Daffodil Polytechnic Institute

System Analysis Design

Chapter 5

Understand the tools of structured analysis(DFD)

Learning Goals

In this module we will learn

1. What are Data Flow Diagrams (DFDs)?
2. Why they are useful?
3. How are they developed?
4. How to level DFDs?
5. Good style conventions in developing DFDs
6. Difference between Logical and Physical DFDs
7. Tools available to draw DFDs

Flow-Oriented Modeling

- ❑Represents how data objects are transformed at they move through the system.
- ❑**Data flow diagram (DFD)** is the diagrammatic form that is used.
- ❑Considered by many to be an “old school” approach, but continues to provide a view of the system that is unique—it should be used to supplement other analysis model elements.

Why DFD?

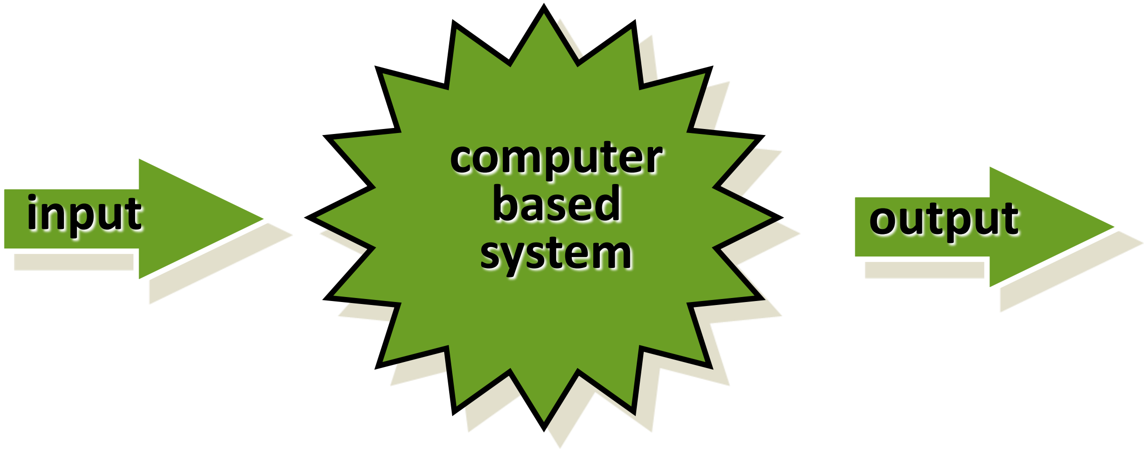
- If a DFD is too detailed it will have too many data flows and will be large and difficult to understand

- Start from a broad overview. Expand to details - Idea similar to using procedures and linking these with a main program

- Each DFD must deal with one aspect of a big system

The Flow Model

Every computer-based system is an information transform ....



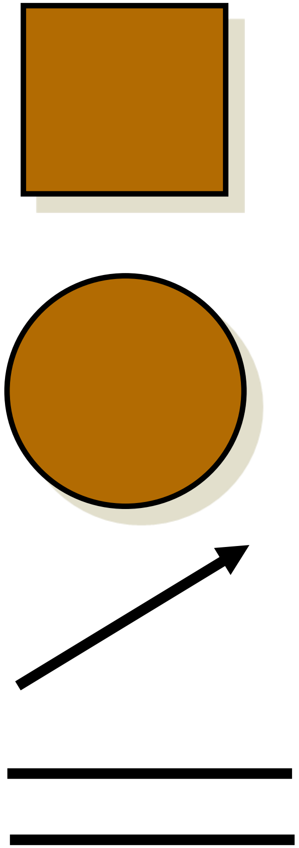
Data Flow Diagrams

WHAT ARE DATA FLOW DIAGRAMS?

❑ DFDs models the system by depicting

- **External entities** from which the data flows and where results terminate
- **Processes** which transform data flows
- **Data stores** from which the data are read or into which data are written by the processes.

Flow Modeling Notation



External Entity

Process

Data flow

Data store

External Entity

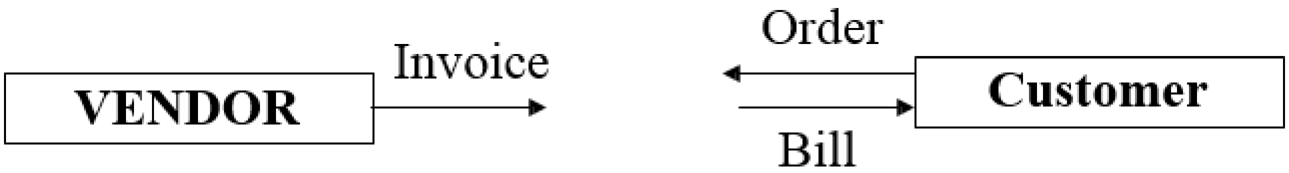
A producer or consumer of data

Examples: **a person, a device, a sensor**

Another example: computer-based system

Data must always originate somewhere and must always be sent to something

Example: External Entity



- A Rectangle represents an external entity
- They either supply data or receive data
- They do not process data

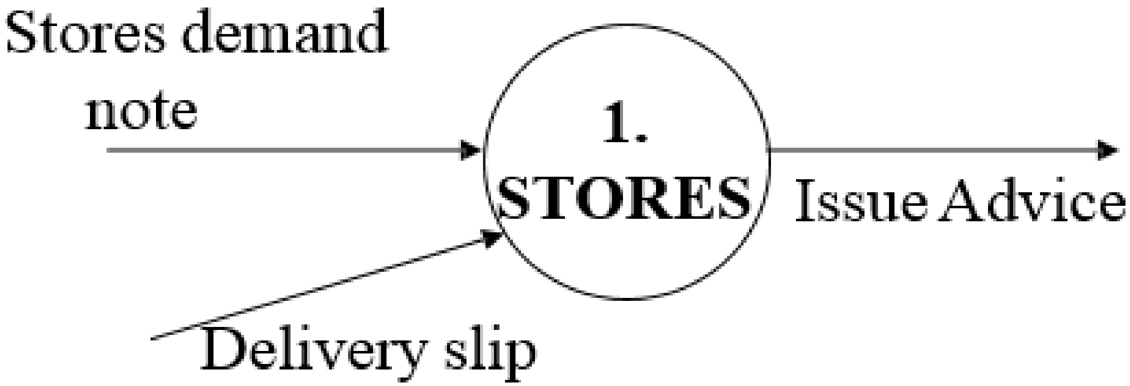
Process

A data transformer (changes input to output)

Examples: *compute taxes, determine area, format report, display graph*

Data must always be processed in some way to achieve system function

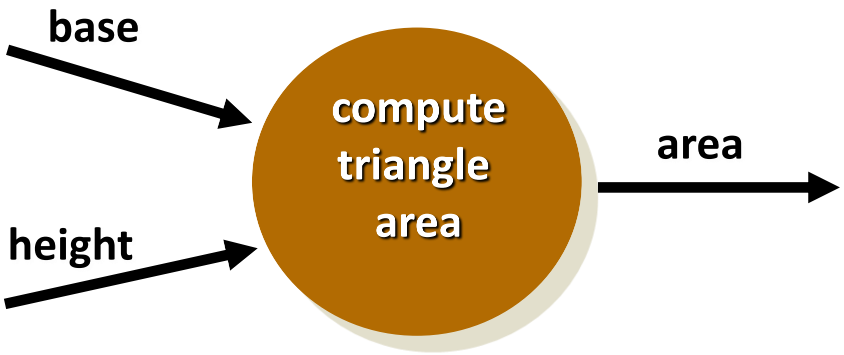
Example: Process



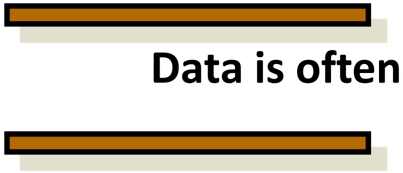
- A circle represents a process
- Straight lines with incoming arrows are input data flows
- Straight lines with outgoing arrows are output data flows
- Processes are given serial numbers for easy reference
- Labels are assigned to Data flow.

Data Flow

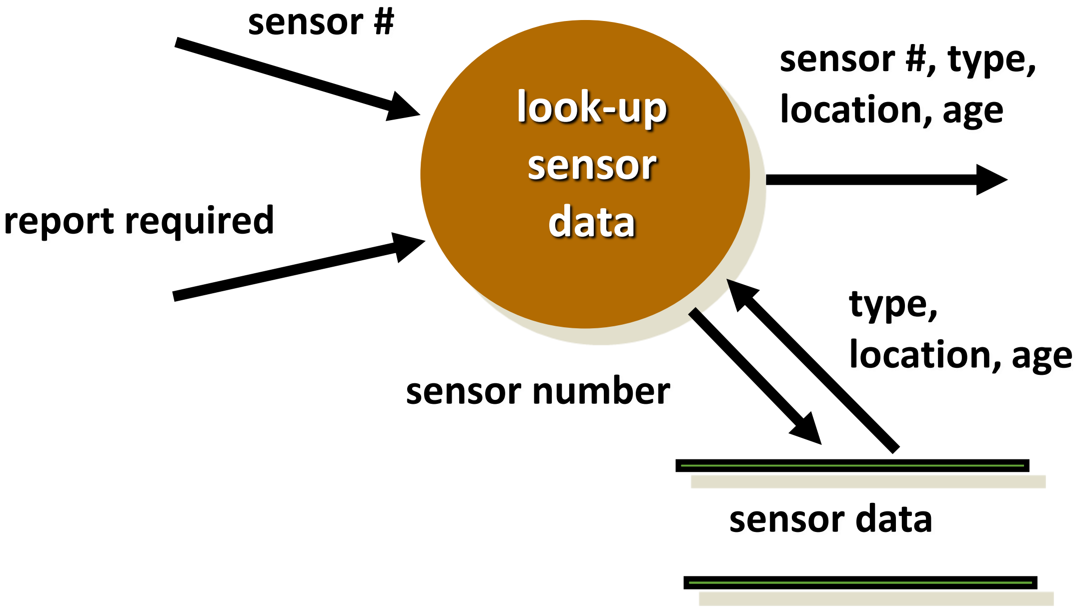
Data flows through a system, beginning as input and transformed into output.



Data Stores

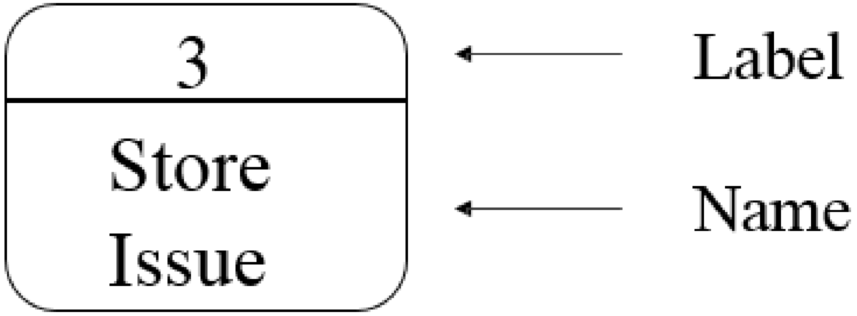


Data is often stored for later use.

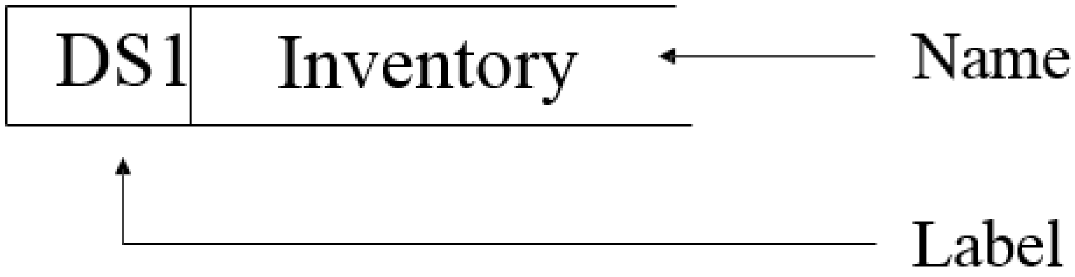


DFD Notation

A Process



A Data store



DFD Rules and Tips

- ❑ Each process should have at least one input and an output.
- ❑ Each data store should have at least one data flow in and one data flow out.
- ❑ Data stored in a system must go through a process.
- ❑ All processes in a DFD go to another process or a data store.

Cont…

❑ 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

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

DFD levels and layers: From context diagrams to pseudocode

- ❑ A data flow diagram can dive into progressively more detail by using levels and layers, zeroing in on a particular piece.
- ❑ DFD levels are numbered 0, 1 or 2, and occasionally go to even Level 3 or beyond.

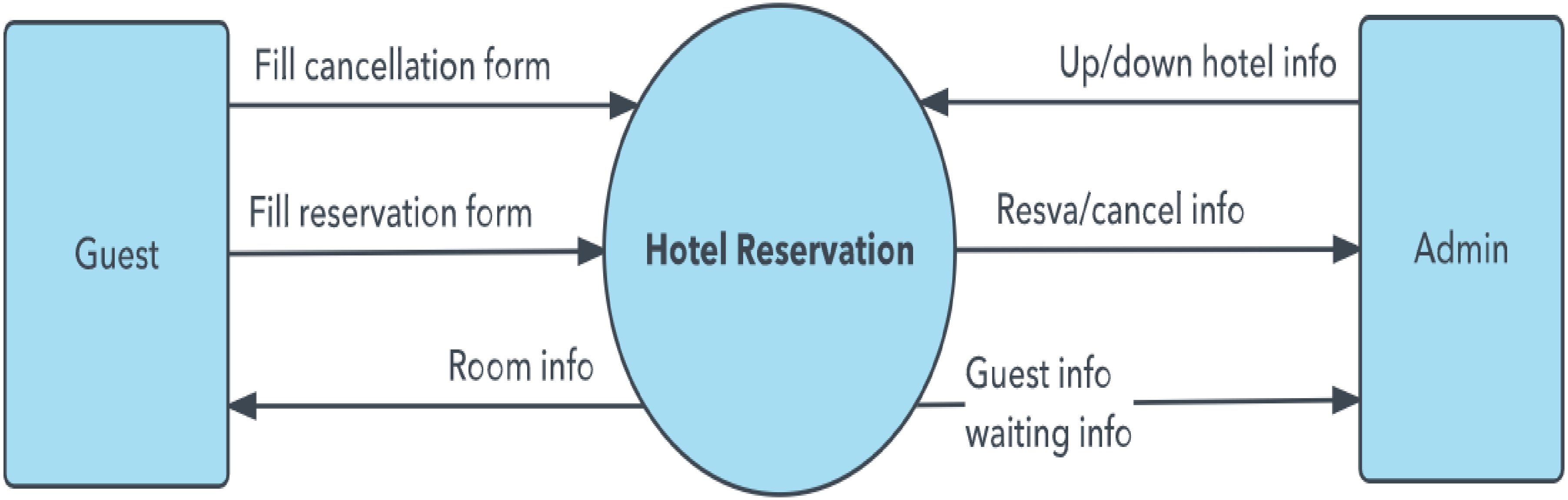
DFD 0

- ❑ DFD Level 0 is also called a **Context Diagram**. It’s a basic overview of the whole system or process being analyzed or modeled.
- ❑ It’s designed to be an at-a-glance view, showing the system as a single high-level process, with its **relationship to external entities.**
- ❑ It should be easily understood by a wide audience, including stakeholders, business analysts, data analysts and developers.

DFD 0

- ❑ DFD Level 0 is also called a **Context Diagram**. It's a basic overview of the whole system or process being analyzed or modeled.
- ❑ It's designed to be **an at-a-glance view**, showing the system as a single high-level process, with its relationship to external entities.
- ❑ It should be easily understood by a wide audience, including stakeholders, business analysts, data analysts and developers.

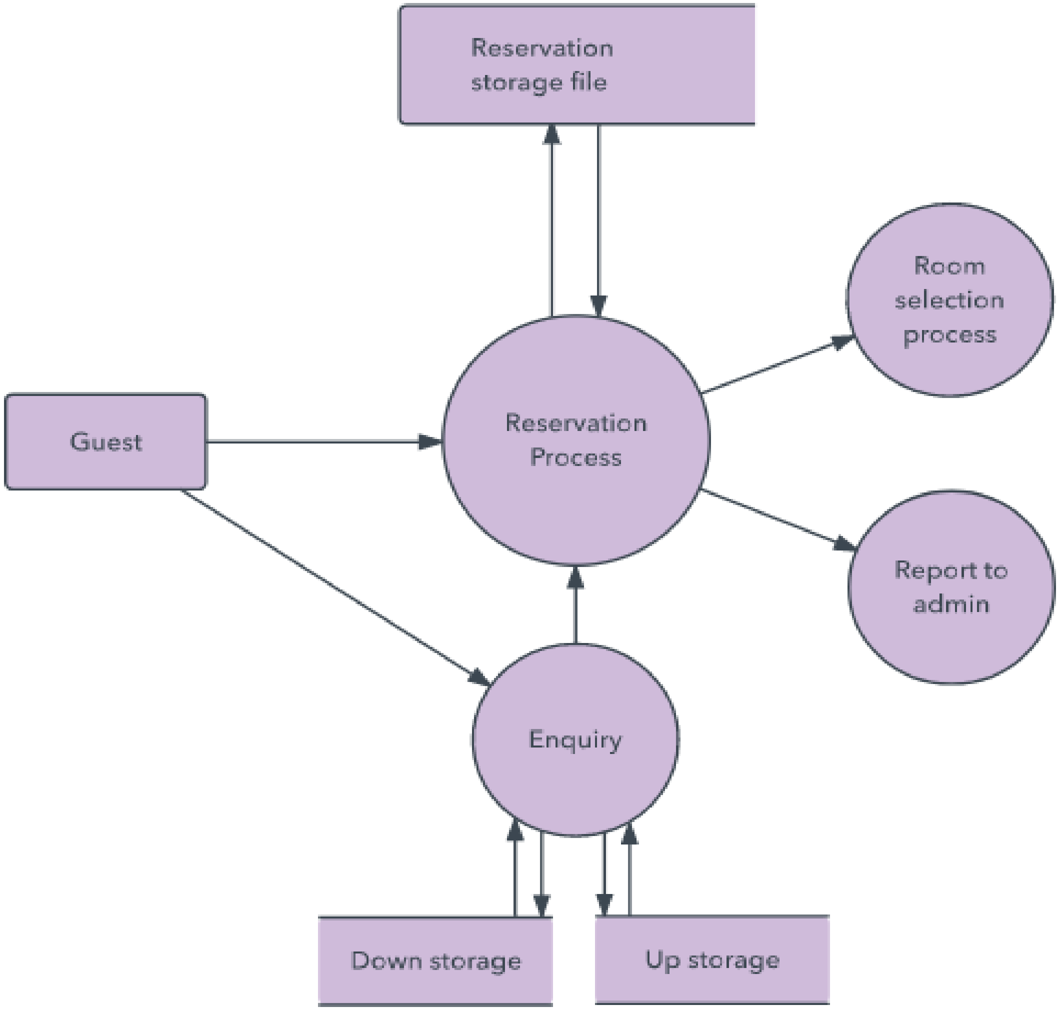
DFD 0


Guest
Fill cancellation form
Fill reservation form
Room info
Hotel Reservation
Up/down hotel info
Resva/cancel info
Guest info
waiting info
Admin

DFD-1

- ❑ DFD Level 1 provides a **more detailed breakout of pieces** of the Context Level Diagram.
- ❑ You will highlight the main functions carried out by the system, as you break down the high-level process of the Context Diagram into its sub-processes.

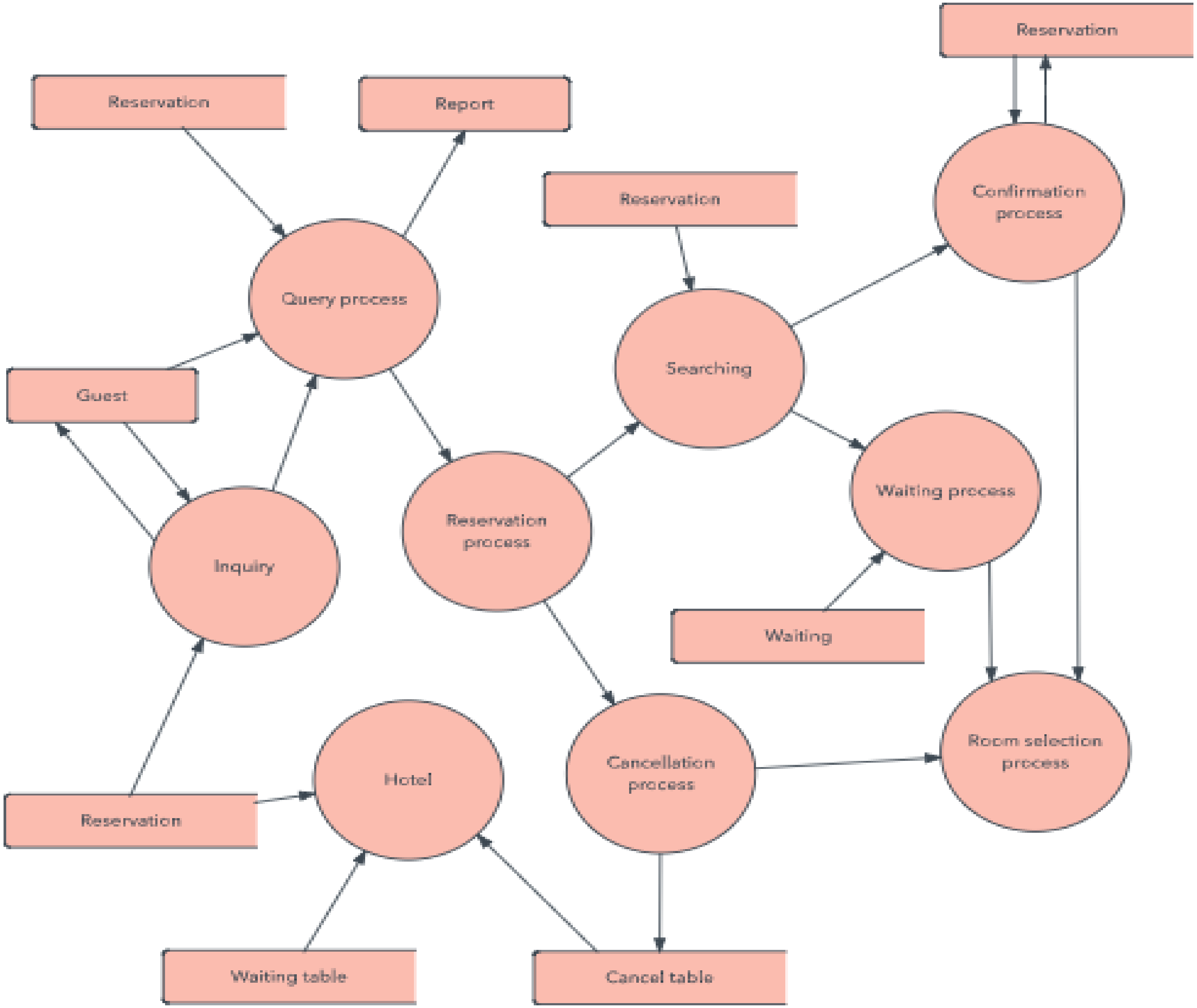
DFD-1


Reservation storage file
Room selection process
Guest
Reservation Process
Report to admin
Enquiry
Down storage
Up storage

DFD-2

- DFD Level 2 then goes one **step deeper into parts of Level 1**.
- It may require more text to reach the necessary level of detail about the system’s functioning.

DFD-2


Reservation
Report
Reservation
Query process
Confirmation process
Searching
Guest
Reservation process
Waiting process
Inquiry
Waiting
Hotel
Cancellation process
Room selection process
Reservation
Waiting table
Cancel table

DFD 3, 4

- ❑ Progression to Levels 3, 4 and beyond is possible, but going beyond Level 3 is uncommon.
- ❑ Doing so can create complexity that makes it difficult to communicate, compare or model effectively.

Leveling Rules

- **If process *p* is expanded, the process at the next level are labeled as *p.1,p.2* etc.**

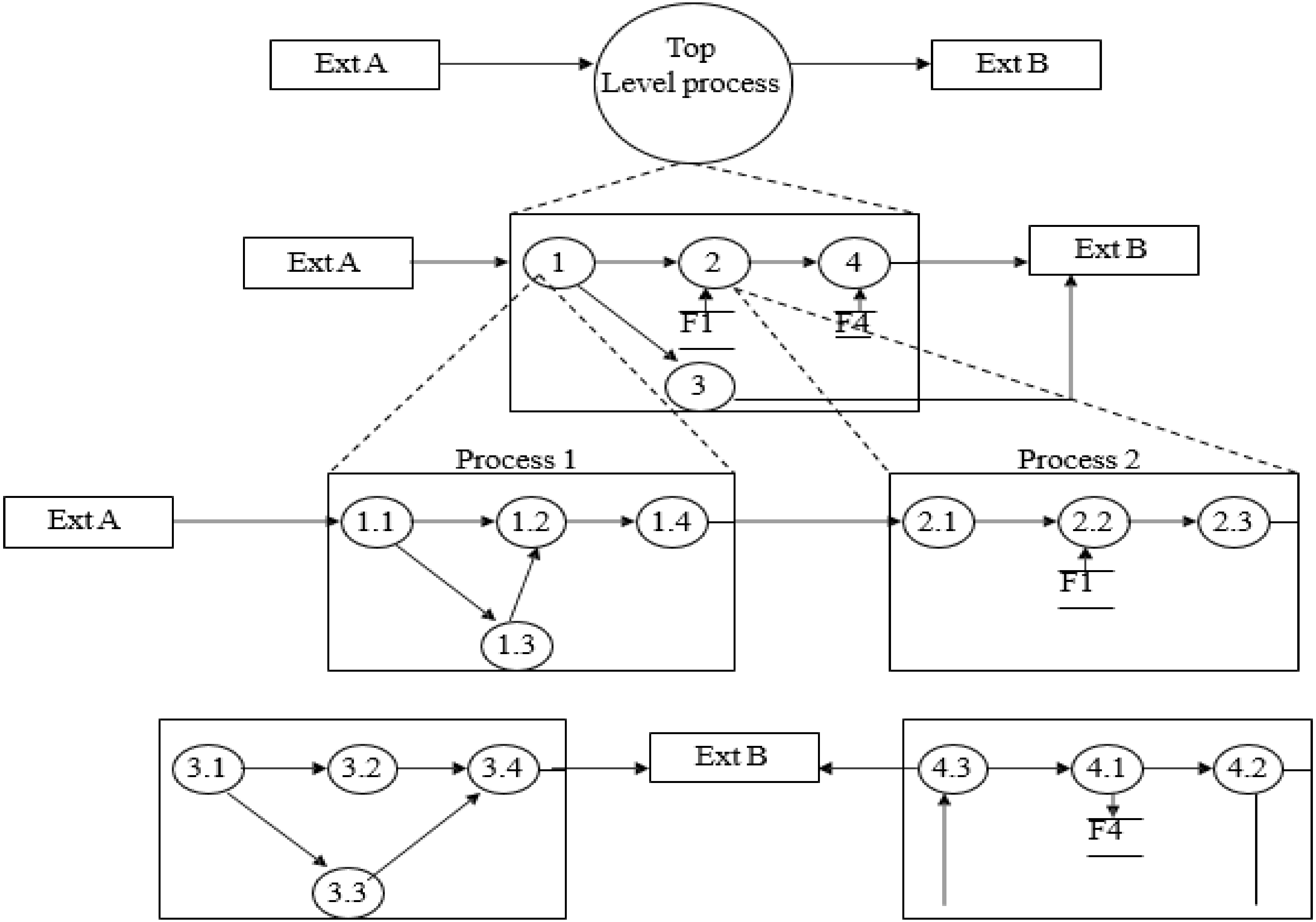
- **All data flow entering or leaving *p* must also enter or leave its expanded version.**

- **Expanded DFD may have data stores**

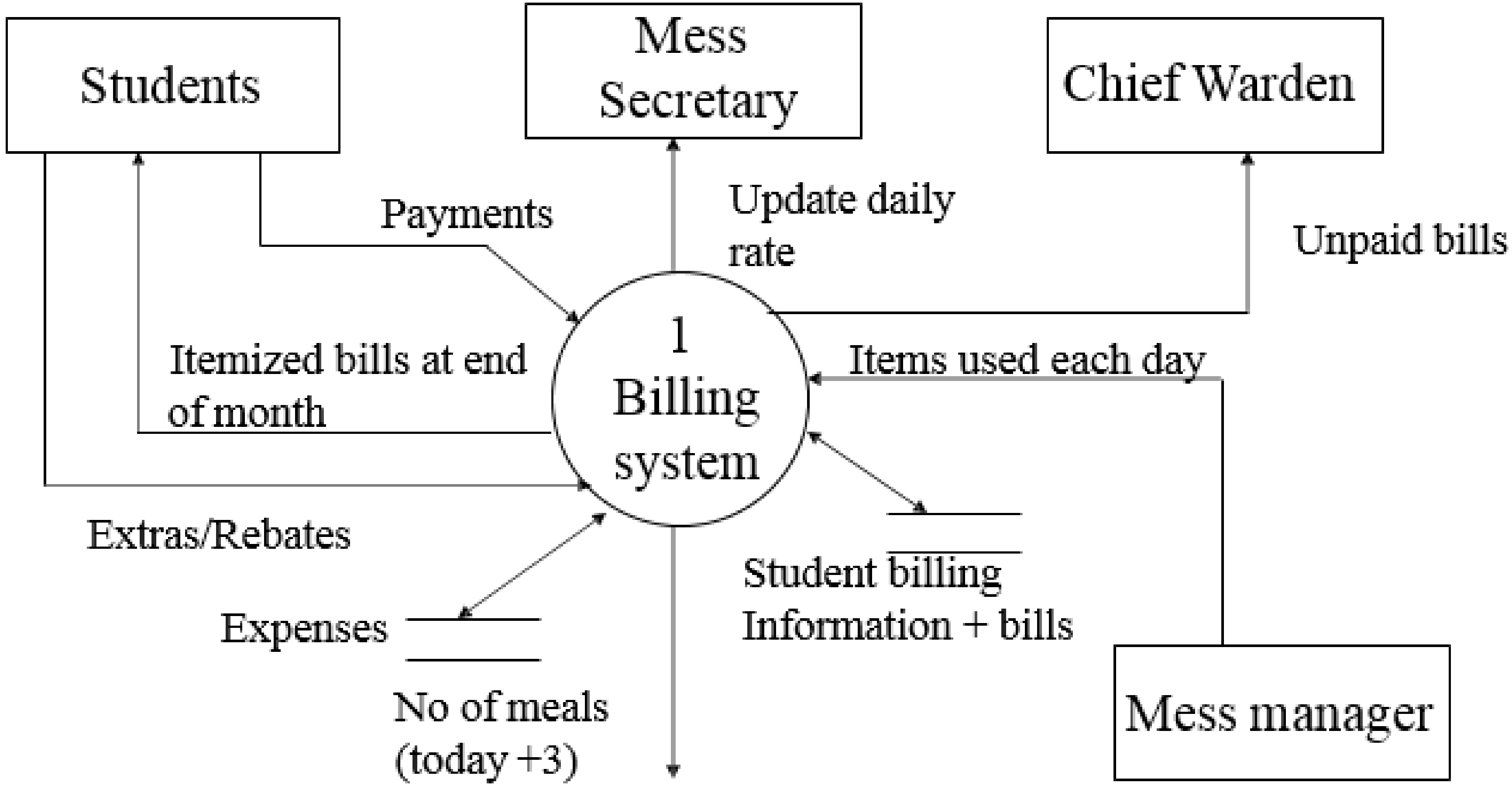
- **No external entity can appear in expanded DFD**

- **Keep the number of processes at each level less than**

DFD levels and layers

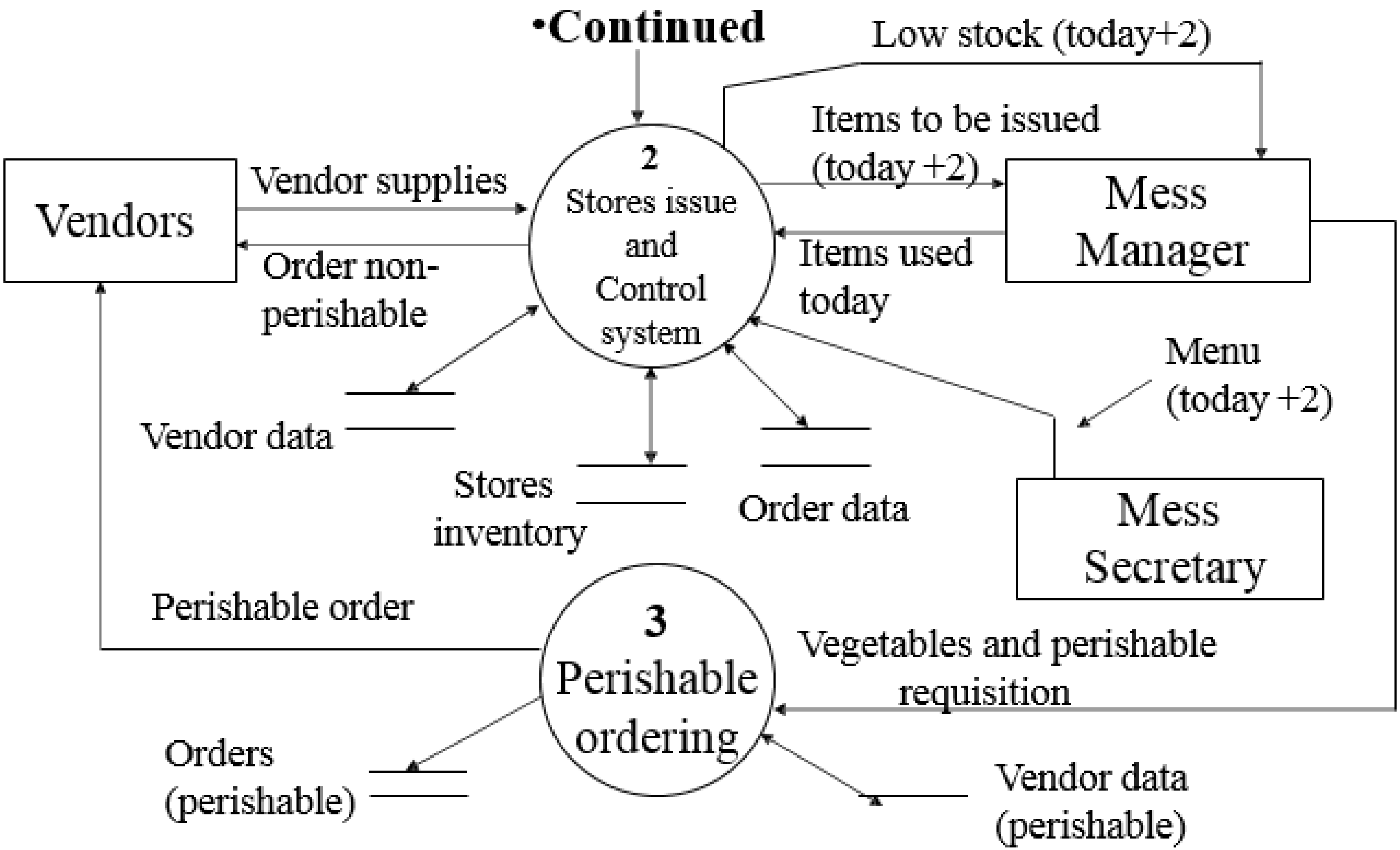


Example: Billing System

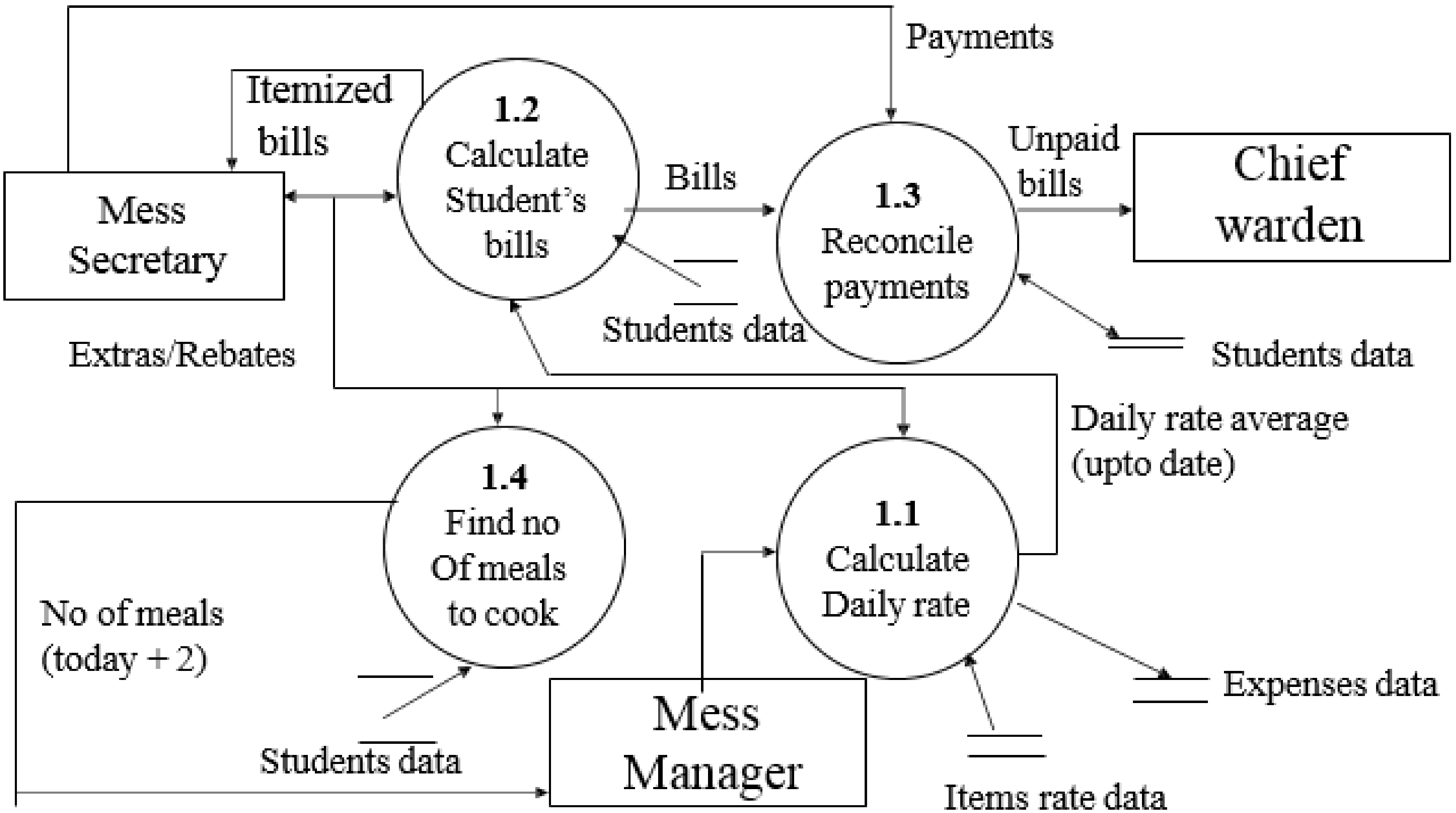


- **Going to next process (Continued in next slide)**

Example: Billing System(Cont…)



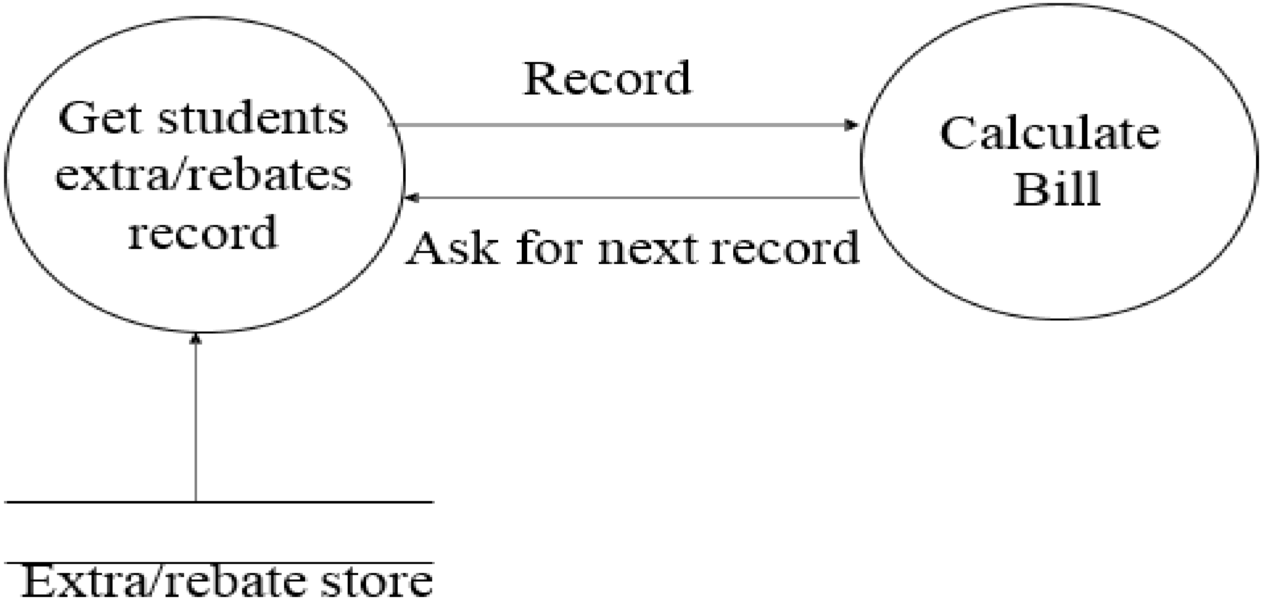
Example: Billing System(Cont…)



Illegal Constructs In DFD

- No loops are allowed in DFD
- A process cannot be a pure decision
- A single data flow should not be split into many flows with different labels
- No data flow allowed between data stores

Illegal Constructs In DFD



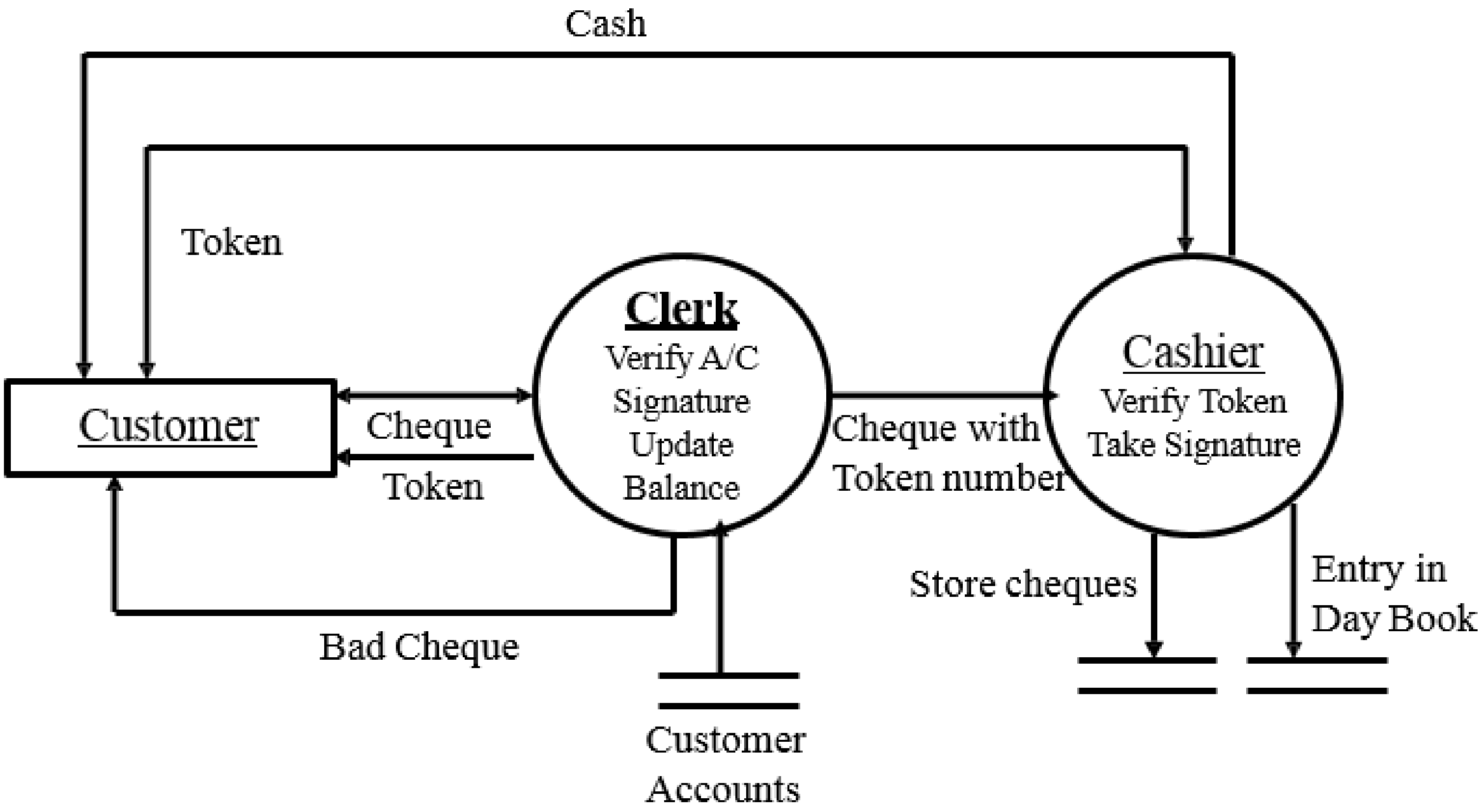
Not correct as loop is formed

Logical DFD vs. Physical DFD

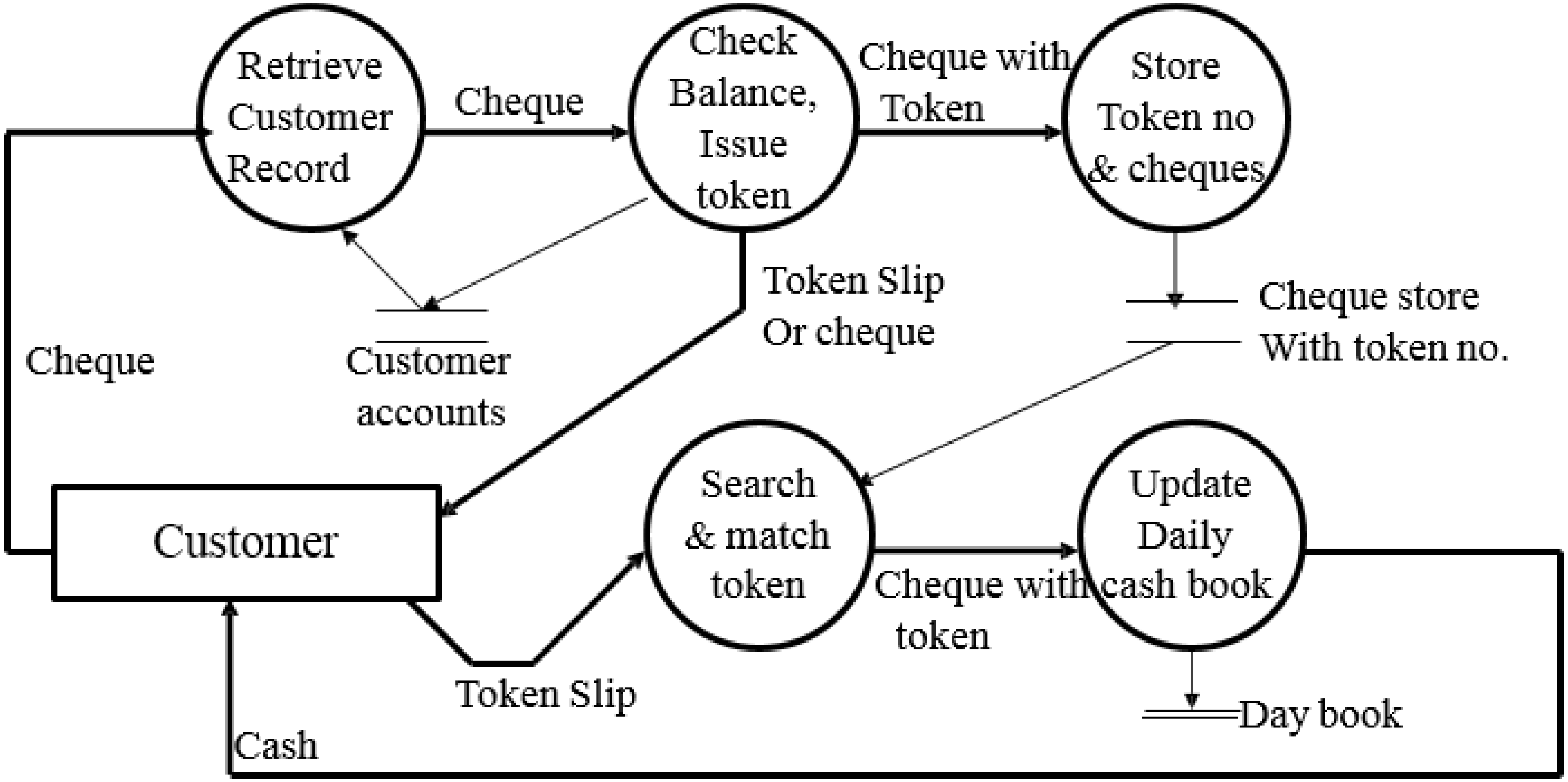
- ❑ A Logical DFD visualizes the data flow that is essential for a business to operate.
  - ❑ It specifies who does the operations specified by the logical DFD

- ❑ A Physical DFD shows how the system is actually implemented now.
  - ▪ A physical DFD is similar to a document flow diagram.
  - ▪ 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


Retrieve Customer Record
Cheque
Check Balance, Issue token
Cheque with Token
Store Token no & cheques
Cheque
Customer accounts
Token Slip Or cheque
Cheque store With token no.
Customer
Search & match token
Cheque with token
Update Daily cash book
Token Slip
Day book
Cash