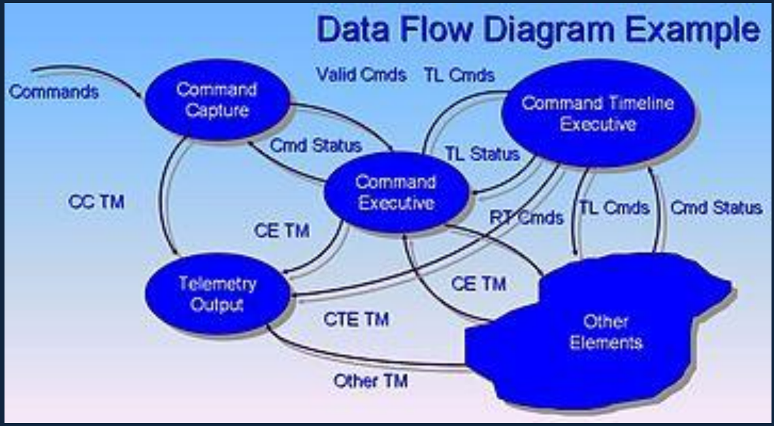
A Quick DFD Tutorial

05: Requirements Engineering

DISCS
SY 2013-2014

DFD

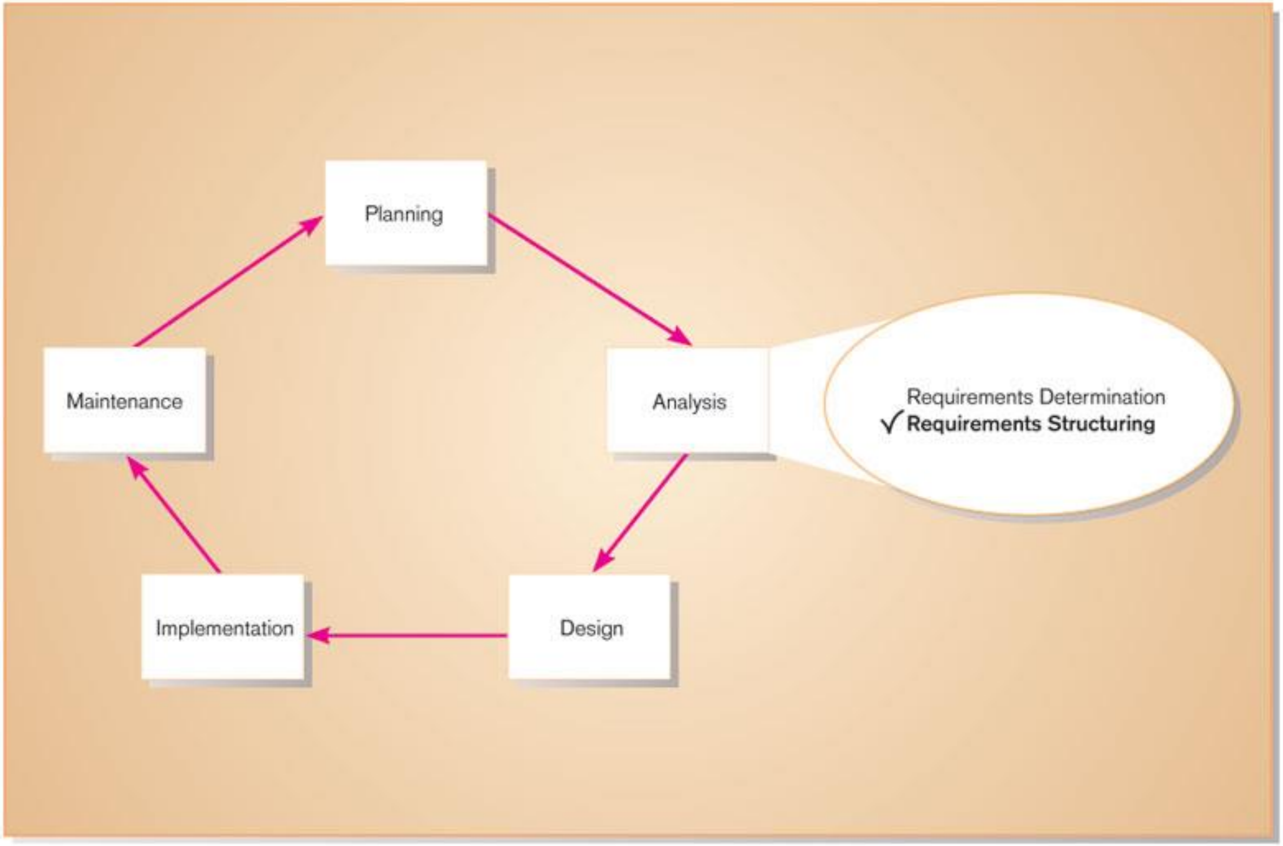
- Data Flow Diagram
- Data-centric : focus is on how data flows within a system
  - Specifies what data is available to a component
  - Specifies what data a component produces, if any
- Essential during pre-OO period, but still sometimes used today
  - Helps in understanding what a component is supposed to do (like input-output of an algorithm)



http://upload.wikimedia.org/wikipedia/commons/thumb/0/0f/Data_Flow_Diagram_Example.jpg/360px-Data_Flow_Diagram_Example.jpg

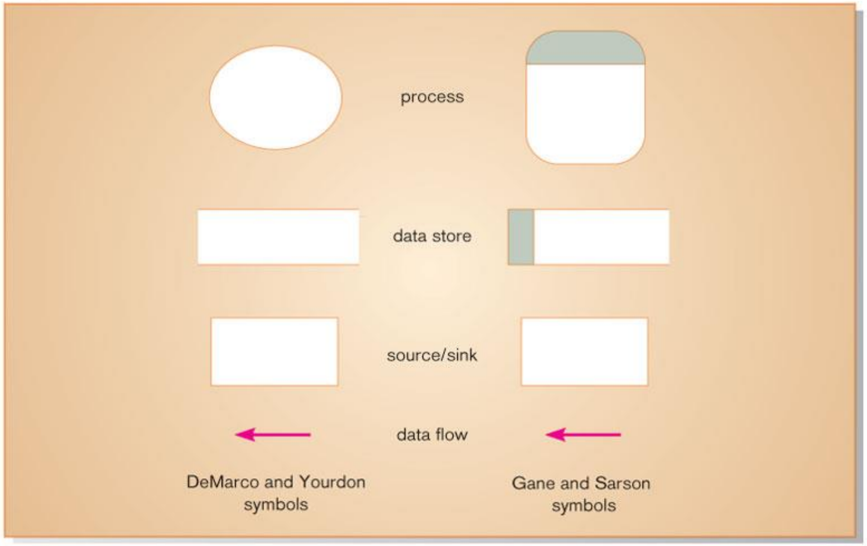
DFDs in Software Life Cycle

Figure 7-1 Systems development life cycle with the analysis phase highlighted



Various Symbols are used

Figure 7-2 Comparison of DeMarco and Yourdon and Gane and Sarson DFD symbol sets



Process: work or actions performed on data (inside the system)

Data flow: arrows depicting movement of data

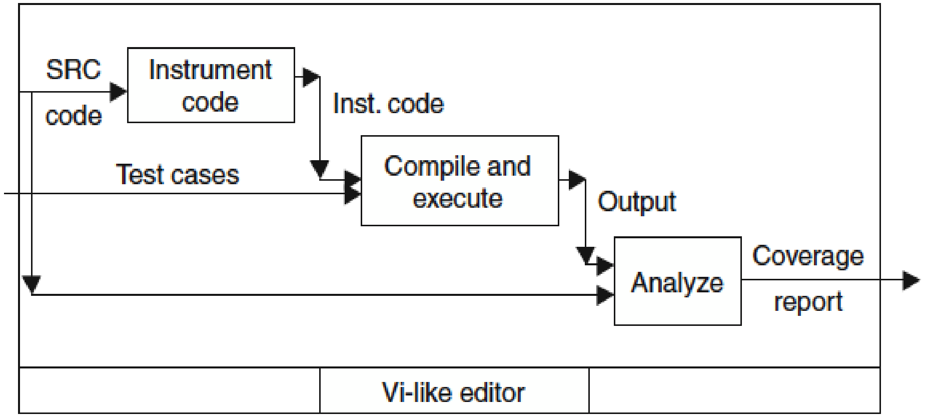
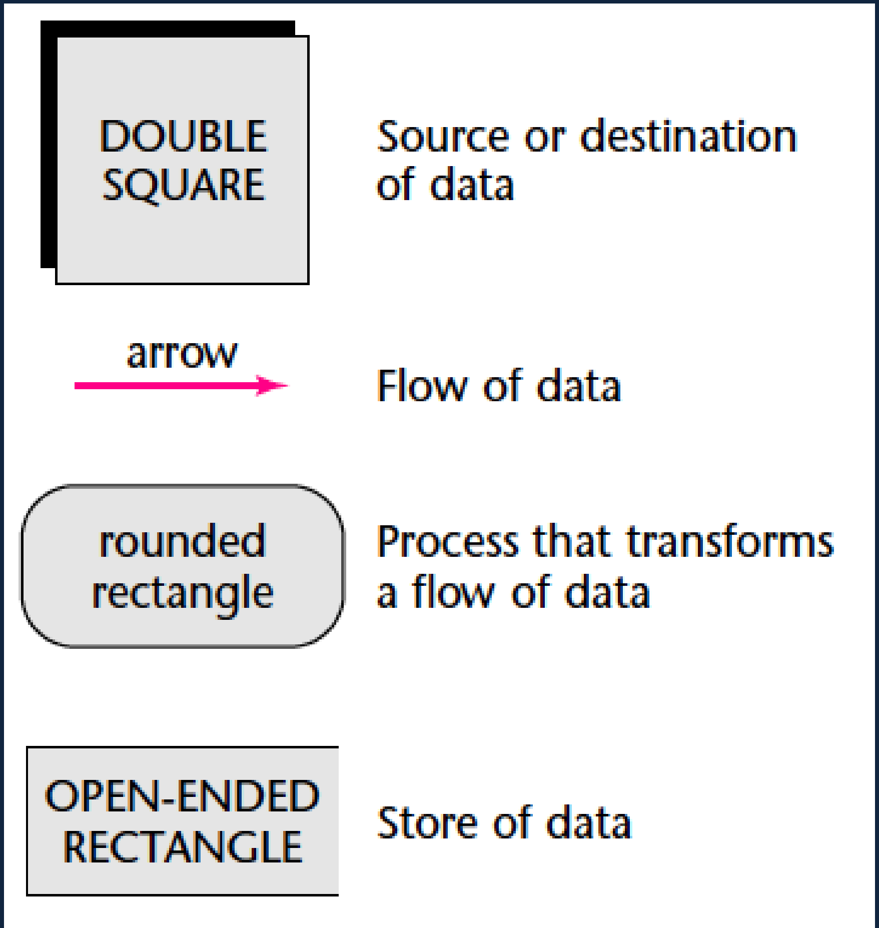


Fig. 8-3. Data flow diagram for Unix, vi-like editor.

Data store: data at rest (inside the system)

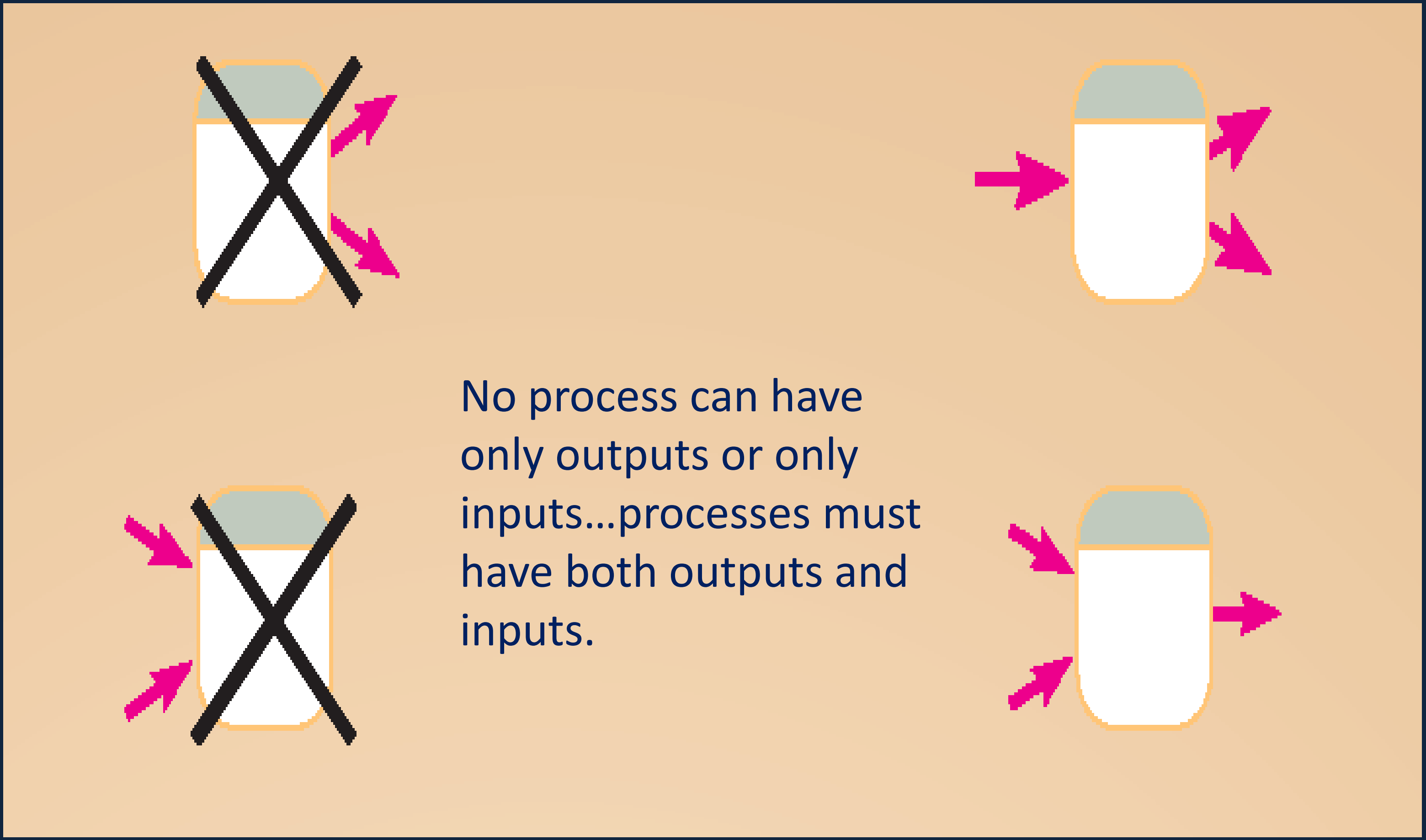
Source/sink: external entity that is origin or destination of data (outside the system)

Various Symbols are used



DOUBLE SQUARE	Source or destination of data
arrow	Flow of data
rounded rectangle	Process that transforms a flow of data
OPEN-ENDED RECTANGLE	Store of data

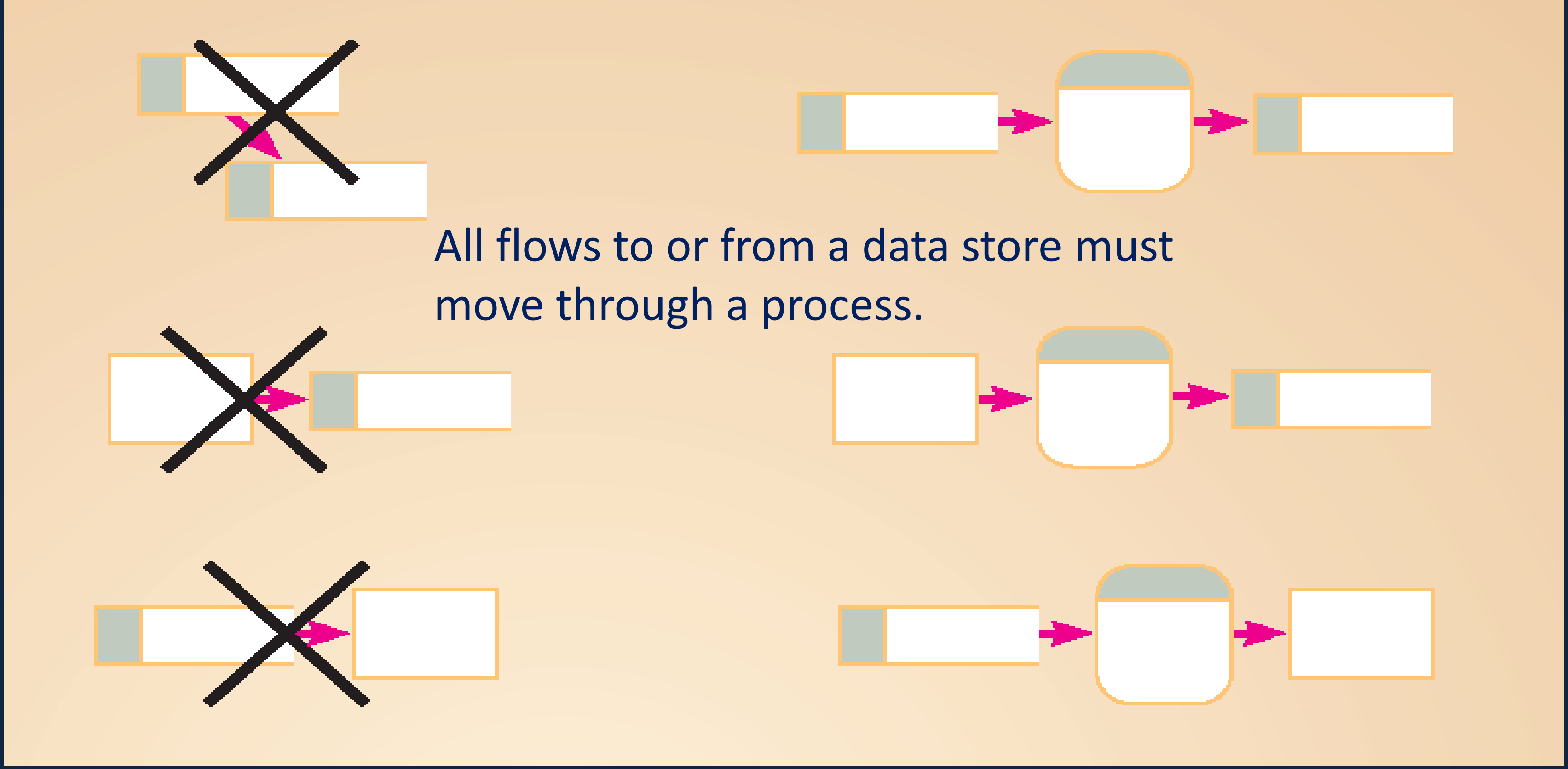
DFD Rules (Process)



No process can have only outputs or only inputs…processes must have both outputs and inputs.

Process labels should be verb phrases.

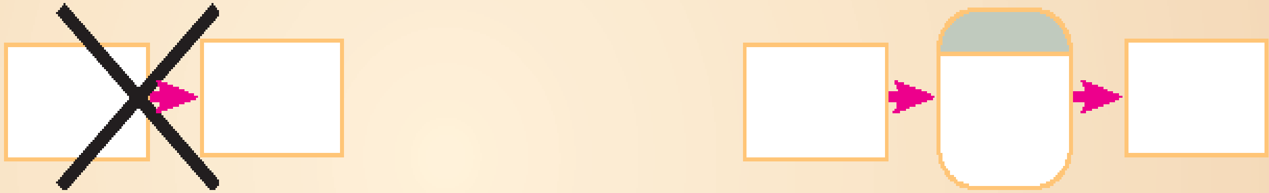
DFD Rules (Data Store)



All flows to or from a data store must move through a process.

Data store labels should be noun phrases.

DFD Rules (Source/Sink)

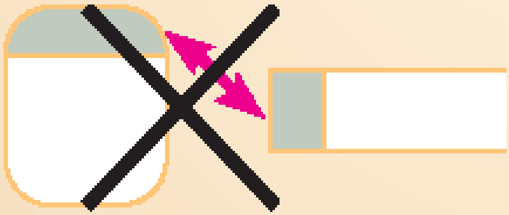


No data moves directly between external entities without going through a process.

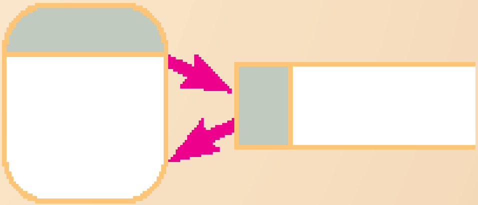
Interactions between external entities without intervening processes are outside the system and therefore not represented in the DFD.

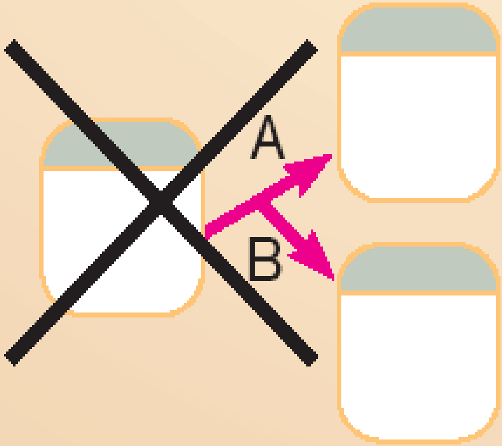
Source and sink labels should be noun phrases.

DFD Rules (Data Flow)

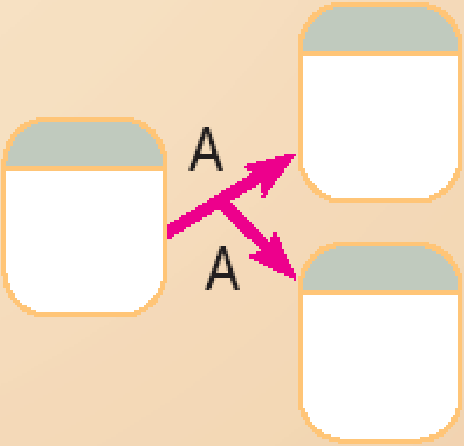


Bidirectional flow between process and data store is represented by two separate arrows.

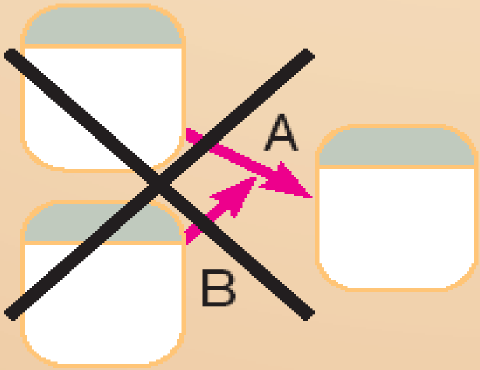




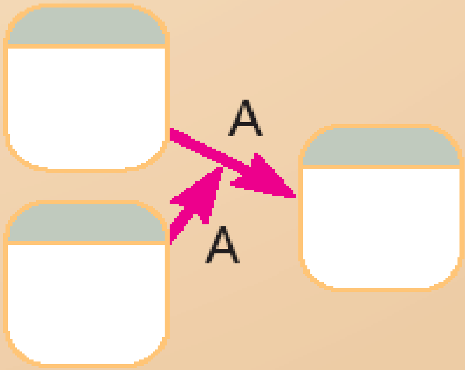
Forked data flow must refer to exact same data item (not different data items) from a common location to multiple destinations.

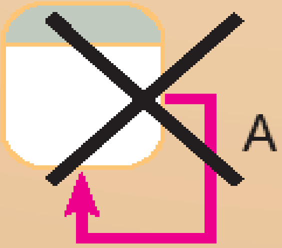


DFD Rules (Data Flow)

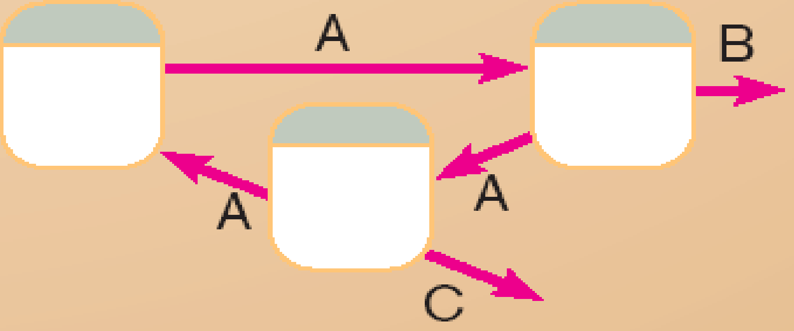


Joined data flow must refer to exact same data item (not different data items) from multiple sources to a common location.





Data flow cannot go directly from a process to itself, must go through intervening processes.



DFD Rules (Data Flow)

- Data flow from a process to a data store means update (insert, delete or change).
- Data flow from a data store to a process means retrieve or use.
- Data flow labels should be noun phrases.

Functional Decomposition

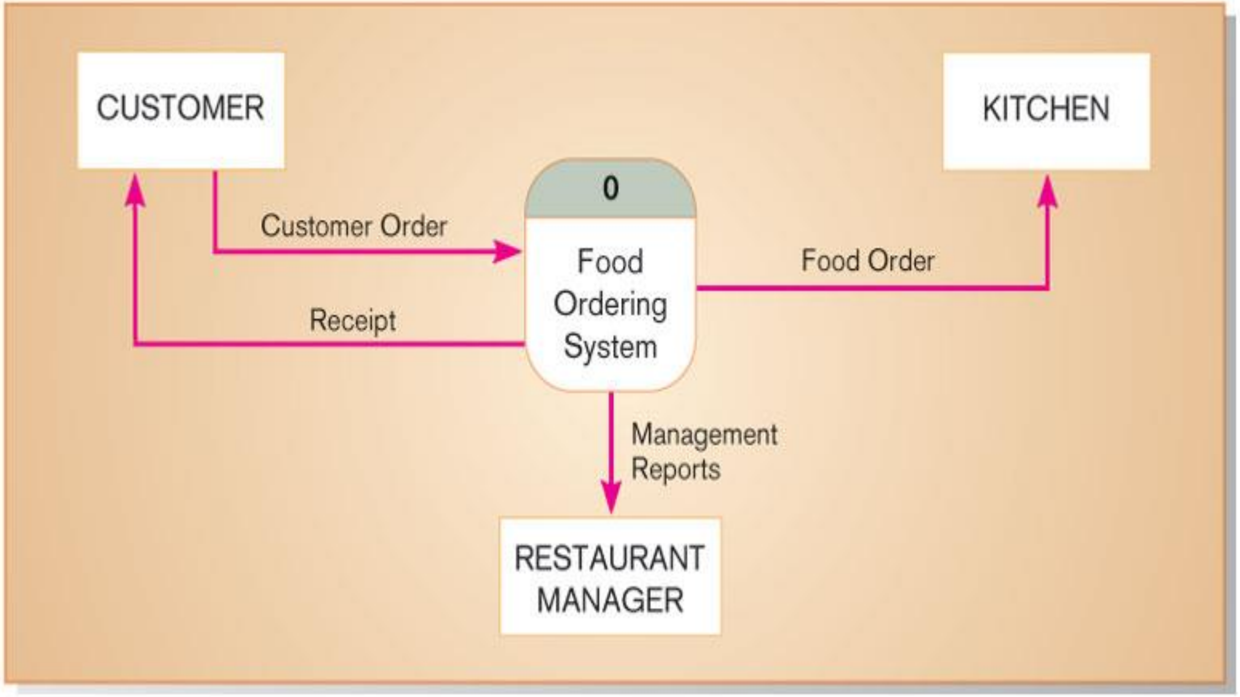
- An iterative process of breaking a system description down into finer and finer detail
- High-level processes described in terms of lower-level sub-processes
- DFD charts created for each level of detail

DFD Levels

- Context DFD
  - Overview of the organizational system
- Level-0 DFD
  - Representation of system’s major processes at high level of abstraction
- Level-1 DFD
  - Results from decomposition of Level 0 diagram
- Level-*n* DFD
  - Results from decomposition of Level *n*-1 diagram

Context Diagram

Figure 7-4 Context diagram of Hoosier Burger's food ordering system

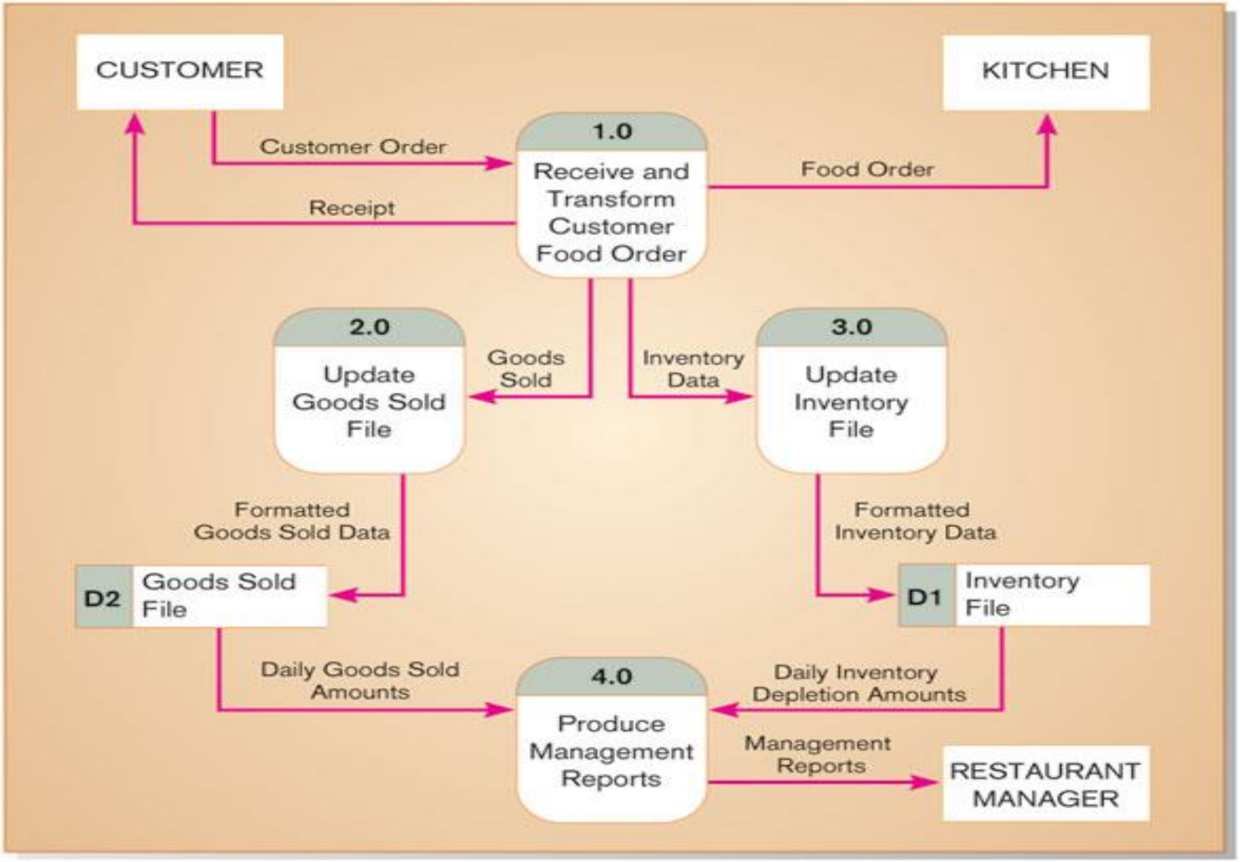


Context diagram shows the system boundaries, external entities that interact with the system, and major information flows between entities and the system.

NOTE: only one process symbol, and no data stores shown.

Level-0 DFD

Figure 7-5 Level-0 DFD of Hoosier Burger's food ordering system

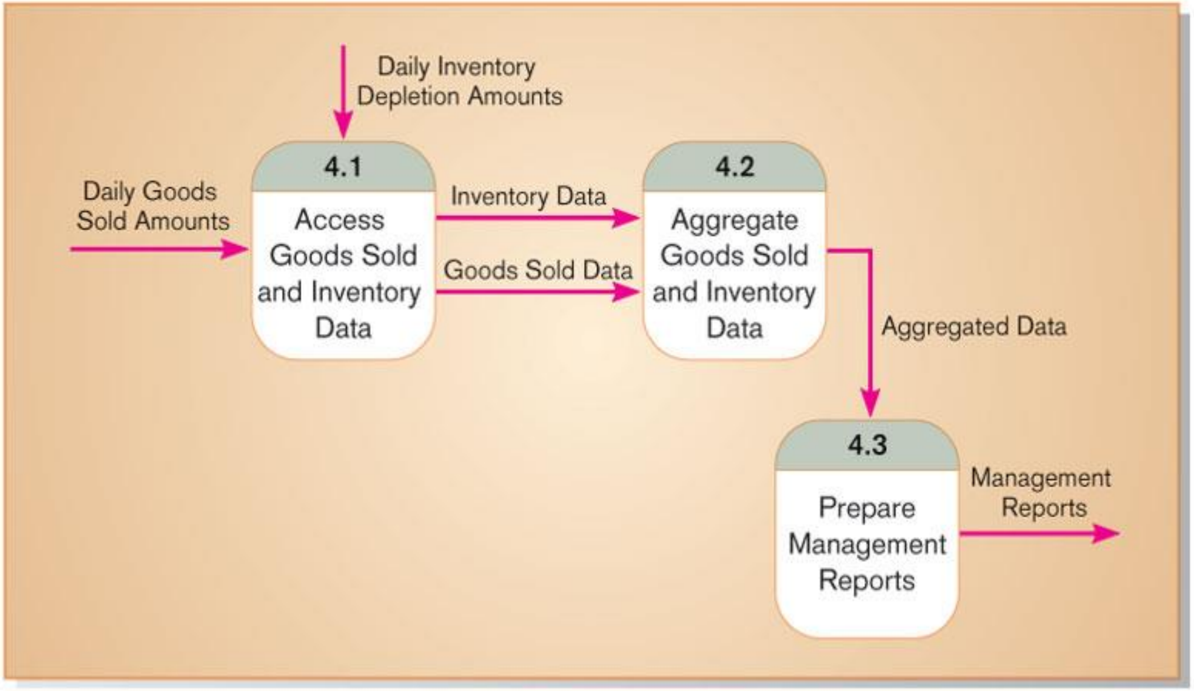


Level-0 DFD shows the system’s major processes, data flows, and data stores at a high level of abstraction.

Processes are labeled 1.0, 2.0, etc. These will be decomposed into more primitive (lower-level) DFDs.

Level-1 DFD

Figure 7-8 Level-1 diagram showing the decomposition of Process 4.0 from the level-0 diagram for Hoosier Burger's food ordering system


Daily Inventory Depletion Amounts
Daily Goods Sold Amounts
4.1
Access Goods Sold and Inventory Data
Inventory Data
Goods Sold Data
4.2
Aggregate Goods Sold and Inventory Data
Aggregated Data
4.3
Prepare Management Reports
Management Reports

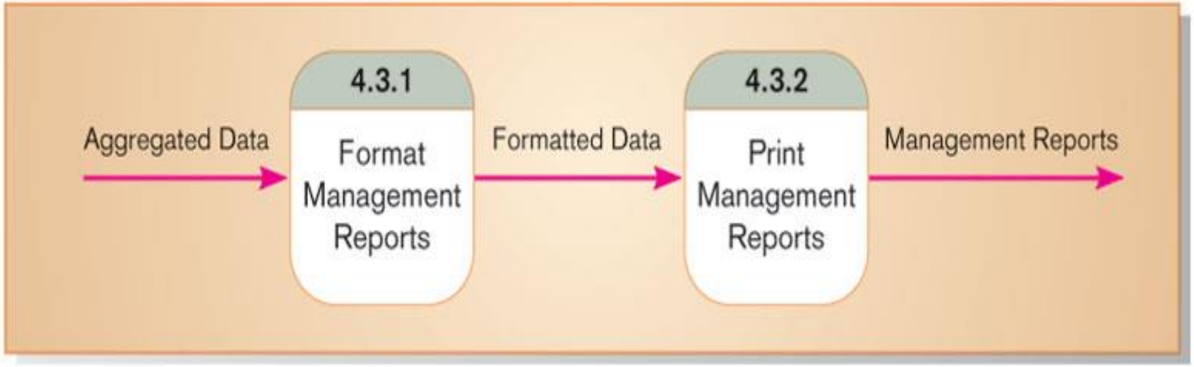
Level-1 DFD shows the sub-processes of one of the processes in the Level-0 DFD.

This is a Level-1 DFD for Process 4.0.

Processes are labeled 4.1, 4.2, etc. These can be further decomposed in more primitive (lower-level) DFDs if necessary.

Level-*n* DFD

Figure 7-9 Level-2 diagram showing the decomposition of Process 4.3 from the level-1 diagram for Process 4.0 for Hoosier Burger's food ordering system



Level-*n* DFD shows the sub-processes of one of the processes in the Level *n-1* DFD.

This is a Level-2 DFD for Process 4.3.

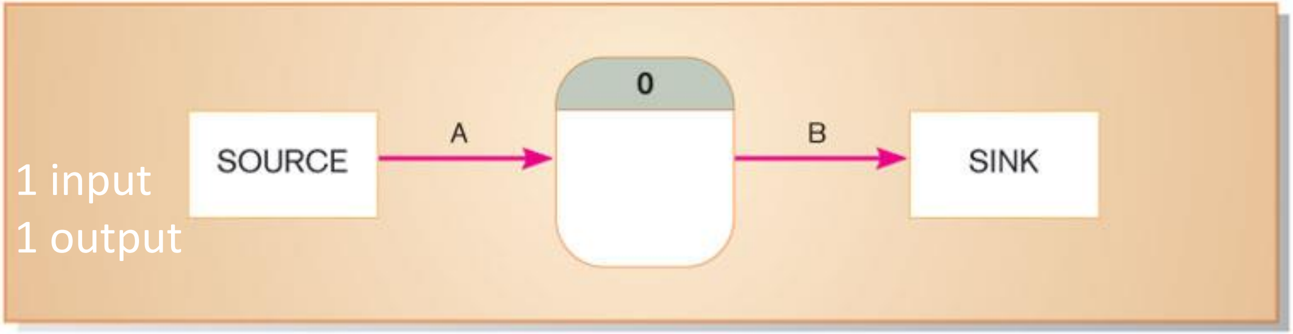
Processes are labeled 4.3.1, 4.3.2, etc. If this is the lowest level of the hierarchy, it is called a *primitive DFD.*

DFD Balancing

- The conservation of inputs and outputs to a data flow process when that process is decomposed to a lower level
- Balanced means:
  - Number of inputs to lower level DFD equals number of inputs to associated process of higher-level DFD
  - Number of outputs to lower level DFD equals number of outputs to associated process of higher-level DFD

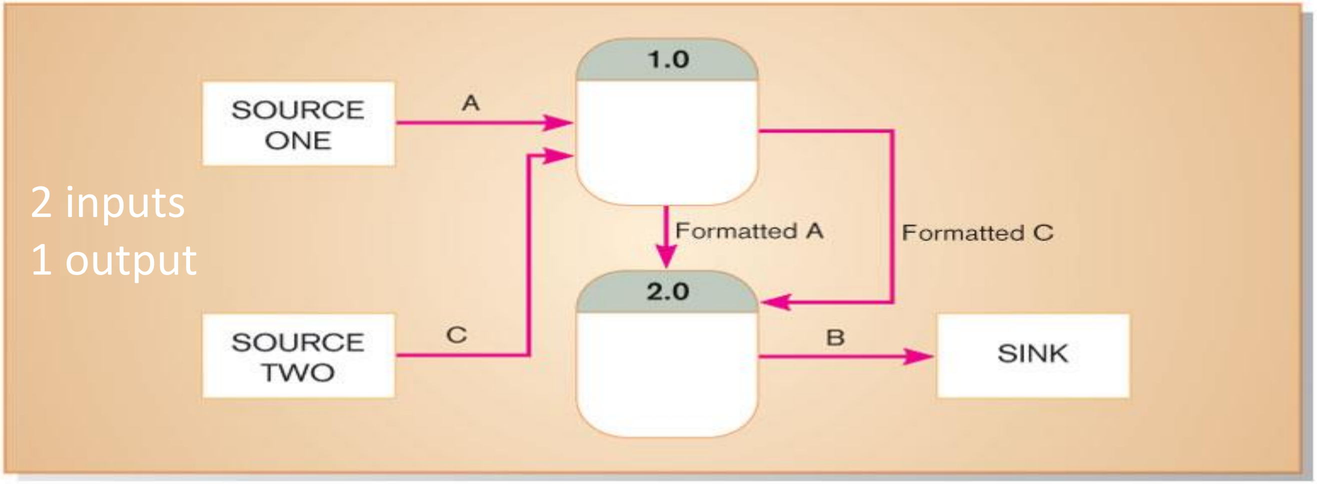
Unbalanced DFD

Figure 7-10a An unbalanced set of data flow diagrams - Context diagram



1 input
1 output

Figure 7-10b An unbalanced set of data flow diagrams - Level-0 diagram

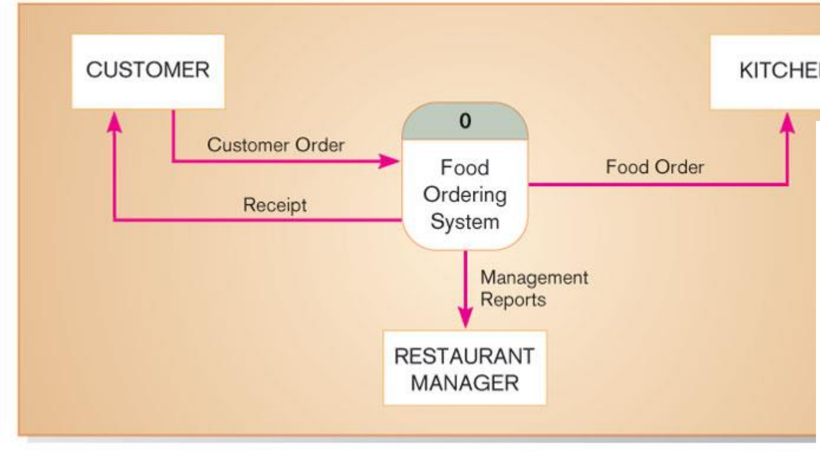


2 inputs
1 output

This is unbalanced because the process of the context diagram has only one input but the Level-0 diagram has two inputs.

Balanced DFD

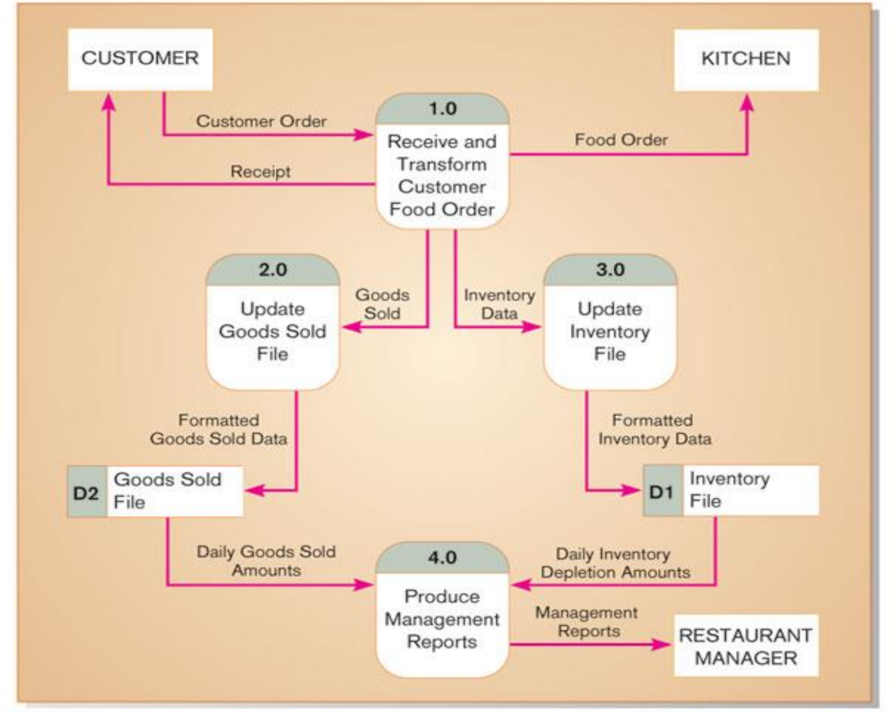
Figure 7-4 Context diagram of Hoosier Burger's food ordering system



1 input
3 outputs

These are balanced because the numbers of inputs and outputs of context diagram process equal the number of inputs and outputs of Level-0 diagram.

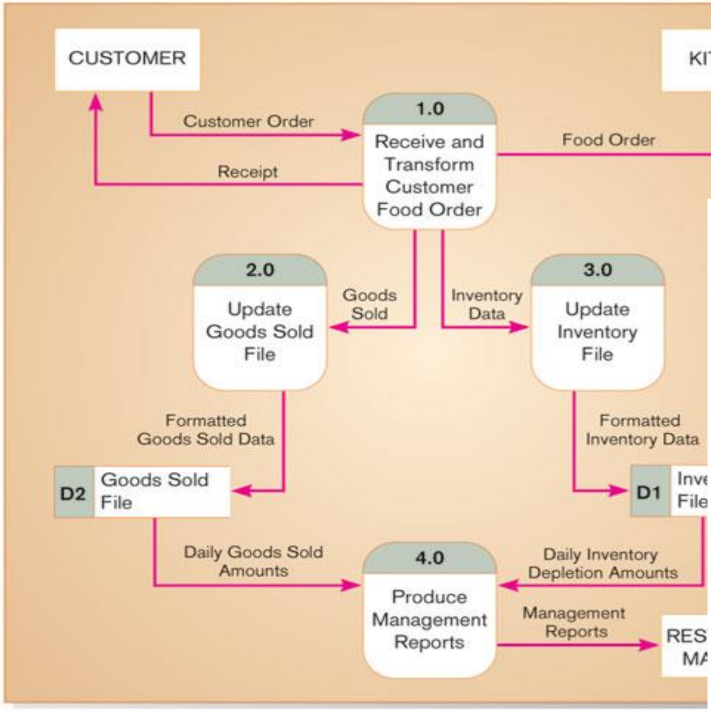
Figure 7-5 Level-0 DFD of Hoosier Burger's food ordering system



Balanced DFD (cont.)

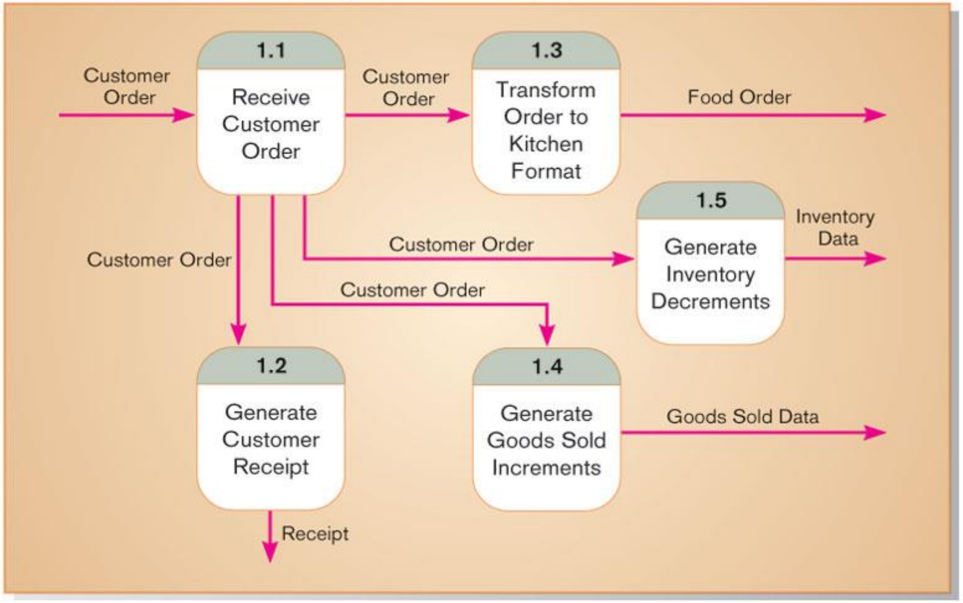
These are balanced because the numbers of inputs and outputs to Process 1.0 of the Level-0 diagram equals the number of inputs and outputs to the Level-1 diagram.

Figure 7-5 Level-0 DFD of Hoosier Burger's food ordering system



1 input
4 outputs

Figure 7-7 Level-1 diagram showing the decomposition of Process 1.0 from the level-0 diagram for Hoosier Burger's food ordering system



Example #1a

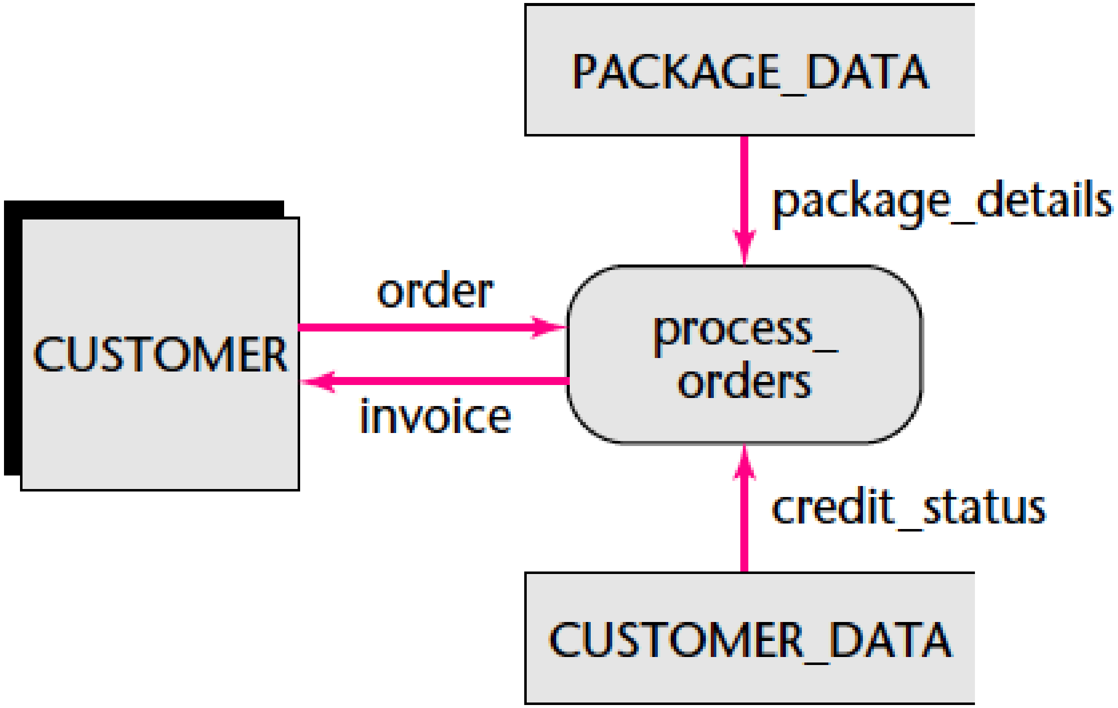
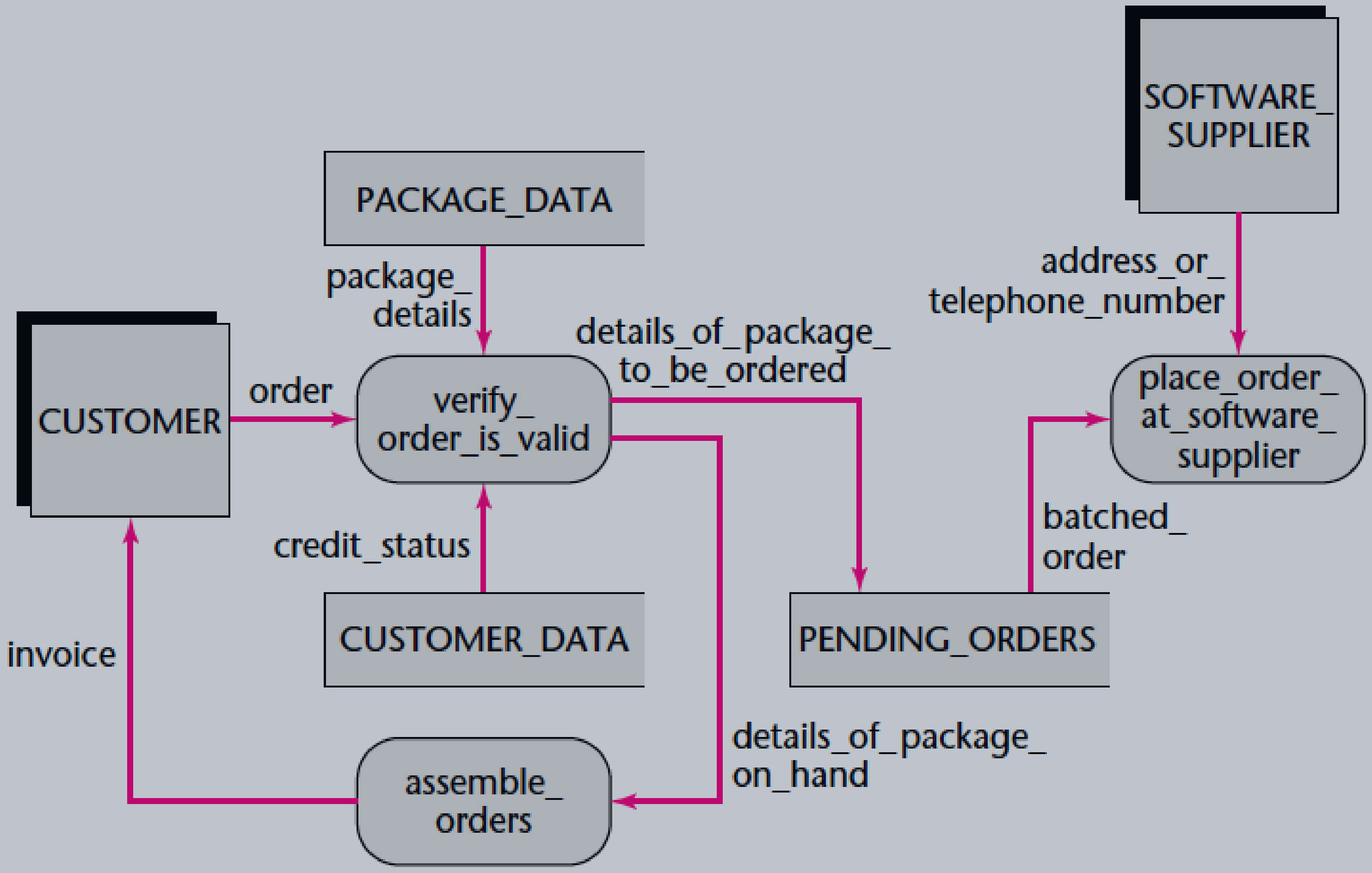


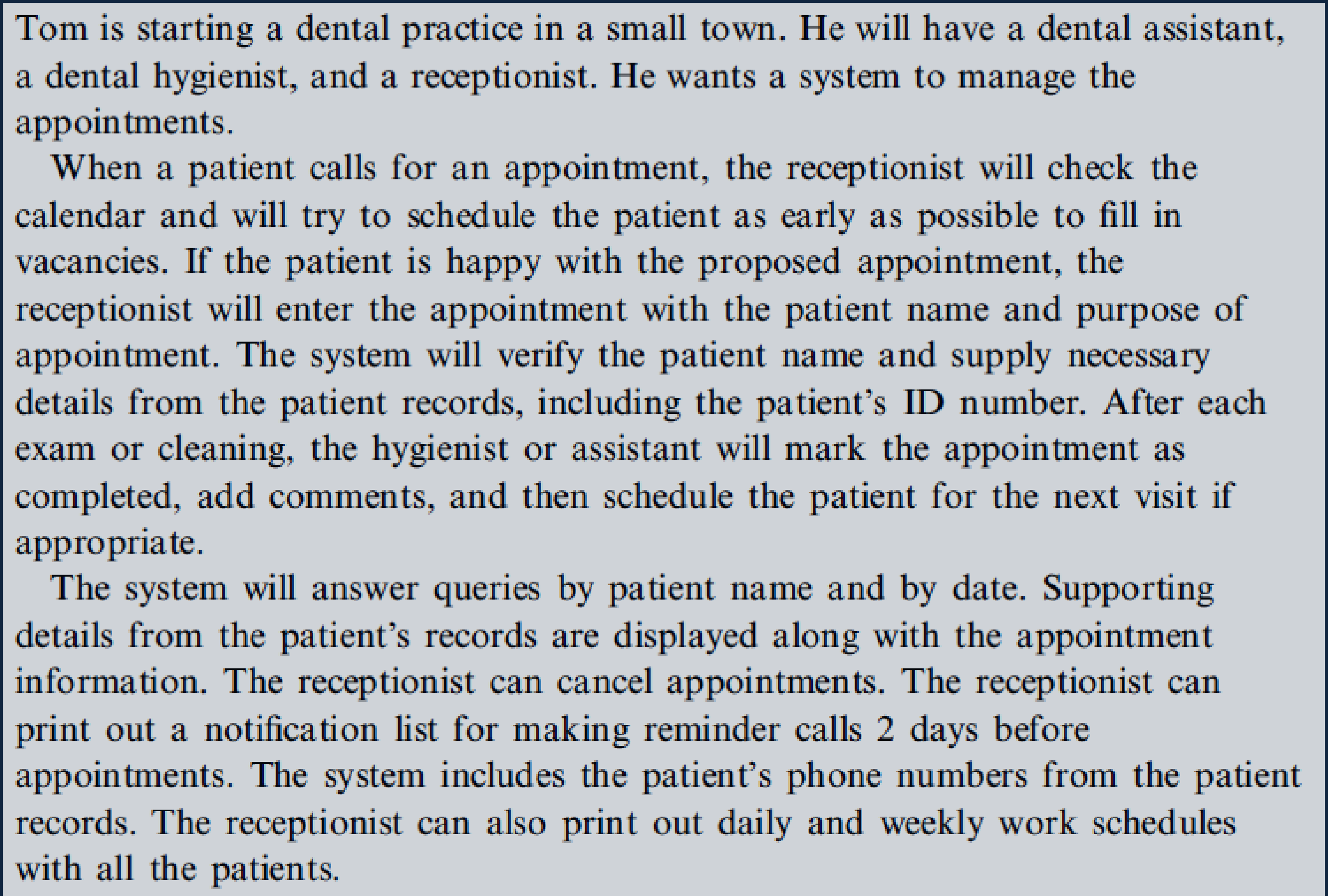
FIGURE 12.2
The data flow diagram for Sally's Software Shop: first refinement.

Example #1b

FIGURE 12.3 The data flow diagram for Sally's Software Shop: second refinement.



Practice #1 – Dental Office

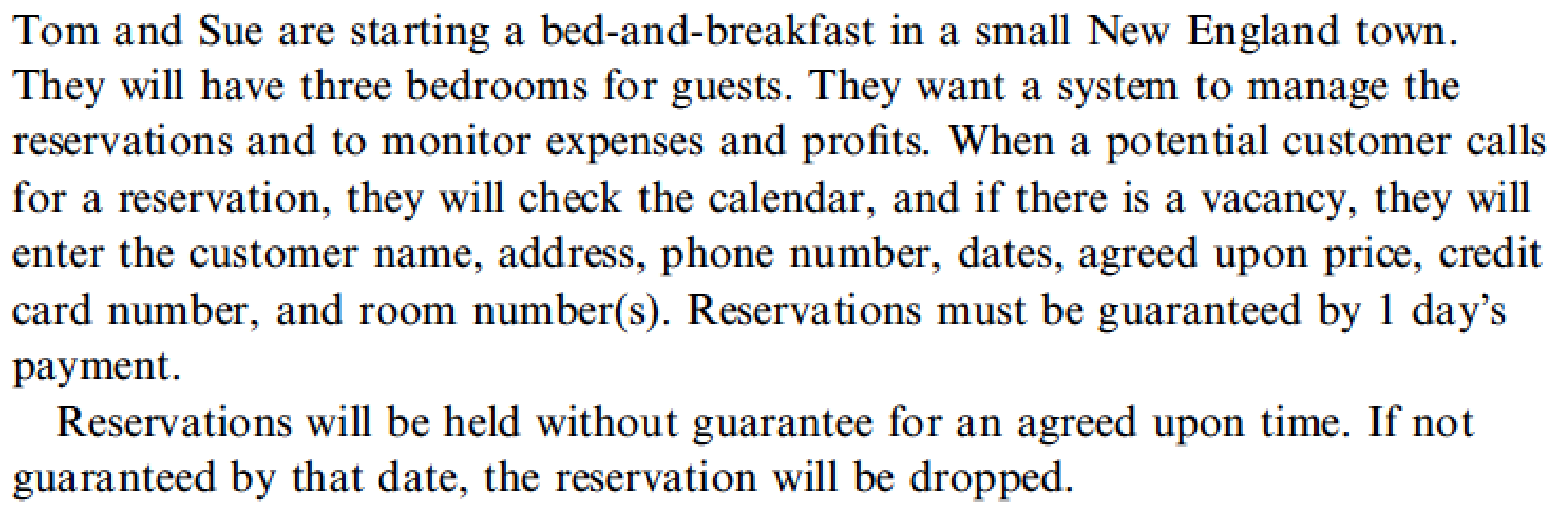


Tom is starting a dental practice in a small town. He will have a dental assistant, a dental hygienist, and a receptionist. He wants a system to manage the appointments.

When a patient calls for an appointment, the receptionist will check the calendar and will try to schedule the patient as early as possible to fill in vacancies. If the patient is happy with the proposed appointment, the receptionist will enter the appointment with the patient name and purpose of appointment. The system will verify the patient name and supply necessary details from the patient records, including the patient's ID number. After each exam or cleaning, the hygienist or assistant will mark the appointment as completed, add comments, and then schedule the patient for the next visit if appropriate.

The system will answer queries by patient name and by date. Supporting details from the patient's records are displayed along with the appointment information. The receptionist can cancel appointments. The receptionist can print out a notification list for making reminder calls 2 days before appointments. The system includes the patient's phone numbers from the patient records. The receptionist can also print out daily and weekly work schedules with all the patients.

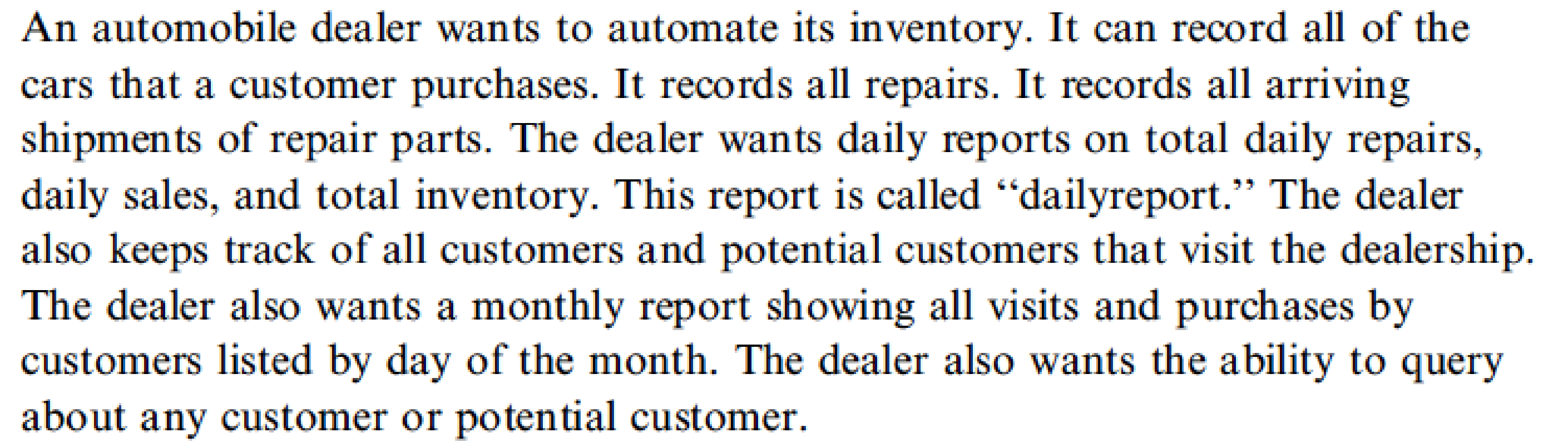
Practice #2 – B&B



Tom and Sue are starting a bed-and-breakfast in a small New England town. They will have three bedrooms for guests. They want a system to manage the reservations and to monitor expenses and profits. When a potential customer calls for a reservation, they will check the calendar, and if there is a vacancy, they will enter the customer name, address, phone number, dates, agreed upon price, credit card number, and room number(s). Reservations must be guaranteed by 1 day's payment.

Reservations will be held without guarantee for an agreed upon time. If not guaranteed by that date, the reservation will be dropped.

Practice #3 – Automobile Dealership



An automobile dealer wants to automate its inventory. It can record all of the cars that a customer purchases. It records all repairs. It records all arriving shipments of repair parts. The dealer wants daily reports on total daily repairs, daily sales, and total inventory. This report is called "dailyreport." The dealer also keeps track of all customers and potential customers that visit the dealership. The dealer also wants a monthly report showing all visits and purchases by customers listed by day of the month. The dealer also wants the ability to query about any customer or potential customer.

References

- DFD Rules : J. Hoffer, J. George, J. Valacich, *Modern Systems Analysis and Design (4th Ed)*, Prentice Hall, 2005
- Examples : S. Schach, *Object Oriented and Classical Software Engineering (8th Ed)*, McGraw-Hill, 2011
- Practice Problems : D. Gustafson, *Schaum's Outline of Theory and Problems of Software Engineering*, McGraw-Hill, 2002