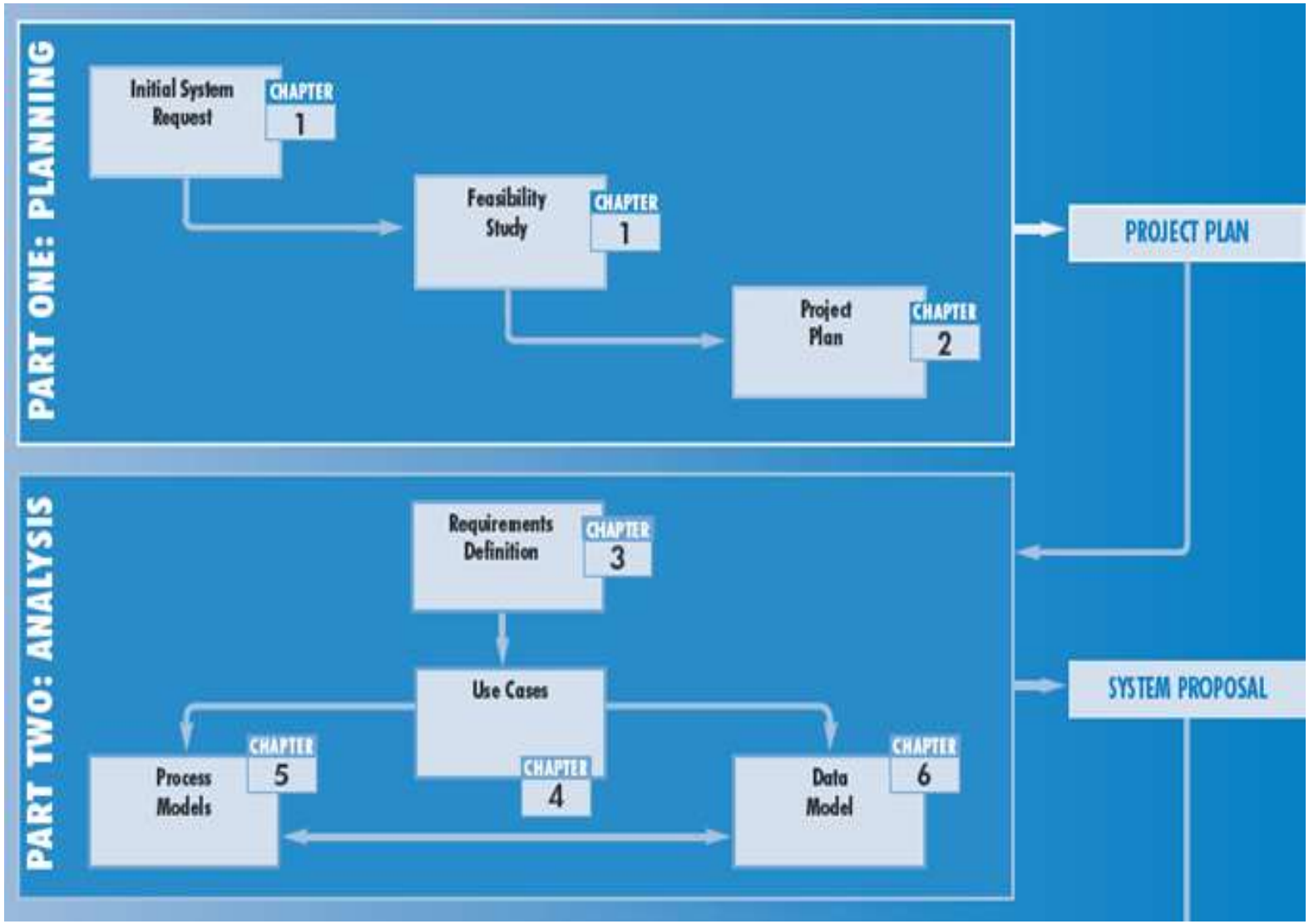
Systems Analysis and Design
5th Edition

Chapter 5. Creating Data Flow Diagrams (DFD)

Roberta Roth, Alan Dennis, and Barbara Haley Wixom

Phase	Chapter	Step	Technique	Deliverable
Planning	1	Identify opportunity	Project identification	System request
Focus: Why build this system?	1	Analyze feasibility	Technical feasibility Economic feasibility Organizational feasibility	Feasibility study
How to structure the project?	2	Develop workplan	Time estimation Task identification Work breakdown structure PERT chart Gantt chart Scope management	Project plan — work plan
Primary outputs: — System Request with feasibility study — Project plan	2	Staff project	Project staffing Project charter	— Staffing plan
	2	Control and direct project	CASE repository Standards Documentation Timeboxing Risk management	— Standards list — Risk assessment

Phase	Chapter	Step	Technique	Deliverable
Analysis Focus: Who, what, where, and when for this system? Primary output — System proposal	3	Develop analysis strategy	Business process automation Business process improvement Business process reengineering	System proposal
	3	Determine business requirements	Interview JAD session Questionnaire Document analysis Observation	— Requirements definition
	4	Create use cases	Use case analysis	— Use cases
	5	Model processes	Data flow diagramming	— Process models
	6	Model data	Entity relationship modeling Normalization	— Data model


PART ONE: PLANNING
Initial System Request
CHAPTER 1
Feasibility Study
CHAPTER 1
Project Plan
CHAPTER 2
PROJECT PLAN
PART TWO: ANALYSIS
Requirements Definition
CHAPTER 3
Use Cases
CHAPTER 4
Process Models
CHAPTER 5
Data Model
CHAPTER 6
SYSTEM PROPOSAL

Chapter 5 Outline

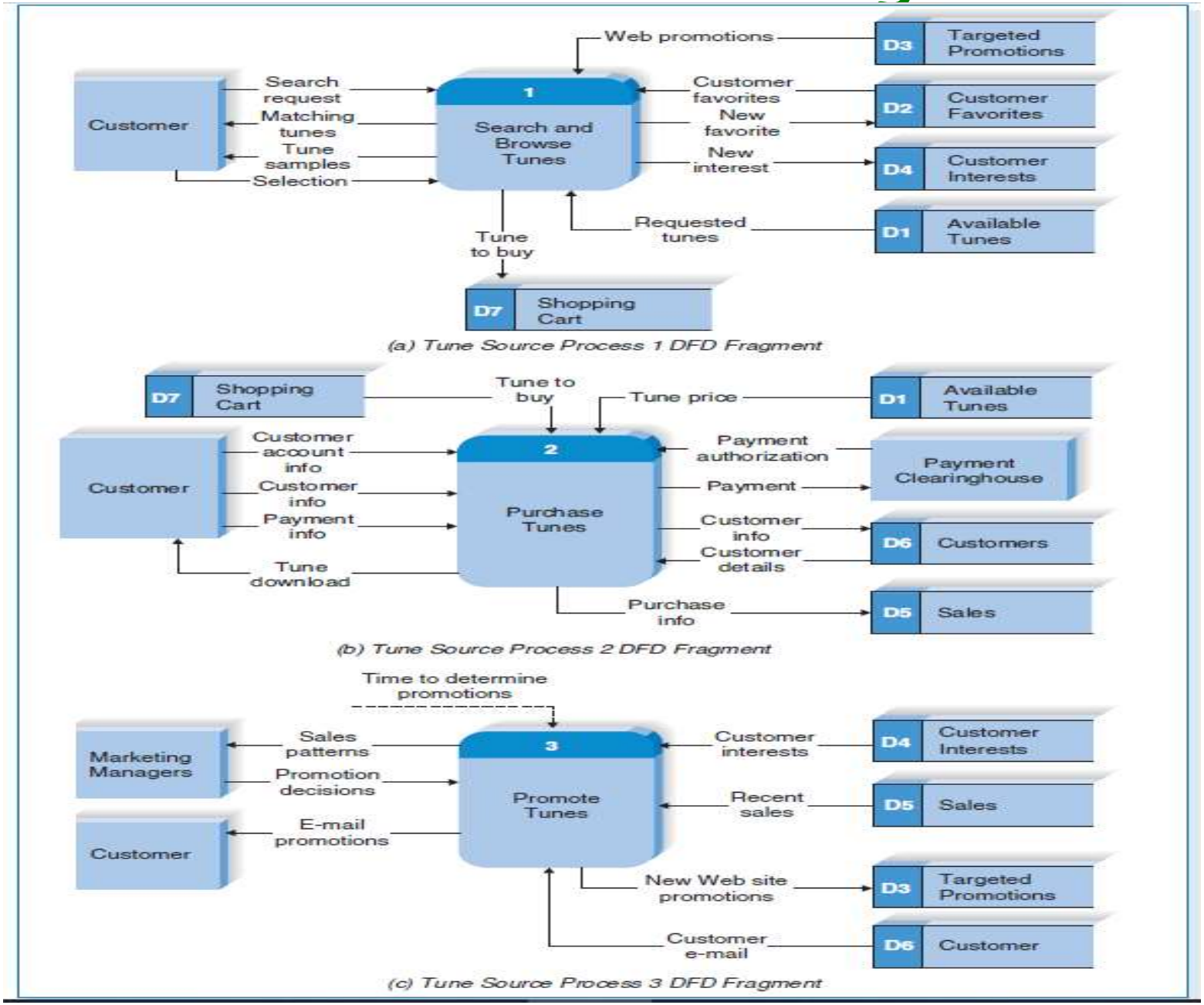
- Explain the rules and style guidelines for data flow diagrams.
- Describe the process used to create data flow diagrams.
- Create data flow diagrams.

Introduction

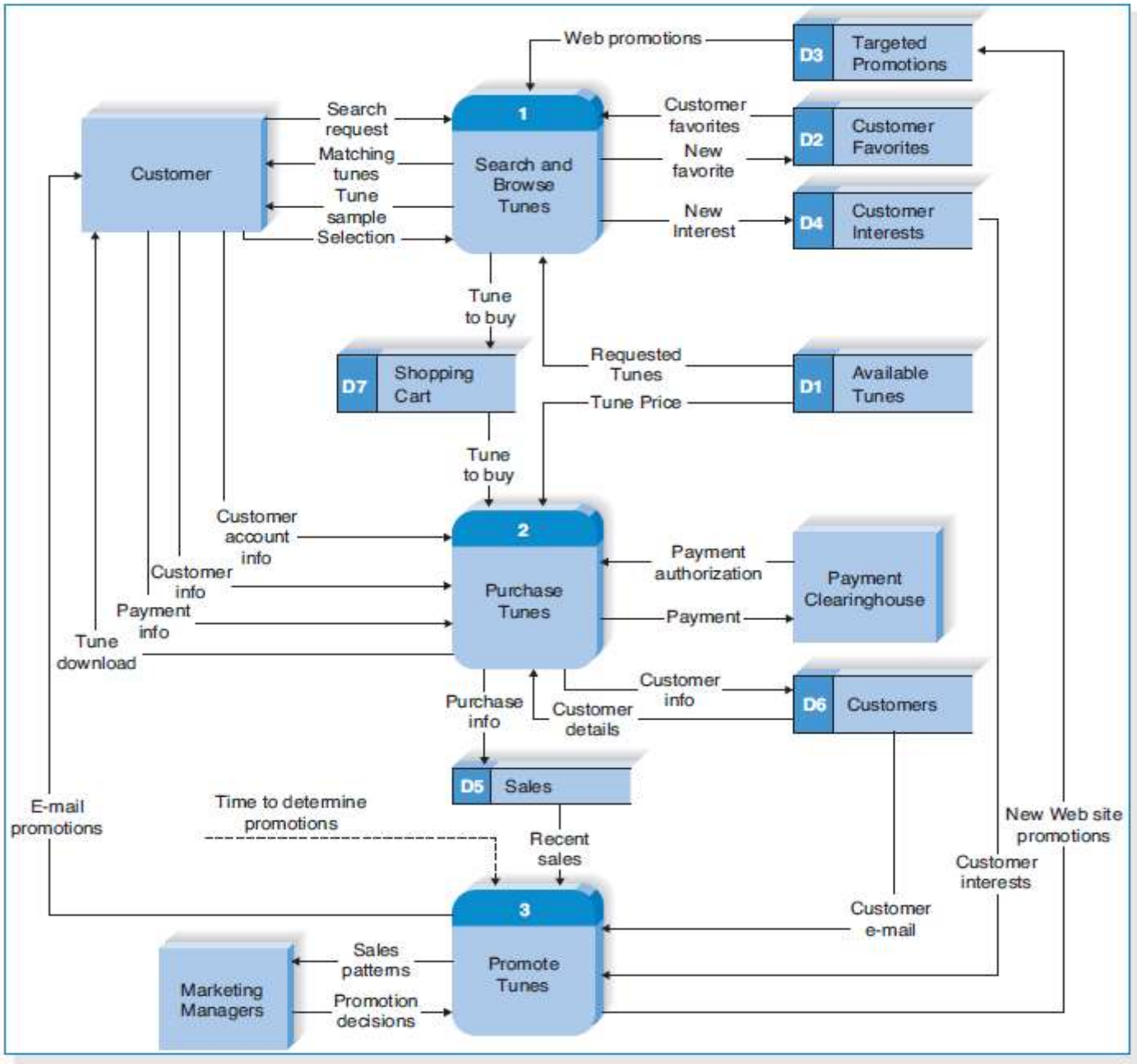
- Process model
  - It is a graphical way of representing how a business system should operate.
  - It illustrates the processes or activities that are performed and how data move among them.

- Data flow diagramming
  - is a common technique that diagrams the business processes and the data that pass among them. DFD is a logical process model.
- Logical process models describe processes without suggesting how they are conducted.
- These logical models are refined into physical models, which provide information that is needed to ultimately build the system during the design phase .
- Explain the rules and style guidelines for data flow diagrams.

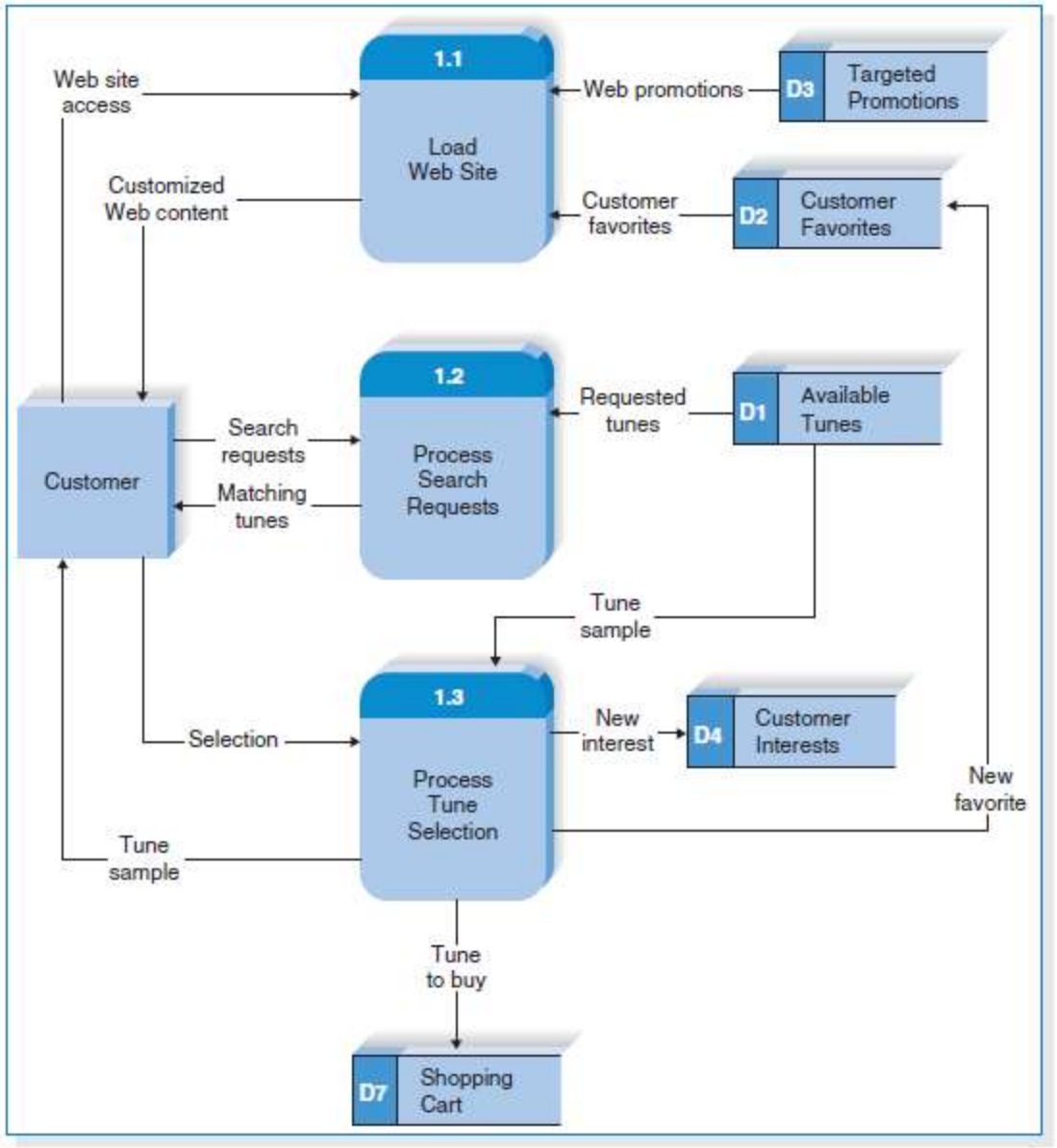
Tune Source DFD Fragments



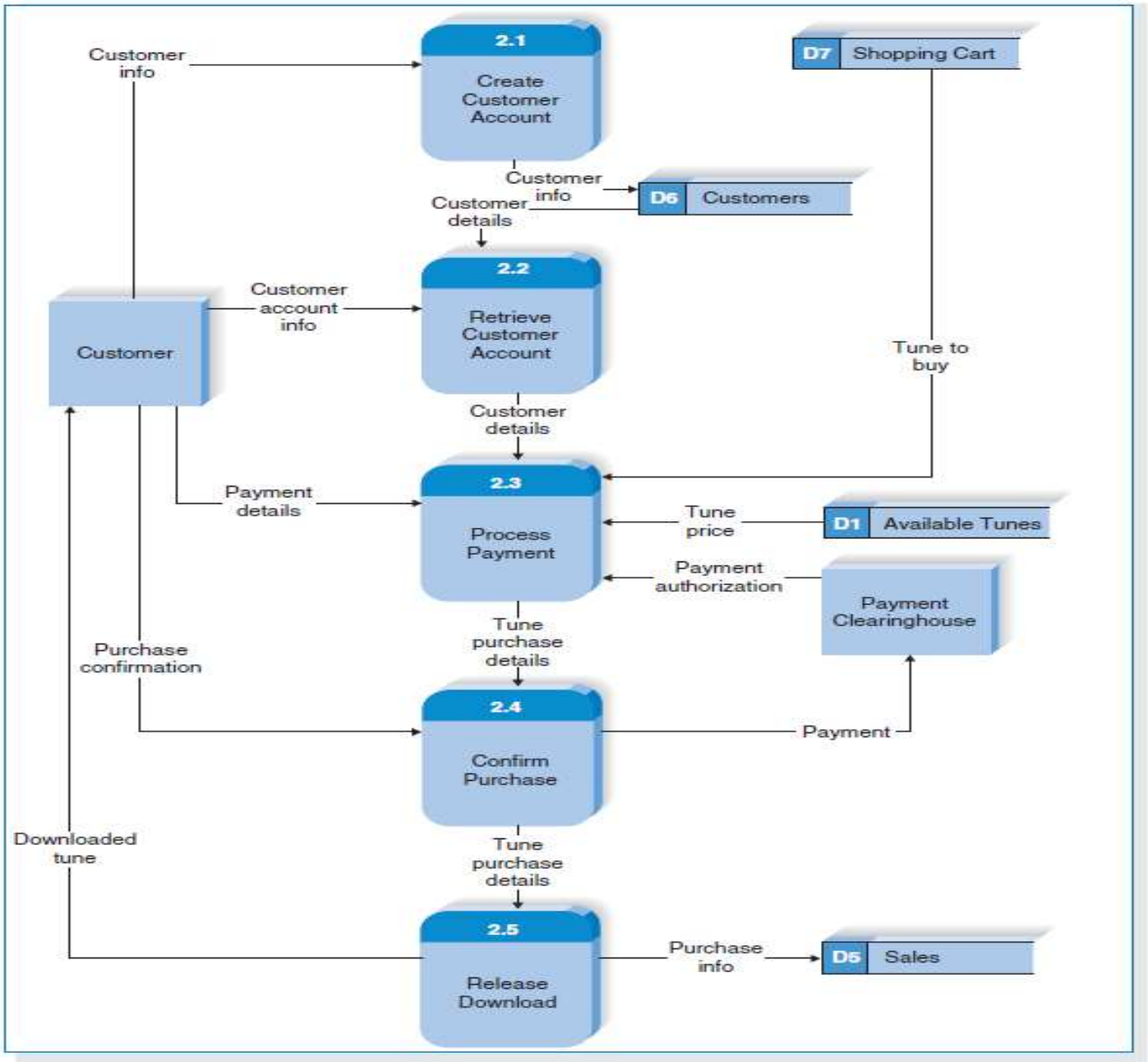
Tune Source Level 0 DFD



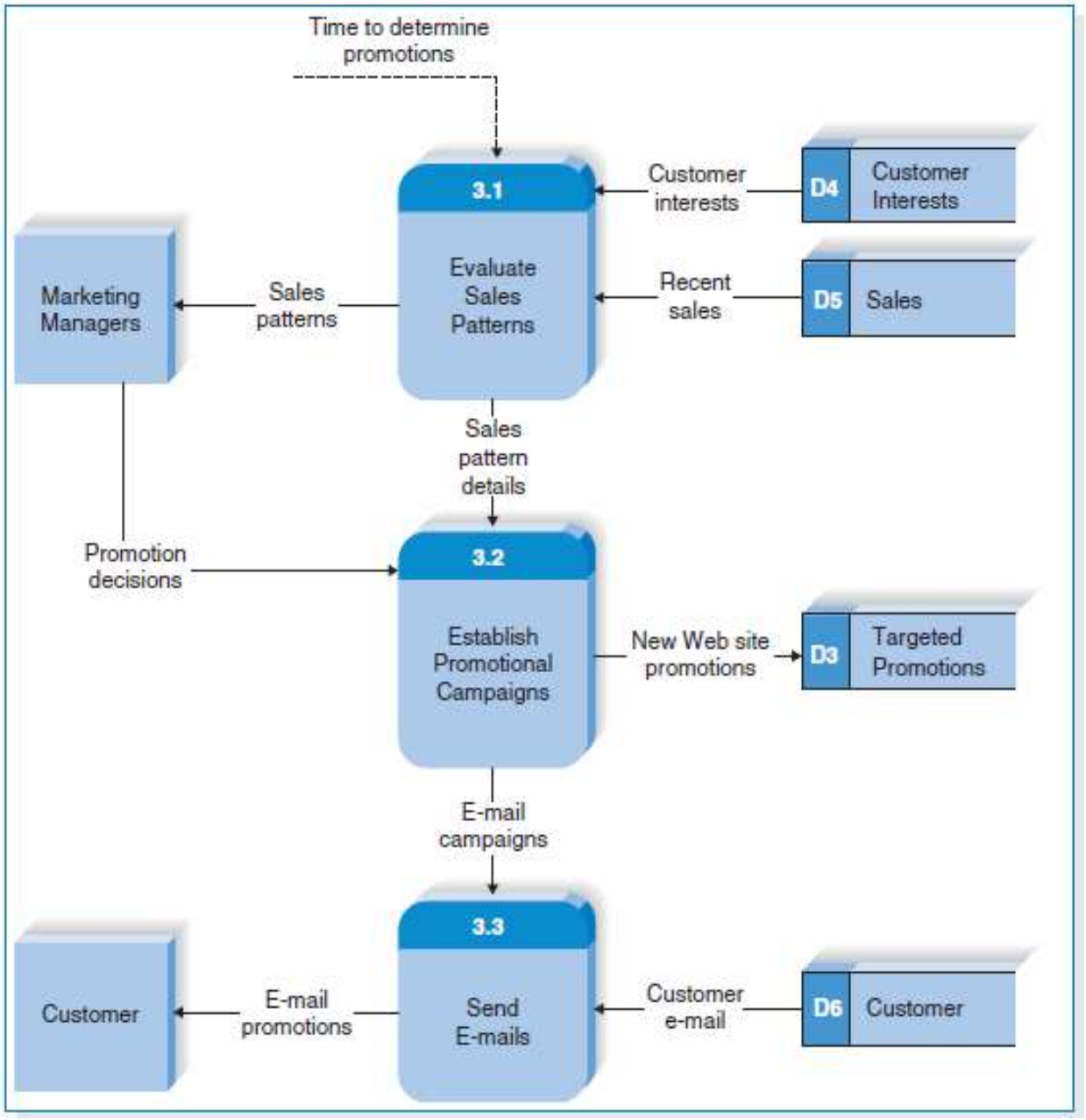
Level 1 DFD for Tune Source Process 1: Search and Browse Tunes



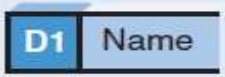
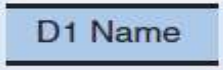
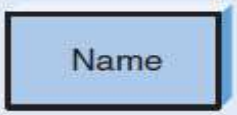
Level 1 DFD for Tune Source Process 2: Purchase Tunes



Level 1 DFD for Tune Source Process 3: Promote Tunes



Elements of Data Flow Diagram (DFD)

Data Flow Diagram Element	Typical Computer-Aided Software Engineering Fields	Gane and Sarson Symbol	DeMarco and Yourdon Symbol
Every *process* has a number a name (verb phase) a description at least one output data flow at least one input data flow	Label (name) Type (process) Description (what is it) Process number Process description (structured English) Notes		
Every *data flow* has a name (a noun) a description one or more connections to a process	Label (name) Type (flow) Description Alias (another name) Composition (description of data elements) Notes		
Every *data store* has a number a name (a noun) a description one or more input data flows one or more output data flows	Label (name) Type (store) Description Alias (another name) Composition (description of data elements) Notes		
Every *external entity* has a name (a noun) a description	Label (name) Type (entity) Description Alias (another name) Entity description Notes		

Elements of Data Flow Diagram (DFD)

- Process
  - A process is an activity or a function that is performed for some specific business reason.
  - **Should be named starting with a verb** and **ending with a noun**
  - **Names should be short** yet contain enough information so that the reader can easily understand exactly what they do.
  - In general, **each process performs only one activity**, so most system analysts **avoid using the word "and" in process names** because it suggests that the process performs several activities.
  - Every process **must have at least one input data flow and at least one output data flow.**
  - Every process **has a unique identification number** a name, and a description.

Elements of Data Flow Diagram (DFD)

- Data Flow
  - data flow is a **single piece of data** (e.g., Tune price) (sometimes called a data element), or **a logical collection of several pieces of information** (e.g., Customer info).
  - **Should be named with a noun**. The description of a data flow lists exactly what data elements the flow contains.
  - Data flows **are the glue** that holds the processes together.
  - One end of every data flow will always **come from or go to a process**, with the **arrow showing the direction into or out of the process**.
  - Data flows show **what inputs go into each process** and **what outputs each process produces**.
  - Every process must **create at least one output data flow**, because if there is no output, **the process does not do anything**.
  - Likewise, each process has **at least one input data flow**, because it is difficult, if not impossible, to produce an output with no input.

Elements of Data Flow Diagram (DFD)

Data Store

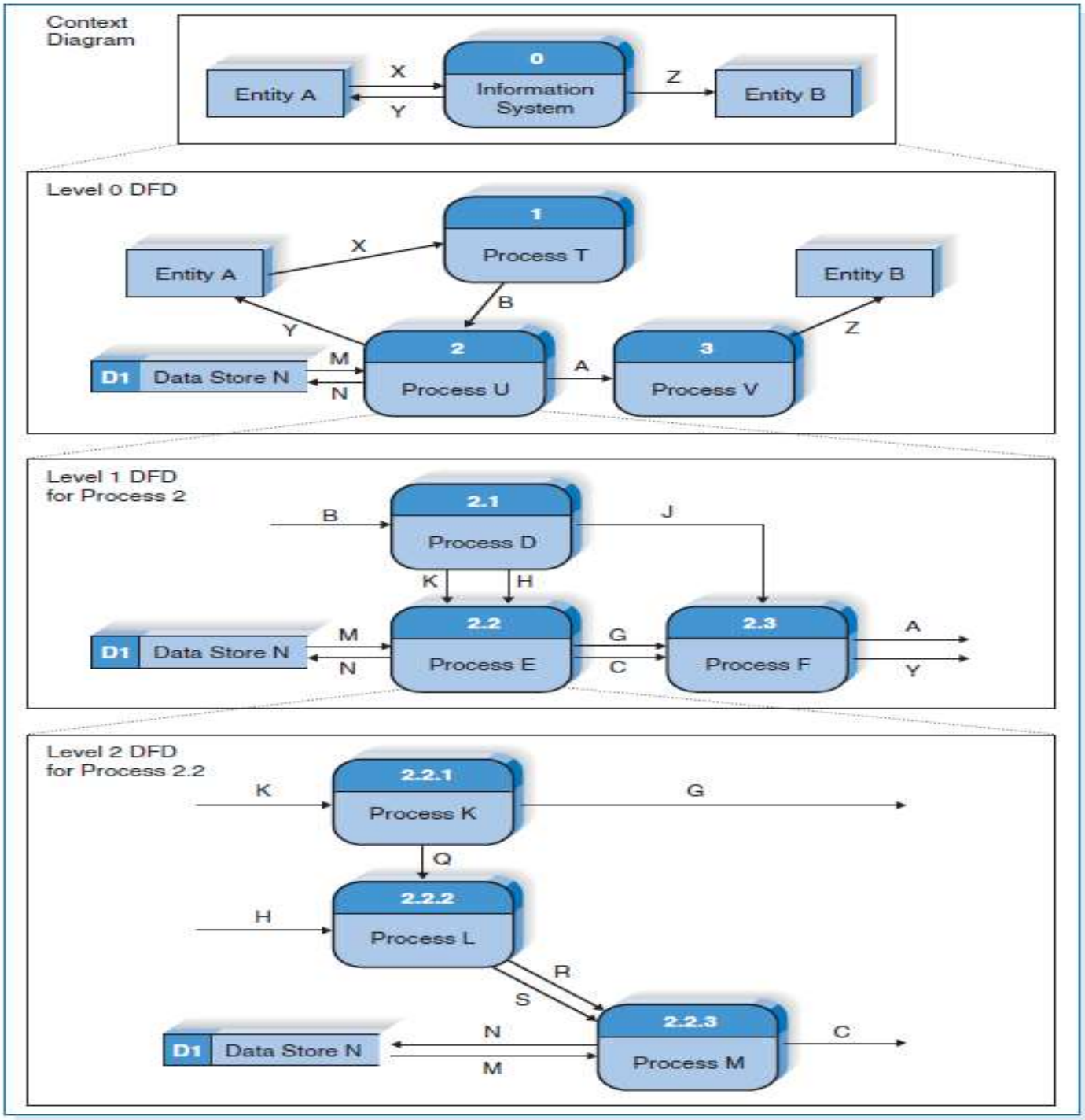
- A data store is a collection of data that is stored in some way (which is determined later when creating the physical model).
- Every data store **is named with a noun** and **is assigned an identification number and a description.**
- Data stores form the starting point for the data model (discussed in the next chapter) and are the **principal link between the process model and the data model**.
- **Data flows coming out of a data store** indicate that **information is retrieved from the data store**.
- Data flows going into a data store indicate that information is added to the data store.
- Finally, data flows going both into and out of a data store indicate that information in the data store is changed (e.g., by retrieving data from a data store, changing it, and storing it back) (e.g., Update Available Product Quantity).
- All data stores must have at least one input data flow (or else they never contain any data), unless they are created and maintained by another information system or on another page of the DFD. Likewise, they have at least one output data flow on some page of the DFD.

Elements of Data Flow Diagram (DFD)

- External Entity
  - **An external entity** is a person, organization, organization unit, or system that is external to the system, but interacts with it (e.g., customer, clearinghouse, government organization, accounting system).
  - Every **external entity has a name and a description**.
  - The external entity typically corresponds to the primary actor identified in the use case.
  - External entities provide data to the system or receive data from the system and serve to establish the system boundaries.
  - The key point to remember about an external entity is that it is external to the system, but may or may not be part of the organization. People who use the information from the system to perform other processes or who decide what information goes into the system are documented as external entities (e.g., managers, staff).

Using Data Flow Diagrams to Define Business Processes

- Most business processes are too complex to be explained in one DFD.
- one important principle in process modeling with DFDs **is the decomposition of the business process into a hierarchy of DFDs**,
- with each level down the hierarchy representing less scope but more detail.
- When a parent process is decomposed into children, its children must completely perform all of its functions.
- Another key principle in creating sets of DFDs is balancing. Balancing means ensuring that all information presented in a DFD at one level is accurately represented in the next-level DFD.



Relationships among Levels of Data Flow Diagrams (DFDs) (Figure 5-3)

Context Diagram

- The **first DFD in every business process model**, whether a manual system or a computerized system, is the **context diagram** (see Figure 5-3).
- As the name suggests, the context diagram shows the **entire system in context with its environment.**
- All process models have **one context diagram.**
- The context diagram shows the overall business process as **just one process (i.e., the system itself)** and **shows the data flows to and from external entities.**
- **Data stores usually are not included on the context diagram**, unless they are "owned" by systems or processes other than the one being documented. **Many organizations, however, would show this as an external entity not as a data store.**
- **None of the data stores** inside the process/system that are created by the process or system itself are included in the context diagram, because they are "inside" the system.
- **The DFDs combine several small inputs and outputs in the use cases into larger data flows** (e.g., combining three separate inputs, such as "customer name," "customer address," and "customer phone number," into one data flow, such as "customer information").

Level 0 Diagram

- **The next DFD is called the level 0 diagram or level 0 DFD.** (See Figure 5-3)
- The level 0 diagram shows all the processes at the first level of numbering (i.e., processes numbered 1 through 3), the data stores, external entities, and data flows among them.
- **The purpose of the level 0 DFD** is **to show** all the **major high-level processes** of the system and how they are interrelated.
- All process **models have one and only one level 0 DFD.**
- The **context diagram** deliberately **hides** some of the system's **complexity** in order to make it easier for the reader to understand.
- Only after the reader understands the context diagram does **the analyst "open up" process 0 to display its internal operations by decomposing the context diagram into the level 0 DFD, which shows more detail about the processes and data flows inside the system.**

Level 1 Diagrams

- The level 0 DFD shows only how the major high-level processes in the system interact.
- **Each process on the level 0 DFD can be decomposed into a more explicit DFD, called a level 1 diagram, or level 1 DFD, which shows how it operates in greater detail.** (See Figure 5-3) (See Figure 5-1 on the Textbook p.212)
- In general, **all process models have as many level 1 diagrams as there are processes on the level 0 diagram**; every process in the **level 0 DFD would be decomposed into its own level 1 DFD.**
- The level 0 DFD in **Figure 5-3** would have three level 1 DFDs (one for process 1, one for process 2, one for process 3). **For simplicity**, we have chosen to show only one level 1 DFD in this figure, **the DFD for process 2.**
- The **processes in level 1 DFDs are numbered on the basis of the process being decomposed**. In this example **Figure 5-3** , we are **decomposing process 2**, so the **processes in this level 1 DFD are numbered 2.1, 2.2, and 2.3**.
- Processes 2.1, 2.2, and 2.3 are the children of process 2, and process 2 is the parent of processes 2.1, 2.2, and 2.3.

Level 2 Diagrams

- The bottom of **Figure 5-3 shows** the next level of decomposition: **a level 2 diagram, or level 2 DFD**, for process 2.2.
- This DFD shows that process 2.2 is decomposed into three processes (2.2.1, 2.2.2, and 2.2.3).
- It is sometimes difficult to remember which DFD level is which. **It may help to remember that the level numbers refer to the number of decimal points in the process numbers on the DFD.**
- **level 0 DFD** has process numbers with **no decimal points (e.g., 1, 2)**, **whereas a level 1 DFD has process numbers with one decimal point (e.g., 2.3, 5.1)**, a **level 2 DFD has numbers with two decimal points (e.g.,1.2.5, 3.3.2)**, and so on.

Creating Data Flow Diagrams

- Data flow diagrams start with the information in the use cases and the requirements definition.
- Although the use cases are created by the users and project team working together, the DFDs typically are created by the project team and then reviewed by the users.
- The project team takes the use cases and rewrites them as DFDs.
- The project team sometimes has to revise some of the information in the use cases to make them conform to the DFD rules.
- The most common types of changes are to the names of the use cases that become processes and the inputs and outputs that become data flows.
- The second most common type of change is to combine several small inputs and outputs in the use cases into larger data flows in the DFDs (e.g., combining three separate inputs, such as "customer name," "customer address," and "customer phone number," into one data flow, such as "customer information").

Creating Data Flow Diagrams (cont'd)

- Project teams usually use process modeling tools or CASE tools to draw process models. Simple tools such as Visio contain DFD symbol sets and enable easy creation and modification of diagrams.
- Building a process model that has many levels of DFDs usually **entails several steps**.
  - it useful to first build the context diagram showing all the external entities and the data flows that originate from or terminate in them.
  - Second, the team creates a DFD fragment for each use case that shows how the use case exchanges data flows with the external entities and data stores.
  - Third, these DFD fragments are organized into a level 0 DFD.
  - Fourth, the team develops level 1 DFDs, based on the steps within each use case, to better explain how they operate. In some cases, these level 1 DFDs are further decomposed into level 2 DFDs, level 3 DFDs, level 4 DFDs, and so on.
  - Fifth, the team validates the set of DFDs to make sure that they are complete and correct

In the following sections, process modeling is illustrated with the Holiday Travel Vehicles information system.

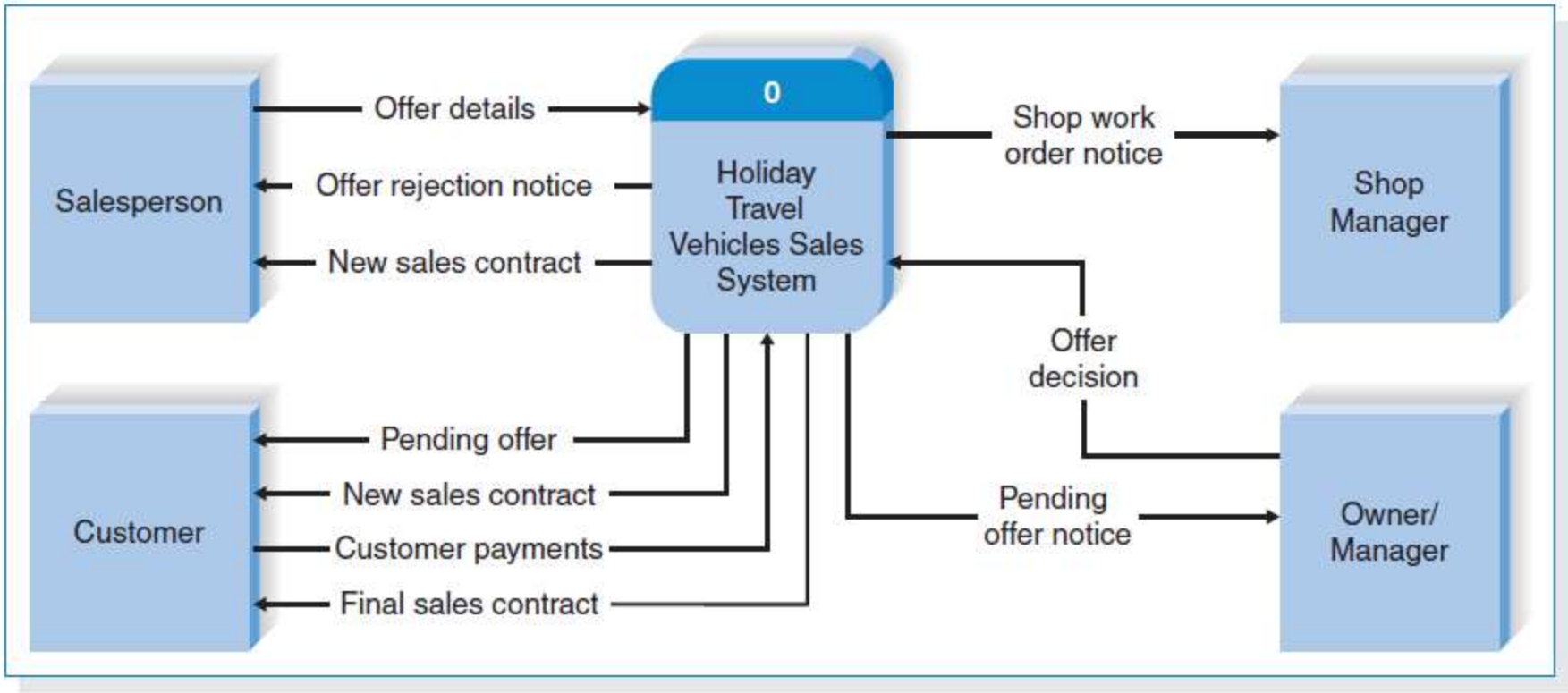
Create Context Diagram

- To create the context diagram, you simply **draw one process symbol** for the **business process or system** being modeled (**numbered 0 and named for the process or system**).
- You read through the **use cases** and add the **inputs and outputs listed on the form**, as well as their sources and destinations.
- Usually, **all the inputs and outputs** will come from or go to **external entities such as a person, organization, or other information system**.
- If any inputs and outputs connect directly to data stores in an external system, **it is best practice to create an external entity which is named for the system that owns the data store.**
- The Next slide has a Sample of the Context Diagram.

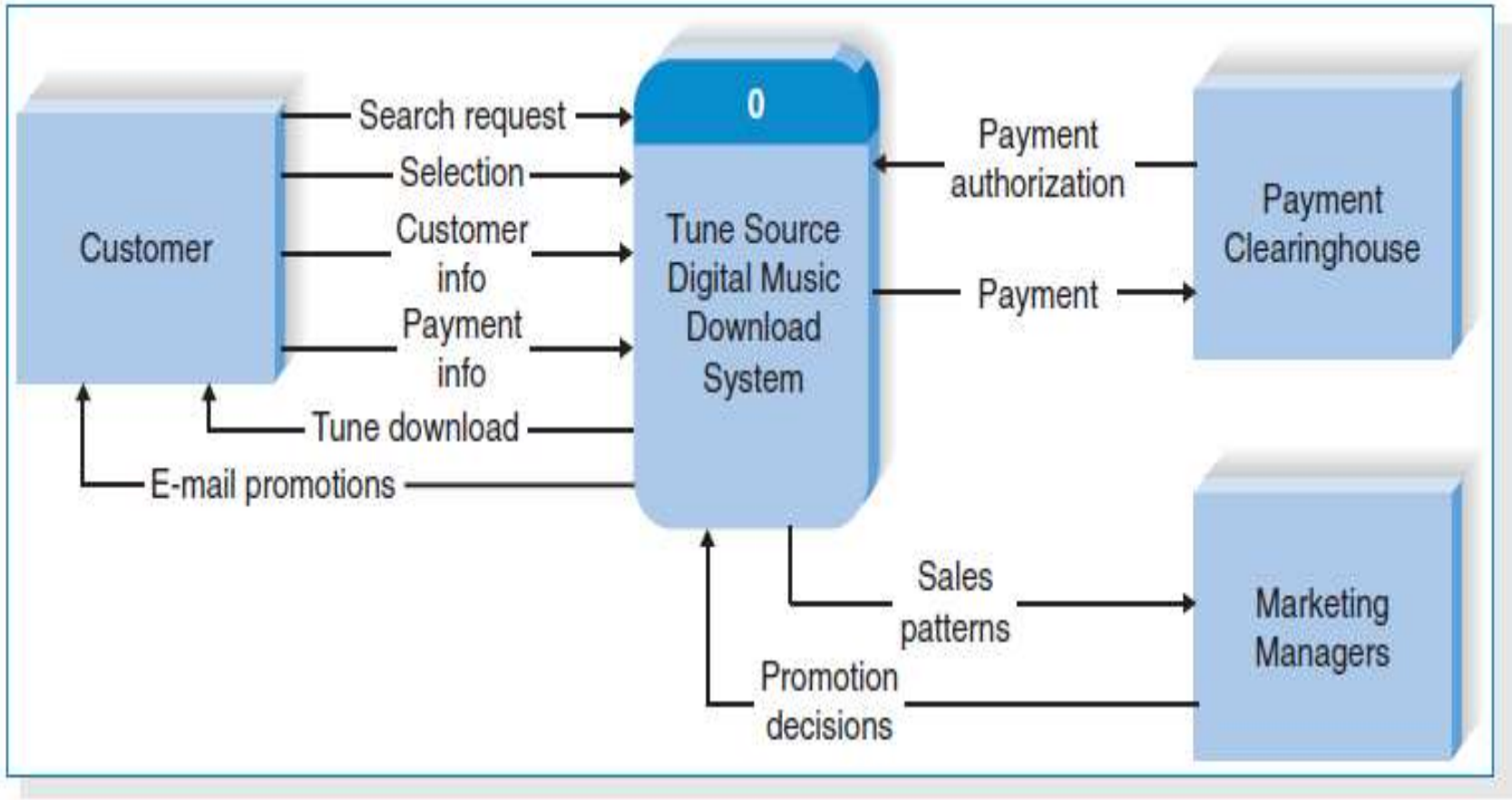
Context Diagram For The Holiday Travel Vehicles System

- Figure 5-5 shows the context diagram for the Holiday Travel Vehicles system focusing on vehicle sales. Take a moment to review this system as described in Chapter 4 and review the use cases in Figure 4-11. You can see from the Major Inputs and Outputs sections in the Figure 4-11 use cases that the system has many interactions with the salesperson external entity. We have simplified these inflows and outflows to just three primary data flows on the context diagram. If we had included each small data flow, the context diagram would become too cluttered. The smaller data flows will become evident as we decompose the context diagram into more detailed levels. Notice that we have established three external entities to represent parts of the Holiday Travel Vehicle organization which receive information from or supply information to this system. Salespeople provide key inputs to the system, and shop work orders flow to the shop manager. The company owner or manager provides information to the system.

Holiday Travel Vehicles Sales System Context Diagram



Tune Source Context Diagram



Creating Data Flow Diagram Fragments

- ***A DFD fragment*** is one part of a DFD that eventually will be combined with other DFD fragments to form a DFD diagram.
- In this step, each use case is converted into one DFD fragment. You start by taking each use case and drawing a DFD fragment, using the information given on the top of the use case: the name, ID number, and major inputs and outputs.
- The information about the major steps that make up each use case is ignored at this point; it will be used in a later step. Figure 5-6 shows a use case and the DFD fragment that was created from it.
- **There are no formal rules covering the layout of processes, data flows, data stores, and external entities within a DFD. Most systems analysts try to put the process in the middle of the DFD fragment, with the major inputs starting from the left side or top entering the process and outputs leaving from the right or the bottom. Data stores are often written below the process.**

FIGURE 5-6

Holiday Travel Vehicles Process 3 (Record Offer) DFD Fragment

Use Case Name: Record an offer	ID: UC-3	Priority: High

Actor: Salesperson

Description: This use case describes how the salesperson records a customer offer on a vehicle. The offer may be a new offer or a revision of a previously rejected offer.

Trigger: Customer decides to make an offer on a vehicle.

Type: ☑ External ☐ Temporal

Summary Inputs	Source	Outputs	Destination
Vehicle ID Existing Pending Offer Offer Type Offer ID Previous offer details Vehicle datastore Customer details Offer details	Salesperson Pending Offers datastore Salesperson Salesperson Rejected Offers datastore Vehicle details Customer Salesperson	Offer Pending Notice Offer Summary New Pending Offer Pending Offer Pending Offer Notice	Salesperson Customer Pending Offer datastore Customer Owner/Manager

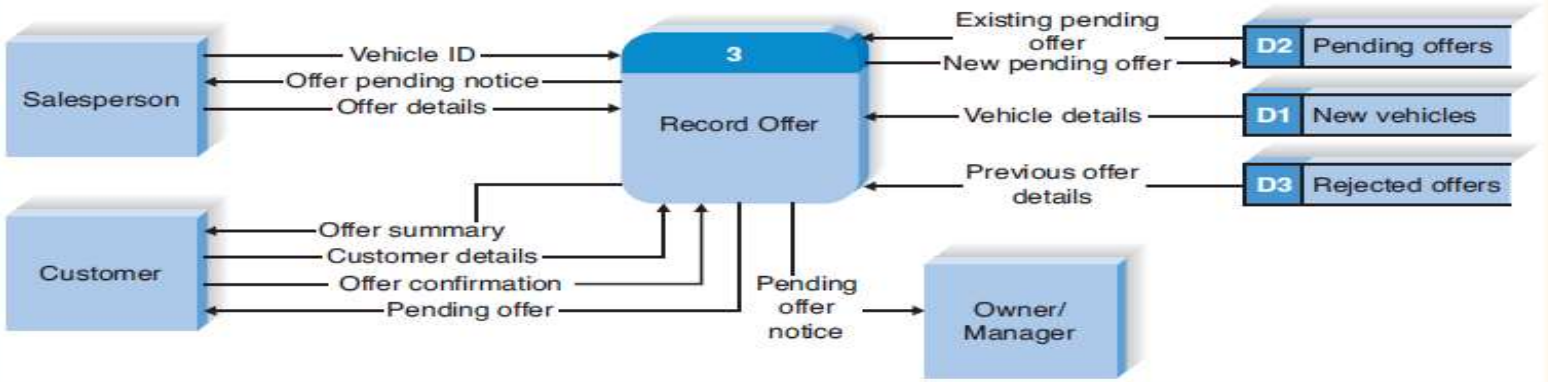
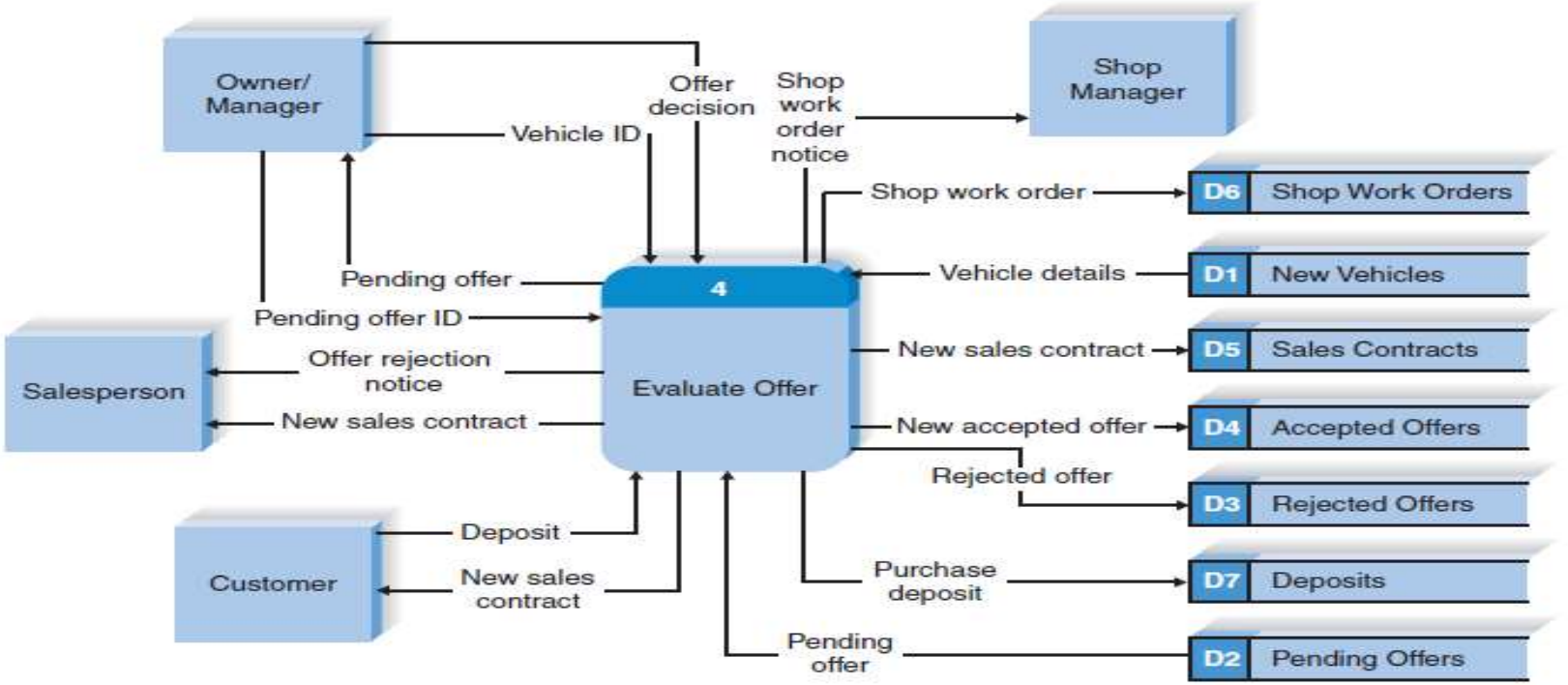


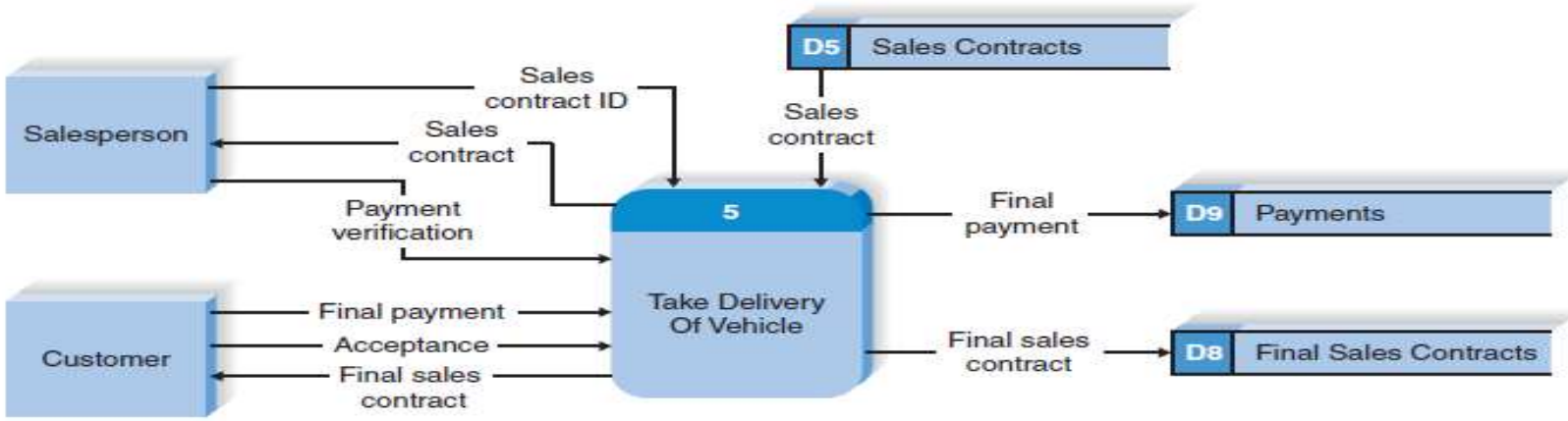
FIGURE 5-7

Additional DFD Fragments for Holiday Travel Vehicles

- Take a moment and draw a DFD fragment for the two other use cases shown in Figure 4-11 (Evaluate Offer and Take Delivery of Vehicle). We have included possible ways of drawing these fragments in Figure 5-7. (Don't look until you've attempted the drawings on your own!)



(a) *Process 4 DFD Fragment*



(b) *Process 5 DFD Fragment*

Creating the Level 0 Data Flow Diagram

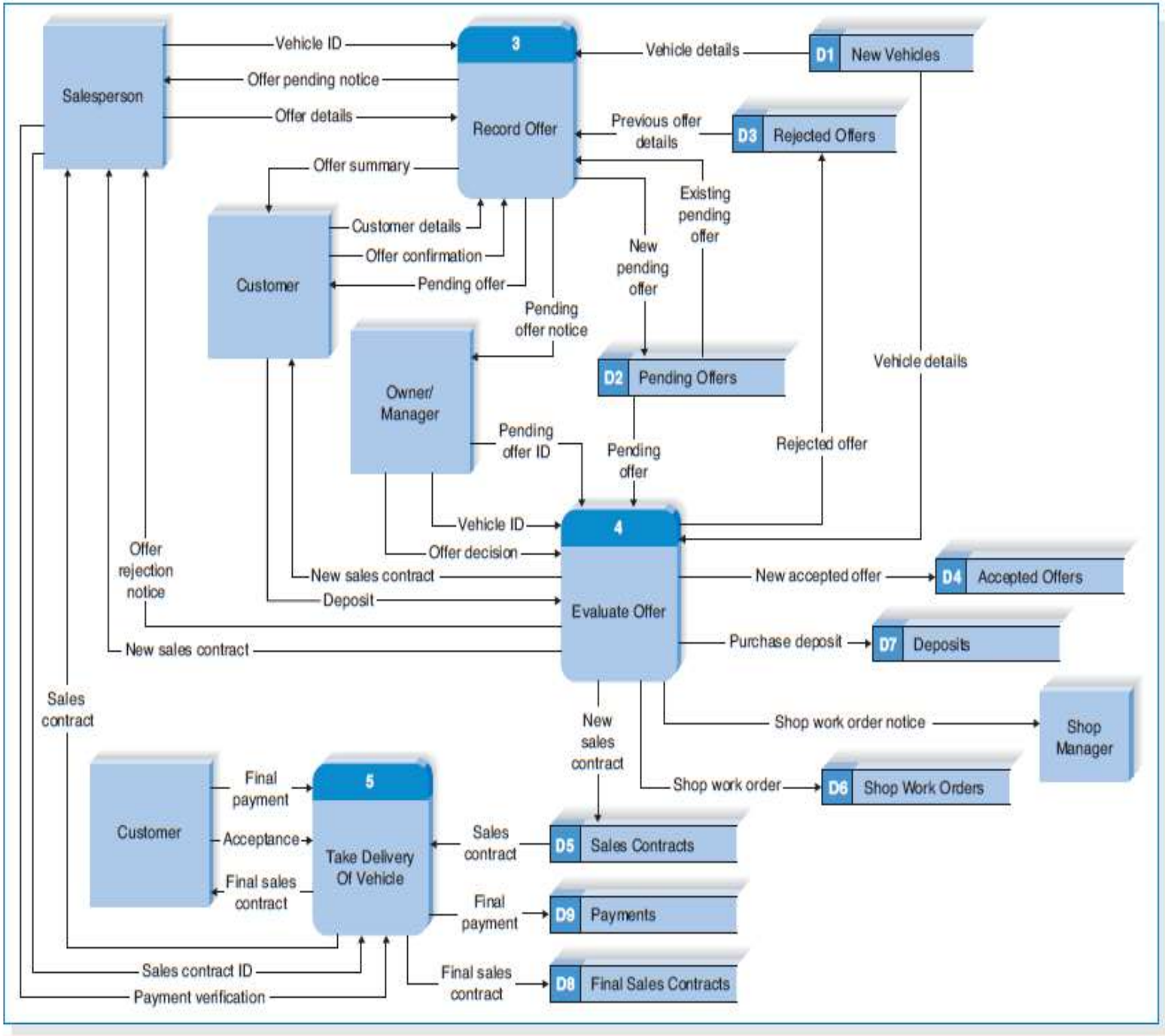
- Once you have the set of DFD fragments (one for each of the major use cases), you combine them into one DFD drawing that becomes the level 0 DFD.
- As mentioned earlier, there are no formal layout rules for DFDs. Most systems analysts try to put the process that is first chronologically in the upper left corner of the diagram and work their way from top to bottom, left to right (e.g., Figure 5-1).
- Generally speaking, most analysts try to reduce the number of times that data flow lines cross or to ensure that when they do cross, they cross at right angles so that there is less confusion. (Many give one line a little "hump" to imply that one data flow jumps over the other without touching it.) Minimizing the number of data flows that cross is challenging.
- Iteration is the cornerstone of good DFD design.
- Even experienced analysts seldom draw a DFD perfectly the first time.
- they draw it once to understand the pattern of processes, data flows, data stores, and external entities and then draw it a second time on a fresh sheet of paper (or in a fresh file) to make it easier to understand and to reduce the number of data flows that cross.
- Often, a DFD is drawn many times before it is finished.

Creating the Level 0 Data Flow Diagram

- Once you have the set of DFD fragments (one for each of the major use cases), you combine them into one DFD drawing that becomes the level 0 DFD.
- In this DFD you will add data stores.
- Level 0 diagram shows the major process within the system, and major process within external entities, which are the sources or destination of data flows.

- Try to out the first chronologically processin the upper left corner of the diagram and work their way from top to bottom, left to right (e.g., Figure 5-1).

Figure 5-8 combines the DFD fragments in Figures 5-6 and 5-7. Take a moment to examine Figure 5-8 and find the DFD fragments from Figures 5-6 and 5-7 contained within it.


Salesperson
Vehicle ID
Offer pending notice
Offer details
3
Record Offer
Vehicle details
D1 New Vehicles
Previous offer details
D3 Rejected Offers
Offer summary
Existing pending offer
Customer
Customer details
Offer confirmation
Pending offer
New pending offer
Pending offer notice
Owner/ Manager
D2 Pending Offers
Vehicle details
Pending offer ID
Pending offer
Rejected offer
Vehicle ID
4
Offer decision
Offer rejection notice
New sales contract
Deposit
Evaluate Offer
New accepted offer
D4 Accepted Offers
Purchase deposit
D7 Deposits
New sales contract
Sales contract
New sales contract
Shop work order notice
Shop Manager
Shop work order
D6 Shop Work Orders
Customer
Final payment
Acceptance
Final sales contract
5
Take Delivery Of Vehicle
Sales contract
D5 Sales Contracts
Final payment
D9 Payments
Final sales contract
D8 Final Sales Contracts
Sales contract ID
Payment verification

Creating Level 1 Data Flow Diagrams (and Below)

- The team now begins to create lower-level DFDs for each process in the level 0 DFD that needs a level 1 DFD.
- Each one of the use cases is turned into its own DFD. The process for creating the level 1 DFDs is to take the steps as written on the use cases and convert them into a DFD in much the same way as for the level 0 DFD.
- Usually, each major step in the use case becomes a process on the level 1 DFD, with the inputs and outputs becoming the input and output data flows.
- Once again, however, sometimes subtle changes are required to go from the informal descriptions in the use case to the more formal process model, such as adding input data flows that were not included in the use case.
- And because the analysts are now starting to think more deeply about how the processes will be supported by an information system, they sometimes slightly change the use case steps to make the process easier to use.

FIGURE 5-9
Holiday Travel Vehicles Process 3 (Record Offer) Level 1 DFD

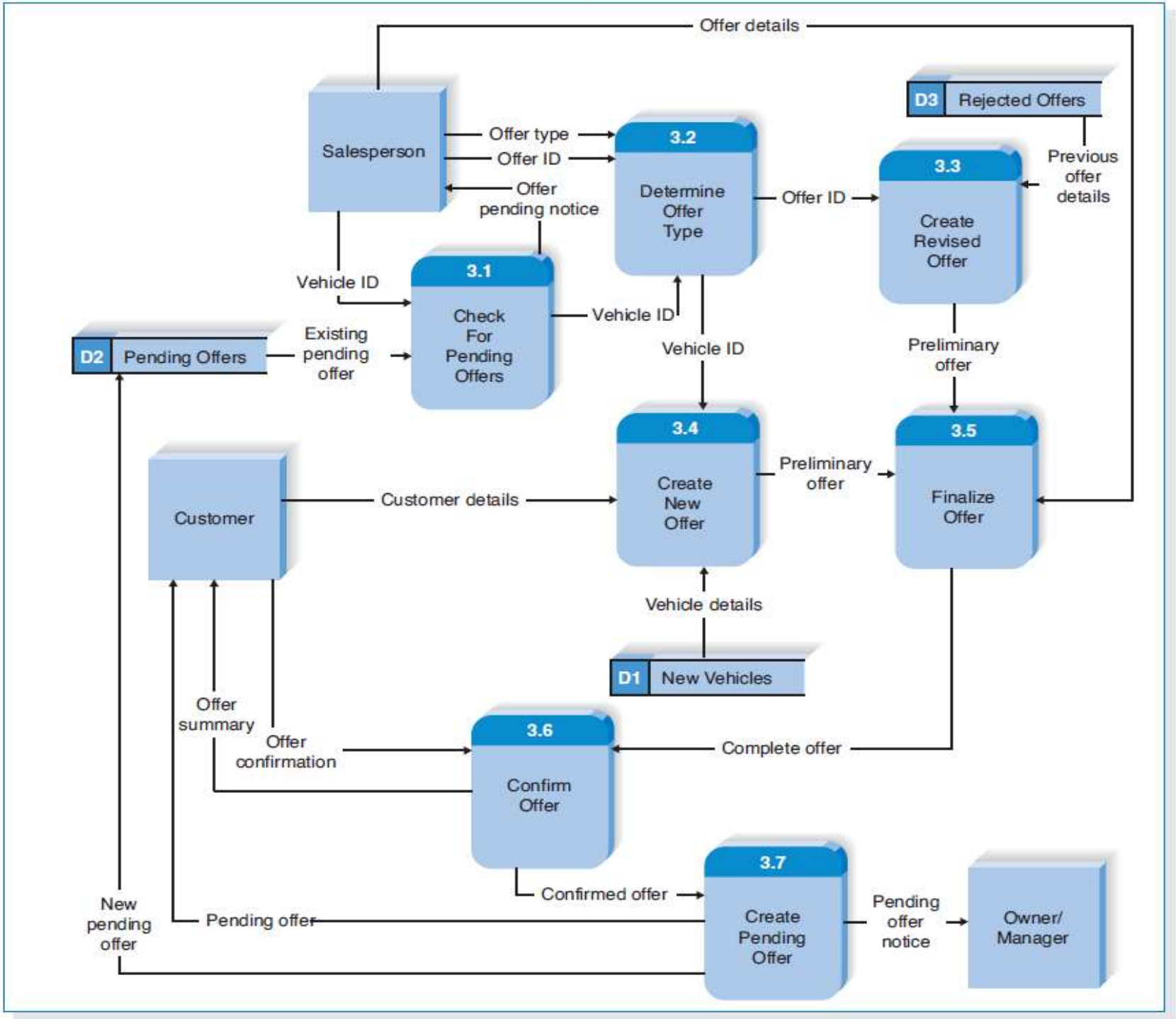
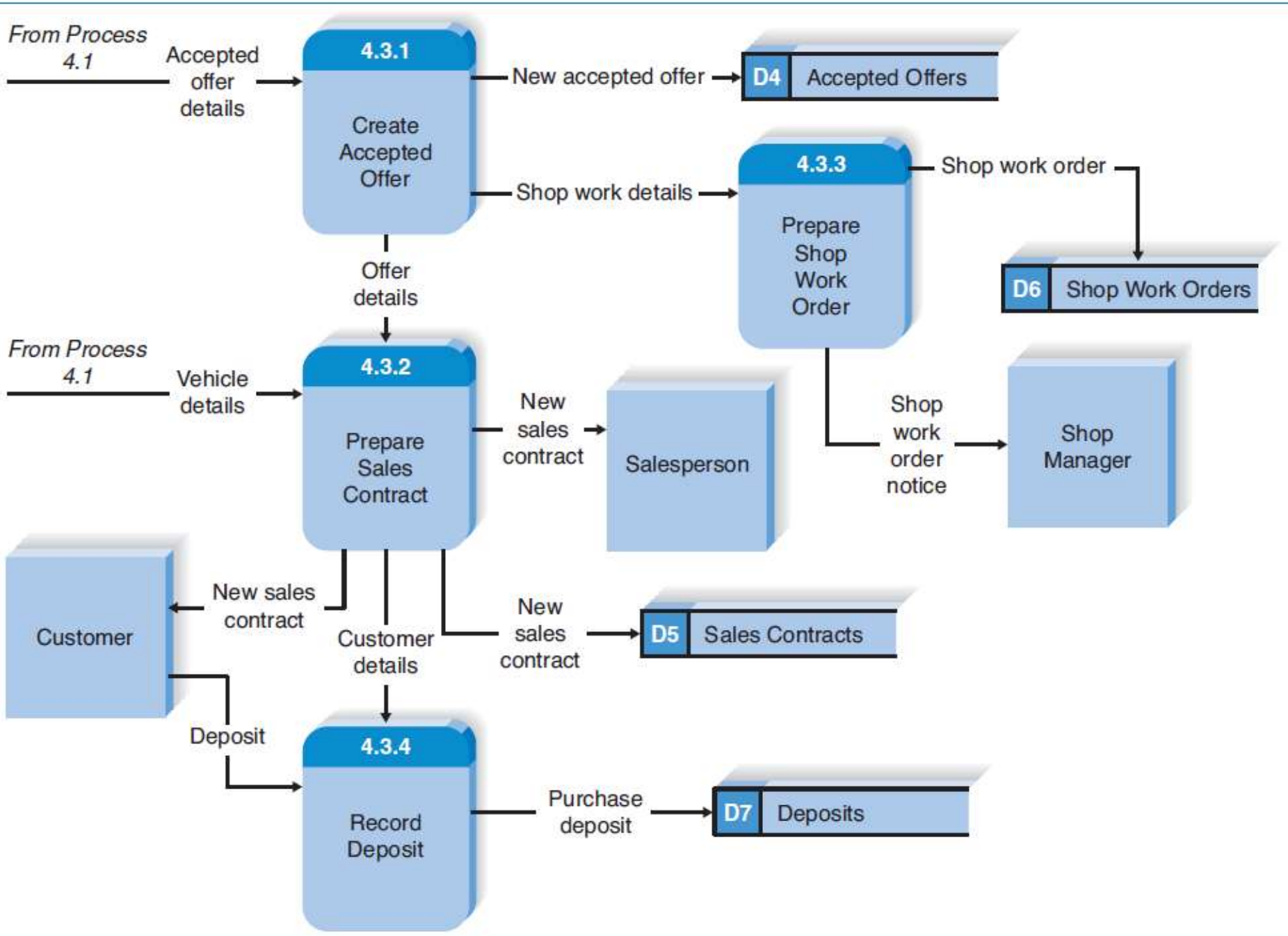

Offer details
Salesperson
Offer type
Offer ID
Offer pending notice
3.2
Determine Offer Type
Offer ID
3.3
Create Revised Offer
D3
Rejected Offers
Previous offer details
Vehicle ID
3.1
Check For Pending Offers
Vehicle ID
D2
Pending Offers
Existing pending offer
Vehicle ID
Preliminary offer
3.4
Create New Offer
Preliminary offer
3.5
Finalize Offer
Customer
Customer details
Vehicle details
D1
New Vehicles
Offer summary
Offer confirmation
3.6
Confirm Offer
Complete offer
Confirmed offer
3.7
Create Pending Offer
Pending offer notice
Owner/ Manager
New pending offer
Pending offer

FIGURE 5-11
Holiday Travel Vehicles Process 4.3 (Process Accepted Offers) Level 2 DFD


From Process 4.1
Accepted offer details
4.3.1
Create Accepted Offer
New accepted offer
D4
Accepted Offers
Shop work details
4.3.3
Prepare Shop Work Order
Shop work order
D6
Shop Work Orders
Offer details
From Process 4.1
Vehicle details
4.3.2
Prepare Sales Contract
New sales contract
Salesperson
Shop work order notice
Shop Manager
Customer
New sales contract
Customer details
New sales contract
D5
Sales Contracts
Deposit
4.3.4
Record Deposit
Purchase deposit
D7
Deposits

Validating the Data Flow Diagrams

- There are two fundamentally different types of problems that can occur in DFDs: syntax errors and semantics errors.
- "Syntax error" refers to the structure of the DFDs and whether the DFDs follow the rules of the DFD language.
- "Semantics error" refers to the meaning of the DFDs and whether they accurately describe the business process being modeled.
- The following table provide a quick checklist for identifying the most common errors.

5.3.4.1 For each DFD:

- Check each data flow for:
  - A unique name: noun; description
  - Connects to at least one process
  - Shown in only one direction(no two-headed arrows)
  - A minimum number of crossed lines

- Check each data store for:
  - A unique name: noun; description
  - at least one input data flow
  - at least one output data flow
- Check each external entity for:
  - A unique name: noun; description
  - at least one input or output data flow

Across DFDs

- Context diagram
  - Every set of DFDs must have one context diagram.
- Viewpoint
  - There is a consistent viewpoint for the entire set of DFDs.
- Decomposition
  - Every process is wholly and completely described by the processes on its children DFDs.
- Balance
  - Every data flow, data store, and external entity on a higher level DFD is shown on the lower-level DFD that decomposes it.

Figure 5-14 shows some common syntax errors. (p.235)

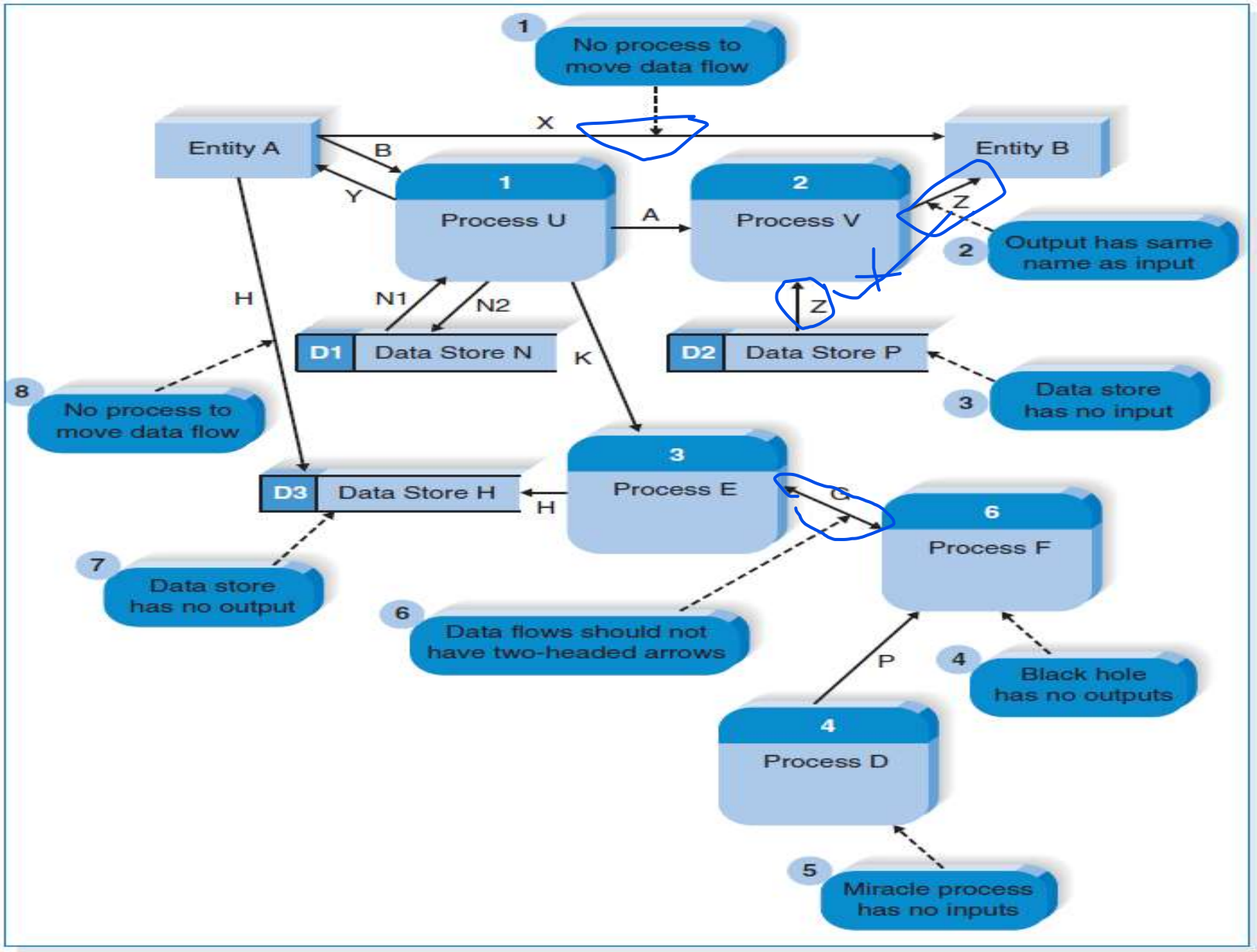
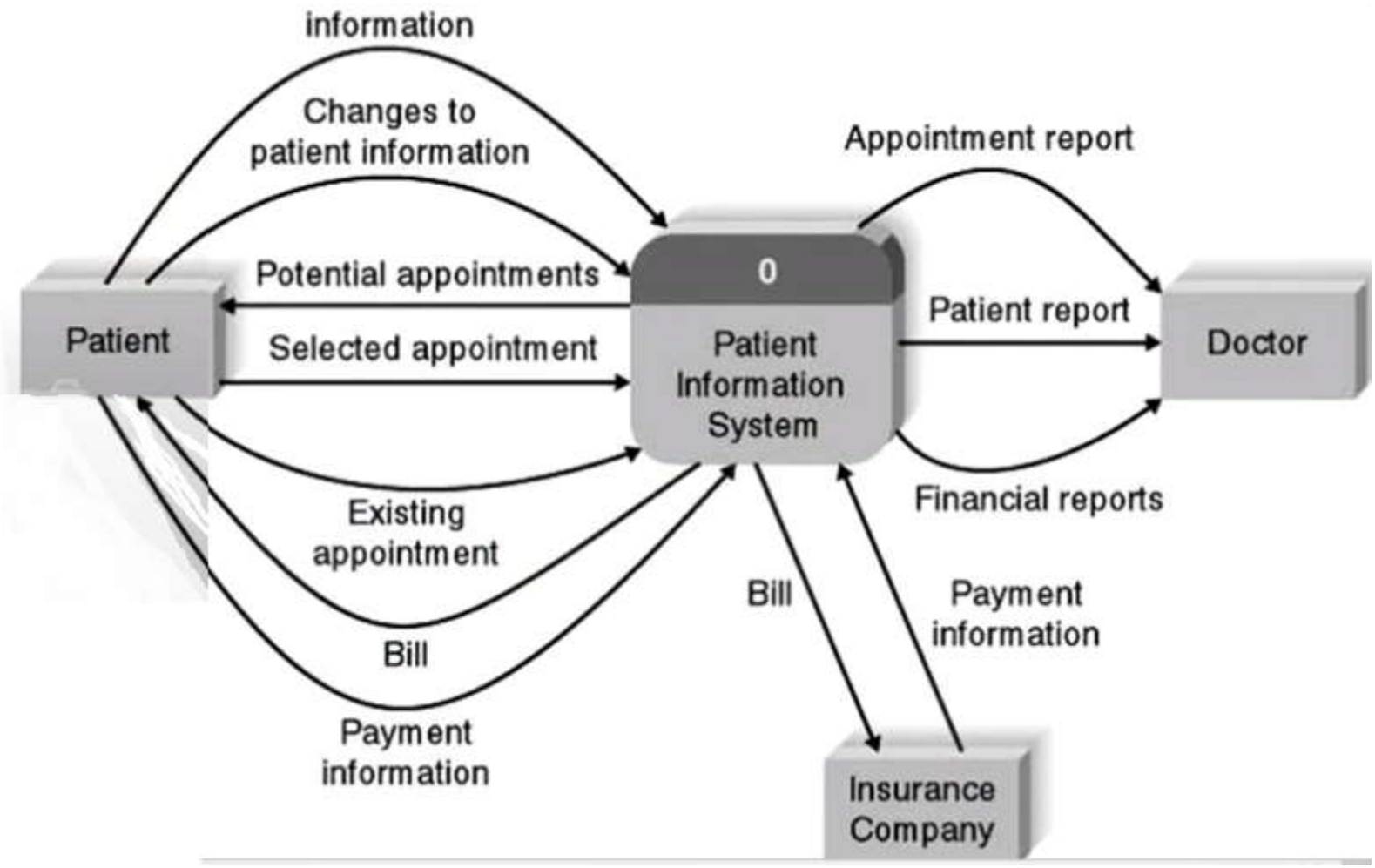
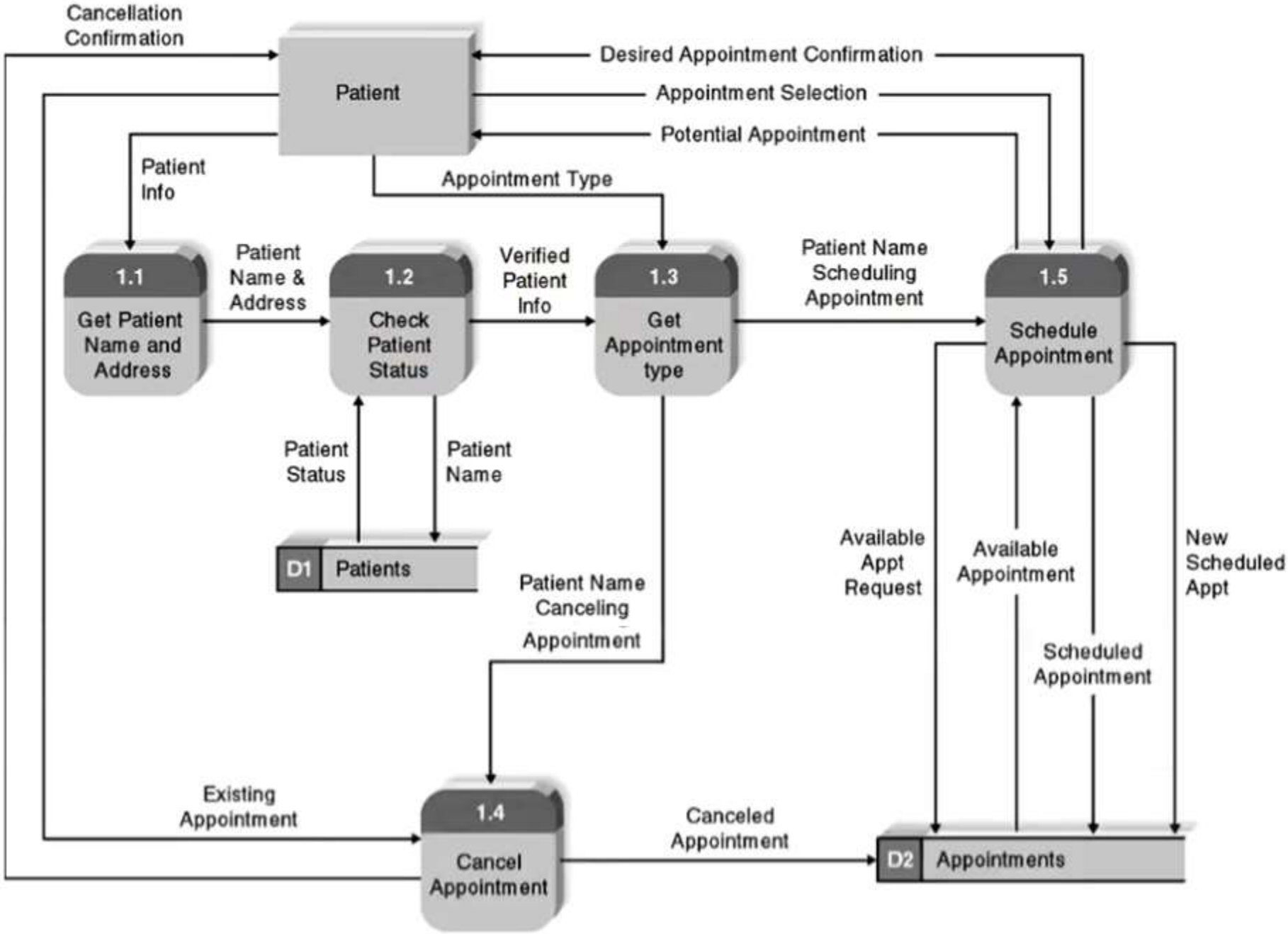


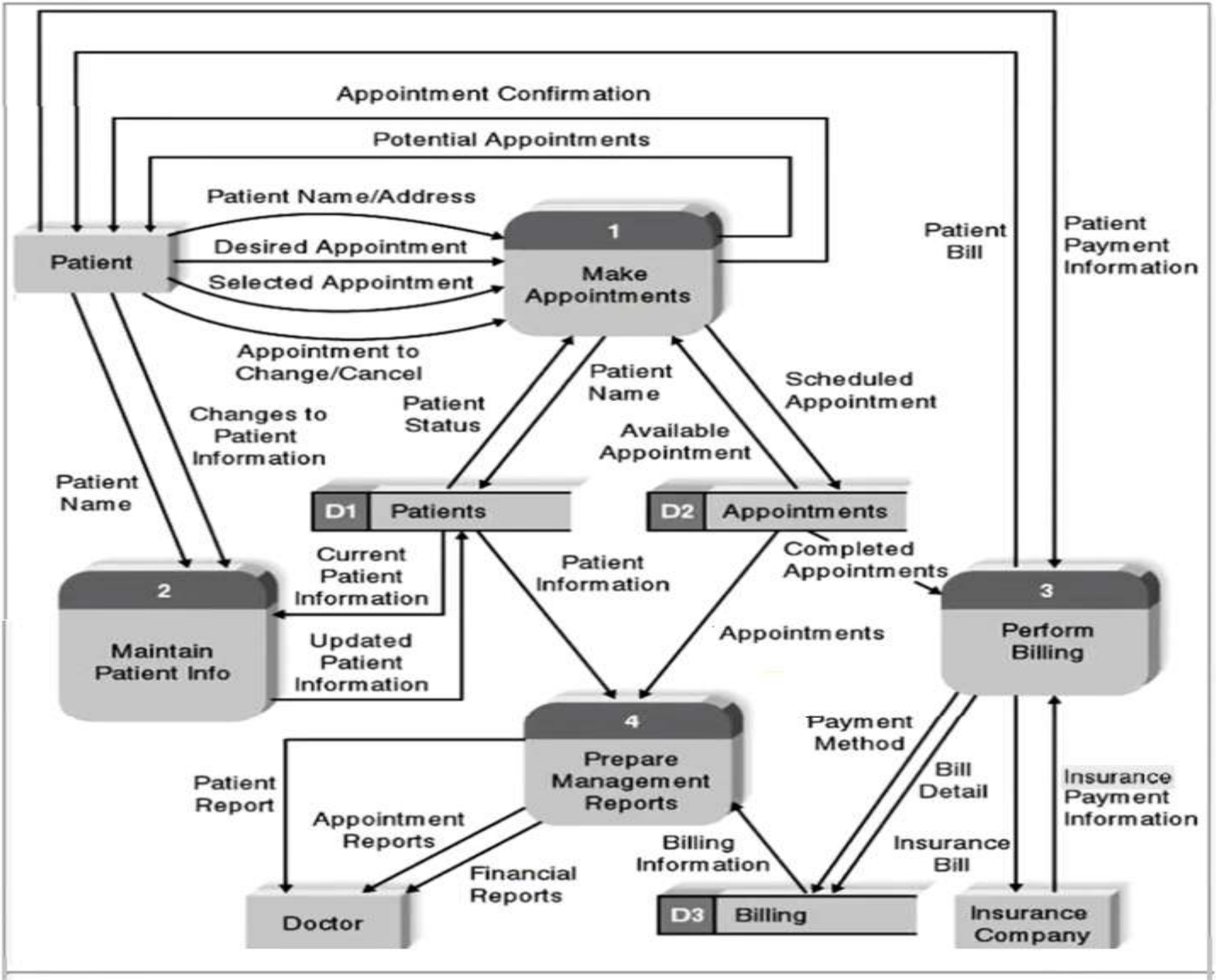
Figure 5-14 shows some common syntax errors. (p.235)

- **First**, we can see data flow X drawn directly from Entity A to Entity B. Remember that a data flow must either originate or terminate at a process; therefore, a process is required.
- **Second**, we see data flow Z retrieved from Data Store P and sent to Entity B. Processes should exist to transform the data in some way, so we usually modify the data flow names to reflect the changes made in the process.
- **Third**, we see that Data Store P has outputs but has no inputs.
- **Fourth**, we can see that Process F receives data but has no outputs. This is considered a black hole since data is received but nothing is produced.
- **Fifth**, Process D is shown producing a data flow but has no inputs. This is termed a miracle process.
- **Sixth**, we see a two-headed arrow depicting Data Flow G between Process E and Process F. Data flows should not be drawn this way, but should flow only in one direction.
- **Seventh**, Data Store H receives Data Flow H as an input, but has no outputs. should be followed up by the analyst to ensure that the data that is stored in Data Store H is used some place in the process model; otherwise, there is no reason to store it.
- **Finally**, we see that a process should be involved between Entity A and Data Store H.

Holiday Travel Vehicles Sales System Context Diagram




Cancellation Confirmation
Patient
Desired Appointment Confirmation
Appointment Selection
Potential Appointment
Patient Info
Appointment Type
1.1
Get Patient Name and Address
Patient Name & Address
1.2
Check Patient Status
Verified Patient Info
1.3
Get Appointment type
Patient Name Scheduling Appointment
1.5
Schedule Appointment
Patient Status
Patient Name
D1
Patients
Patient Name Canceling Appointment
Available Appt Request
Available Appointment
New Scheduled Appt
Scheduled Appointment
Existing Appointment
1.4
Cancel Appointment
Canceled Appointment
D2
Appointments


Appointment Confirmation
Potential Appointments
Patient Name/Address
Patient
Desired Appointment
Selected Appointment
1
Make Appointments
Patient Bill
Patient Payment Information
Appointment to Change/Cancel
Changes to Patient Information
Patient Status
Patient Name
Scheduled Appointment
Available Appointment
Patient Name
D1 Patients
D2 Appointments
Current Patient Information
Patient Information
Completed Appointments
2
Maintain Patient Info
Updated Patient Information
Appointments
3
Perform Billing
4
Prepare Management Reports
Payment Method
Bill Detail
Insurance Payment Information
Patient Report
Appointment Reports
Financial Reports
Billing Information
Insurance Bill
Doctor
D3 Billing
Insurance Company