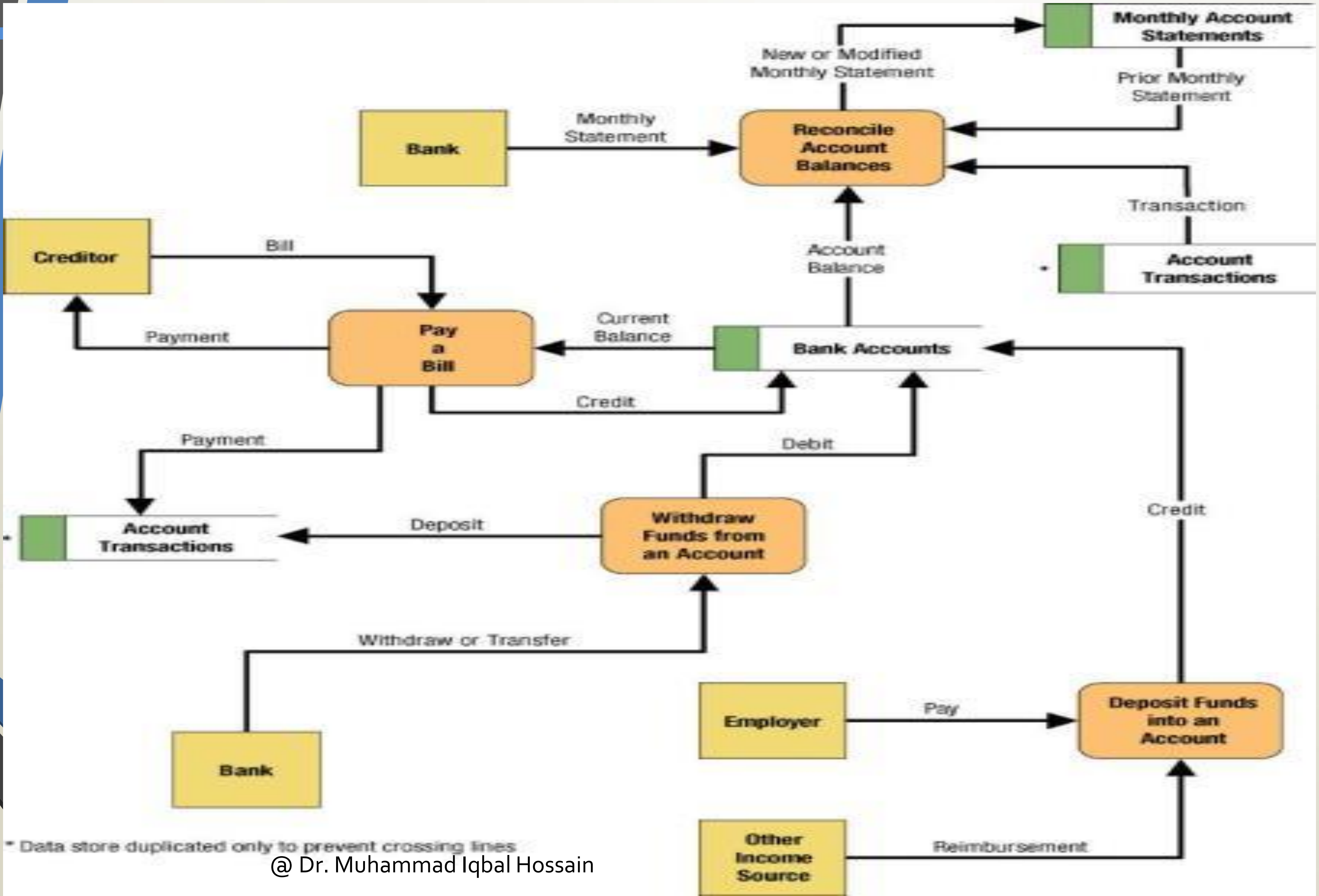
- Used sparingly on DFDs.



Control flow name


Customer Database
Customer details
Open Account
New account details
Bank Manager
Account details
Account Database
Online Banking Login
Login credentials
Account balance update
Account balance update
~~Withdraw~~
Money amount
Third Party
Money amount
Deposit
Money amount
Customer
Other Bank
Money amount
Transfer Funds
Account destination and money amount

Simple Data Flow Diagram

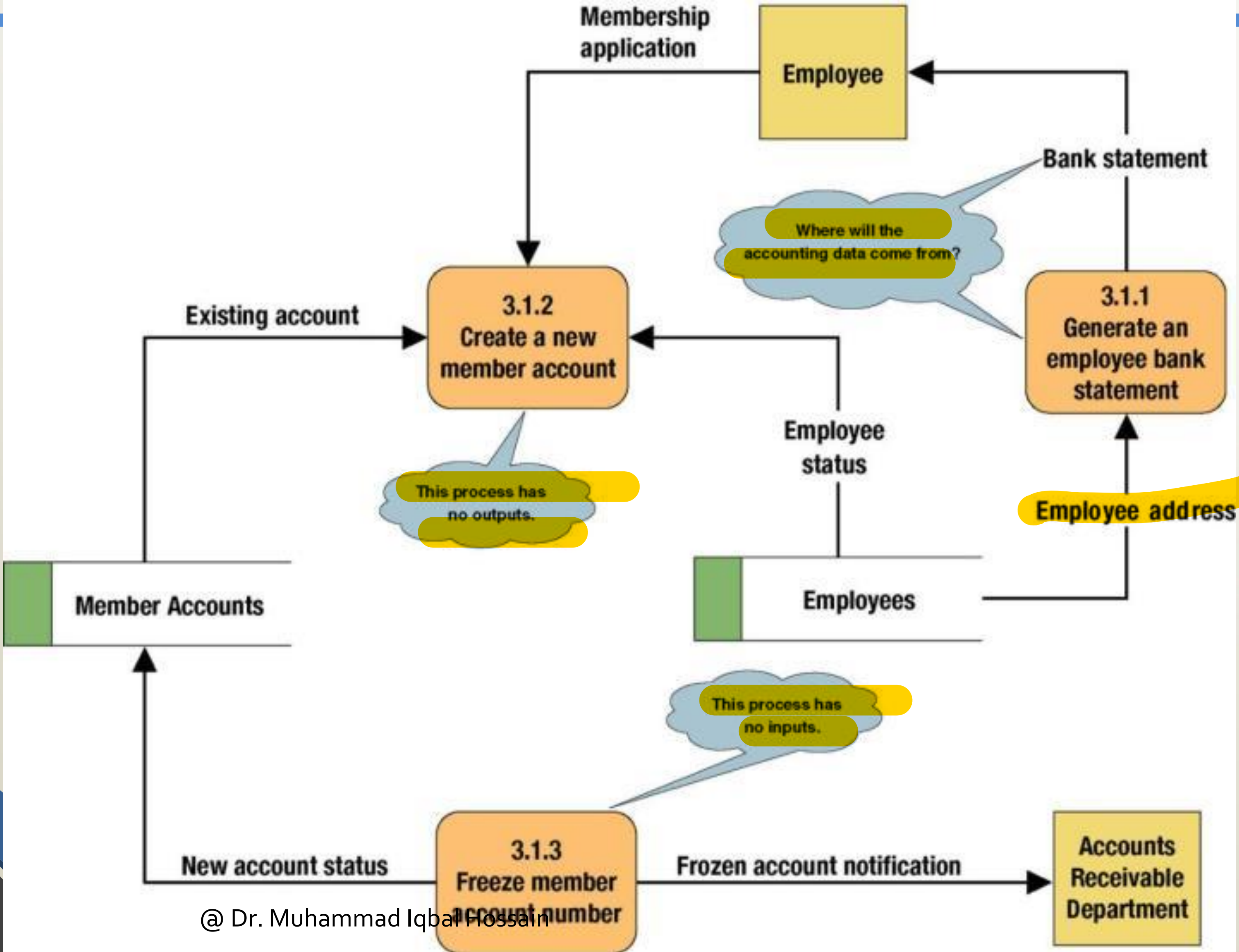


* Data store duplicated only to prevent crossing lines

@ Dr. Muhammad Iqbal Hossain

Common Process Errors on DFDs

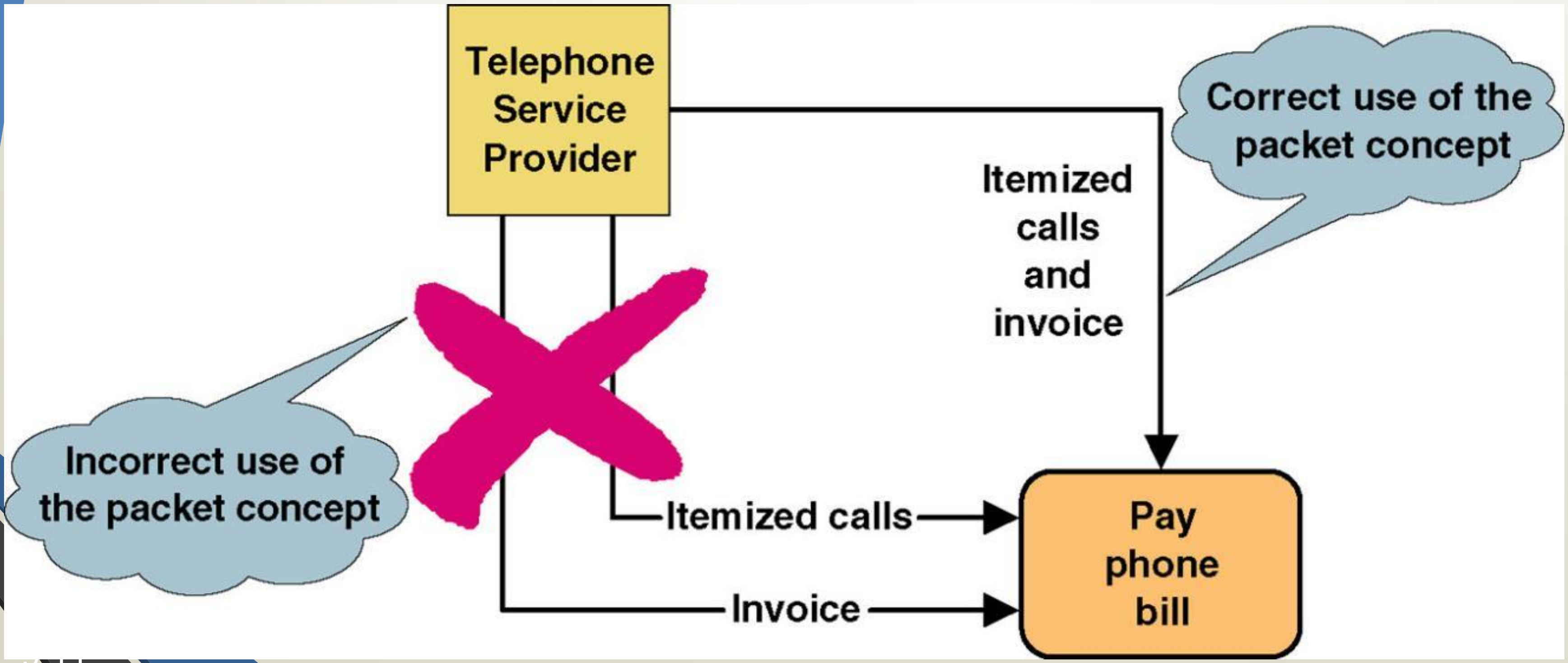
Common Process Errors on DFDs


Membership application
Employee
Bank statement
Where will the accounting data come from?
Existing account
3.1.2 Create a new member account
3.1.1 Generate an employee bank statement
Employee status
This process has no outputs.
Employee address
Member Accounts
Employees
This process has no inputs.
New account status
3.1.3 Freeze member account number
Frozen account notification
Accounts Receivable Department

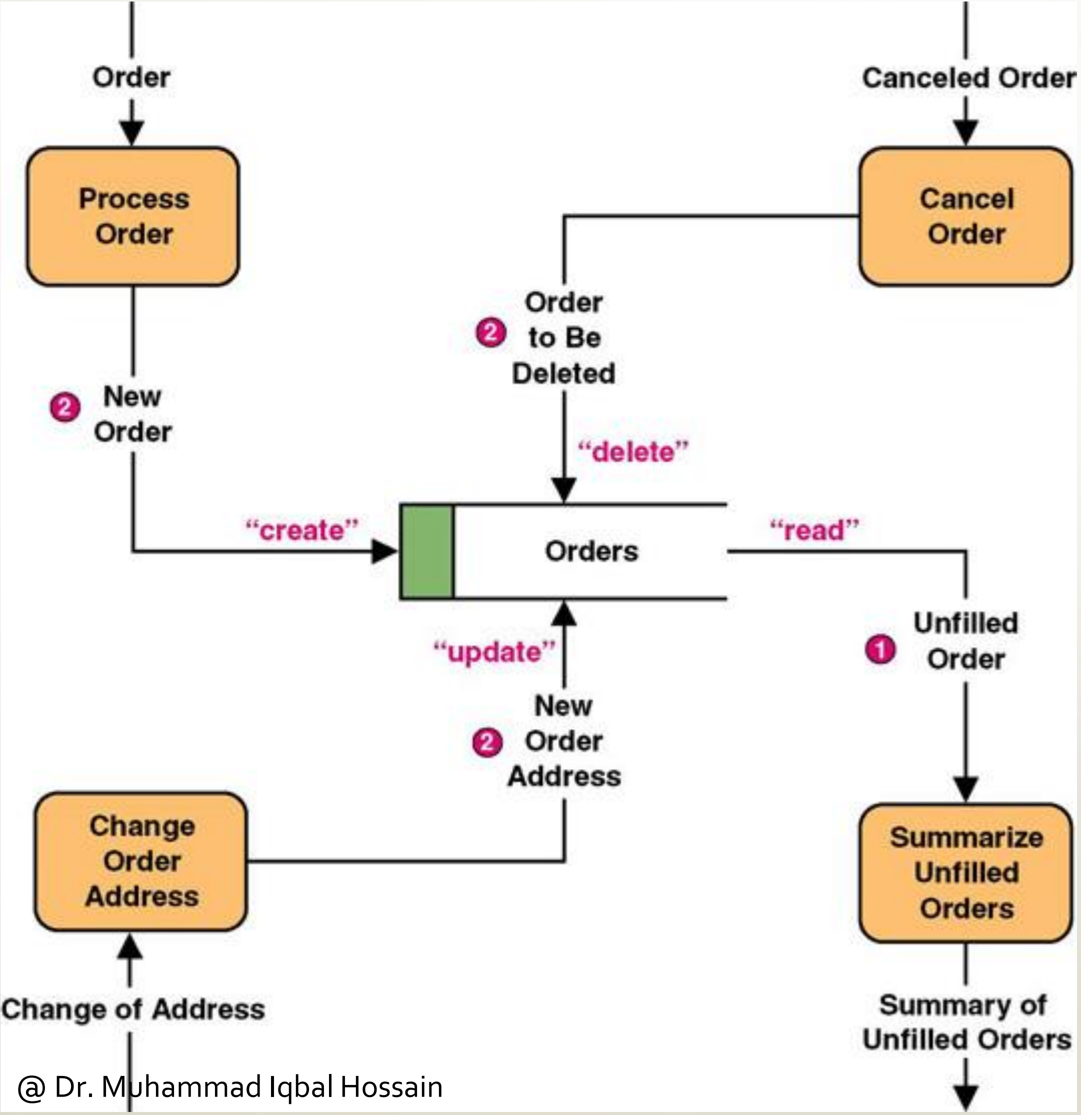
@ Dr. Muhammad Iqbal Hossain

Data Flow Packet Concept

- Data that should travel together should be shown as a single data flow, no matter how many physical documents might be included.


Telephone Service Provider
Itemized calls and invoice
Correct use of the packet concept
Incorrect use of the packet concept
Itemized calls
Invoice
Pay phone bill

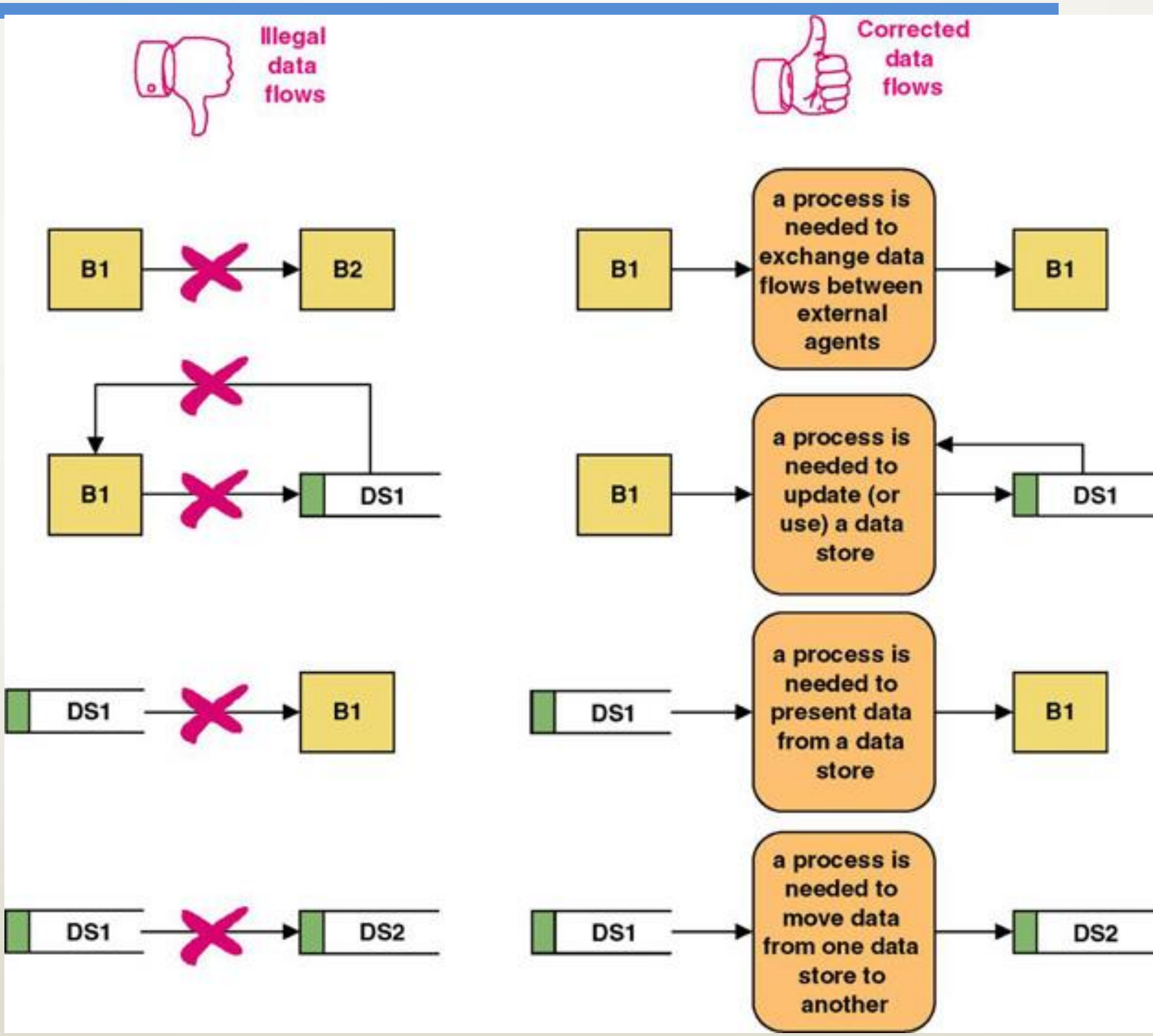
Data Flows to and from Data Stores



@ Dr. Muhammad Iqbal Hossain

Rules for Data Flows

- A data flow should never go unnamed.
- In logical modeling, data flow names should describe the data flow without describing the implementation
- All data flows must begin and/or end at a process.


Illegal data flows
Corrected data flows
B1
B2
B1
a process is needed to exchange data flows between external agents
B1
B1
DS1
B1
a process is needed to update (or use) a data store
DS1
DS1
B1
DS1
a process is needed to present data from a data store
B1
DS1
DS2
DS1
a process is needed to move data from one data store to another
DS2

Diverging and Converging Data Flows

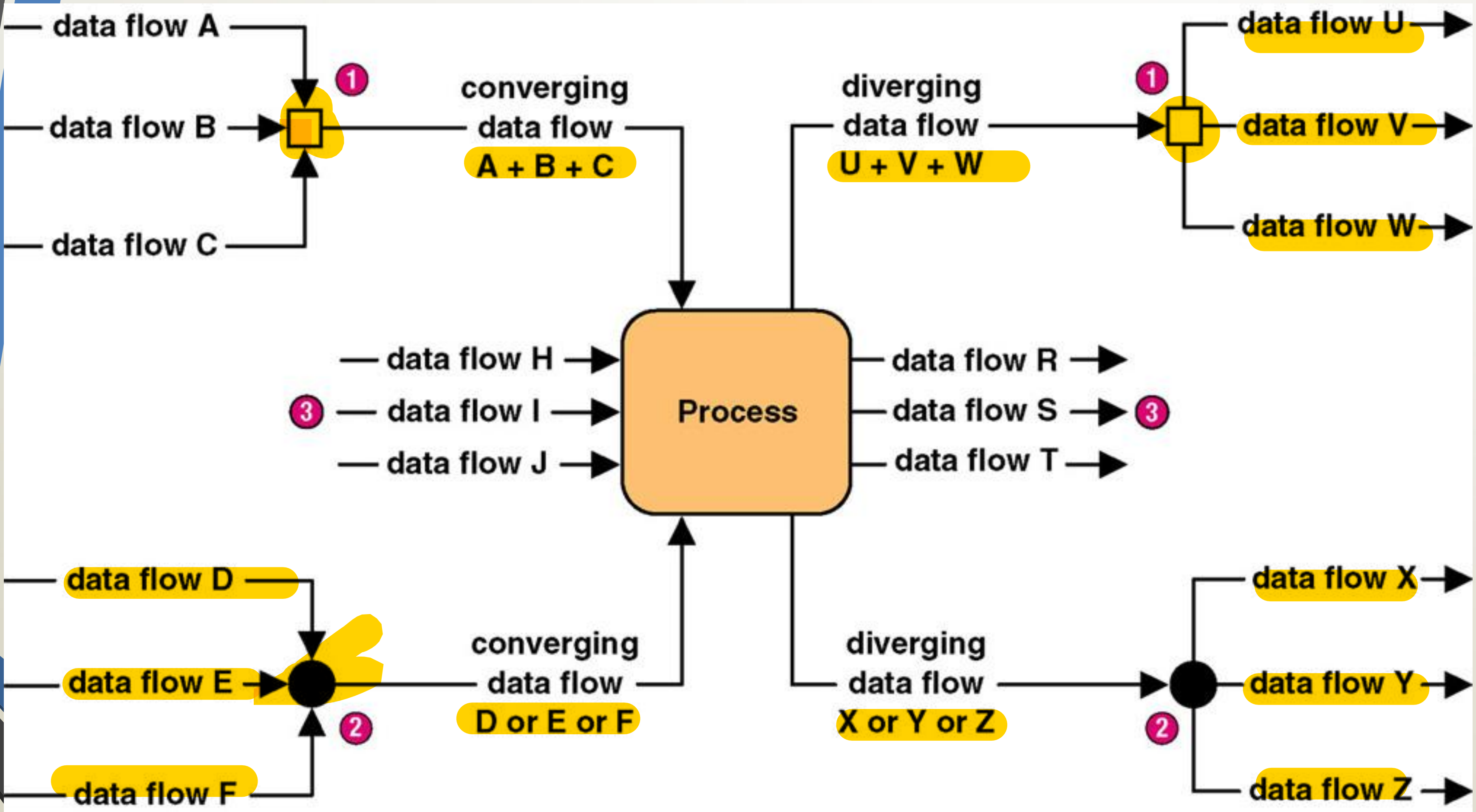
Diverging data flow – a data flow that splits into multiple data flows.

- Indicates data that starts out naturally as one flow, but is routed to different destinations.
- Also useful to indicate multiple copies of the same output going to different destinations.

Converging data flow – the merger of multiple data flows into a single packet.

- Indicates data from multiple sources that can (must) come together as a single packet for subsequent processing.

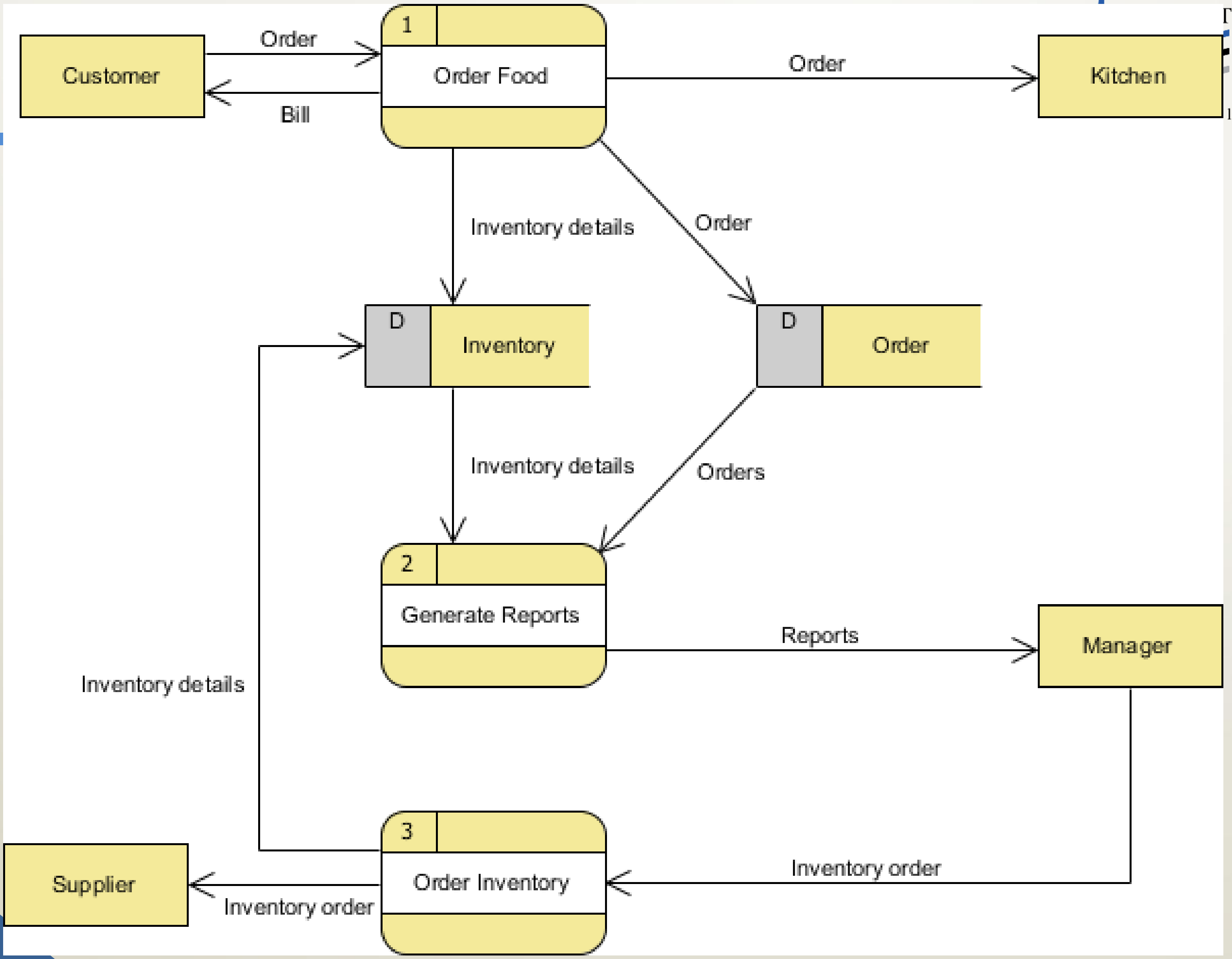
Diverging and Converging Data Flows



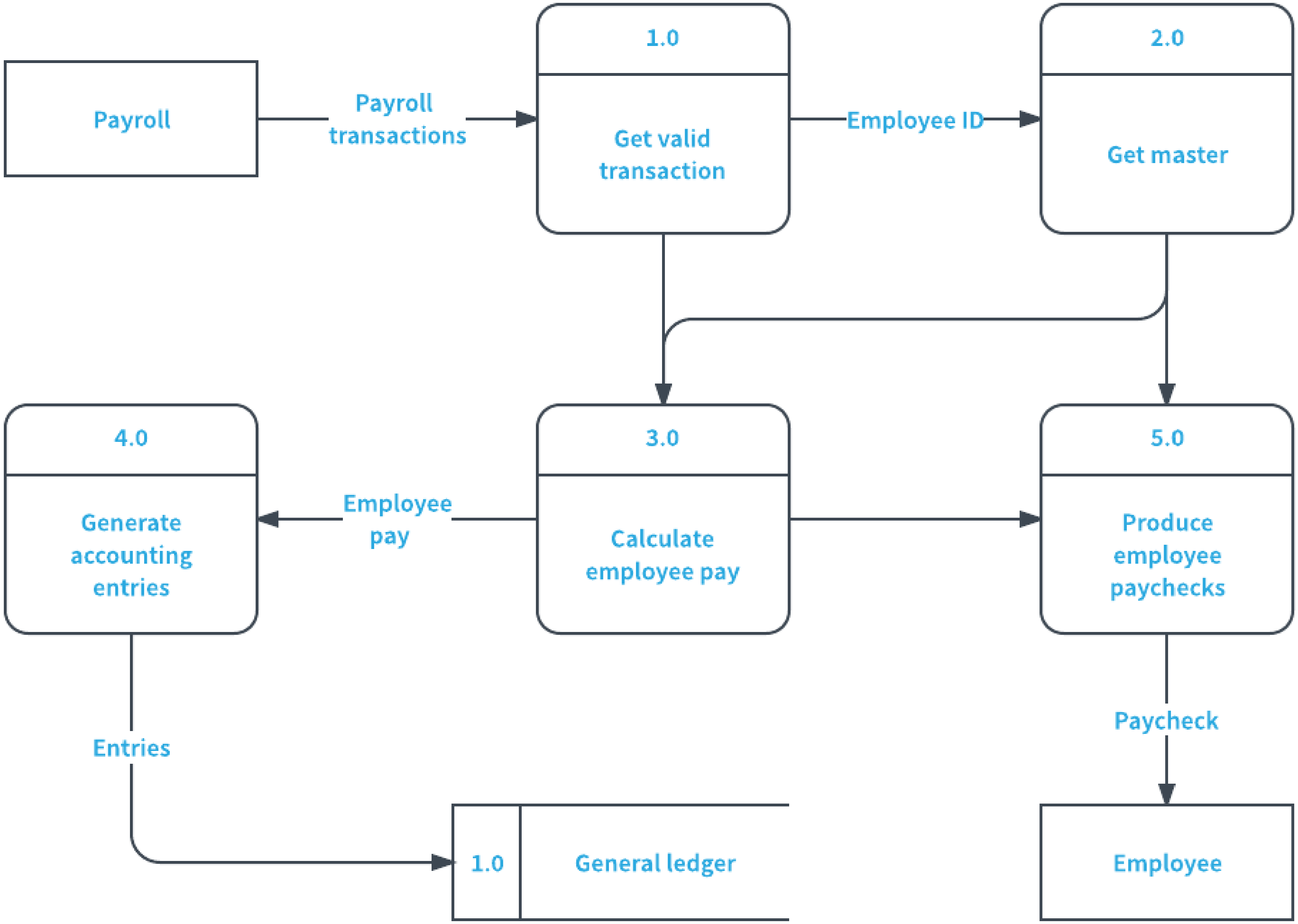
@ Dr. Muhammad Iqbal Hossain

Exercise: Order food

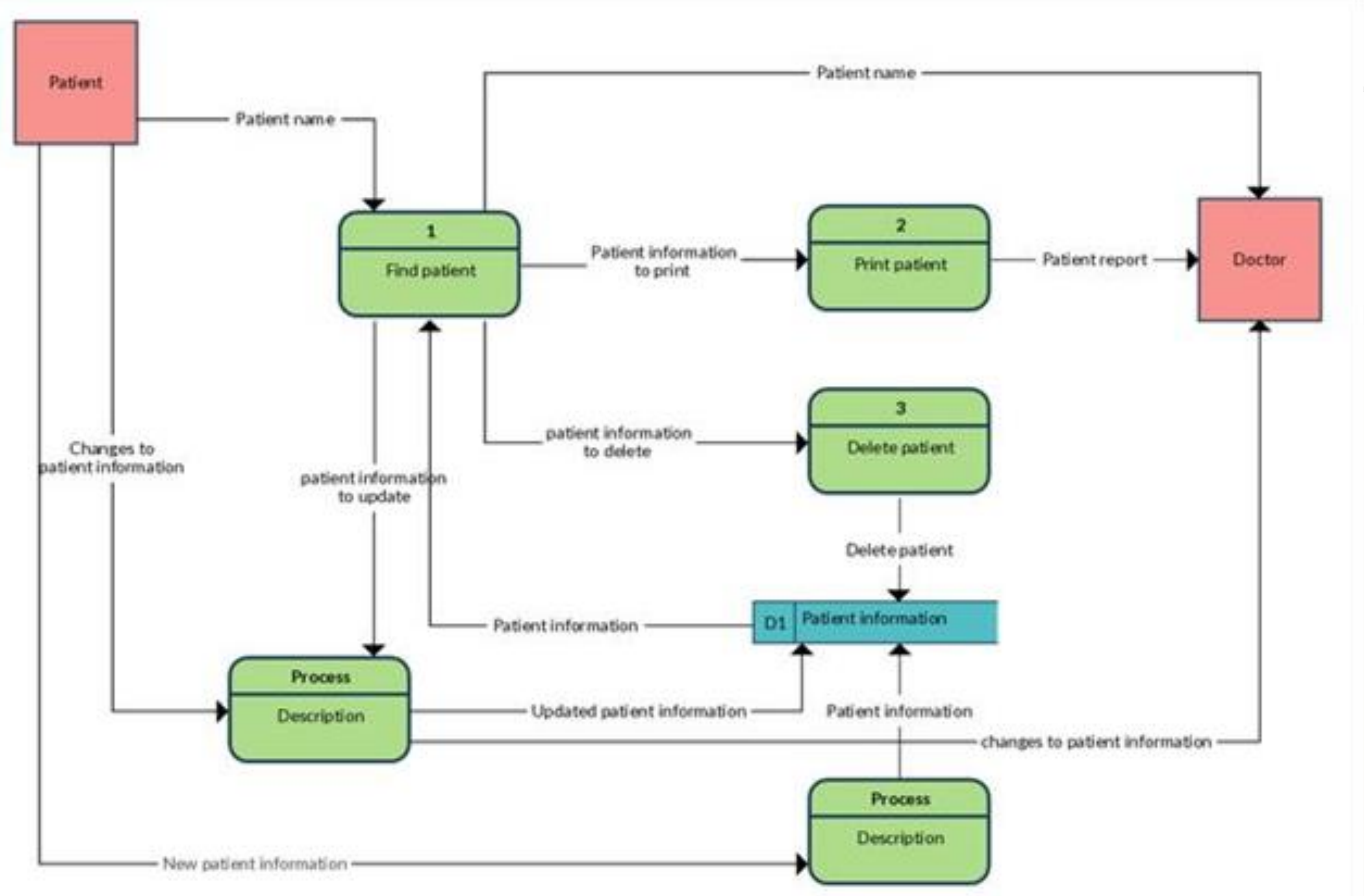
- Draw a Data Flow Diagram (DFD) for "***Food Ordering System"*** considering the following scenarios:
- **Customer** place an *Order*. The Order Food process receives the *Order*, forwards it to the **Kitchen**, store it in the ***Order*** data store, and store the updated *Inventory details* in the ***Inventory*** data store. The process also delivers a *Bill* to the **Customer**. **Manager** can receive *Reports* through the Generate Reports process, which takes *Inventory details* and *Orders* as input from the ***Inventory*** and ***Order*** data store respectively. **Manager** can also initiate the Order Inventory process by providing *Inventory order*. The process forwards the *Inventory order* to the **Supplier** and stores the updated *Inventory details* in the ***Inventory*** data store.

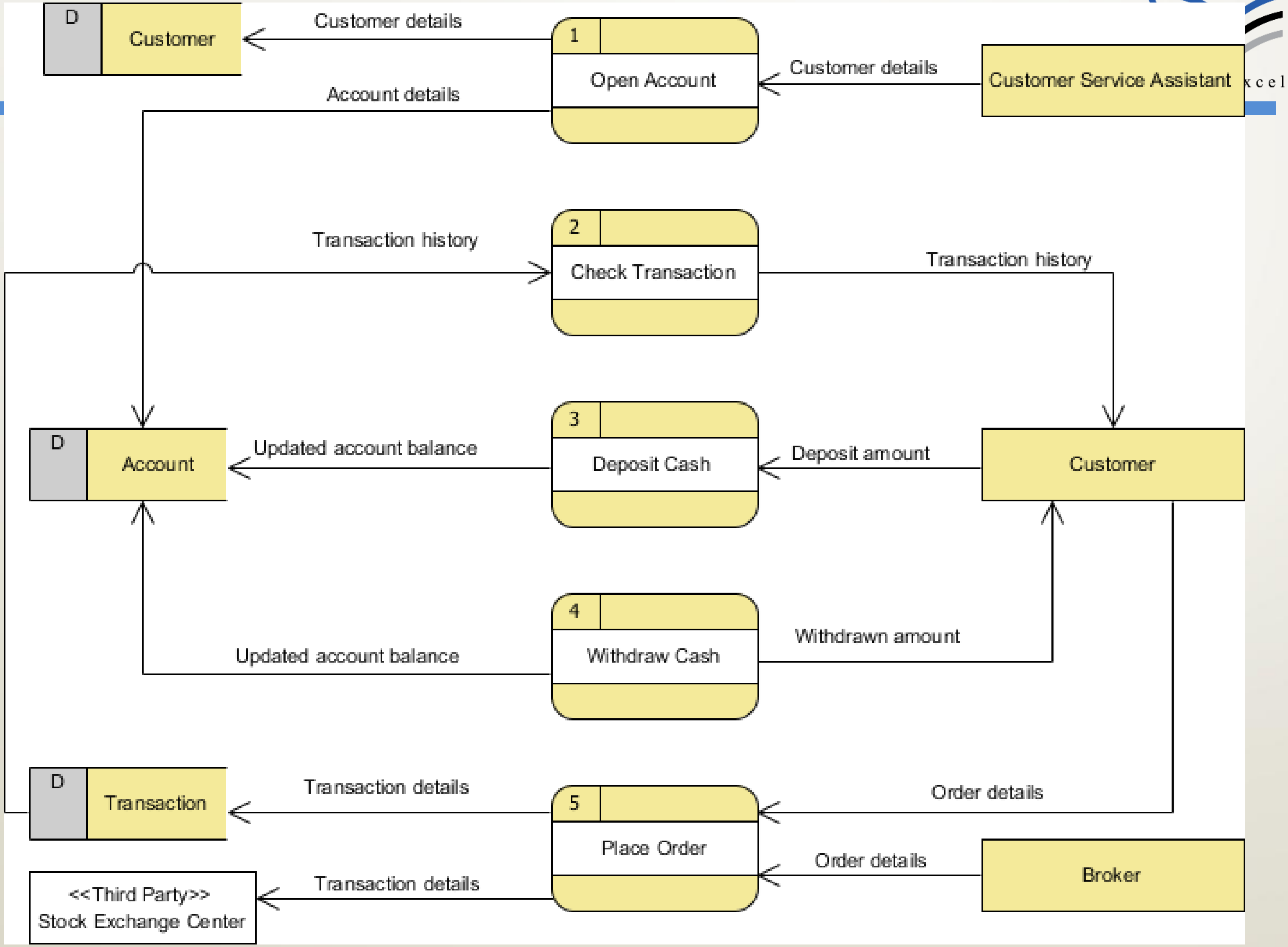

Customer
Order
Bill
1
Order Food
Order
Kitchen
Inventory details
Order
D
Inventory
D
Order
Inventory details
Orders
2
Generate Reports
Reports
Manager
Inventory details
3
Order Inventory
Inventory order
Supplier
Inventory order

Payroll management system



Patient Information system





D Customer

Customer details

1 Open Account

Customer details

Customer Service Assistant

Account details

2 Check Transaction

Transaction history

Transaction history

D Account

Updated account balance

3 Deposit Cash

Deposit amount

Customer

4 Withdraw Cash

Withdrawn amount

Updated account balance

D Transaction

Transaction details

5 Place Order

Order details

Order details

Broker

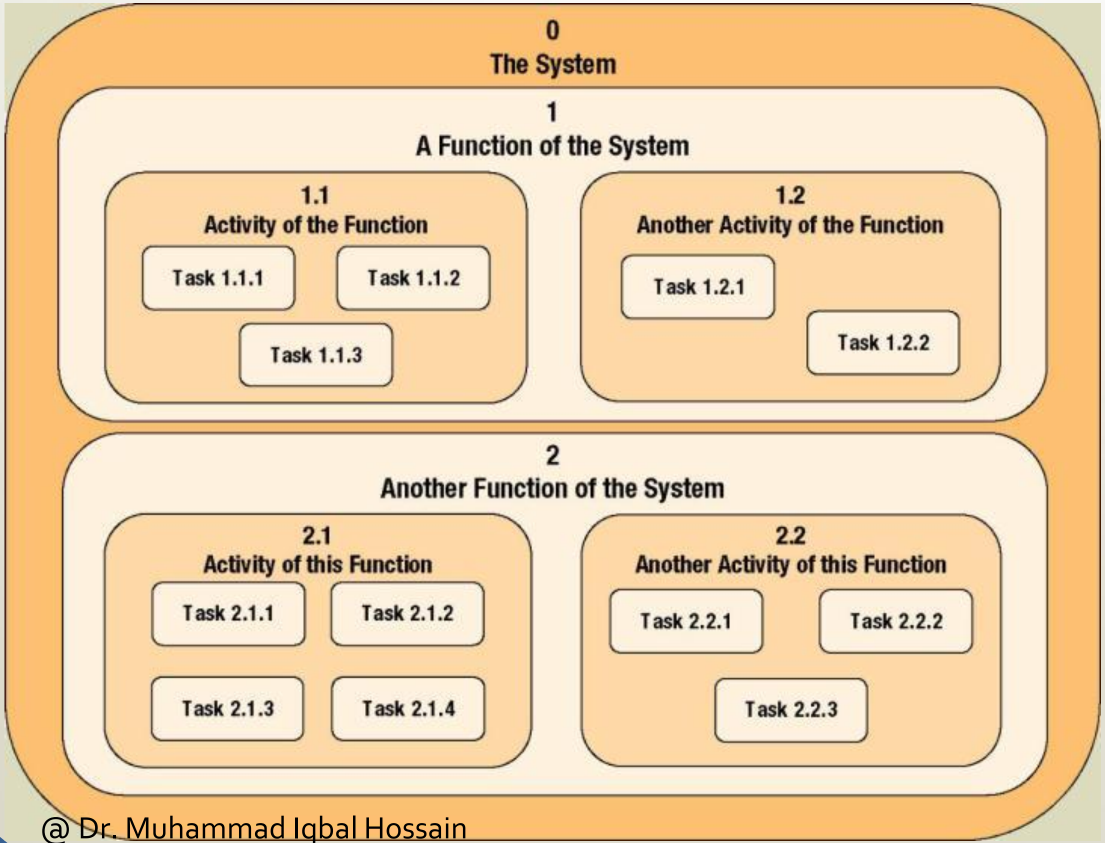
<<Third Party>>
Stock Exchange Center

Transaction details

Process Decomposition

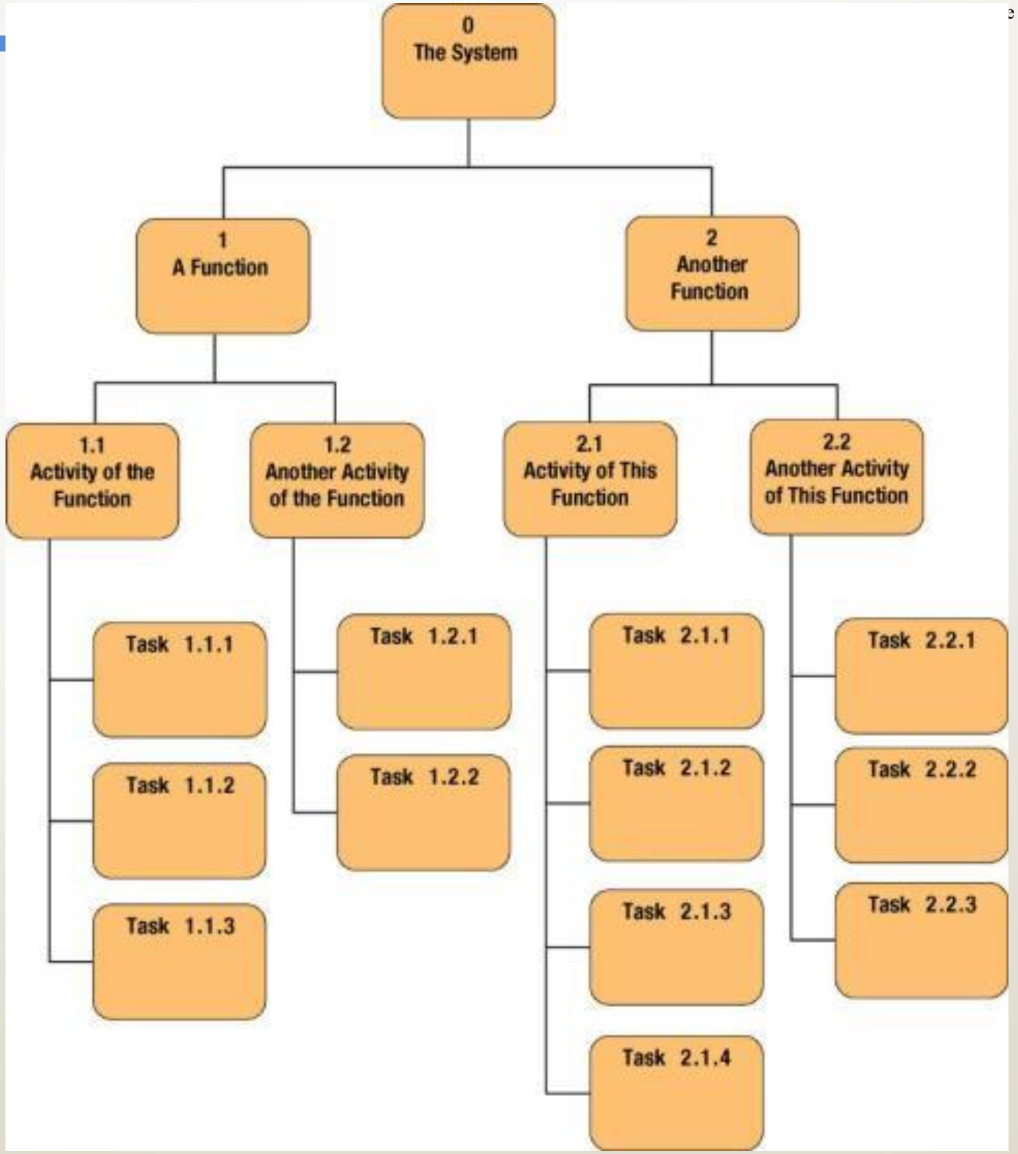
Process Decomposition

Decomposition – the act of breaking a system into sub-components. Each level of abstraction reveals more or less detail.



Decomposition Diagrams

Decomposition diagram – a tool used to depict the decomposition of a system. Also called hierarchy chart.



Level o Diagram

- Shows all the processes that comprise the overall system
- Shows how information moves from and to each process
- Adds data stores

Level 1 Diagrams

- Shows all the processes that comprise a single process on the level 0 diagram
- Shows how information moves from and to each of these processes
- Shows in more detail the content of higher level process
- Level 1 diagrams may not be needed for all level 0 processes

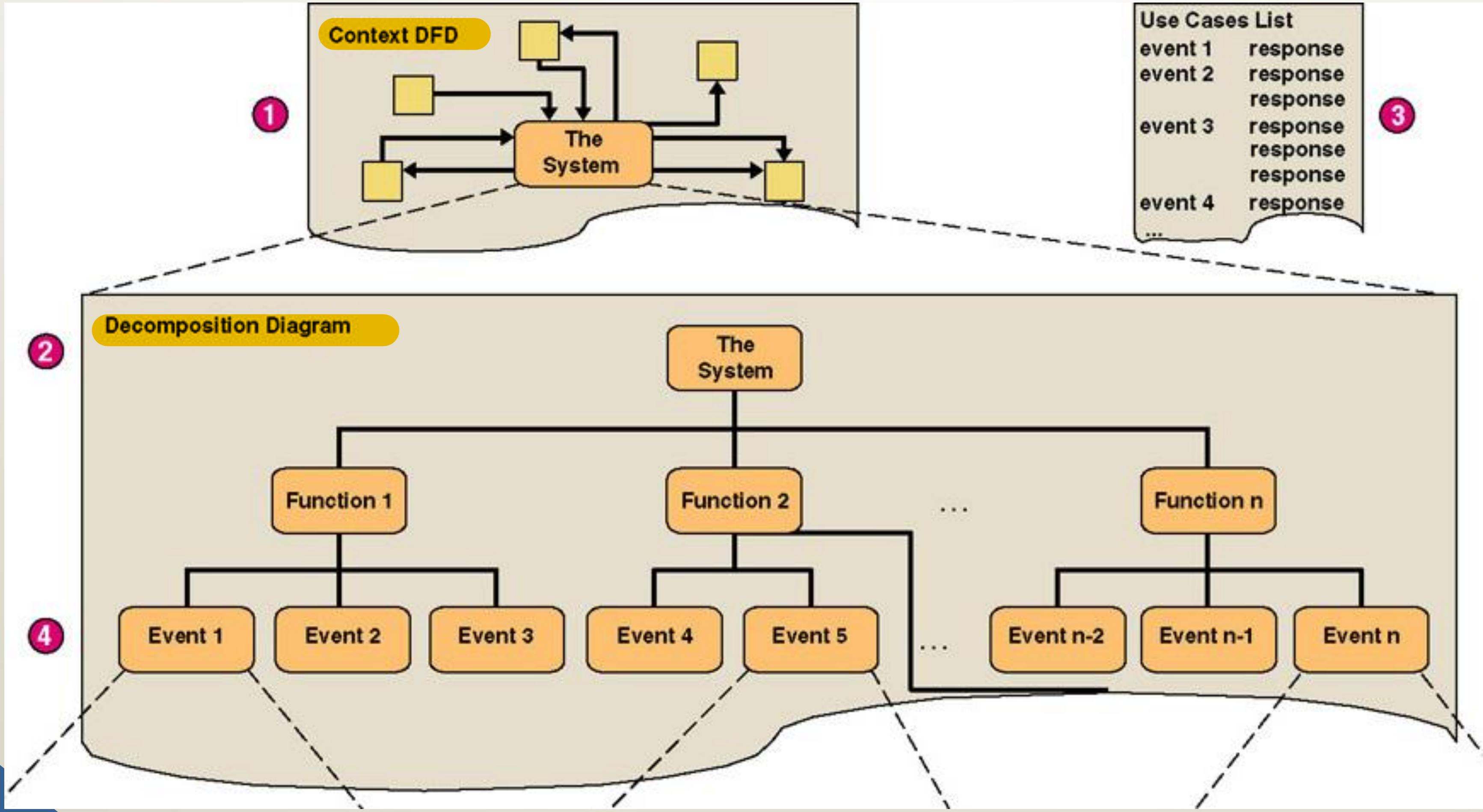
Level 2 Diagrams

- Shows all processes that comprise a single process on the level 1 diagram
- Shows how information moves from and to each of these processes
- Level 2 diagrams may not be needed for all level 1 processes
- Correctly numbering each process helps the user understand where the process fits into the overall system

Modern Structured Analysis
(More Commonly Practiced)

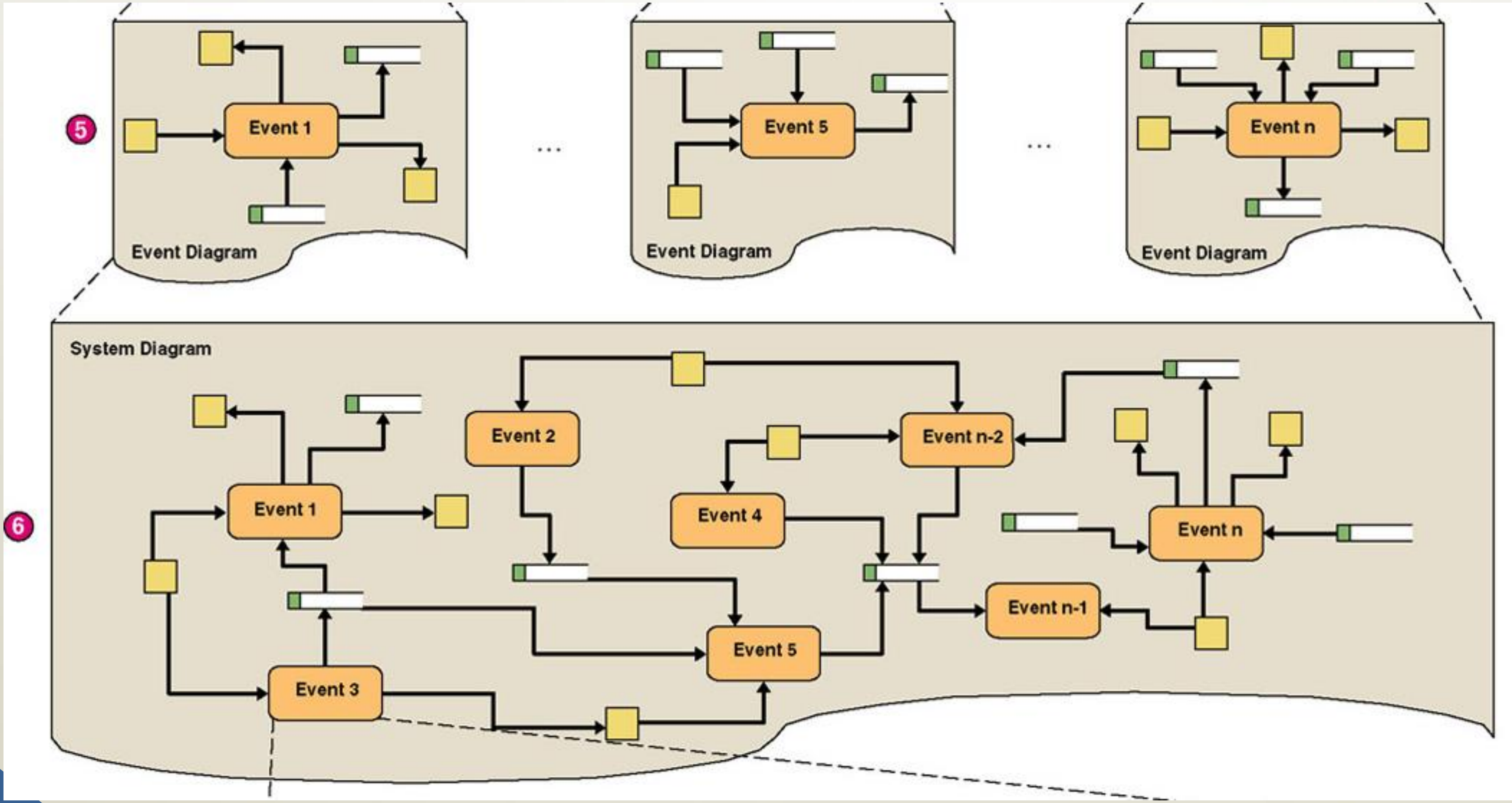
1. Draw context DFD to establish initial project scope.
2. Draw functional decomposition diagram to partition the system into subsystems.
3. Create event-response or use-case list for the system to define events for which the system must have a response.
4. Draw an event DFD (or event handler) for each event.
5. Merge event DFDs into a system diagram (or, for larger systems, subsystem diagrams).
6. Draw detailed, primitive DFDs for the more complex event handlers.
7. Document data flows and processes in data dictionary.

Structured Analysis Diagram Progression (1 of 3)

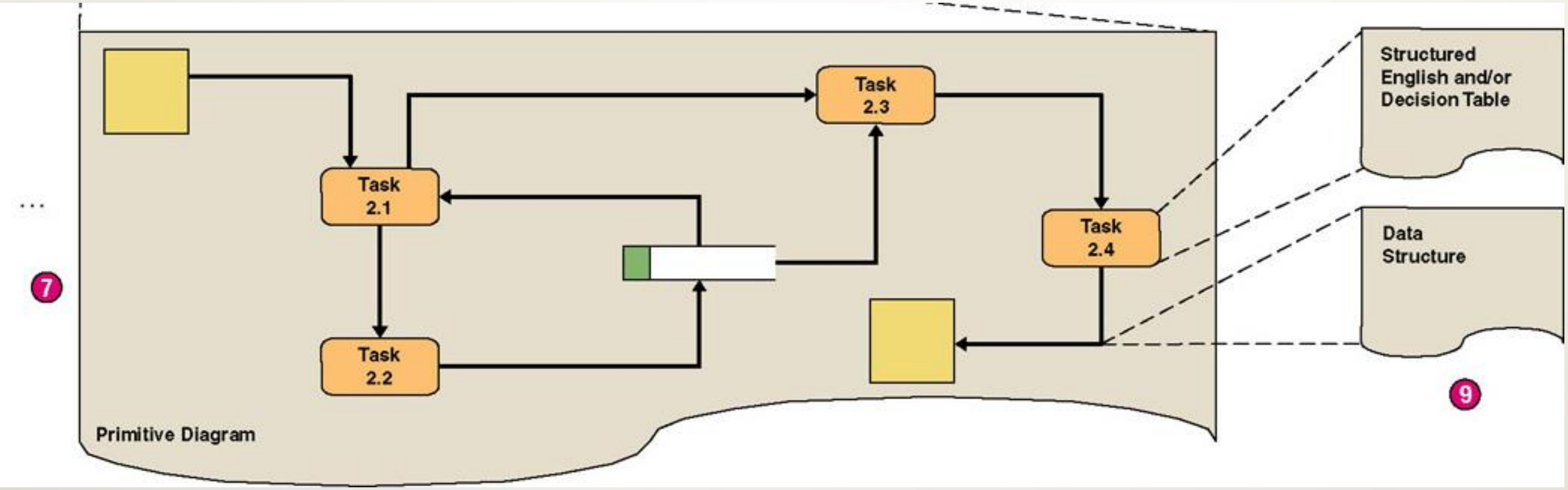


@ Dr. Muhammad Iqbal Hossain

Structured Analysis Diagram Progression (2 of 3)



Structured Analysis Diagram Progression (3 of 3)



@ Dr. Muhammad Iqbal Hossain

Context Data Flow Diagram

Context Data Flow Diagram

- **Context data flow diagram** - a process model used to document the scope for a system. Also called the environmental model.

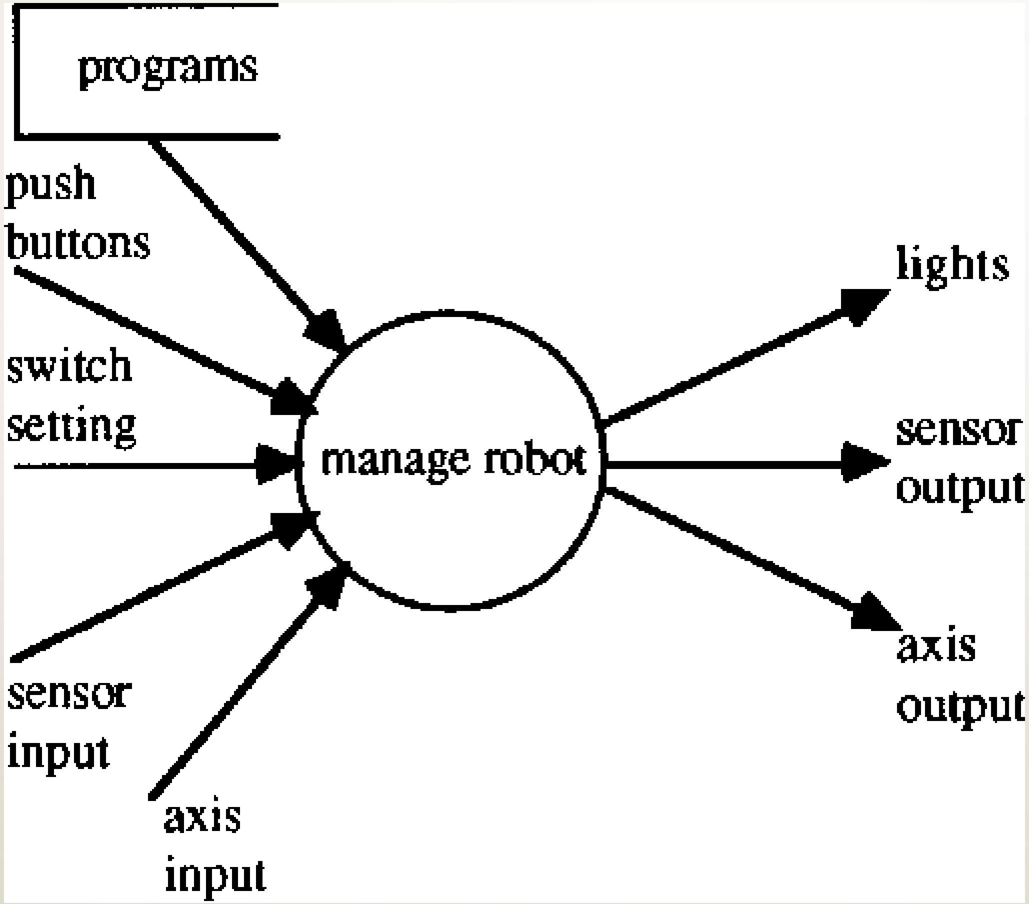
1. Think of the system as a "black box."
2. Ask users what business transactions the system must respond to. These are inputs, and the sources are external agents.
3. Ask users what responses must be produced by the system. These are outputs, and the destinations are external agents.
4. Identify any external data stores, if any.
5. Draw a context diagram.

Balancing

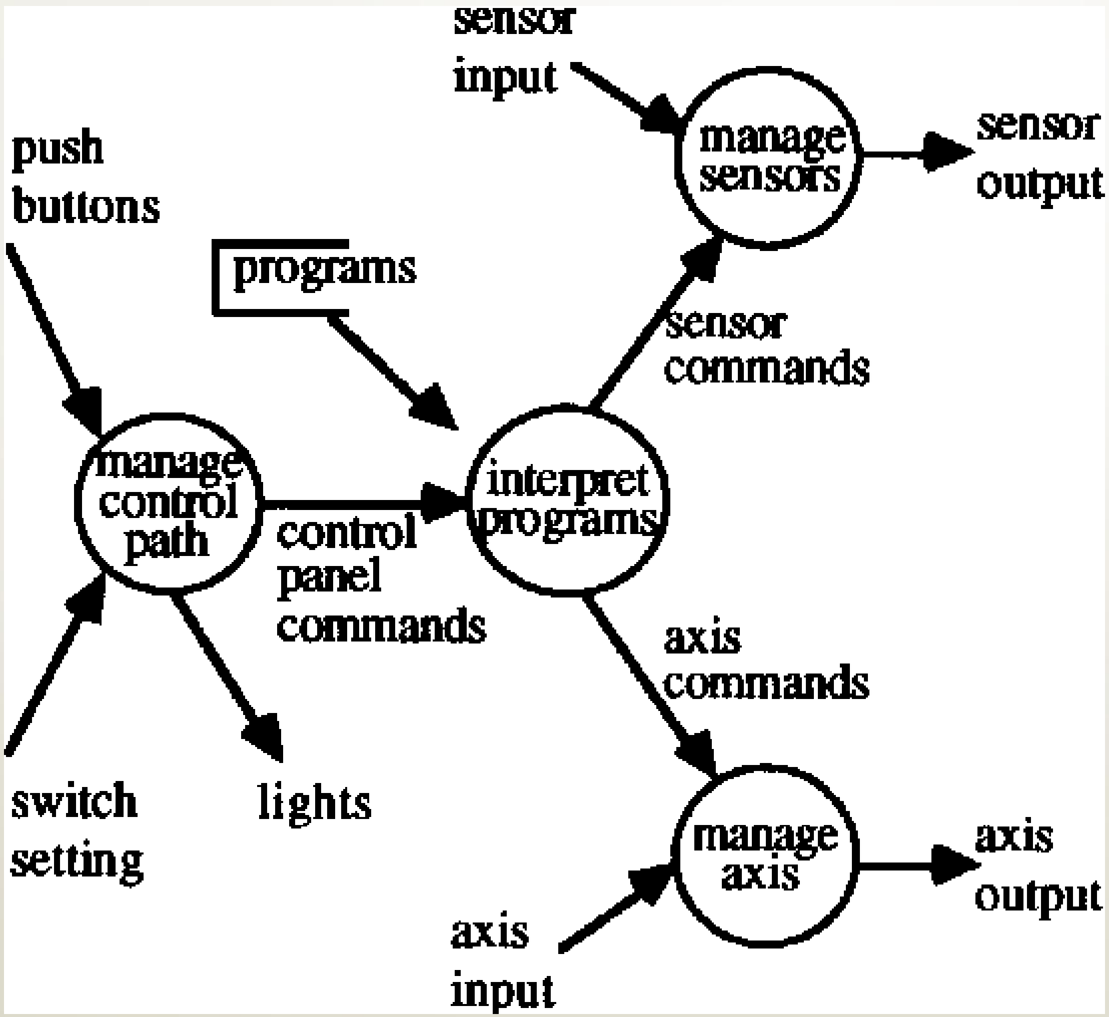
Balancing - a concept that requires that data flow diagrams at different levels of detail reflect consistency and completeness

- Quality assurance technique
- Requires that if you explode a process to another DFD to reveal more detail, you must include the same dta flows and data stores

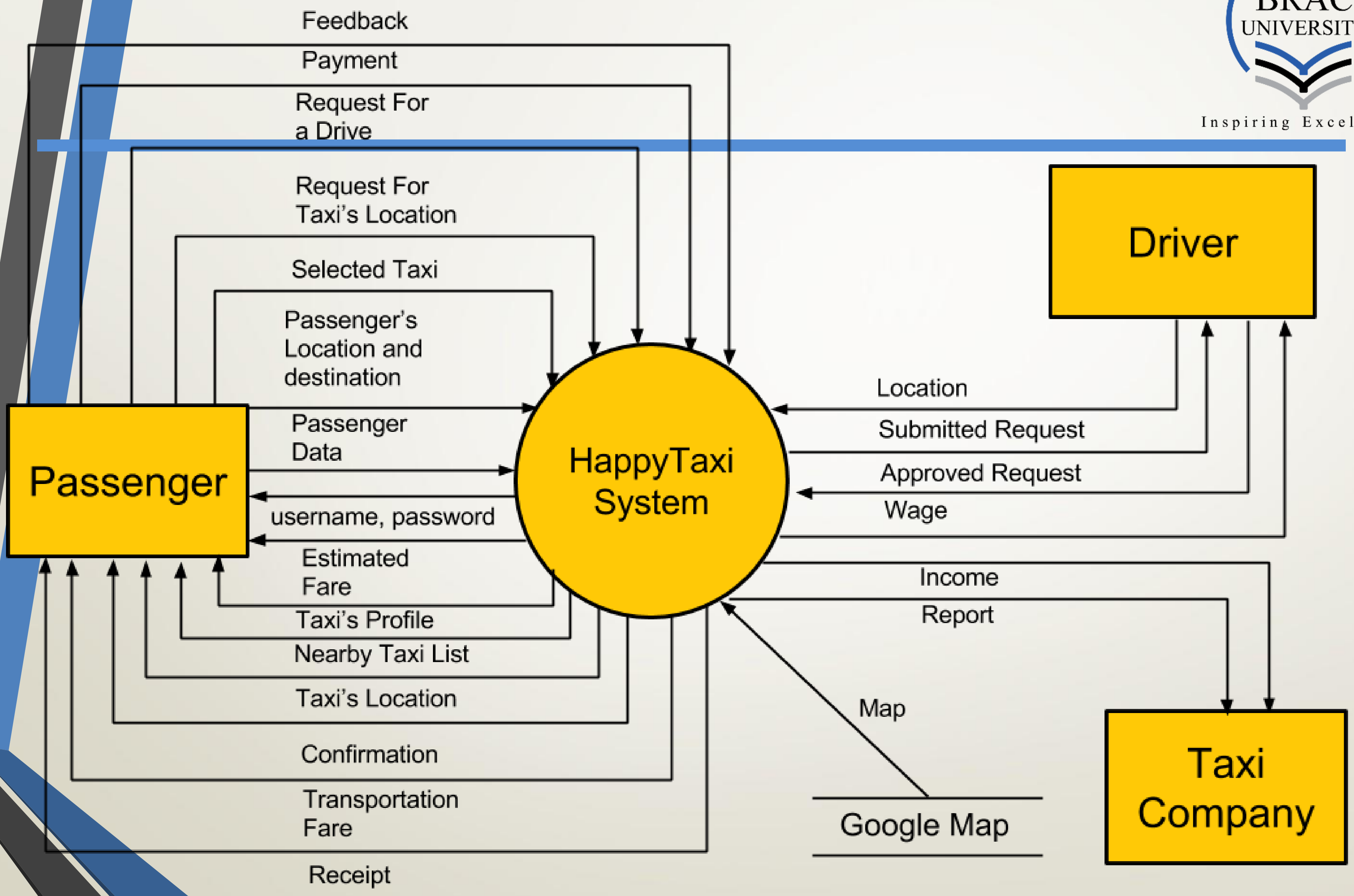
Manage Robot



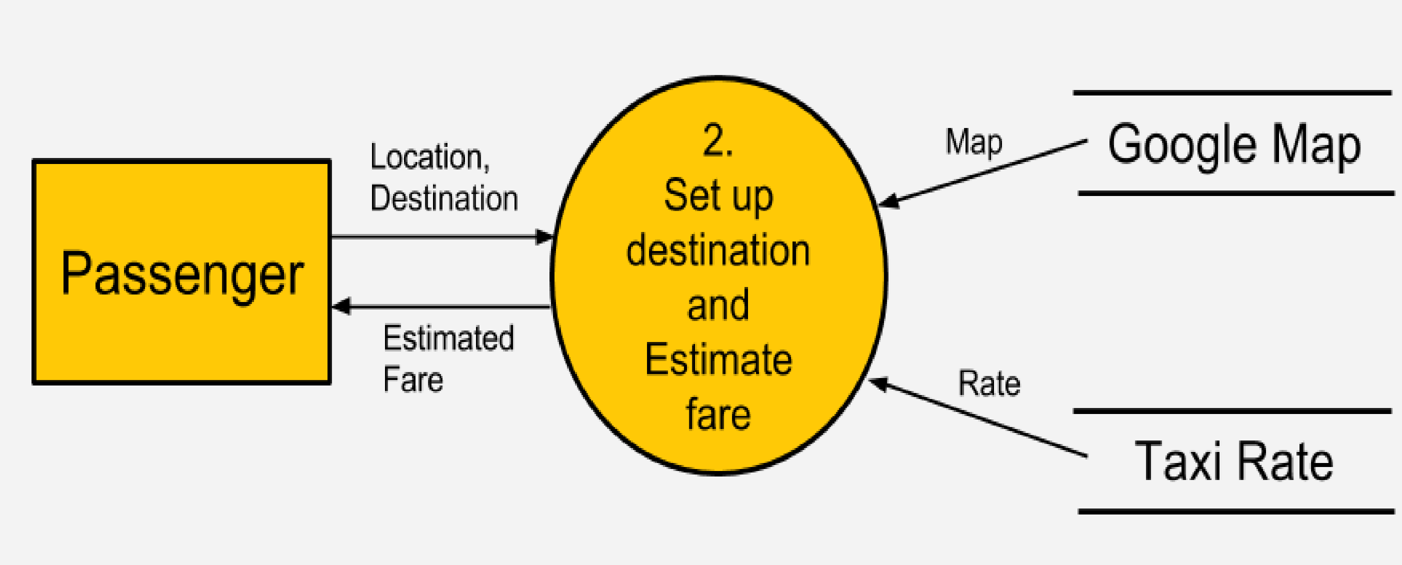
Manage Robot decomposed



@ Dr. Muhammad Iqbal Hossain


Feedback
Payment
Request For a Drive
Request For Taxi's Location
Selected Taxi
Passenger's Location and destination
Passenger Data
username, password
Estimated Fare
Taxi's Profile
Nearby Taxi List
Taxi's Location
Confirmation
Transportation Fare
Receipt
Passenger
HappyTaxi System
Driver
Location
Submitted Request
Approved Request
Wage
Income
Report
Map
Google Map
Taxi Company

Here passenger input current location and destination address to set up destination and estimate fare process and the process send estimate fare to the passenger. The same process gather map information from Google map and rate from taxi rate data base.

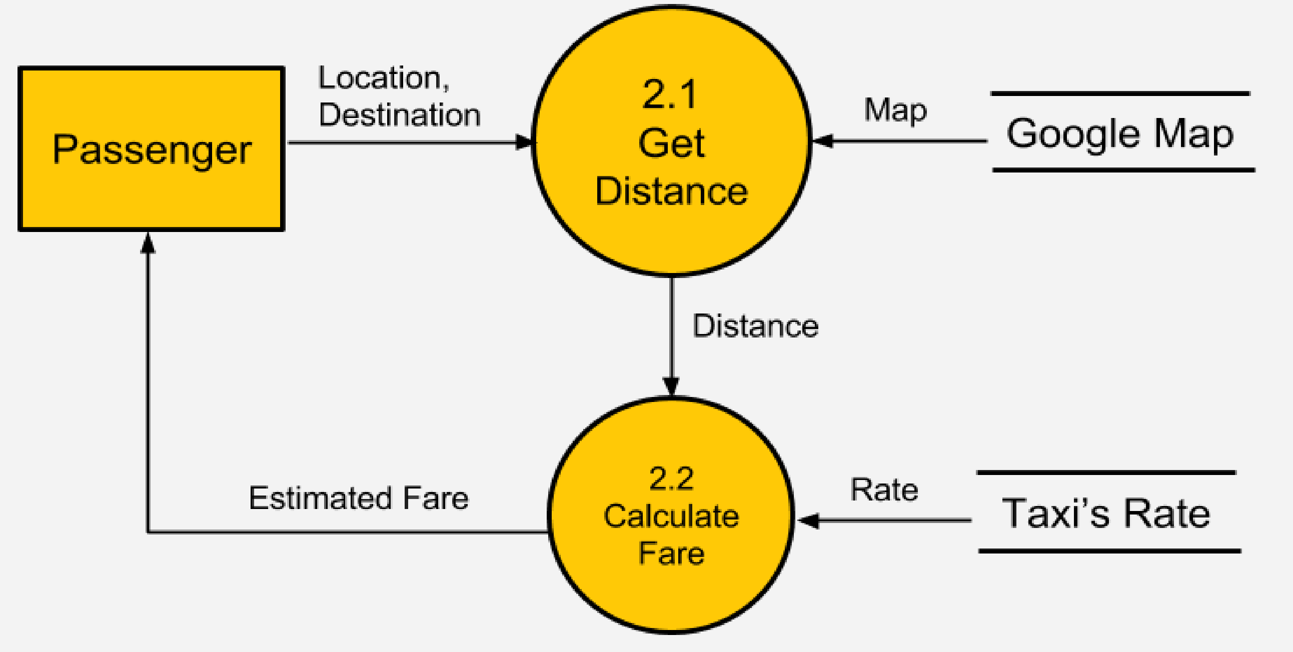


Class work

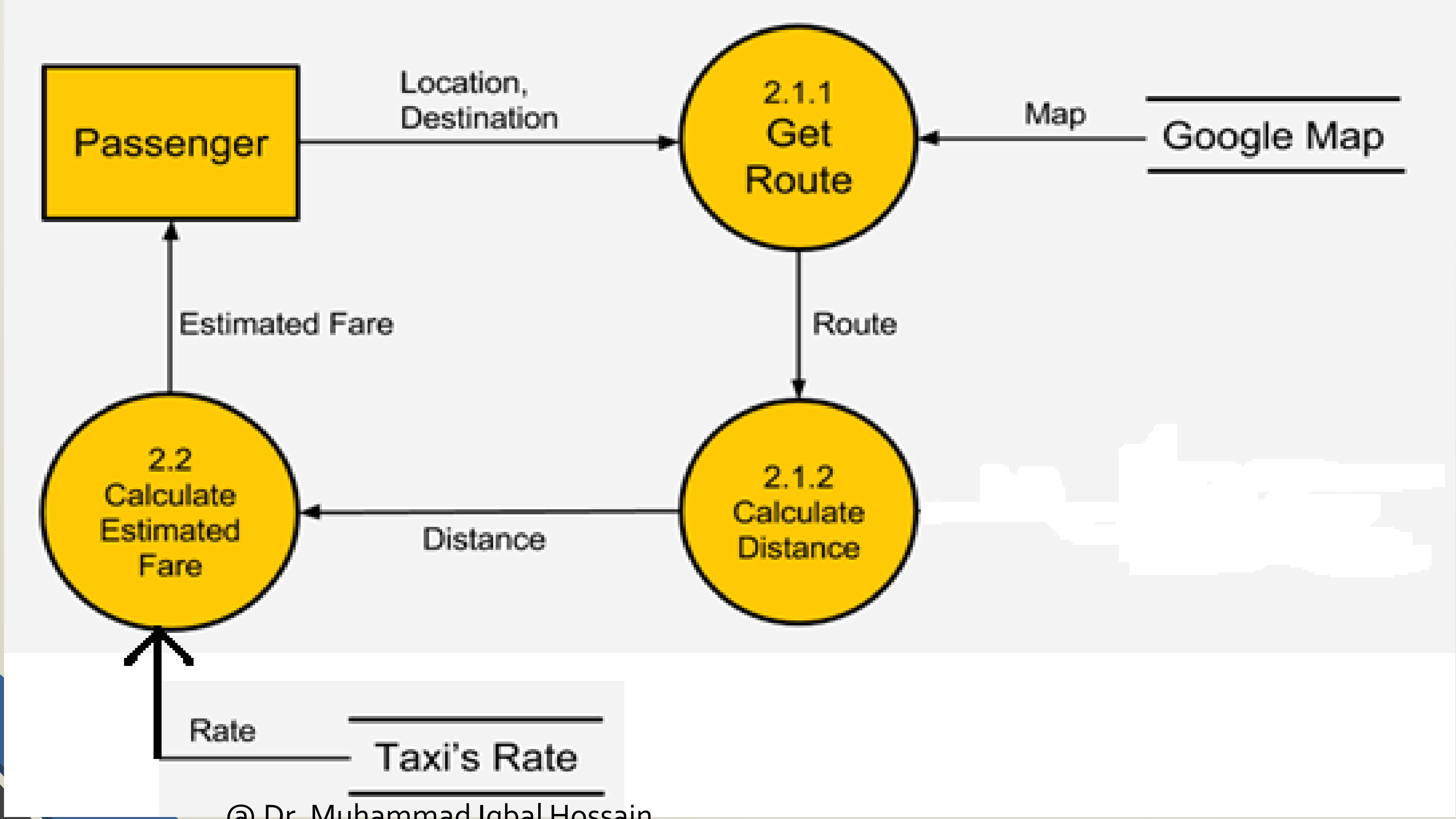
i. **Draw level 1 DFD** where set up destination and estimate fare process is decomposed into two sub process as get distance and calculate fare. Data of distance will sent from get distance to calculate fare.

i. **Draw Level 2 DFD** for get distance sub process. Decompose get distance in two sub process as get route and calculate distance. Data of route will sent from get route to calculate distance.

i. **Draw level 1 DFD** where set up destination and estimate fare process is decomposed into two sub process as get distance and calculate fare. Data of distance will sent from get distance to calculate fare.



Draw Level 2 DFD for get distance sub process. Decompose get distance in two sub process as get route and calculate distance. Data of route will sent from get route to calculate distance.


Passenger
Location, Destination
2.1.1 Get Route
Map
Google Map
Estimated Fare
Route
2.2 Calculate Estimated Fare
Distance
2.1.2 Calculate Distance
Rate
Taxi's Rate