MODULE 5

DATA FLOW DIAGRAMS

Contents

DATA FLOW DIAGRAMS

MOTIVATION

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

Structure of DFD allows starting from a broad overview and expands it to a hierarchy of detailed diagrams

LEARNING GOALS

At the end of this module you will know

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

LEARNING UNIT 1

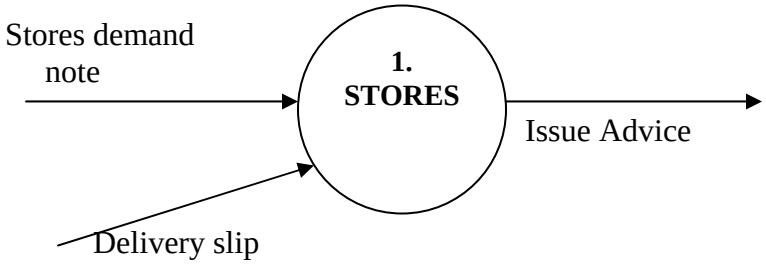
Developing Data Flow Diagrams(DFD)

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.

SYMBOLS USED IN DFD

PROCESS

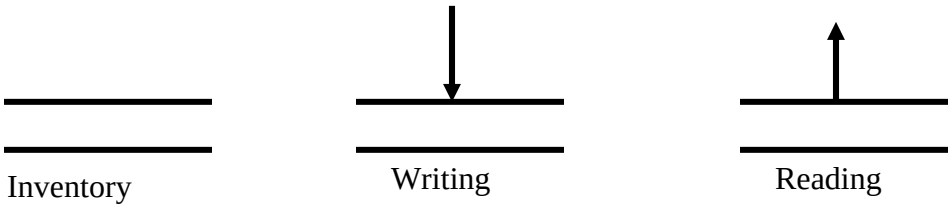


- A circle represents a process
- Straight lines with incoming arrows are input data flows
- Straight lines with outgoing arrows are output data flows
- Processes are given serial numbers for easy reference
- Labels are assigned to Data flow. These aid documentation

EXTERNAL ENTITIES

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

DATA STORES



A Data Store is a repository of data
Data can be written into the data store and this is depicted by an incoming arrow.
Data can be read from a data store and this is depicted by an outgoing arrow
External entity cannot read or write to the data store. Two data stores cannot be connected by a data flow

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

GOOD STYLE IN DRAWING DFD

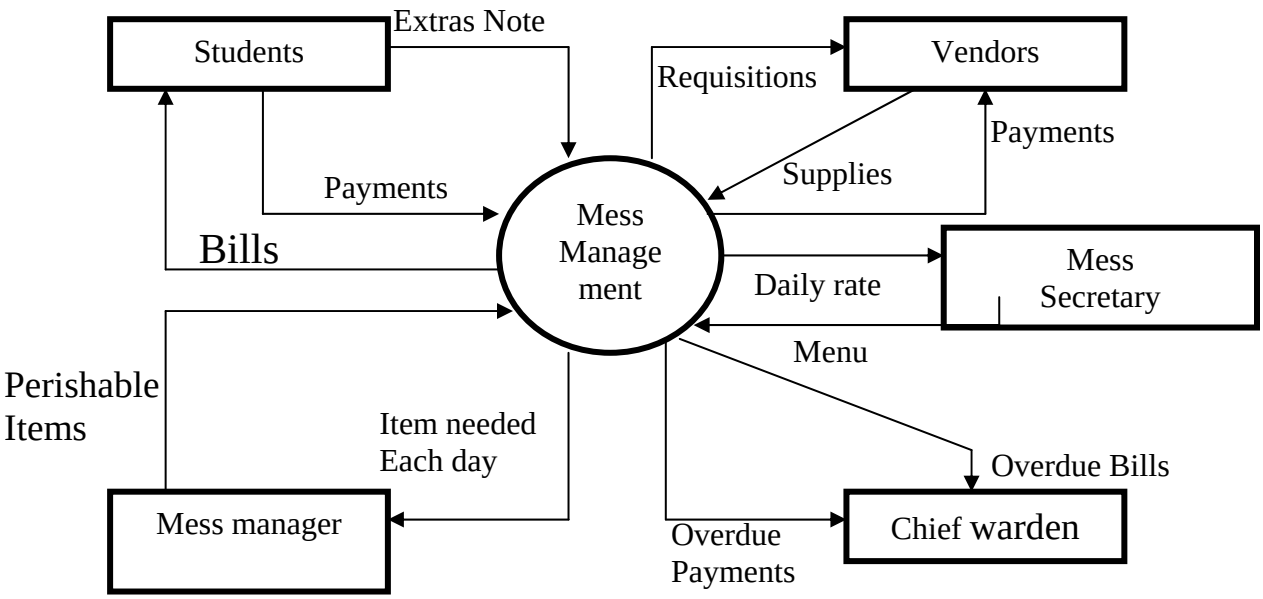
- Use meaningful names for data flows, processes and data stores.
- Use top down development starting from context diagram and successively levelling DFD
- Only previously stored data can be read
- A process can only transfer input to output.It cannot create new data
- Data stores cannot create new data

LEARNING UNIT 2

Describing systems with DFD & Levelling DFDs

An entire system is represented by one DFD which gives the system's overview . It is called a context diagram. It gives little detail & is also known as the top level DFD.

Context diagram of mess management is shown



Note: This diagram gives very little detail

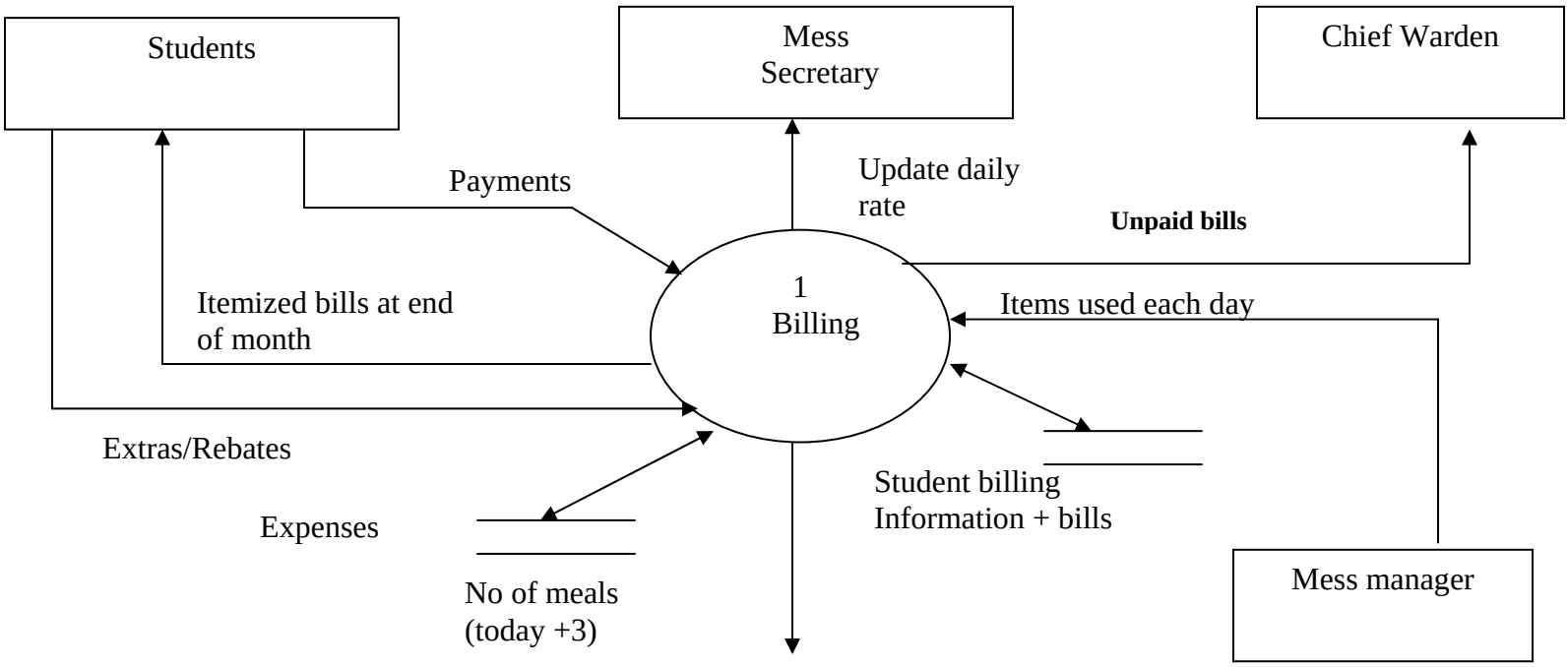
LEVELLING DFD

A context diagram gives an overview, it should be split into major processes which give greater detail. Each major process is further split to give more detail. Each major process is further split to give more detail

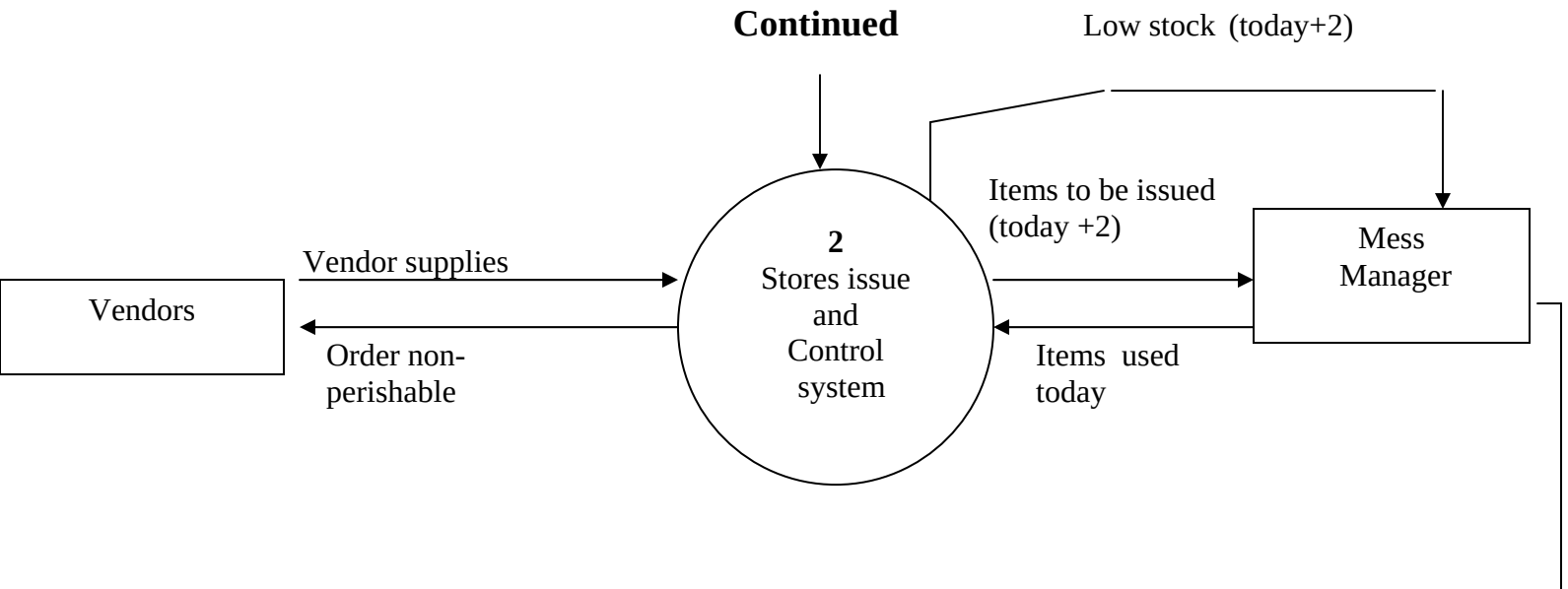
WHY LEVEL DFD?

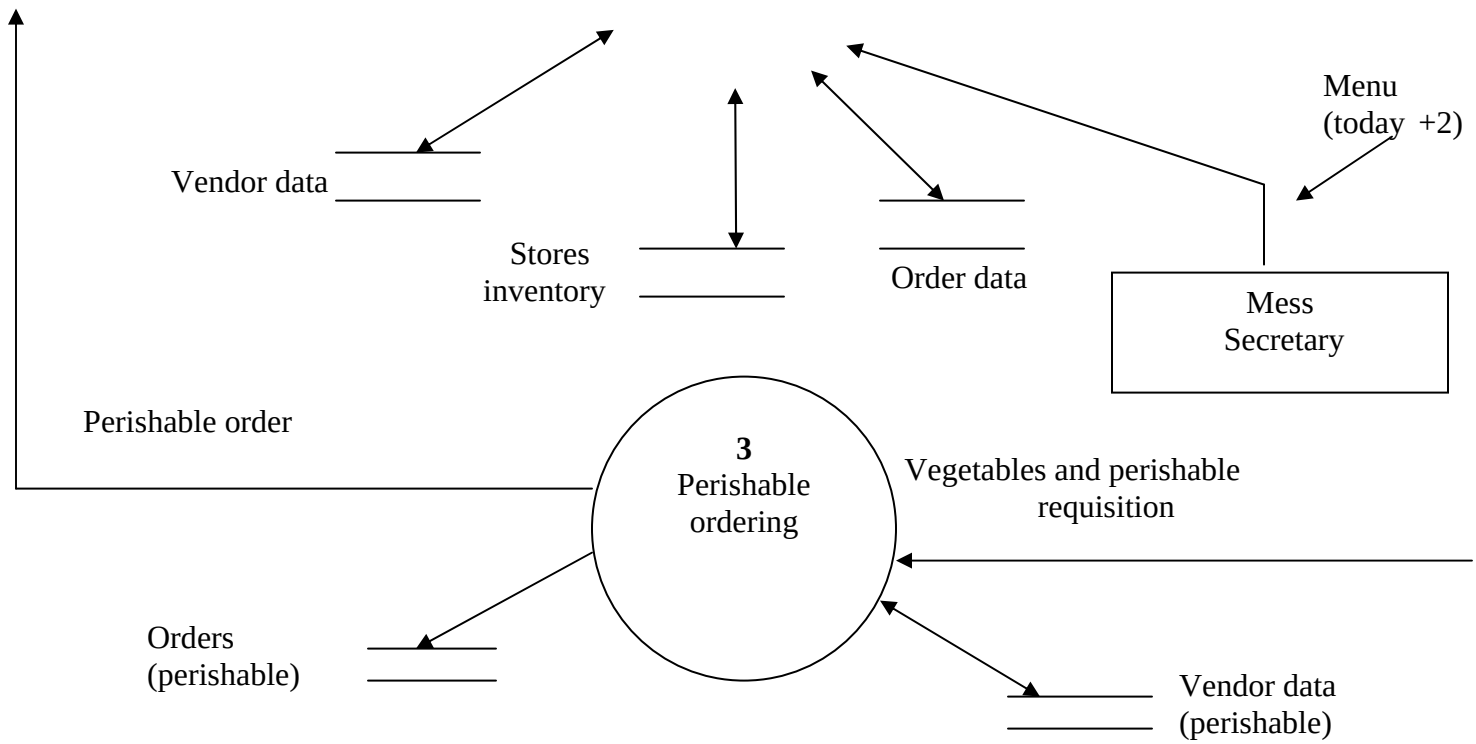
If a DFD is too detailed it will have too many data flows and will be large and difficult to understand. Therefore start from a broad overview. Expand the details - Idea similar to using procedures and linking these with a main program. Each DFD must deal with one aspect of a big system

EXPANDED DFD FOR HOSTEL MESS MANAGEMENT



•Going to next process (Continued in next page)





LEVELLING RULES

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

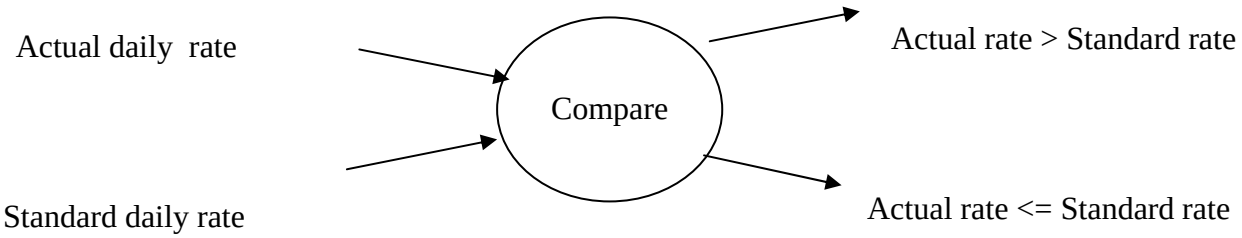
- All data flow entering or leaving *p* must also enter or leave its expanded version.

- Expanded DFD may have data stores

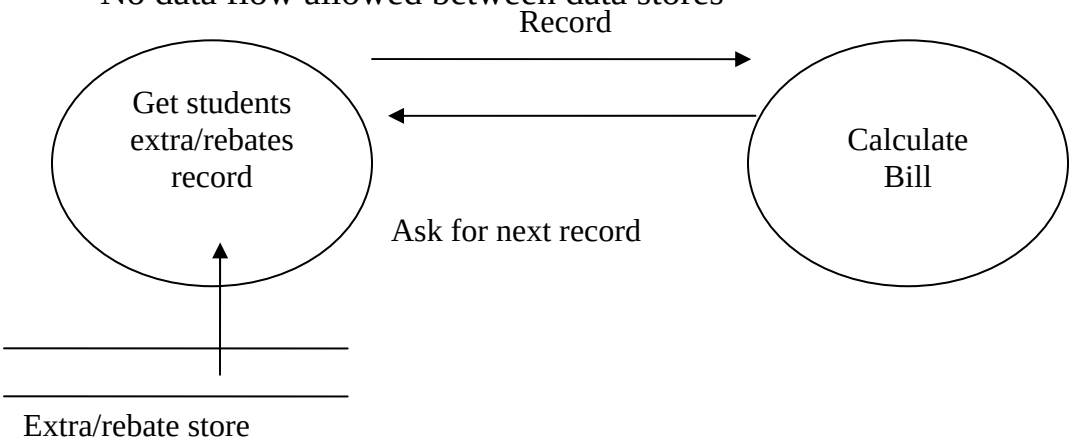
- No new external entity can appear in expanded DFD
- Keep the number of processes at each level less than 7.

ILLEGAL CONSTRUCTS IN DFD

- No loops are allowed in DFD
- A process cannot be a pure decision

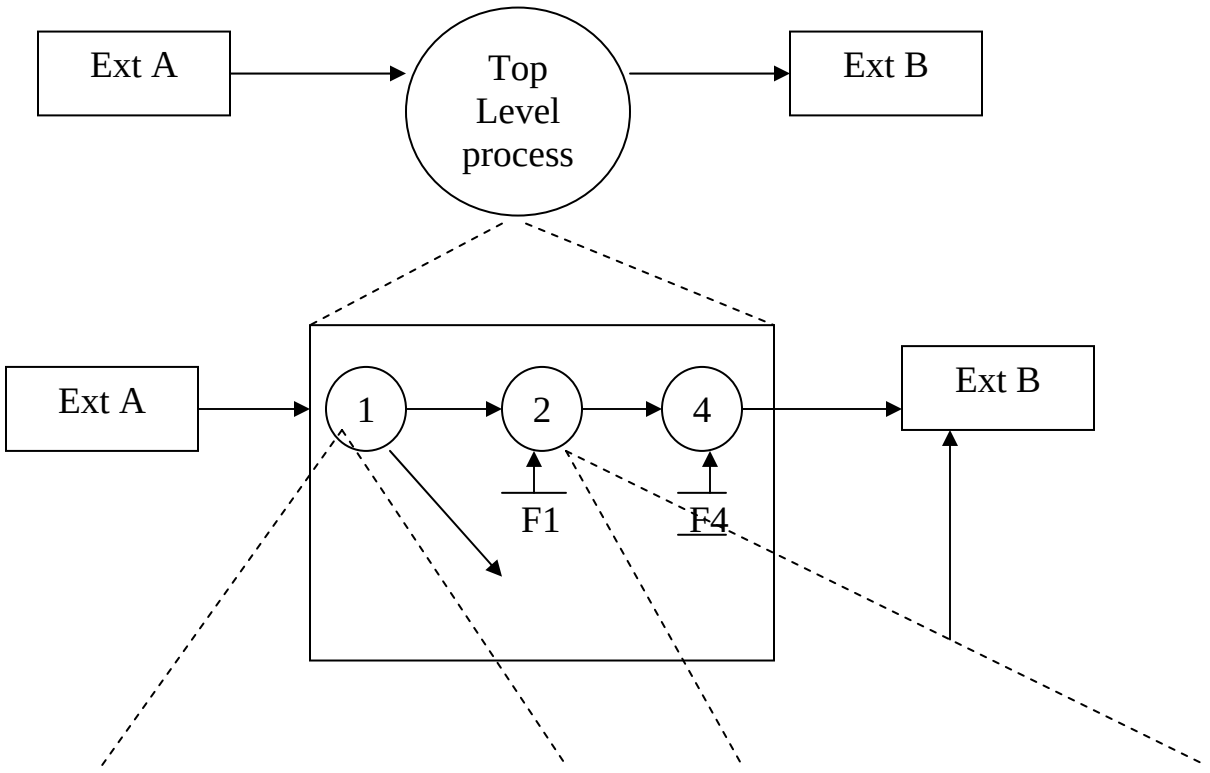


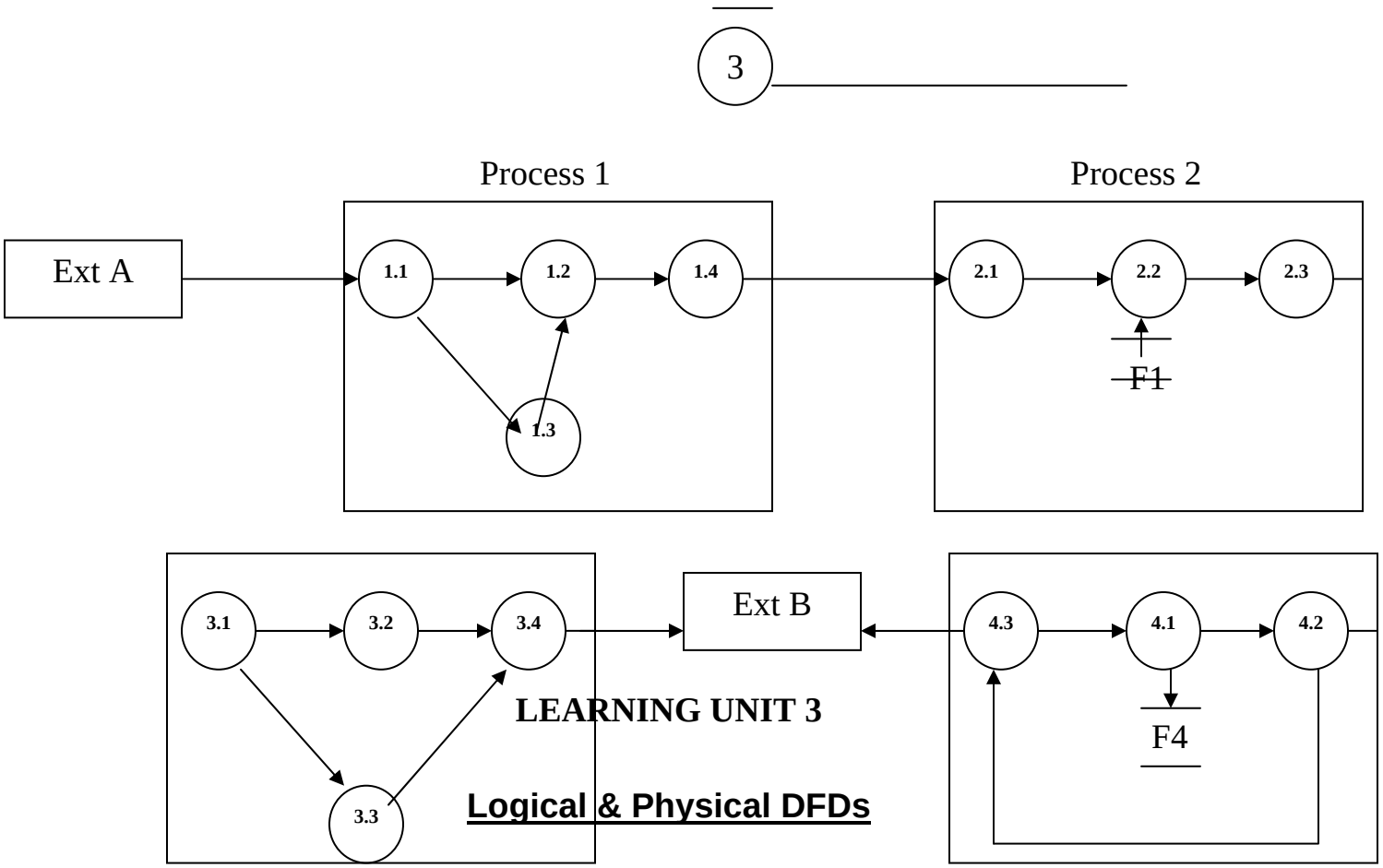
- A single data flow should not be split into many flows with different labels
- No data flow allowed between data stores



- Above DFD not correct as loop is formed

LEVELLING EXAMPLES

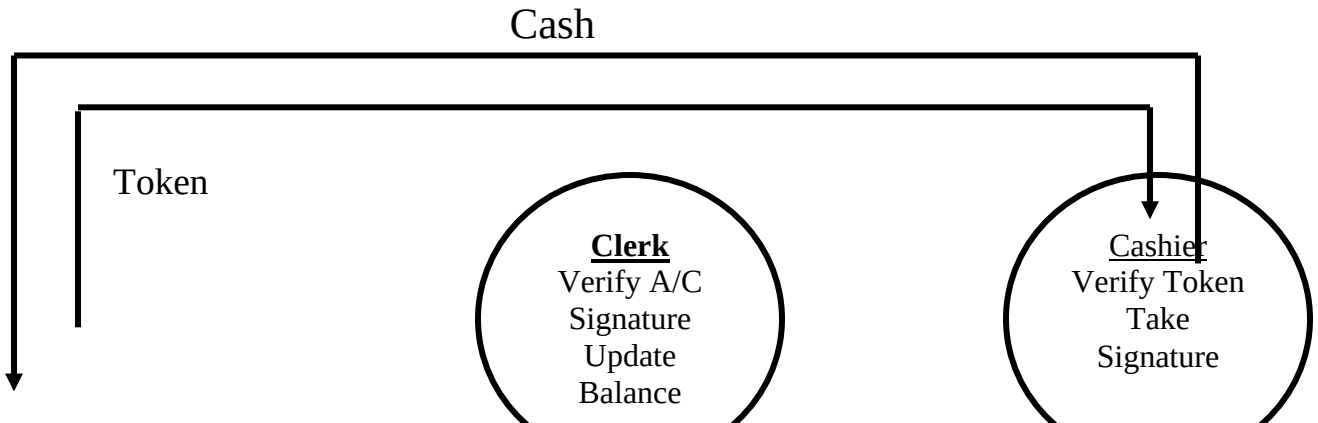


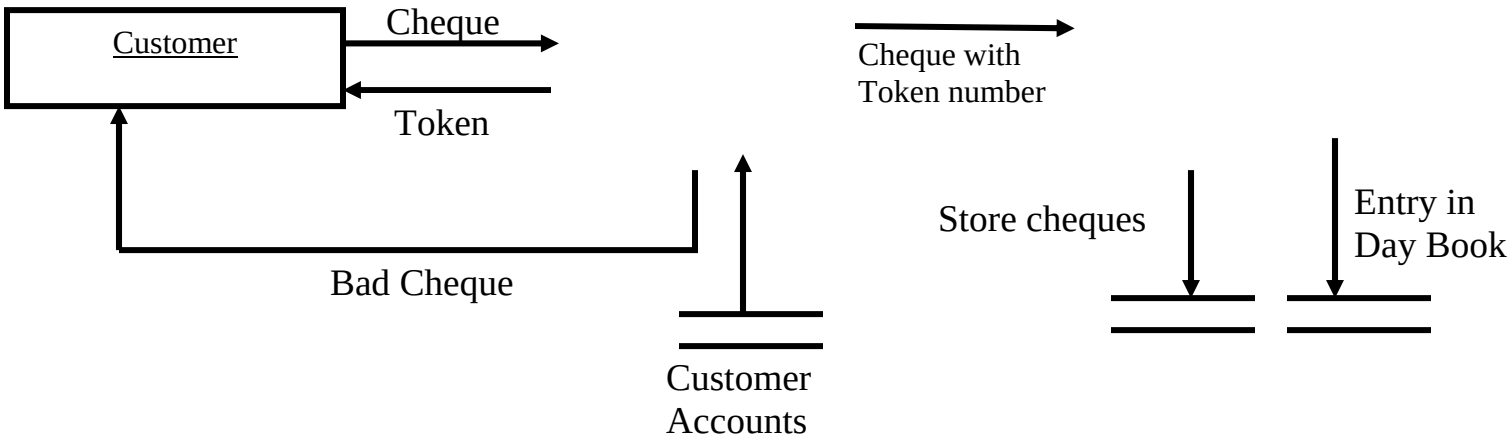


LOGICAL AND PHYSICAL DFD

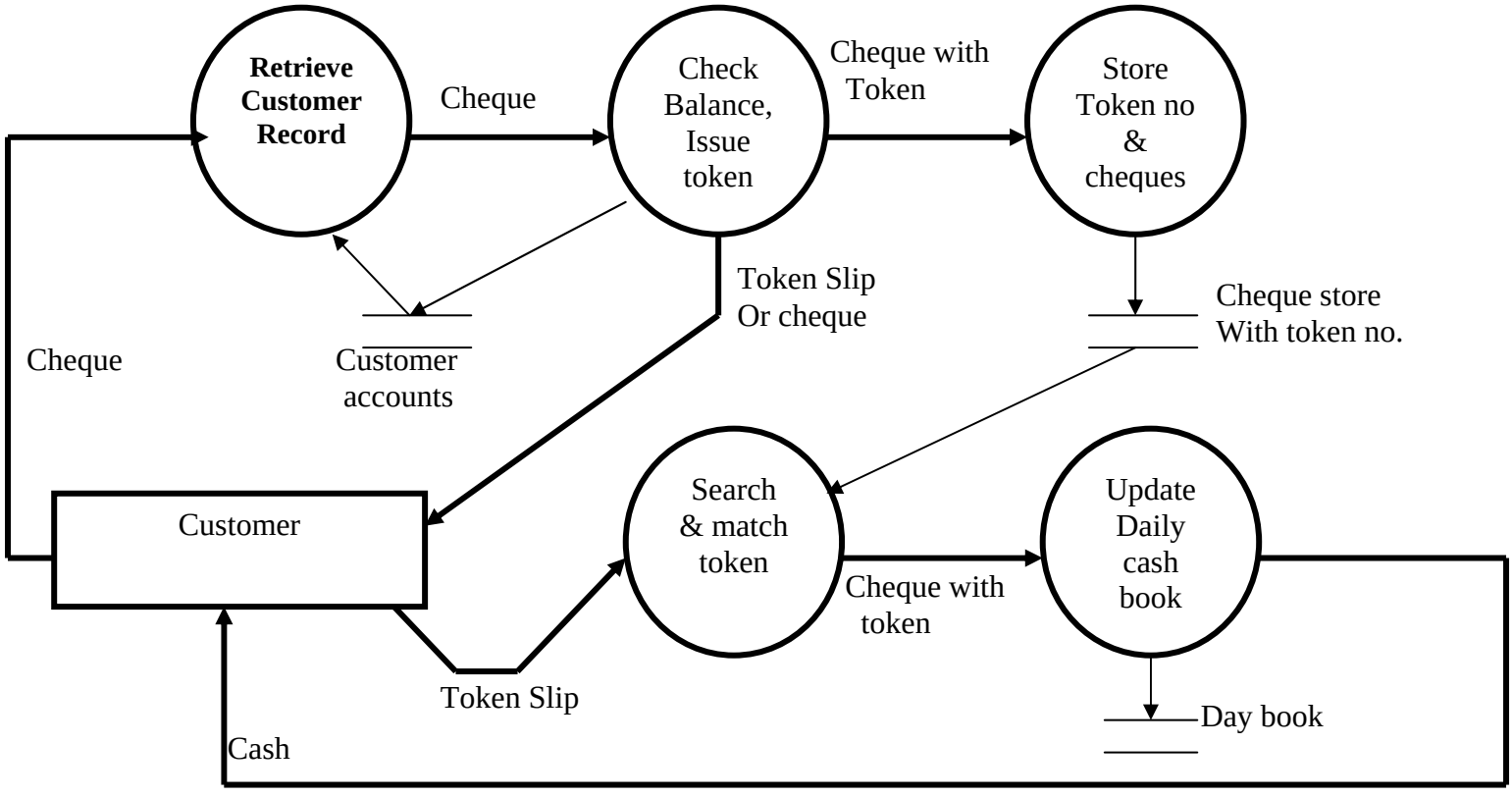
DFD'S considered so far are called logical DFDs.
A physical DFD is similar to a document flow diagram. It specifies who does the operations specified by the logical DFD. Physical DFD may depict physical movements of the goods. Physical DFDs can be drawn during fact gathering phase of a life cycle.

PHYSICAL DFD FOR ENCASHING CHEQUE





LOGICAL DFD FOR CHEQUE ENCASHMENT



REFERENCES

1. This module is based on Chapter 7 of Analysis and Design of information systems, V.Rajaraman, Prentice Hall of India, New Delhi, 2002.

2. Ian Sommerville, "Software Engineering", 5th Edition, Addison-Wesley, 1996, has a brief discussion of Data Flow Models on pp.101 to 103.

3. T.DeMarco, "Structured Analysis and System Specification", Yourdon Press, 1978. this book written by the original developer of DFD modeling is a well written book. It is a good reference book.

4. E.Yourdon, "Modern Structured Analysis", Prentice Hall of India, New Delhi, 1996. Chapter 9 (pp.139 to 187) is a good treatment of Data Flow Diagrams. All the topics covered in this module are discussed in this chapter.

5. Hoffer, J.A., George, J.F. and Valacich J.S., "Modern Systems Analysis and Design", 3rd Edition, Pearson Education Asia, New Delhi, 2002. Chapter 8 (pp.241 to 271) has a good treatment of

DFDs with a running example of a quick service restaurant. Different types of DFDs and logical checking of DFDs are discussed well.

6. K.E.Kendall and J.E.Kendall, "Systems Analysis and Design", 5th Edition, Pearson Eduation Asia, New Delhi, 2003. Chapter 9 is devoted to Data Flow Diagrams (pp.241 to 285). Has a number of problems at the end of the chapter which are quite interesting.