King Saud University

College of Engineering

IE – 462: "Industrial Information Systems"

Fall – 2024 (1st Sem. 1446H)

Chapter 4:

Structured Analysis and Functional Architecture Design – p2 – DFD – iii – Case Studies

Prepared by: Ahmed M. El-Sherbeeny, PhD

Lesson Overview

- Modeling IIS – (p1)
- Integrated Computer-Aided Manufacturing Definition 0 (IDEF0) – (p1)
- **Data Flow Diagram (DFD) – (p2)**
  i. Fundamentals
  ii. Diagramming Rules
  iii. Case Studies

DFD – part iii – Case Studies

1. Inventory Control System
2. Business Process Reengineering (BPR)
3. Electronic Commerce Application
4. Converting IDEF0 Model to DFD

Functional/Process Modeling:

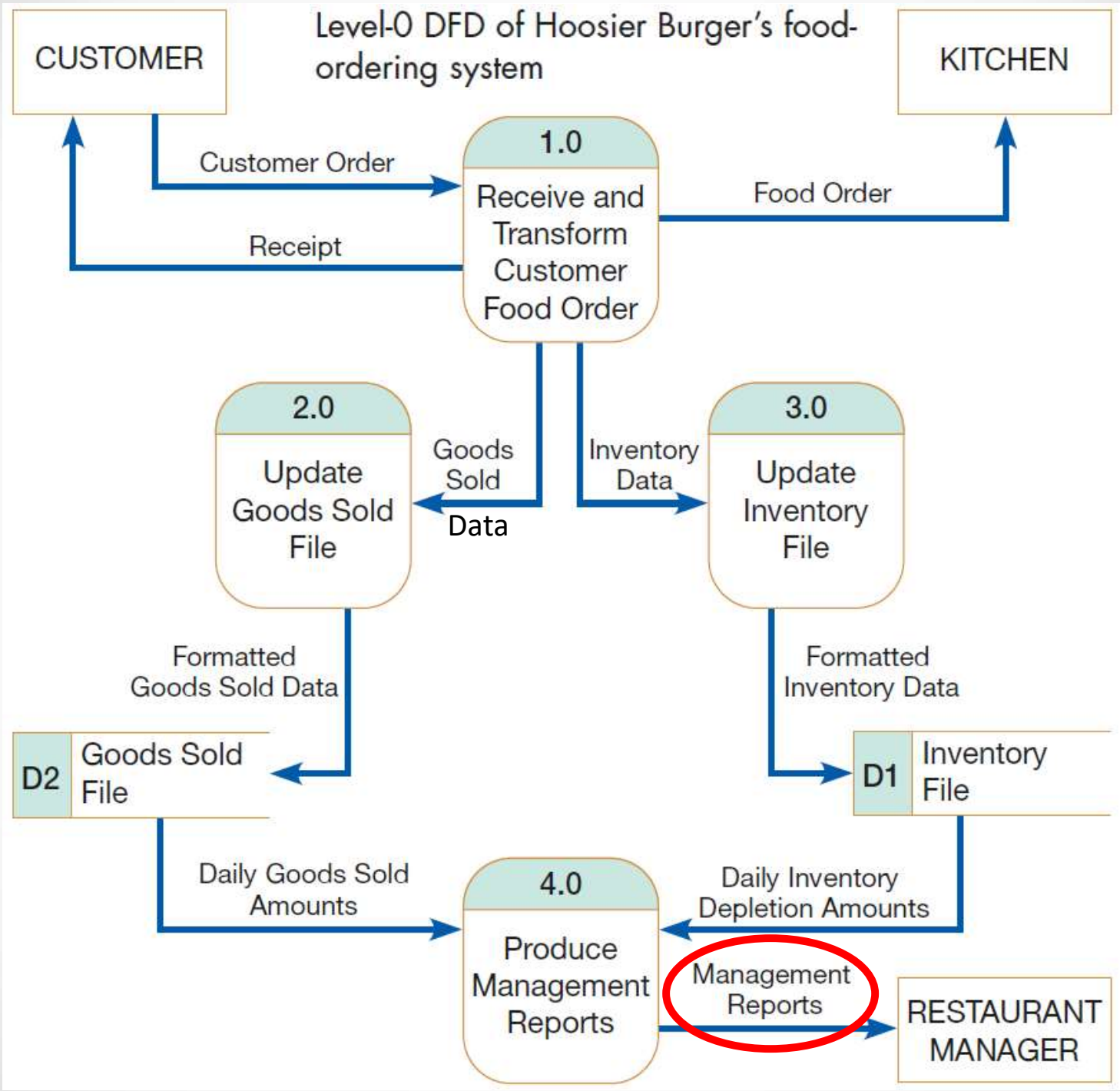
2. Data Flow Diagram (DFD) – cont'd

Case Study 1 – Using DFD in Inventory Control System – "Hoosier Burger"

Case Study 1: HB Inventory Control System

Introduction:

- Remember: Hoosier Burger food-ordering system generates two types of usage data,
  - goods sold and
  - inventory
- At end of each day, manager, (Bob) generates inventory report that tells him
  - how much inventory *should* have been used,
  - items associated with each sale
- Bob uses a *manual* inventory control system


Level-0 DFD of Hoosier Burger's food-ordering system
CUSTOMER
KITCHEN
1.0
Receive and Transform Customer Food Order
Customer Order
Receipt
Food Order
2.0
Update Goods Sold File
3.0
Update Inventory File
Goods Sold Data
Inventory Data
Formatted Goods Sold Data
Formatted Inventory Data
D2
Goods Sold File
D1
Inventory File
Daily Goods Sold Amounts
Daily Inventory Depletion Amounts
4.0
Produce Management Reports
Management Reports
RESTAURANT MANAGER

Case Study 1: HB Inventory Control System

FIGURE 7-12
List of activities involved in Bob Mellankamp's inventory control system for Hoosier Burger

1. Meet delivery trucks before opening restaurant.
2. Unload and store deliveries.
3. Log invoices and file in accordion file.
4. Manually add amounts received to stock logs.
5. After closing, print inventory report.
6. Count physical inventory amounts.
7. Compare inventory report totals to physical count totals.
8. Compare physical count totals to minimum order quantities. If the amount is less, make order; if not, do nothing.
9. Pay bills that are due and record them as paid.

Case Study 1: HB Inventory Control System

Bob's Hoosier Burger Inventory System:

- 3 sources of input data (i.e. from outside system):
  1. suppliers
     - provide *invoices* (i.e. system input)
  2. food-ordering system inventory report
     - provide inventory *counts* (i.e. system inputs)
  3. stock on hand
     - also provides inventory *counts* (i.e. system inputs)
- System output:
  - Suppliers: *payments* and *orders*
- We can now create the context diagram shown in Figure 7-14 for the system

Case Study 1: HB Inventory Control System

Bob's Hoosier Burger Inventory System (cont.):

- When Bob receives invoices from suppliers,
  - he records their receipt on an *invoice log sheet*,
  - and files the actual invoices in his *accordion file*

- Using the invoices, Bob records the amount of stock delivered on *stock logs*:
  - these are paper forms posted near the *point of storage* for each inventory item

Case Study 1: HB Inventory Control System

FIGURE 7-13

Hoosier Burger's stock log form

Stock Log					
Date:		Jan 1			Jan 2
Item	*Reorder Quantity*	Starting Amount	Amount Delivered	Amount Used	Starting Amount
Hamburger buns	50 dozen	5	50	43	12
Hot dog buns	25 dozen	0	25	22	3
English muffins	10 dozen	6	10	12	4
Napkins	2 cases	10	0	2	8
Straws	1 case	1	0	1	0

Case Study 1: HB Inventory Control System

Hoosier Burger's Stock Log Form:

- *Minimum order quantities** appear on the log form:
  - stock level at which orders *must be placed* in order to avoid running out of an item
- Stock log also has spaces for entering:
  a) starting amount:
     - entered on the sheet when Bob logs stock deliveries
  b) amount delivered, and
  c) amount used for each item:
     - entered on sheet *after* Bob has compared amounts of stock used according to a *physical count* and according to the numbers on the *inventory report* (generated by the food-ordering system)**

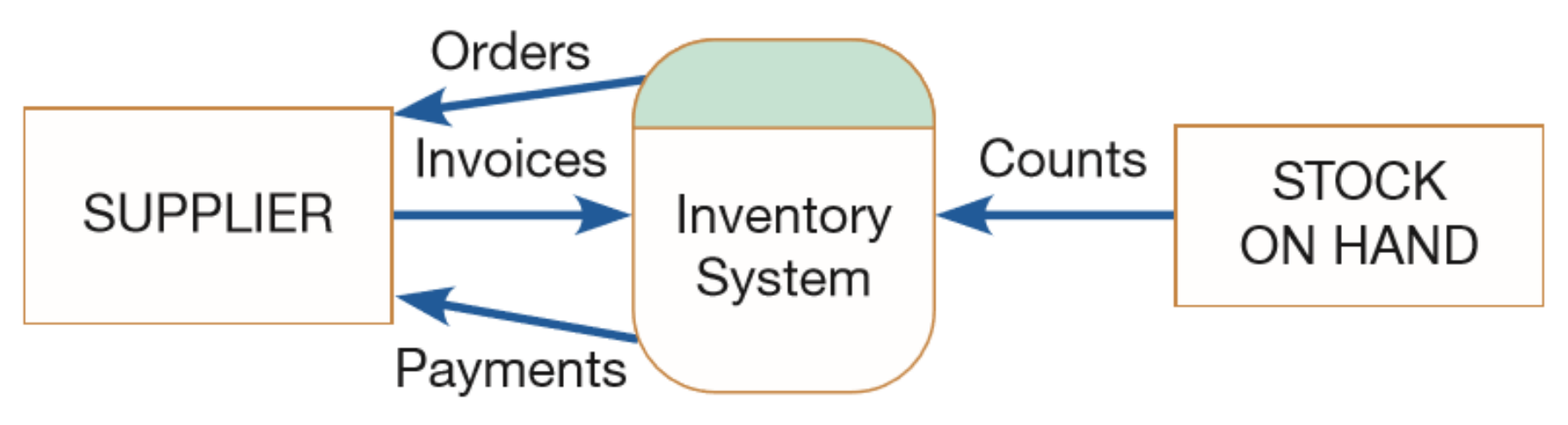
Case Study 1: HB Inventory Control System

Hoosier Burger's Stock Log Form (cont.):

- Hoosier Burger has *standing daily delivery orders*
  - for some perishable items that are used every day (e.g. burger buns, meats, and vegetables)
- Bob determines which orders need to be placed by comparing,
  - minimum order quantities and
  - the amount of stock on hand
- Bob uses the invoices,
  - determines which bills need to be paid, and
  - carefully records each payment

Case Study 1: HB Inventory Control System

FIGURE 7-14
(a) Context diagram for Hoosier Burger's inventory control system



Case Study 1: HB Inventory Control System

Main Elements of Bob's Inventory System:

- Key processes (see level-0 DFD):
  1. account for anything *added to* inventory
  2. account for anything *taken from* inventory
  3. place orders
  4. pay bills
- Key data *used* by the system:
  - inventories counts (used by the system) and
  - stock-on-hand counts
- Key data *output* by the system:
  - orders
  - payments

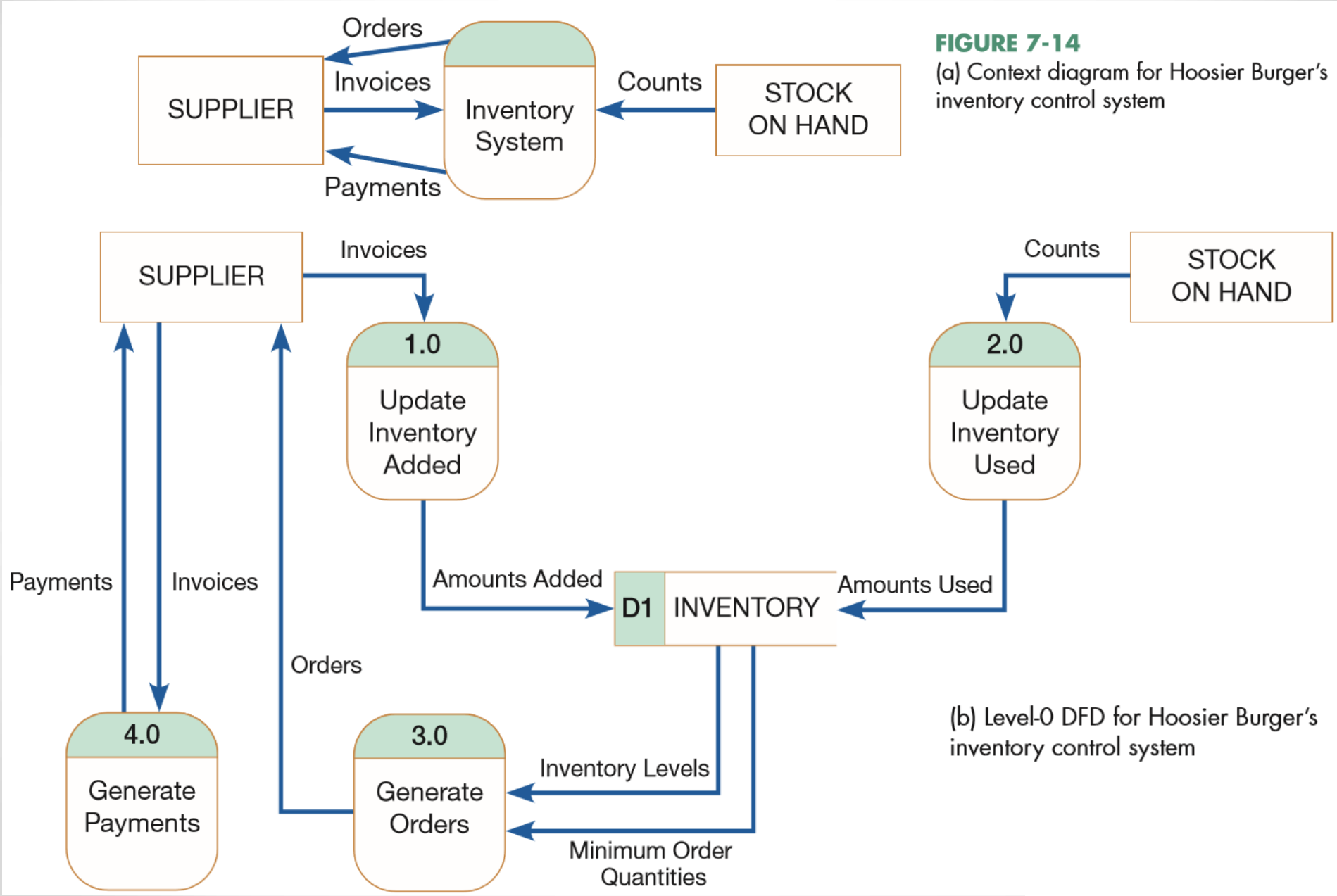


FIGURE 7-14

(a) Context diagram for Hoosier Burger's inventory control system

(b) Level-0 DFD for Hoosier Burger's inventory control system

Case Study 1: HB Inventory Control System

Revised DFD for Bob's Inventory System:

- Bob would like to add 3 additional functions:
  1. data on *new shipments* should be entered into an *automated* system, thus:
     - no more paper stock log sheets
     - shipment data will stay as current as possible (i.e. will be entered into the system *as soon* as the new stock arrives at the restaurant)
  2. system should determine automatically whether a *new order* should be placed,
     - i.e. Bob would no longer worry whether Hoosier Burger has enough of everything in stock at all times*

Case Study 1: HB Inventory Control System

Revised DFD for Bob's Inventory System (cont.):

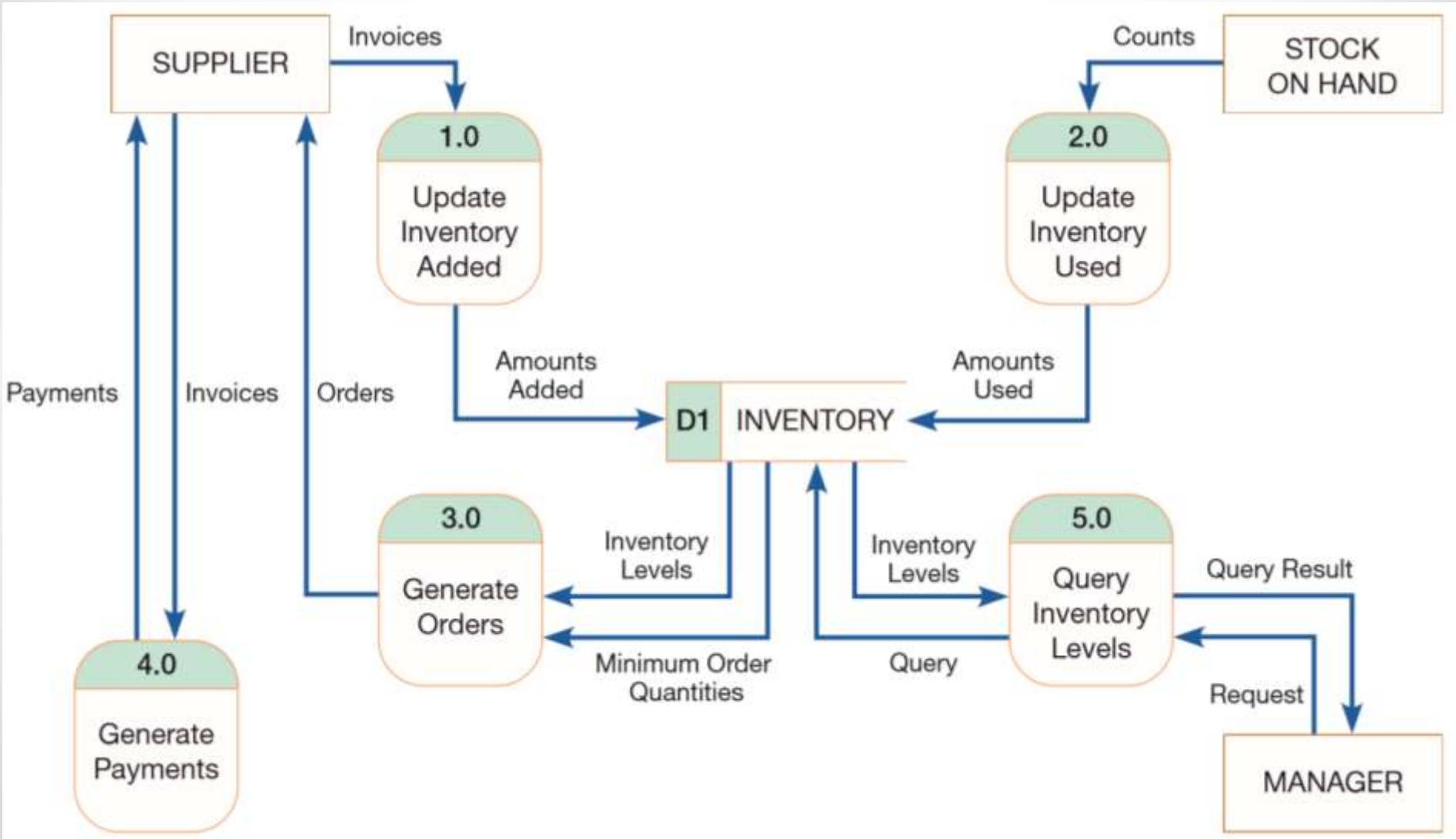
- Bob would like to add 3 additional functions (cont.):
  3. Bob would like to be able to know, at any time, the *approximate inventory level* for each good in stock
     - for some items (e.g. buns), Bob can *visually inspect* the amount in stock and determine approximately how much is left and how much more is needed before closing time
     - for other items, however, Bob may need a *rough estimate* of what is in stock more quickly than he can estimate via a visual inspection

Case Study 1: HB Inventory Control System

Revised DFD for Bob's Inventory System (cont.):

- Compare between original and revised DFDs:
  - new Process 5.0 allows for *querying** the inventory data to get an estimate of how much of an item is in stock
  - Bob's 2 other requests for change can both be handled *within the existing* logical view of the inventory system

FIGURE 7-15
Revised level-0 DFD for Hoosier Burger's inventory control system



Case Study 2 – Using DFDs in Business Process Reengineering (BPR) – "IBM Credit Corporation"

Case Study 2: BPR – IBM Credit

IBM Credit Corporation:

- Case study by *Hammer and Champy* (1993)
- IBM Credit Corporation
  - provides financing for customers making large purchases of IBM computer equipment
  - analyzes deals proposed by salespeople and writes the final contracts governing those deals
  - it typically took six business days to process each financing deal

Case Study 2: BPR – IBM Credit

Steps in processing each financing deal:

1. salesperson calls in with a proposed deal
   - clerk **logs** it and writes details on a piece of paper

2. second person:
   - enters data into a computer system and
   - checks client's **creditworthiness**
   - writes details on a piece of paper and carries paper (along with original documentation) to a loan officer

3. loan officer:
   - modifies standard IBM **loan agreement** for the customer
   - involves separate computer system from one used in step 2

Case Study 2: BPR – IBM Credit

Steps in processing each financing deal (cont.):

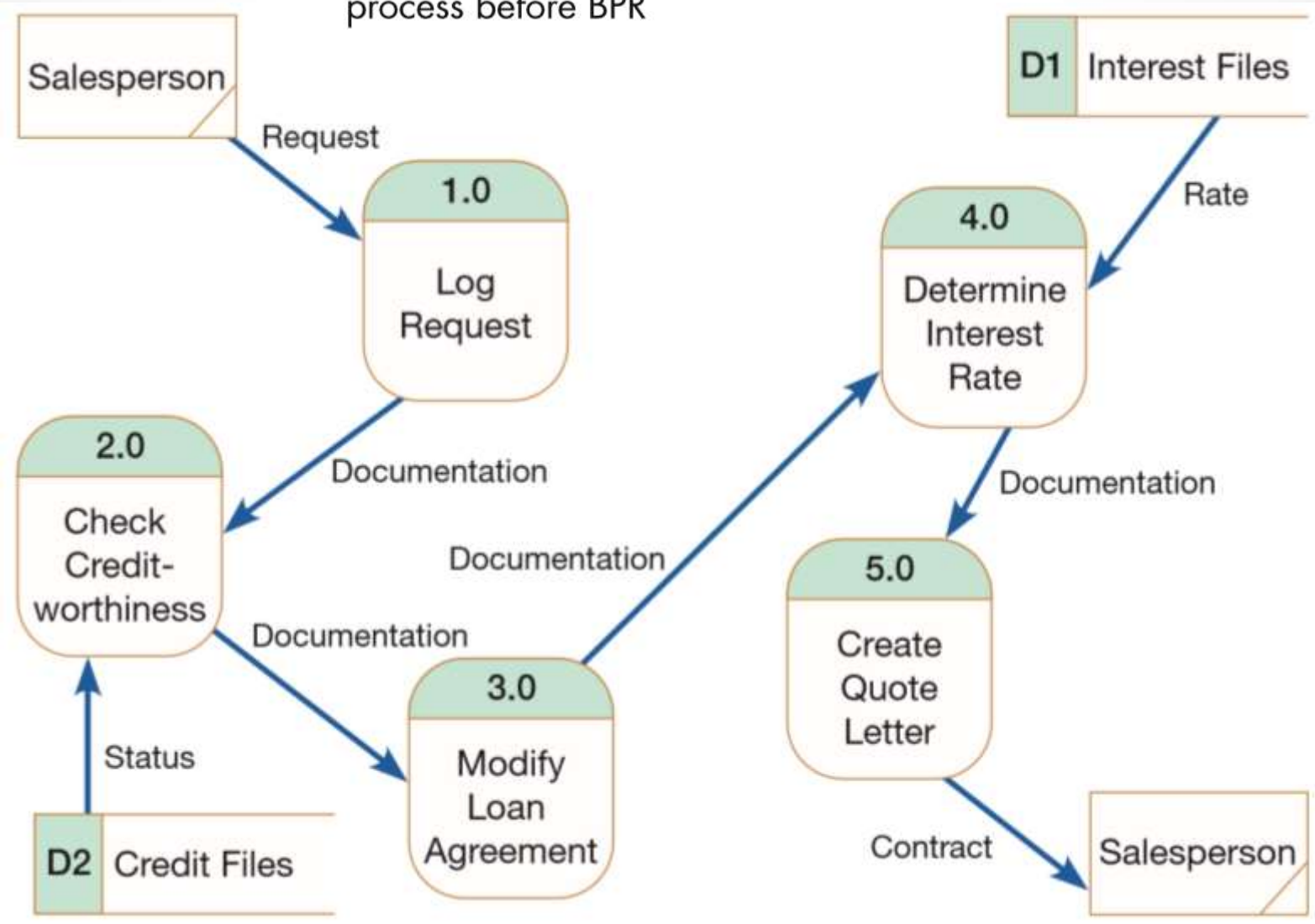
4. details of modified loan agreement:
   - sent on to the next station in the process
   - different clerk **determines interest rate** for loan
   - again, this involves its own information system

5. **quote letter** is created at the next stop:
   - using resulting interest rate and
   - all of the paper generated up to this point
   - quote letter is sent via overnight mail back to the salesperson

- DFD illustrates the overall process, people, computers, and shows it is not that complicated

Case Study 2: BPR – IBM Credit

FIGURE 7-16
IBM Credit Corporation's primary work process before BPR



Case Study 2: BPR – IBM Credit

Steps after applying Business Process Reengineering:

- five sets of task specialists were replaced with *generalists*:
  - call from the field goes to a *single clerk*
  - clerk does all the work necessary to process the contract
  - i.e. now *only 1 person*: checks for creditworthiness, modifies basic loan agreement, & determines appropriate interest rate
- company still has specialists for few cases that are significantly different from routine encounters
- process is now supported by a single computer system

Case Study 2: BPR – IBM Credit

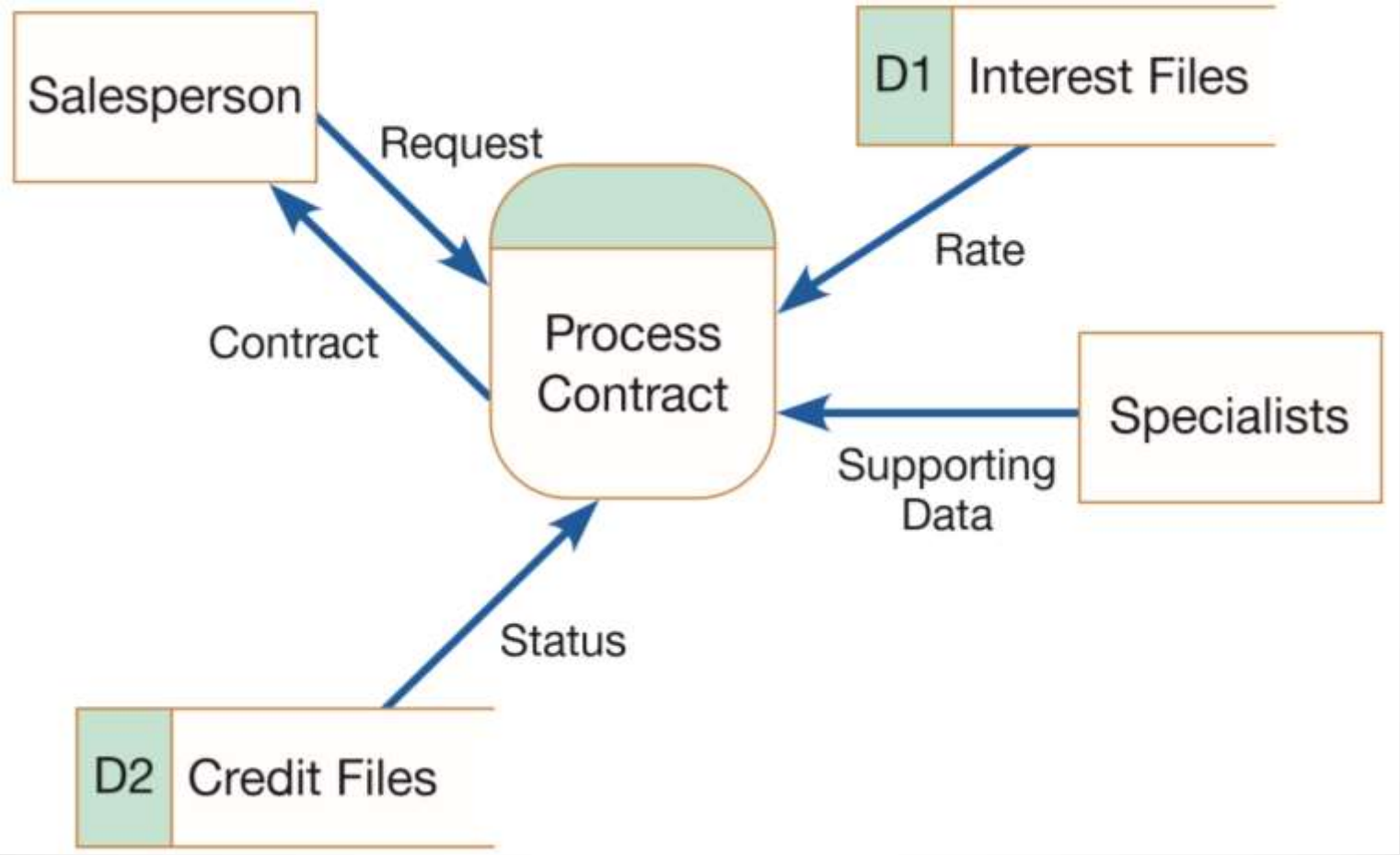
Steps after applying BPR (cont.):

- Note some differences in revised DFD vs original DFD
  - compare the number of *process boxes*
  - *lack of documentation flow* in revised DFD (⇒ process is much simpler, cuts down dramatically on any chance of documentation getting lost between steps)

- Redesigning process from beginning to end:
  - allowed IBM Credit Corporation to increase the number of contracts it can handle by *100-fold*!
  - i.e. allowed company to handle *100 times* more work in the same amount of time and with fewer people

Case Study 2: BPR – IBM Credit

FIGURE 7-17
IBM Credit Corporation's primary work process after BPR



Case Study 3 – Using DFDs in Electronic Commerce Application – "Pine Valley Furniture" WebStore

Case Study 3: PVF Furniture Webstore

Background:

- Process modeling (using DFD) is similar to the process followed for other applications
- *Pine Valley Furniture* (PVF) made a project to sell furniture products over the Internet (i.e. *webstore*)
- Objectives:
  - analyze webstore's high-level system structure,
  - develop a level-0 DFD for those requirements

Case Study 3: PVF Furniture Webstore

Steps in translating webstore system structure into DFD:

- Senior systems analyst (Jim Woo)
  - first, completed JAD (*Joint Application Design*) session
  - then adopted following steps to create DFD for webstore:

1. Identify level-0 ***–major system– processes***:
   - examined the outcomes of the JAD session, and
   - defined *system structure* of webstore system
   - he identified six *high-level processes*, which were the "work" or "action" parts of the website (see Table 7-4)
   - note how each process corresponds to *major processing items* listed in the system structure

Case Study 3: PVF Furniture Webstore

TABLE 7-4 System Structure of the WebStore and Corresponding Level-0 Processes

WebStore System	Processes
❑ Main Page	Information Display (minor/no processes)
• Product Line (Catalog)	1.0 Browse Catalog
✓ Desks	2.0 Select Item for Purchase
✓ Chairs	
✓ Tables	
✓ File Cabinets	
• Shopping Cart	3.0 Display Shopping Cart
• Checkout	4.0 Check Out Process Order
• Account Profile	5.0 Add/Modify Account Profile
• Order Status/History	6.0 Order Status Request
• Customer Comments	Information Display (minor/no processes)
❑ Company Information	
❑ Feedback	
❑ Contact Information	

Case Study 3: PVF Furniture Webstore

Translating webstore system structure into DFD (cont.):

2. Determine ***existing PVF systems*** with which webstore can *exchange information*:
   1. *Customer Tracking System* (for managing customer information)
      - info. is *passed from webstore* system to this system when customer opens an account
   2. *Purchasing Fulfillment System* (for tracking orders)
      - info. is *stored in this system* when an order is placed, and
      - *retrieves* status info. on a prior order (at customer request)
   - These 2 existing systems will be
     - "**sources**" (providers) of information, and
     - "**sinks**" (receivers) of information

     for webstore system

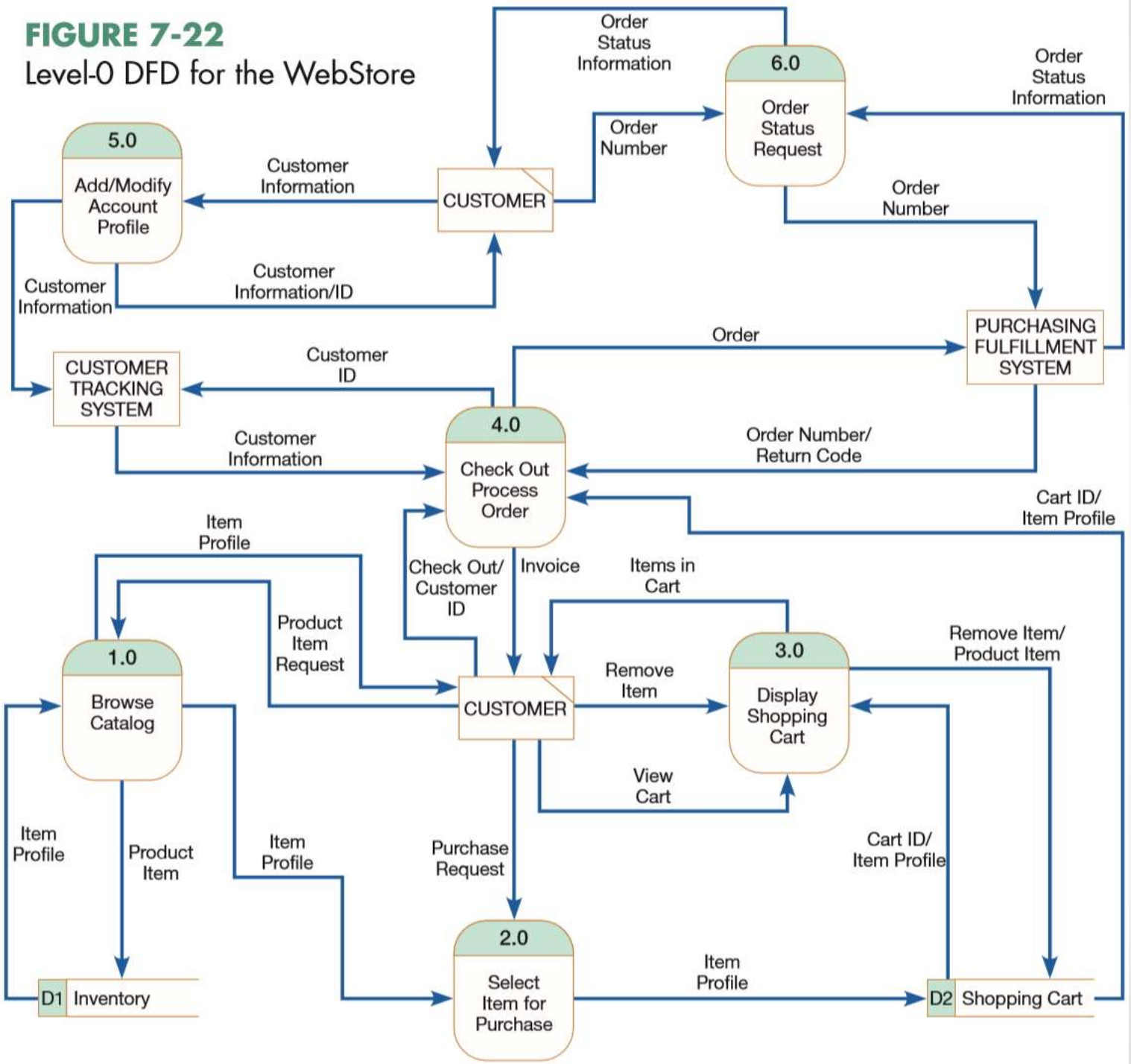
Case Study 3: PVF Furniture Webstore

Translating webstore system structure into DFD (cont.):

3. Determine additional data sources (i.e. **data stores**):
   1. access to *inventory database*
      - to produce an online product catalog
   2. webstore *shopping cart*
      - a temporary database
      - used to store the items a customer wants to purchase
      - shopping cart data can be deleted once transaction is completed

- Jim used these steps to develop level-0 DFD for webstore system

FIGURE 7-22
Level-0 DFD for the WebStore



Case Study 3: PVF Furniture Webstore

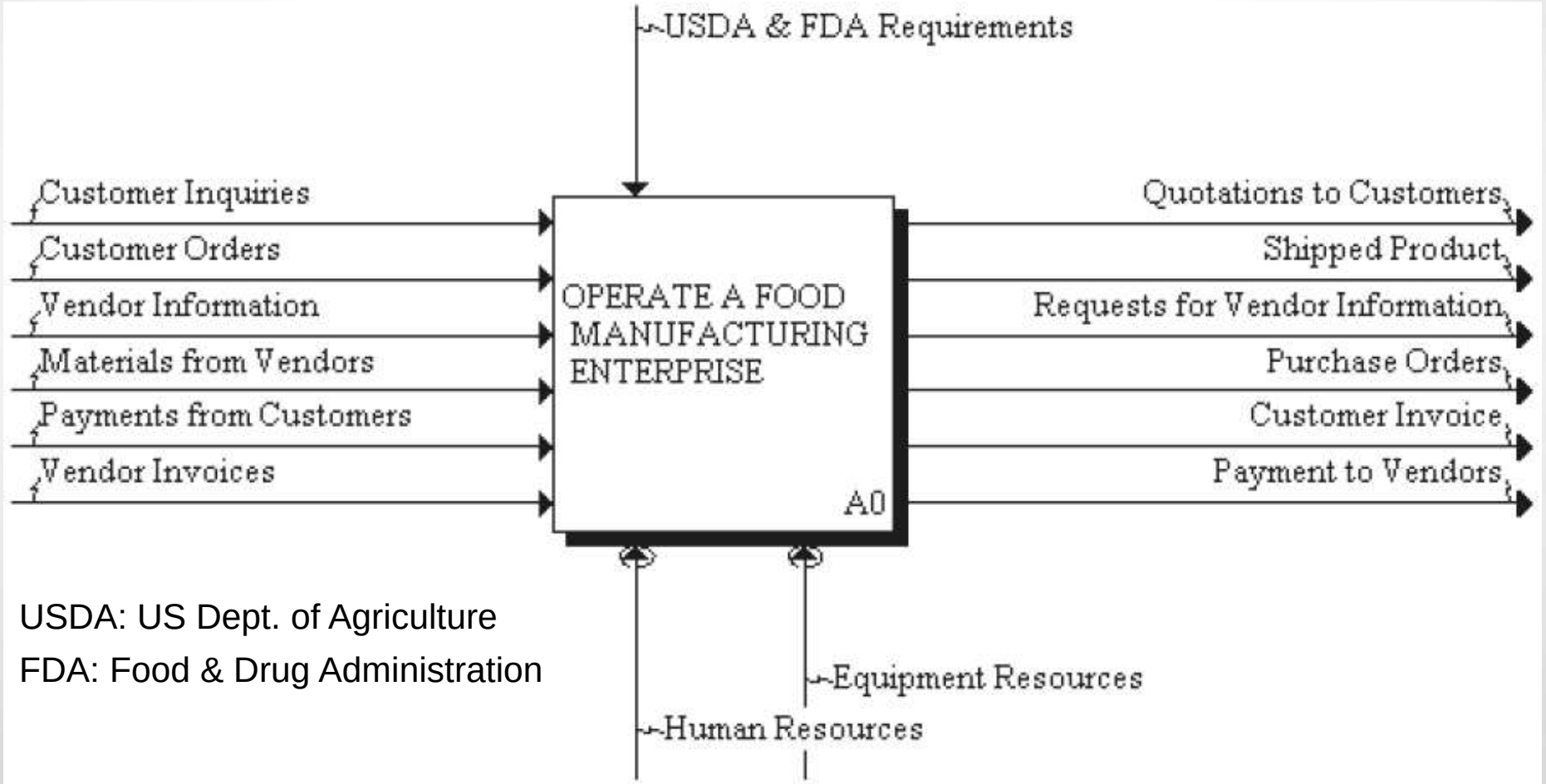
Results of developing DFD for webstore:

- good understanding of *information flow* through the webstore
- good understanding of *customer interaction* with the system
- good understanding of how the webstore *shares information* with existing PVF systems
- Note, each of these high-level processes needs (eventually) to be further decomposed before system design could proceed
- Also, another outcome of this analysis activity is *conceptual data modeling* (discussed later)

Case Study 4 – Converting IDEF0 Model to DFD – Food Manufacturing Company

Integrated IDEF0 Model of Mfg. Enterprise

- Top-level view of the enterprise: Node A0



USDA: US Dept. of Agriculture

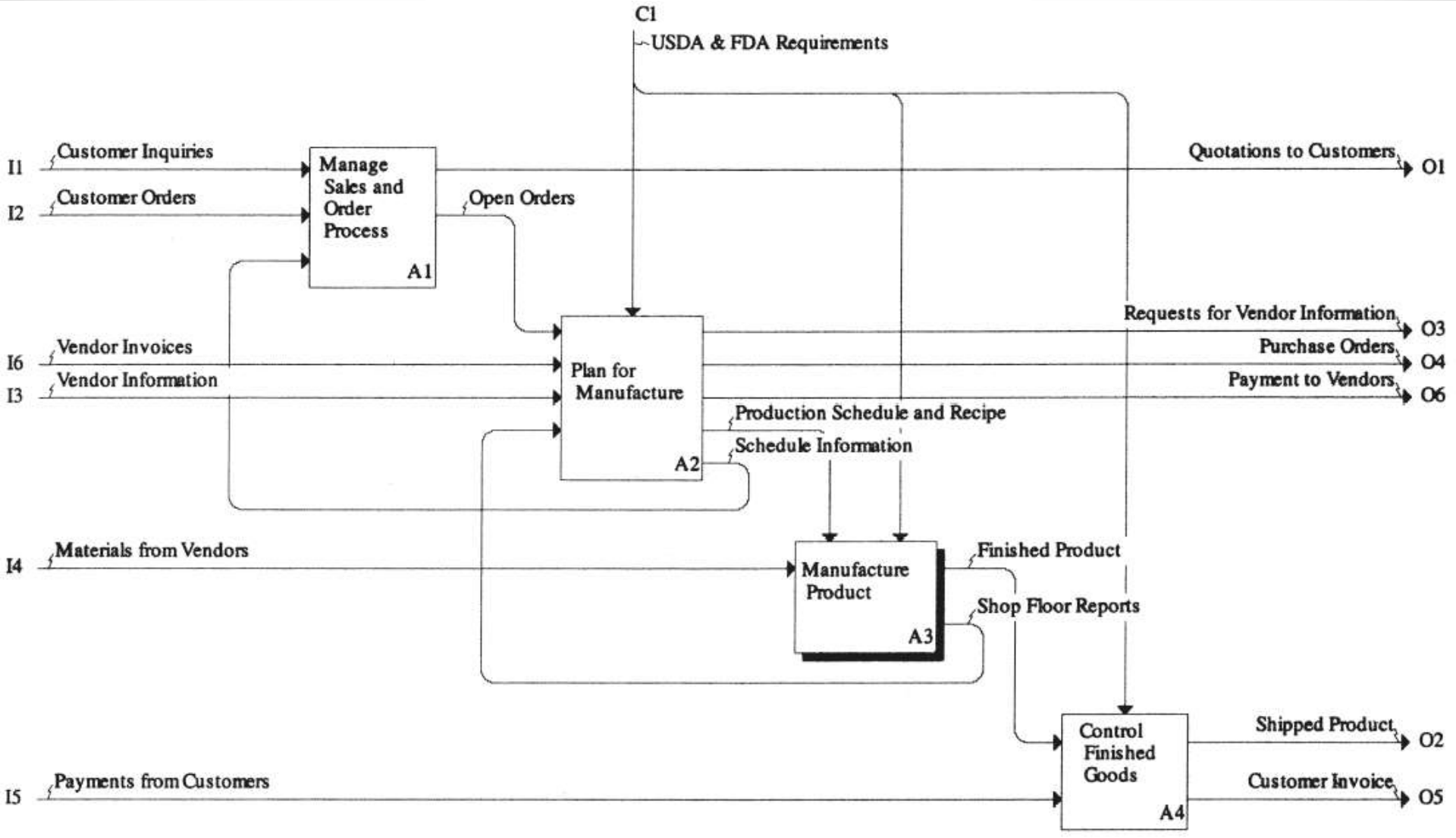
FDA: Food & Drug Administration

Decomposition of Node A0

A0 — Operate a Food Manufacturing Enterprise

- A1 — Manage Sales and Order Process
- A2 — Plan for Manufacture
- A3 — Manufacture Product
- A4 — Control Finished Goods

Decomposition of Node A0

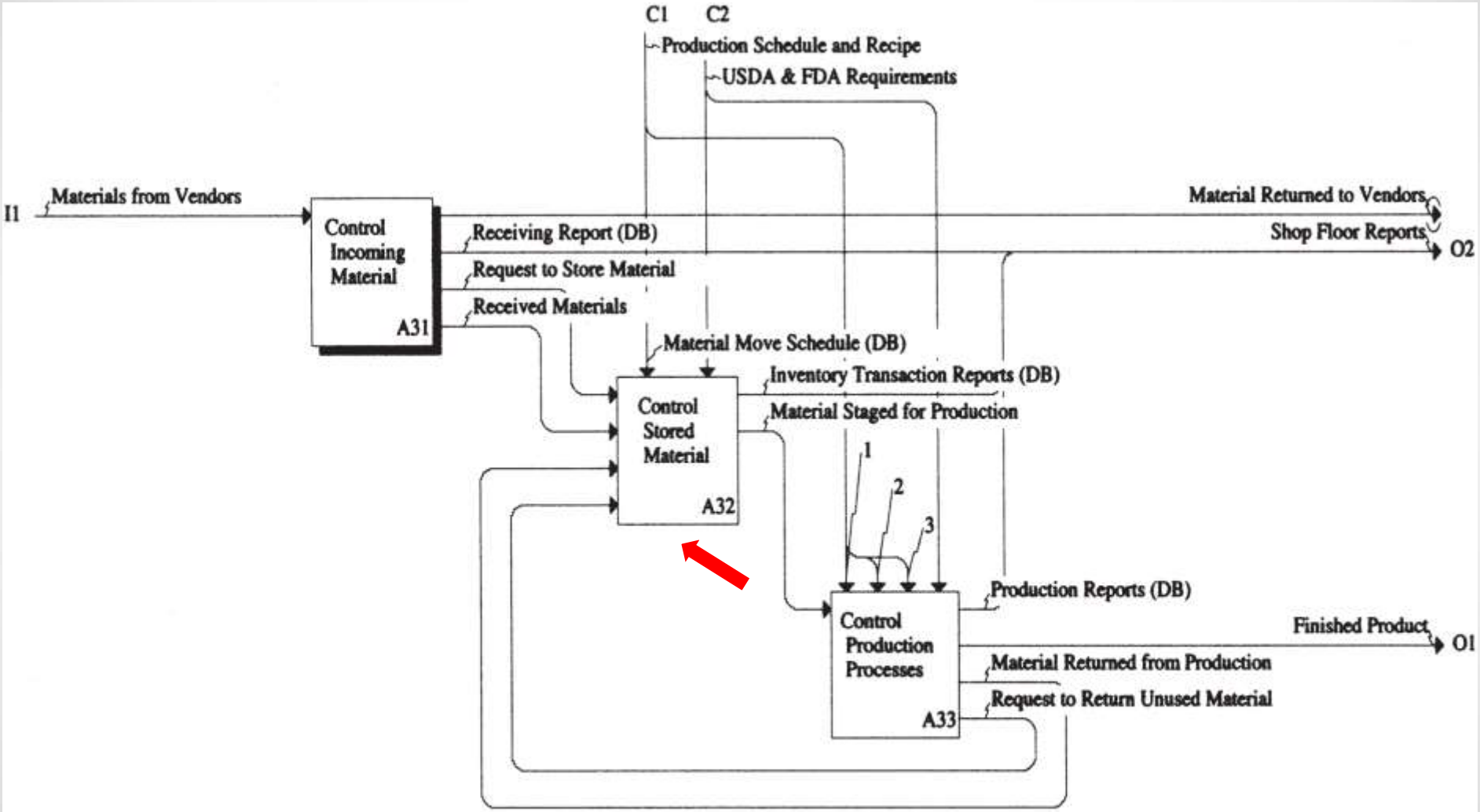


Decomposition of Node A3

A0 — Operate a Food Manufacturing Enterprise

- A1 — Manage Sales and Order Process
- A2 — Plan for Manufacture
- A3 — Manufacture Product
  - **A31 — Control Incoming Materials**
  - **A32 — Control Stored Material**
  - **A33 — Control Production Processes**
- A4 — Control Finished Goods

Hierarchic Decomposition Illustrated: Node A3



1 - Retort Processing Information (DB)
2 - Cook Sheet (DB)
3 - Day Production Schedule (DB)

DFD Symbols/Notation (Reminder)

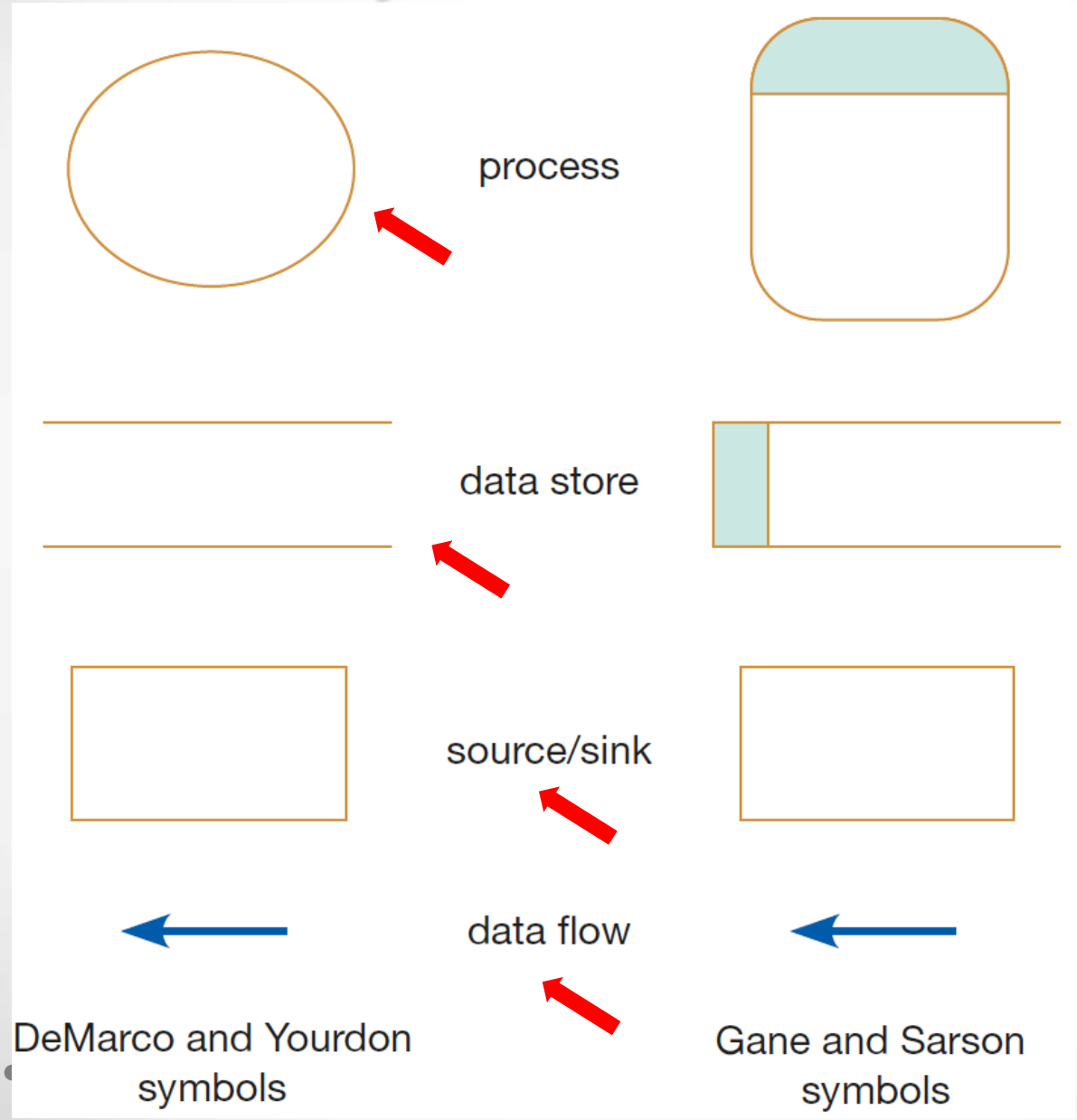


FIGURE 7-2
Comparison of DeMarco and Yourdon and Gane and Sarson DFD symbol sets

Context DFD of Node A32

- There are 3 source entities at boundary of system (A32):
  1. receiving (i.e. input from Node A31)
  2. production planning (i.e. input from Node A2), and
  3. production (i.e. feedback from Node A33)

1. **Receiving** (Node A31):
   - this is the entity in charge of the process "Control Incoming Material"
   - receiving is a "trigger" for process "Control Stored Materials" (i.e. it initiates an action in the process when it makes a "**request to store materials**")

Context DFD of Node A32 (cont.)

- There are 3 source entities at boundary of system (A32):
  1. receiving (i.e. input from Node A31)
  2. production planning (i.e. input from Node A2), and
  3. production (i.e. feedback from Node A33)

2. **Production planning** department (Node A2)
   - source of another trigger
   - trigger ("**material move schedule**") is to move raw material from warehouse to work in process – WIP

Context DFD of Node A32 (cont.)

- There are 3 source entities at boundary of system (A32):
  1. receiving (i.e. input from Node A31)
  2. production planning (i.e. input from Node A2), and
  3. production (i.e. feedback from Node A33)

3. **Production** (Node A33)
   - "**request to return unused materials**" from production supervisor is another trigger to the process
   - raw material that has been moved into production but not used must be returned to storage

- Finally, the process sends (i.e. as output) an "**inventory transaction report**" to a *data store* (see context DFD)

Context DFD of Node A32 (cont.)

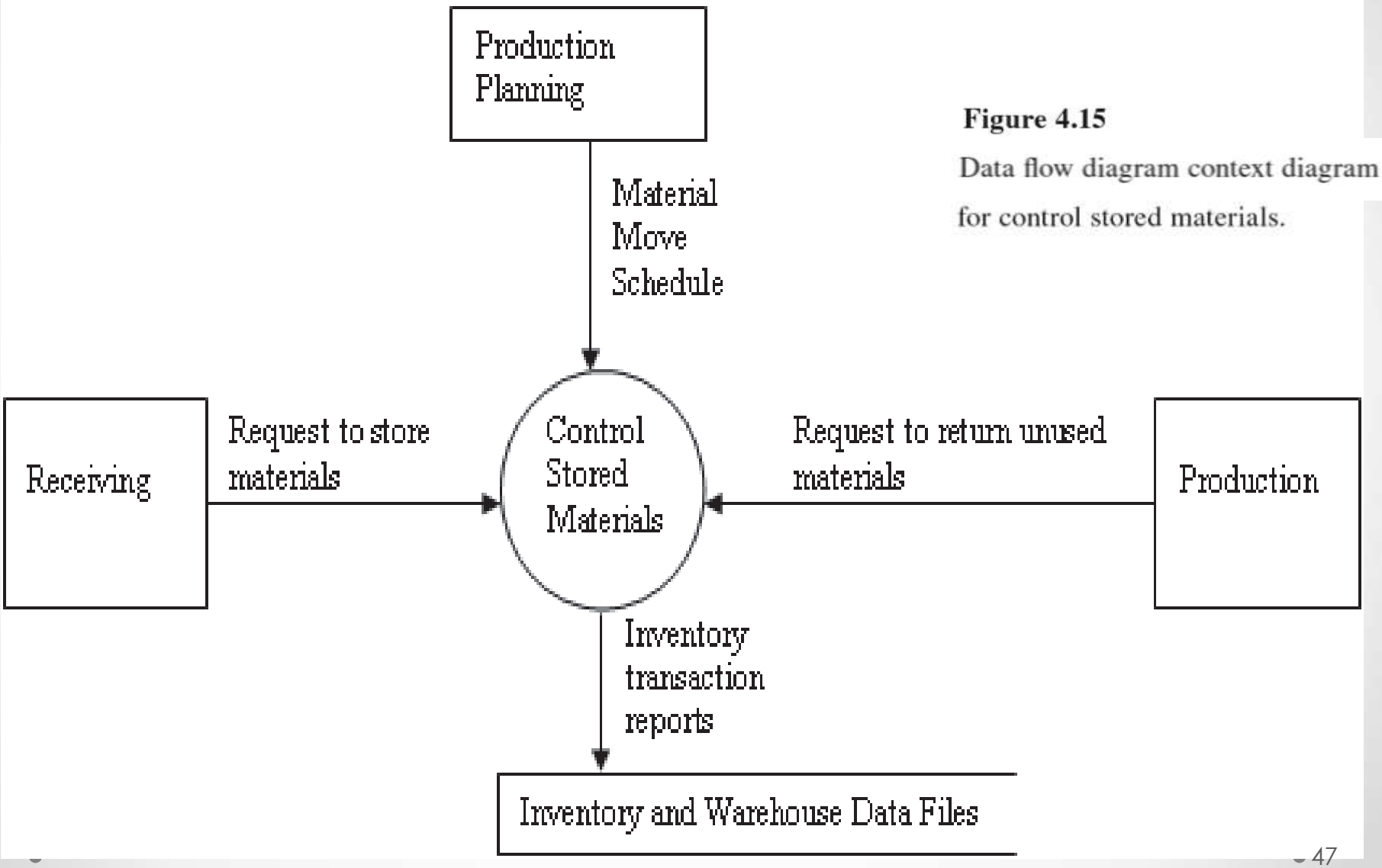


Figure 4.15

Data flow diagram context diagram for control stored materials.

Decomposition of Context Data Flow Diagram

- The context process (Control Stored Materials) is composed of four level-0 processes (see level-0 DFD):
  1. **Store Raw Materials**
  2. **Move Raw Materials to WIP**
  3. **Return Unused Raw Materials to Storage**
  4. **Transfer Daily Records**

Decomposition of Context Data Flow Diagram

- The overall structure of the data flow diagram hierarchy is often shown in a **process hierarchy chart**
- Process hierarchy chart:
  - series of block diagrams
  - show the hierarchic relationship among processes that are documented in the data flow diagrams

Decomposition of Context Data Flow Diagram

The hierarchic process function:

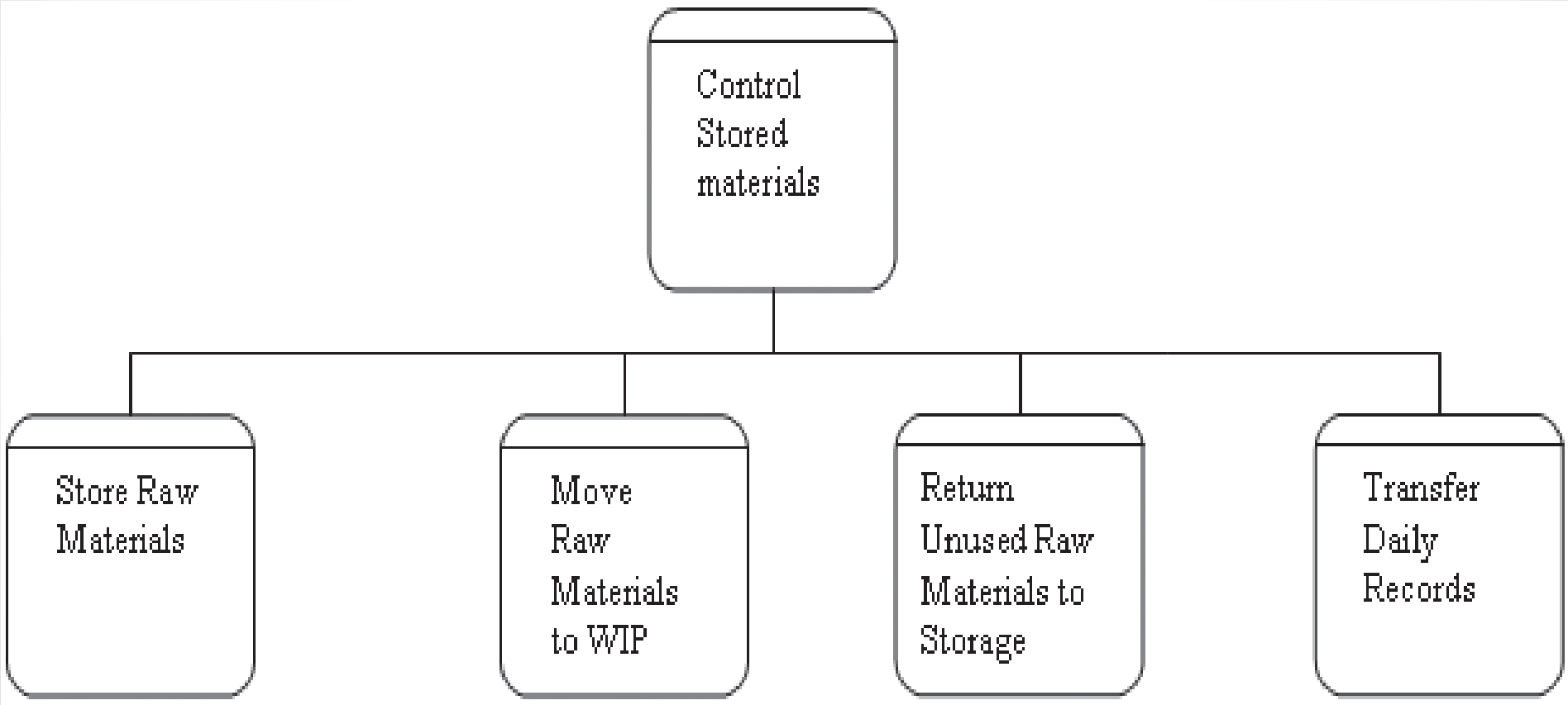
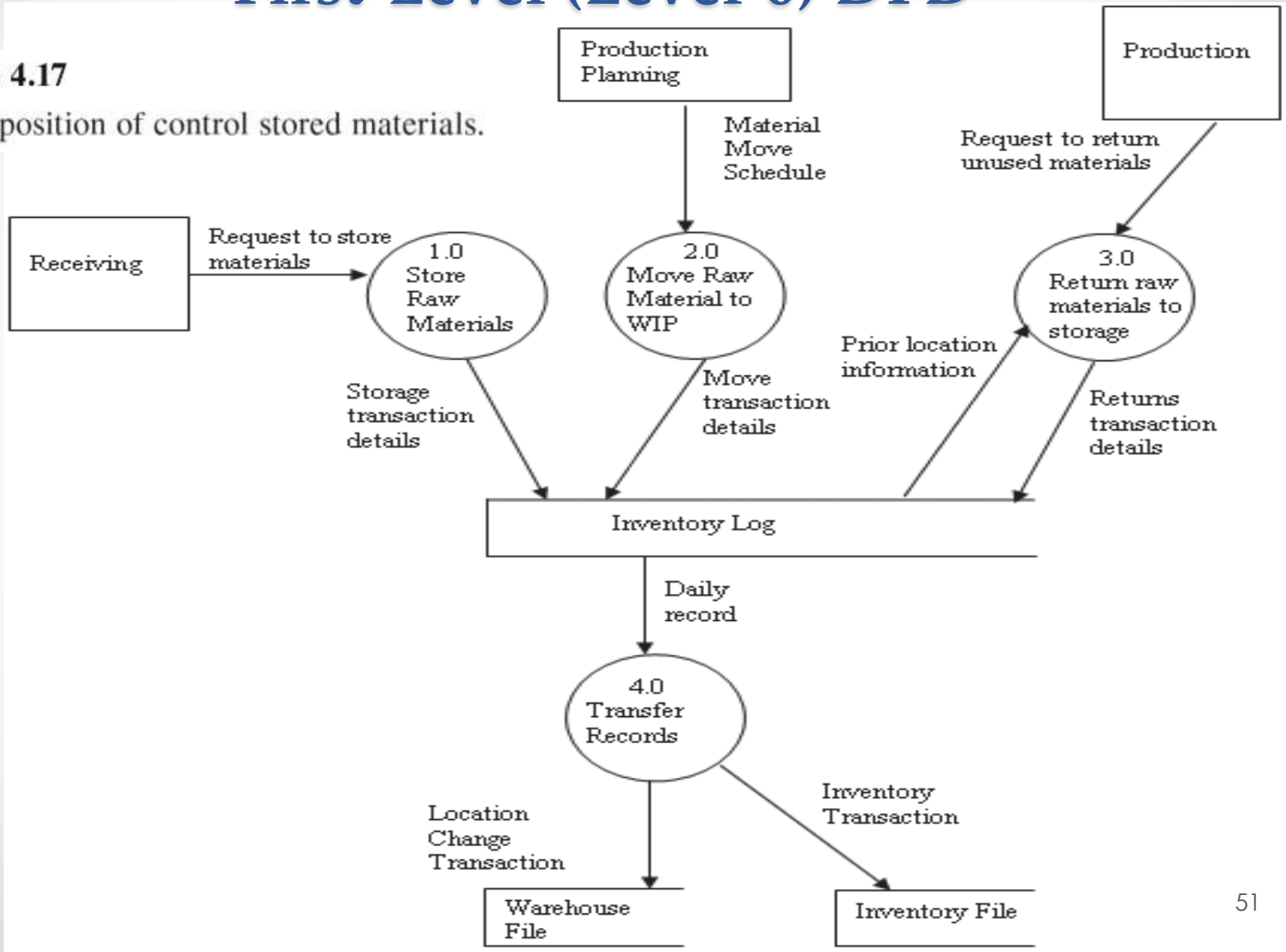


Figure 4.16 Process hierarchy chart.

First-Level (Level-0) DFD

Figure 4.17

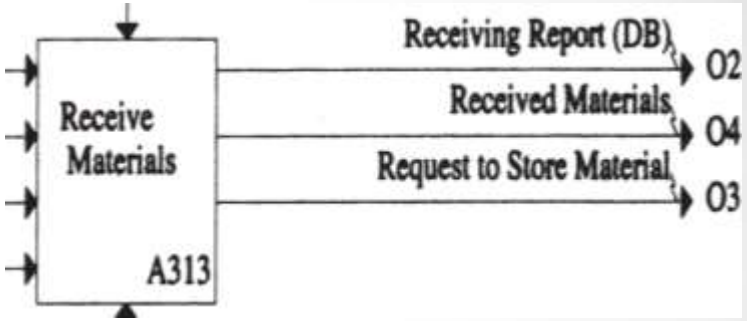
Decomposition of control stored materials.



Description of First-Level (Level-0) DFD

1. **Store raw materials**:

- Receiving requests forklift truck driver to move material from the loading dock to inventory storage



- Driver takes the material to the warehouse
- Driver places the material in the required location
- Driver then records the following in the log,
  - material
  - location used, and
  - date and time of the transaction

Description of First-Level (Level-0) DFD

2. Move raw material to WIP:

- Forklift truck driver is given the schedule of material moves from storage to the factory floor
- Each time the driver makes a move,
  - raw material inventory is *debited* and
  - status of the warehouse location is *updated*
- This is done by indicating a transaction to relieve inventory in the log:
  - recording the material
  - material location
  - date and time of the transaction

Description of First-Level (Level-0) DFD

3. Return unused raw material to storage:

- Some materials that are brought to the factory floor may be returned if they are not used during production
- Upon request from production supervisor,
  - driver takes material *back to storage* and
  - *logs* the credit entry into the log

Description of First-Level (Level-0) DFD

4. **Transfer records:**

- Forklift truck driver's inventory log is used as the primary record for updating
  - warehouse records and
  - inventory records
- This process is done at the end of the shift
- Materials management checks for any discrepancies between the receiving report and actual location of material
  - i.e. by comparing the log with the receiving report

RECEIVING REPORT

Supplier: General Provisions
125 Common St.
Boise, ID 44830

Purchase Order No.: PO3502
Date Received: June 25 2006

Quantity		Mfg. Lot No.	Item Code	Mat'L Lot No.	Description	Storage Location
accepted	not accepted					
1000		1275	RM805	97275	Tomato Paste, 1 gallon cans	Area A, Aisle 1 tier 1, bins 10-18
300		1283	"	97276	" " " "	Area A, Aisle 1 Tier 2, Bins 10-13
	100	"	"		" " " "	returned(1)

Comments: (1) returned due to case damage and badly dented containers.

Received by: J. Bibbs

Videos to Watch

- **What is DFD? Data Flow Diagram Symbols and More**
  https://youtu.be/6VGTvgaJIIM (*Smartdraw*)

- **How to Draw Data Flow Diagram?**
  https://youtu.be/ztZsEI6C-mI (*Visual Paradigm*)

- **DFD Diagram 0**
  https://youtu.be/Ik85hZkyYPA (*Visible Analyst*)

Sources

- **Modern Systems Analysis and Design**. Joseph S. Valacich and Joey F. George. Pearson. Eighth Ed. 2017. Chapter 7.

- **Design of Industrial Information Systems**. Thomas Boucher, and Ali Yalcin. Academic Press. First Ed. 2006. Chapter 4.