Chapter 5. Process Modeling

Chapter 5 Outline

- Data flow diagrams.
  - Reading data flow diagrams
  - Elements of data flow diagrams
  - Using data flow diagrams to define business processes
  - Process descriptions
- Creating data flow diagrams.

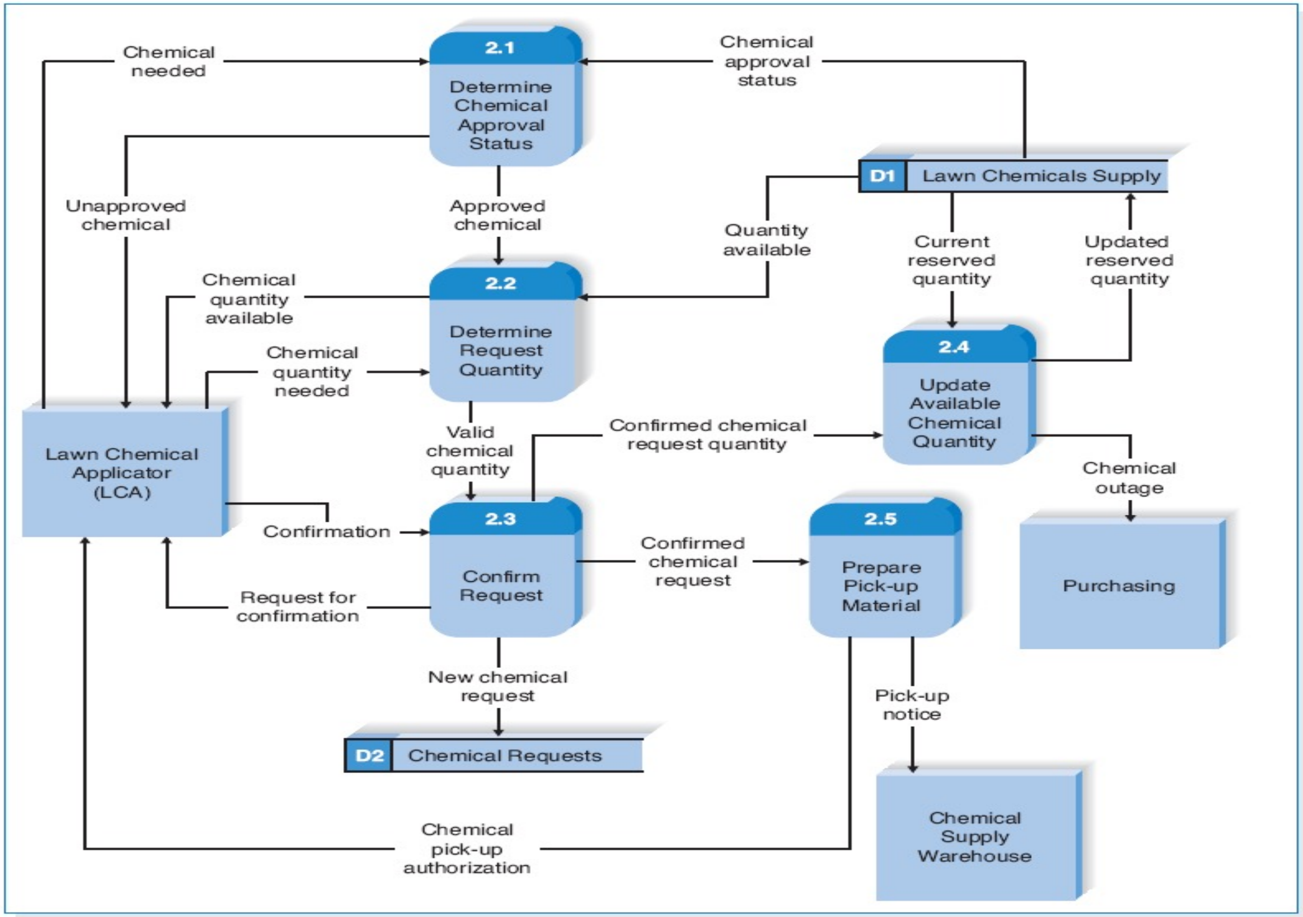
INTRODUCTION

- A process model can be used to further clarify the requirements definition and use cases.
- A process model is a graphical way of representing how a business system should operate.
- A process model can be used to document the as-is system or the to-be system, whether computerized or not.

(cont’d)

- **Data flow diagramming** is a technique that diagrams the business processes and the data that pass among them.
- **Logical process models** describe processes without suggesting how they are conducted.
- **Physical process models** provide information that is needed to build the system.

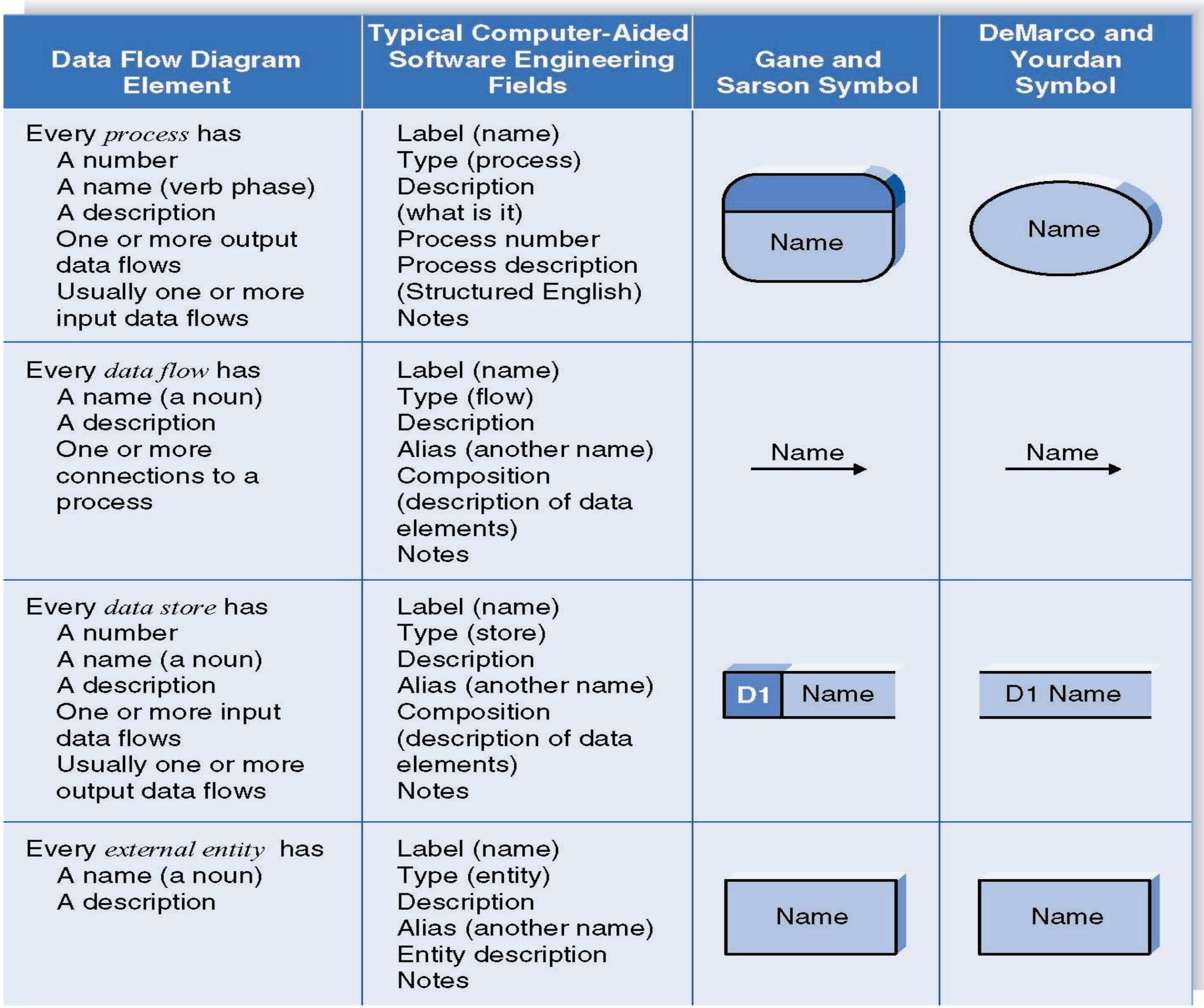
Reading Data Flow Diagrams



Elements of Data Flow Diagrams

- **Process** – A process is an activity or a function performed for some specific business reason.
- **Data Flow** – A data flow is a single piece of data, or a logical collection of several pieces of information.
- **Data Store** – A data store is a collection of data that is stored in some way.
- **External Entity** – An external entity is a person, organization, organization unit, or system that is **external** to the system, but interacts with it.

(cont'd)

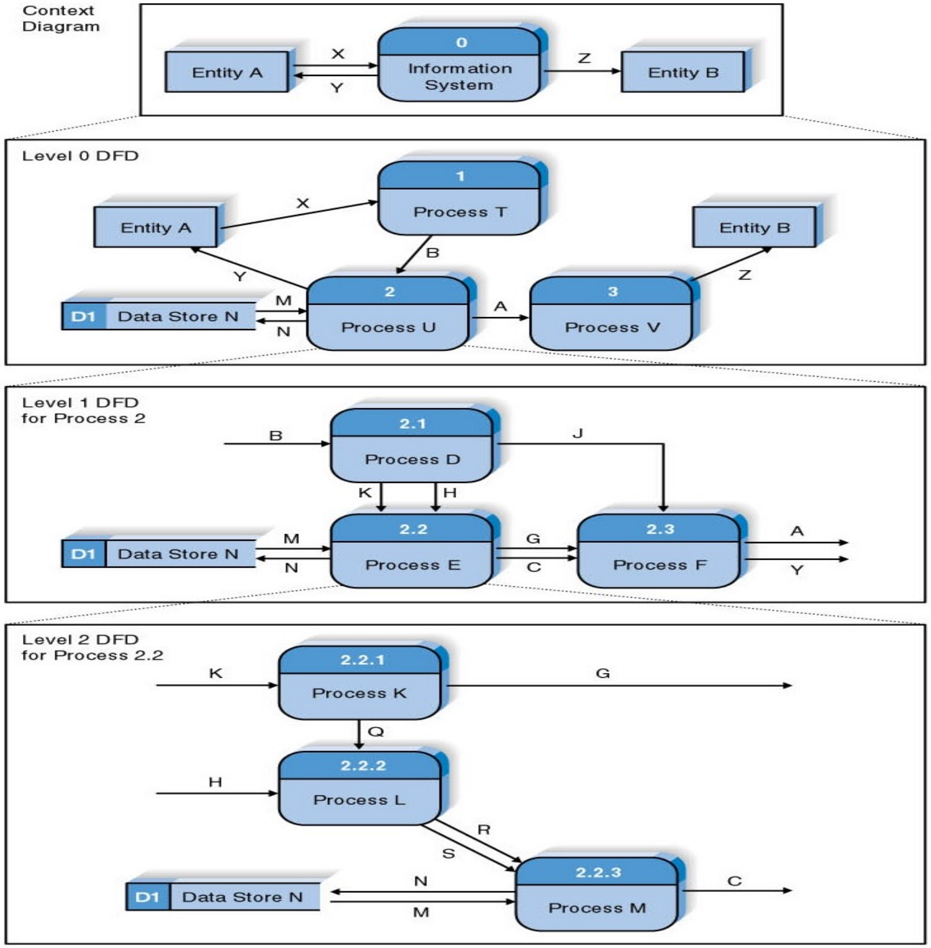


Data Flow Diagram Element	Typical Computer-Aided Software Engineering Fields	Gane and Sarson Symbol	DeMarco and Yourdan Symbol
Every *process* has A number A name (verb phase) A description One or more output data flows Usually one or more input data flows	Label (name) Type (process) Description (what is it) Process number Process description (Structured English) Notes	Name	Name
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	Name →	Name →
Every *data store* has A number A name (a noun) A description One or more input data flows Usually one or more output data flows	Label (name) Type (store) Description Alias (another name) Composition (description of data elements) Notes	D1 Name	D1 Name
Every *external entity* has A name (a noun) A description	Label (name) Type (entity) Description Alias (another name) Entity description Notes	Name	Name

Using DFDs to Define Business Processes

- 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 series of DFDs, each representing a lower level of detail.

(cont'd)


Context Diagram
Entity A
X
Y
0
Information System
Z
Entity B
Level 0 DFD
Entity A
X
1
Process T
Entity B
B
Y
Z
D1
Data Store N
M
N
2
Process U
A
3
Process V
Level 1 DFD for Process 2
B
2.1
Process D
J
K
H
D1
Data Store N
M
N
2.2
Process E
G
C
2.3
Process F
A
Y
Level 2 DFD for Process 2.2
K
2.2.1
Process K
G
Q
H
2.2.2
Process L
R
S
D1
Data Store N
N
M
2.2.3
Process M
C

Context Diagram

- The first DFD in every business process is the **context diagram**.
- It shows the entire system in **context** with its environment.
- The context diagram shows the overall business process as just ***one*** process and shows the data flows to and from external entities.

Level 0 Diagram

- The **level 0 diagram** (or **level 0 DFD**) shows all the major high-level processes of the system and how they are interrelated.
- The Level 0 diagram shows all the *processes* at the first level the numbering, the *data stores*, external entities, and data flows among them.
- A key concept: ***Balancing***

  - Ensuring that all information presented in a DFD at one level is accurately represented in the next-level DFD.
- A process model has one and only one level 0 DFD.

Level 1 Diagrams

- Each process on the level 0 DFD can be ***decomposed*** into a more explicit DFD called **level 1 diagram** (or **level 1 DFD**).
- The set of ***children*** and the ***parent*** are identical; they are simply different ways of looking at the same thing.
- It is important to ensure that level 0 and level 1 DFDs are balanced.

(cont’d)

- All process models have as many level 1 diagrams as there are processes on the level 0 diagram.
- The parent process and the children processes are numbered consistently.

Level 2 Diagrams

- The next level of decomposition: a ***level 2 diagram***, or **level 2 DFD**.
- A level 2 DFD shows all processes, data flows, and data stores that comprise a single process on the level 1 diagram.
- It is important to ensure that level 1 and level 2 DFDs are balanced.

Alternative Data Flows

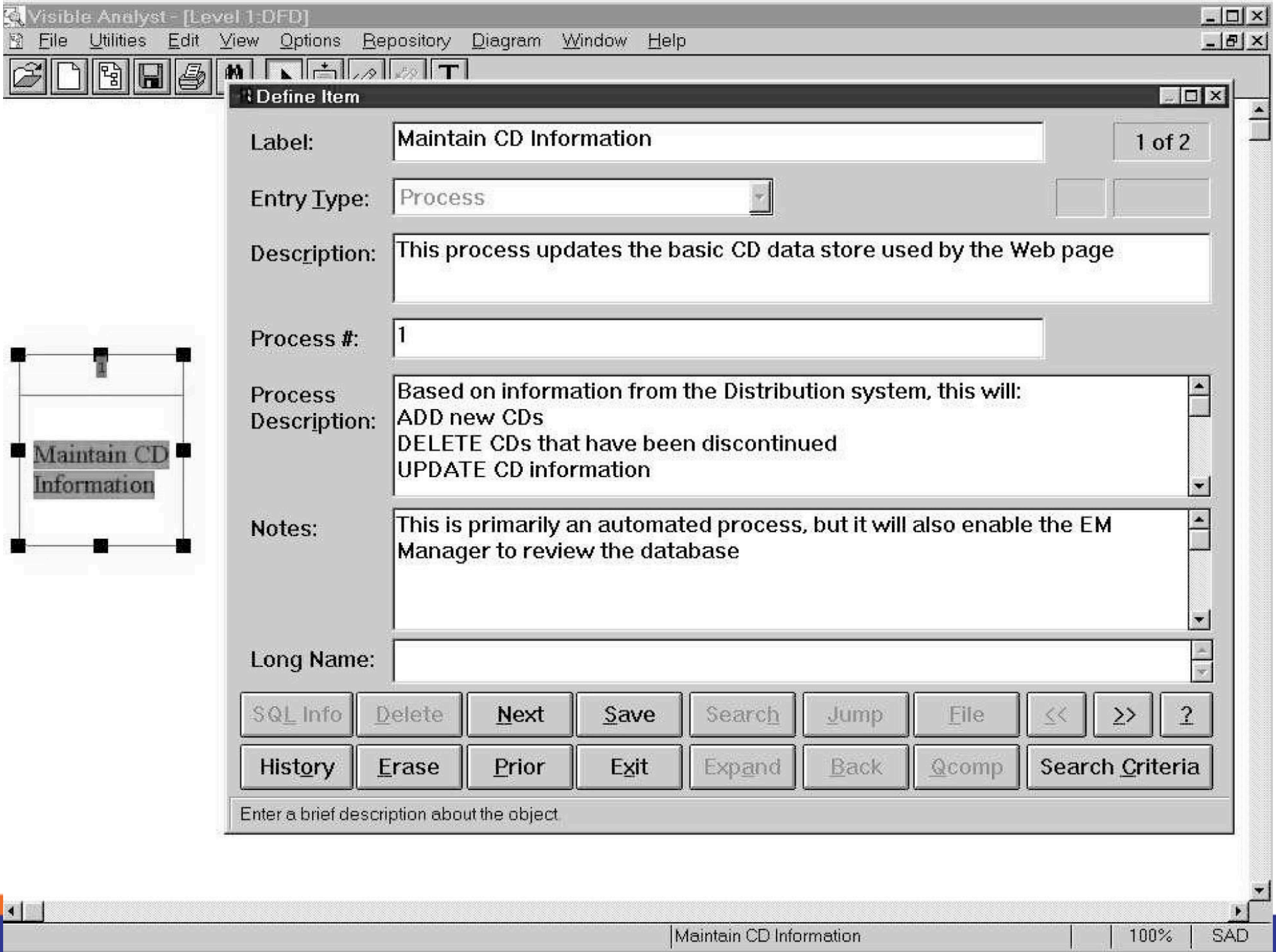
- A process can produce different data flows under different circumstance.
- We show both data flows and use the **process description** to explain why they are alternatives.

Process Descriptions

- The purpose of the **process descriptions** is to explain what the process does and provide additional information that the DFD does not provide.
- Three techniques are commonly used to describe more complex processing logic:
  - Structured English
  - Decision trees
  - Decision tables

CREATING DATA FLOW DIAGRAMS

- DFDs start with the information in the use cases and the requirements definition.
- Generally, the set of DFDs integrates the individual use cases.
- The project team takes the use cases and rewrites them as DFDs, following the DFD formal rules about symbols and syntax.
- CASE tools are used to draw process models.



Label: Maintain CD Information

Entry Type: Process

Description: This process updates the basic CD data store used by the Web page

Process #: 1

Process Description: Based on information from the Distribution system, this will:
ADD new CDs
DELETE CDs that have been discontinued
UPDATE CD information

Notes: This is primarily an automated process, but it will also enable the EM Manager to review the database

Long Name:

Enter a brief description about the object

(cont'd)

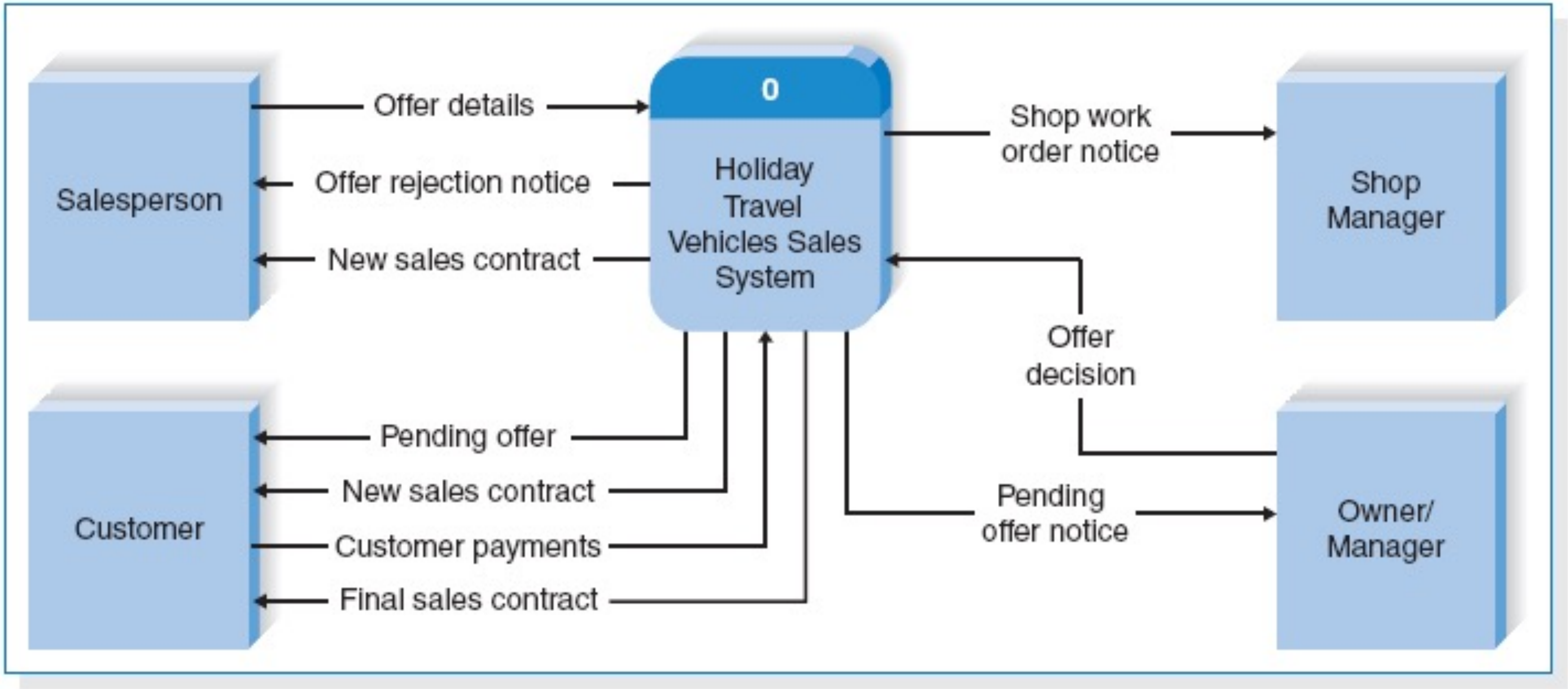
- 1. Build the context diagram.
- 2. Create DFD fragments for each use case.
- 3. Organize the DFD fragments into level 0 diagram.
- 4. Develop level 1 DFDs based on the steps with each use case. In some cases, these level 1 DFDs are further decomposed into level 2 DFDs, level 3 DFDs., and so son.
- 5. Validate the set of DFDs to make sure that they are complete and correct.

Creating the Context Diagram

- The **context diagram** defines how the business process or computer system interacts with its environment.
- Draw one process symbol for the business process or system being modeled (numbered 0 and named for the process or system).
- Add all inputs and outputs listed on the form of the use cases as data flows.
- Draw in external entities as the source or destination of the data flows.
- No data stores are included in the context diagram.

(cont'd)

Example of Context Diagram

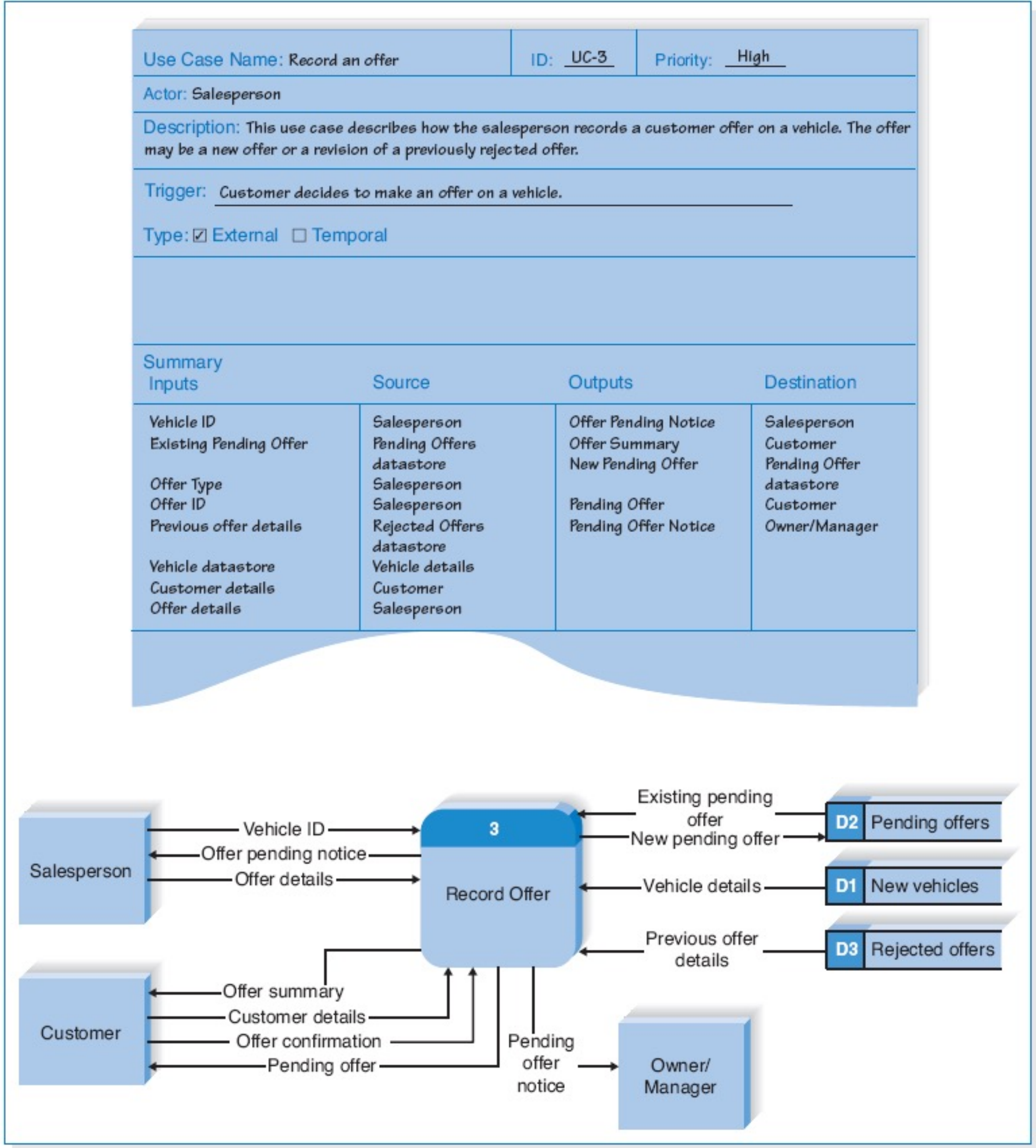


Creating DFD Fragments

- A DFD ***fragment*** is one part of a DFD that eventually will be combined with other DFD fragments to form a DFD.
- Each use case is converted into one DFD fragment using the information given on the form of the use case: the name, the 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.

(cont’d)

E



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.

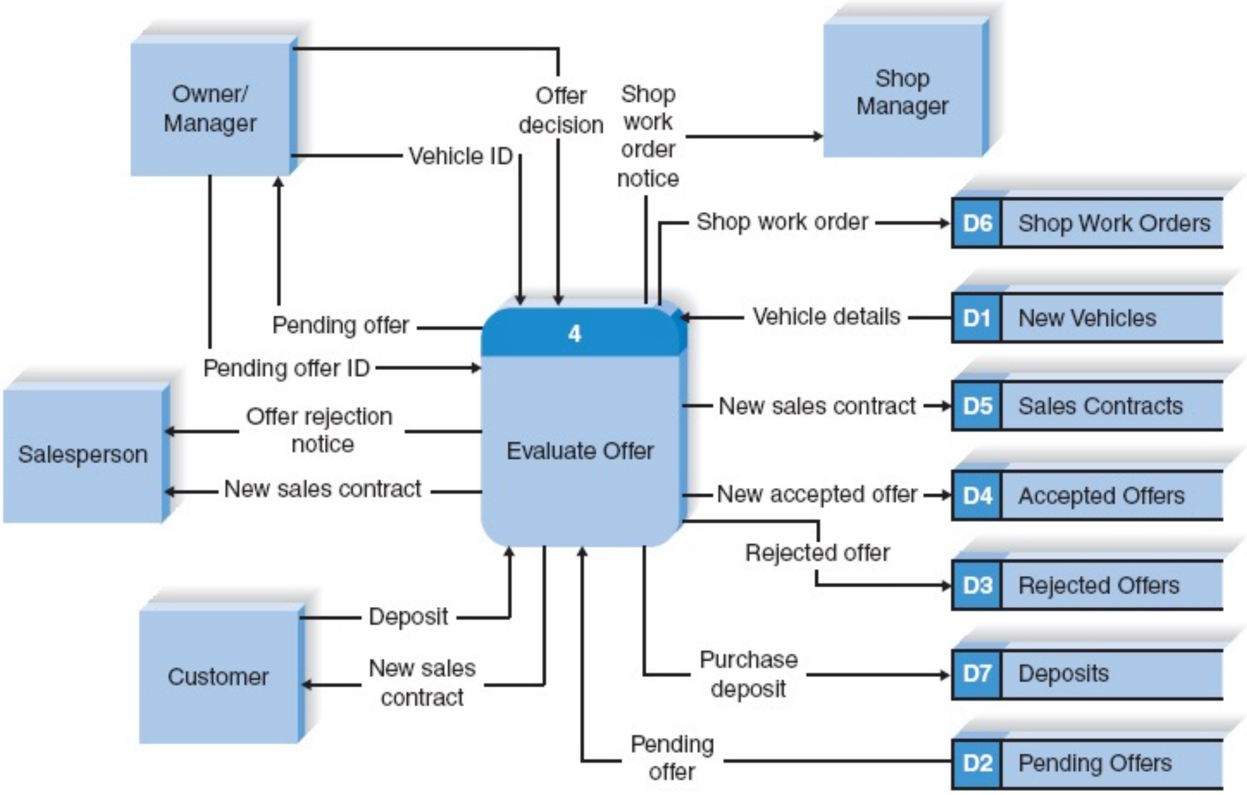
Trigger: Customer decides to make an offer on a vehicle.

Type: ☑ External ☐ Temporal

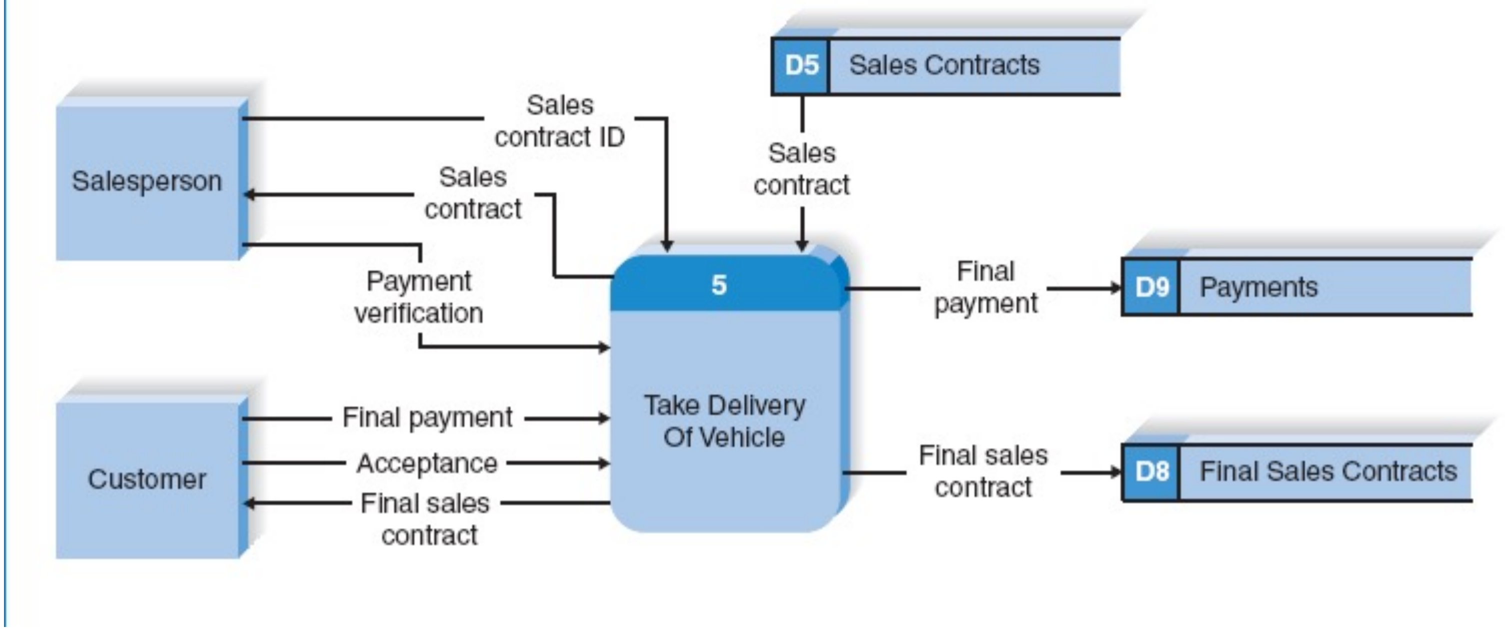
Summary Inputs	Source	Outputs	Destination
Vehicle ID	Salesperson	Offer Pending Notice	Salesperson
Existing Pending Offer	Pending Offers datastore	Offer Summary	Customer
Offer Type	Salesperson	New Pending Offer	Pending Offer datastore
Offer ID	Salesperson	Pending Offer	Customer
Previous offer details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

(cont'd)

Additional Example of Fragment



(cont’d)


D5
Sales Contracts
Salesperson
Sales contract ID
Sales contract
Sales contract
Payment verification
5
Take Delivery Of Vehicle
Final payment
D9
Payments
Customer
Final payment
Acceptance
Final sales contract
Final sales contract
D8
Final Sales Contracts

(cont'd)

- Important changes are often made in converting the use case into a DFD:
  - modifications to the process names
  - the addition of data flows.
- Make sure that any information given to the user is obtained from a **data store**.
- There are not formal rules covering the ***layouts***; typically
  - – place the processes in the middle
  - – inputs start from the left or top
  - – outputs leave from the right or the bottom

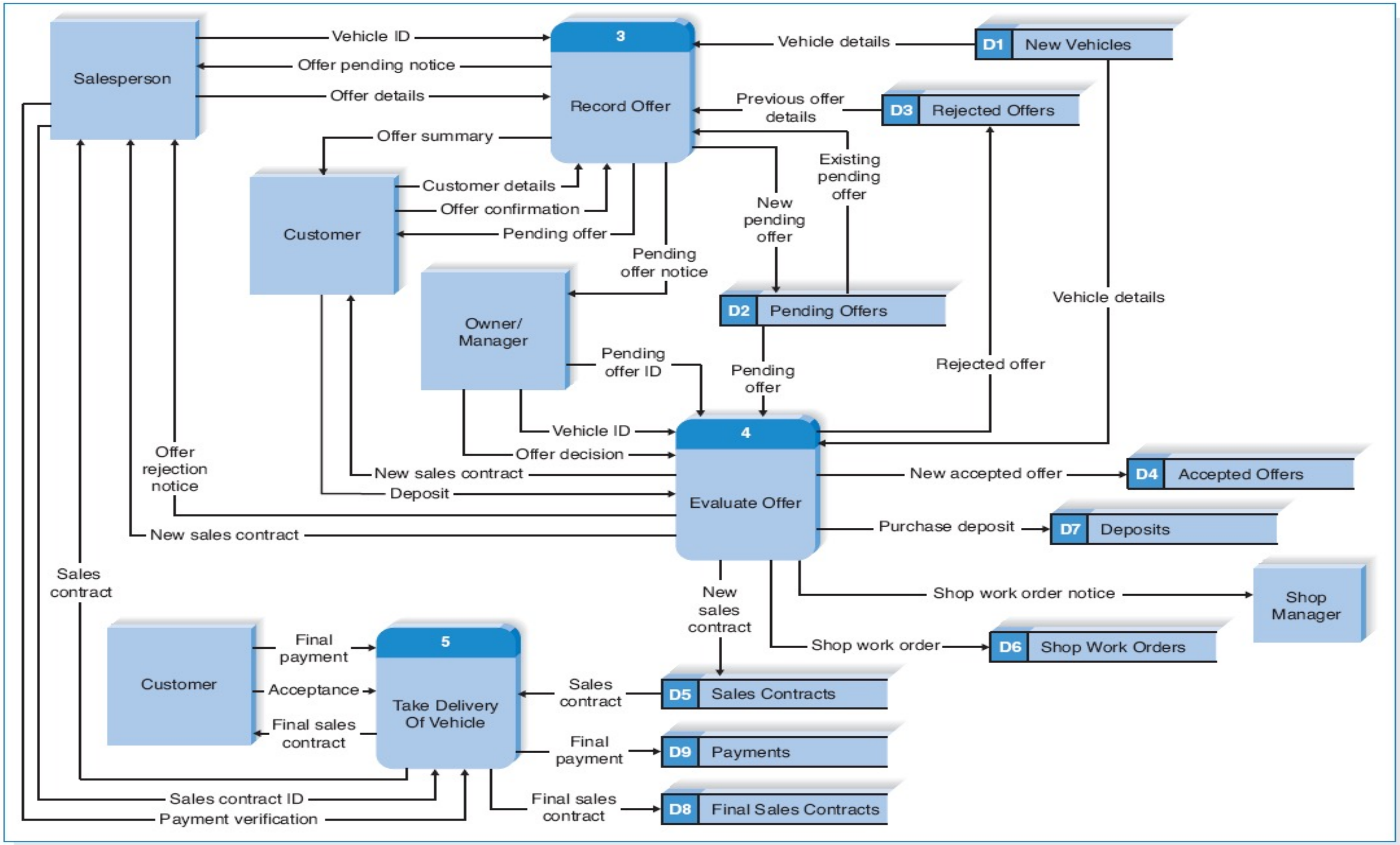
Creating the Level 0 Diagram

- Combine the set of DFD fragments into one diagram – the **level 0 DFD**.
- There are not formal layout rules. Generally,

  - to put the process that is first chronologically in the upper-left corner and work the way from top to bottom, left to right;

  - to reduce the number of crossed data flow lines.
- ***Iteration*** is the cornerstone of good DFD design.

(cont'd)


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

Creating Level 1 Data Flow Diagrams (and Below)

- **Level 1 DFD** – lower-level DFDs for each process in the level 0 DFD.
- Each one of the use cases is turned into its own DFD
- 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.
- Level 1 DFDs include the sources and destinations of data flows for data stores and data flows to processes.
external entities

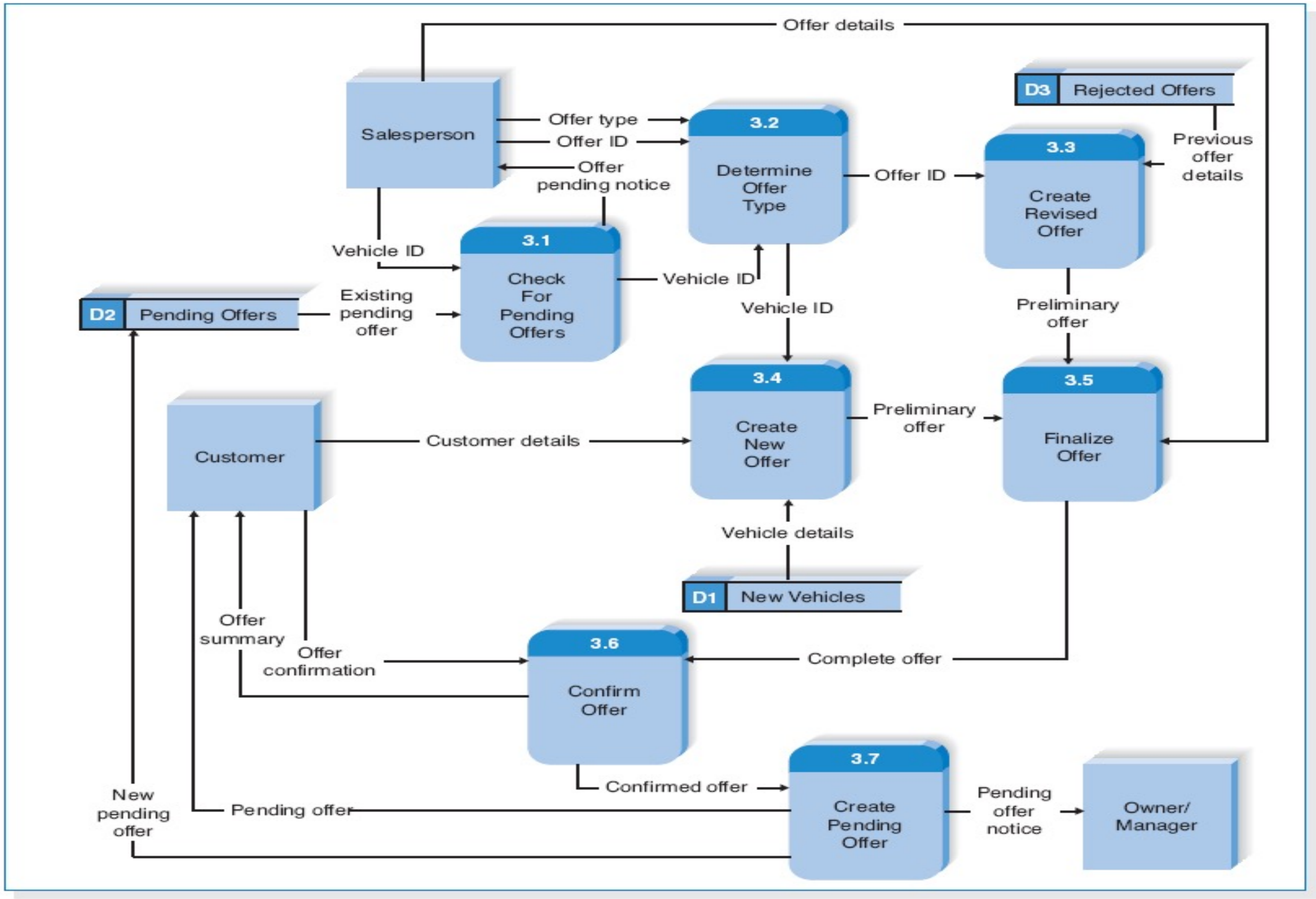
(cont’d)

- There is no simple answer to the “ideal” level of decomposition, because it depends on the complexity of the system or business process being modeled.
- In general, you decompose a process into a lower-level DFD whenever the process is sufficiently complex that additional decomposition can help explain the process.
- Rules of thumb:

  - There should be at least 3, and no more than 7-9, processes on every DFD.

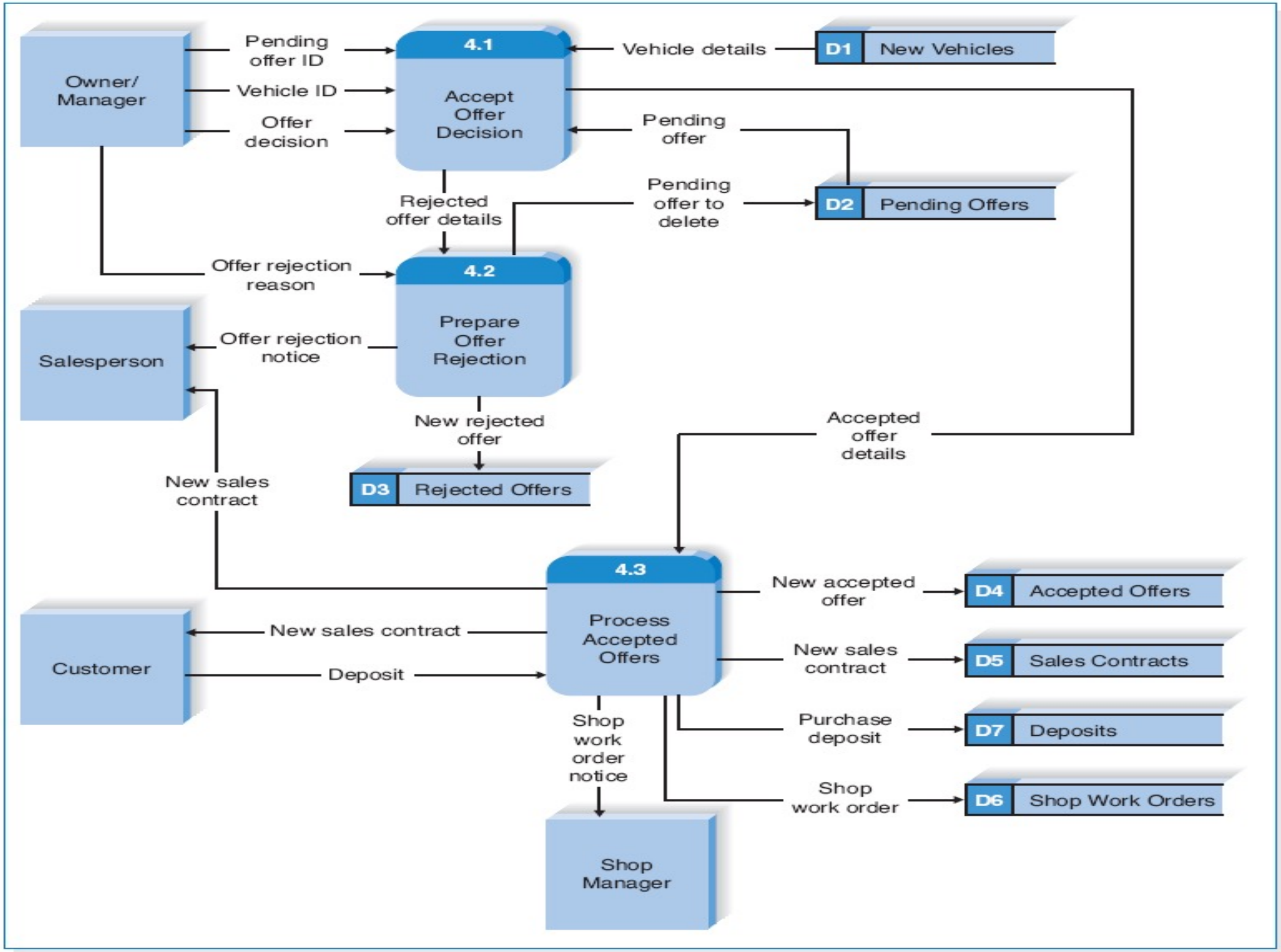
  - Decompose until you can provide a detailed

(cont'd)



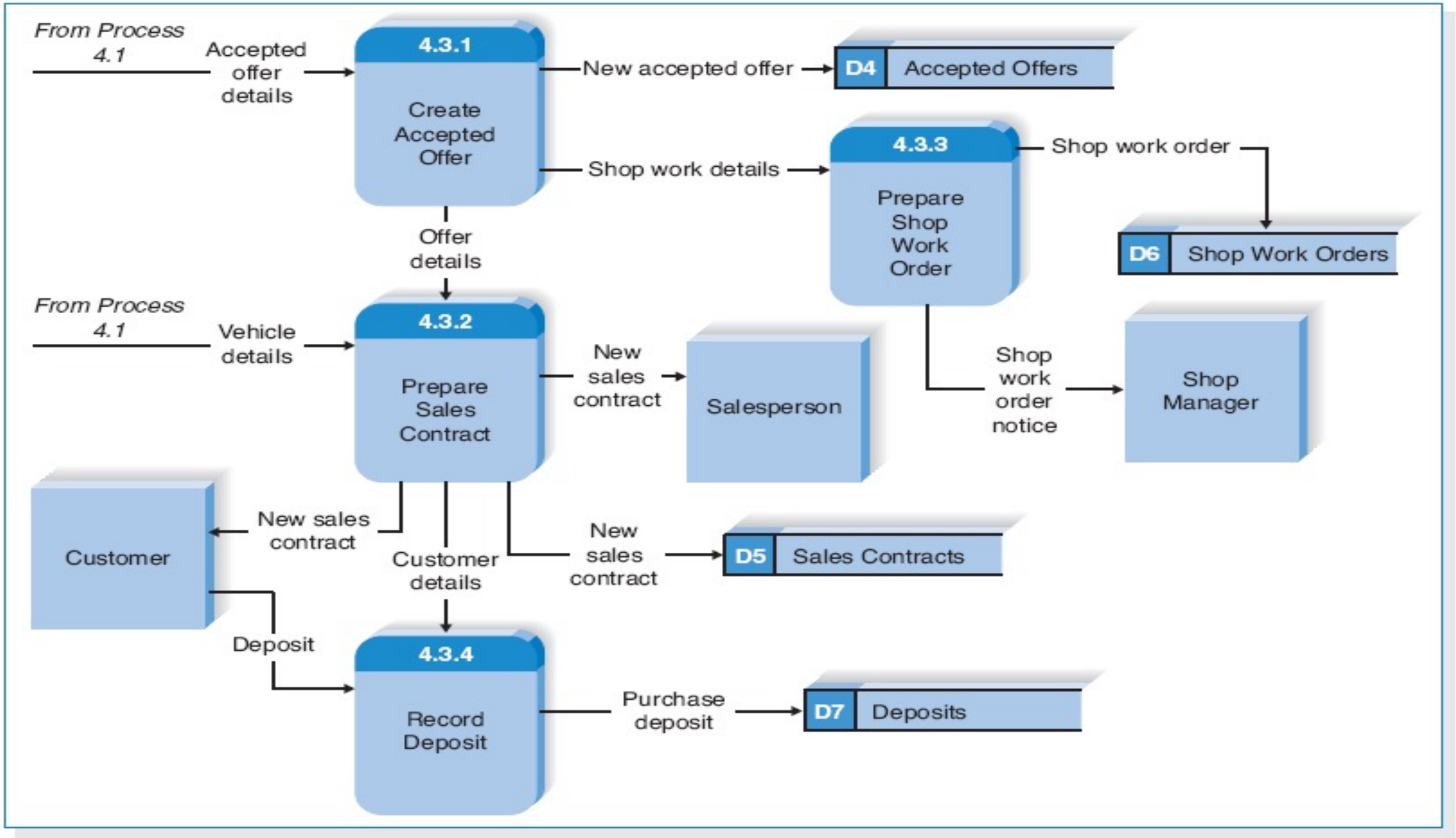
Offer details
D3 Rejected Offers
Salesperson
Offer type
Offer ID
Offer pending notice
3.2
Determine Offer Type
Offer ID
3.3
Create Revised Offer
Previous offer details
Vehicle ID
3.1
Check For Pending Offers
Vehicle ID
D2 Pending Offers
Existing pending offer
Vehicle ID
Preliminary offer
3.4
Create New 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

(cont'd)

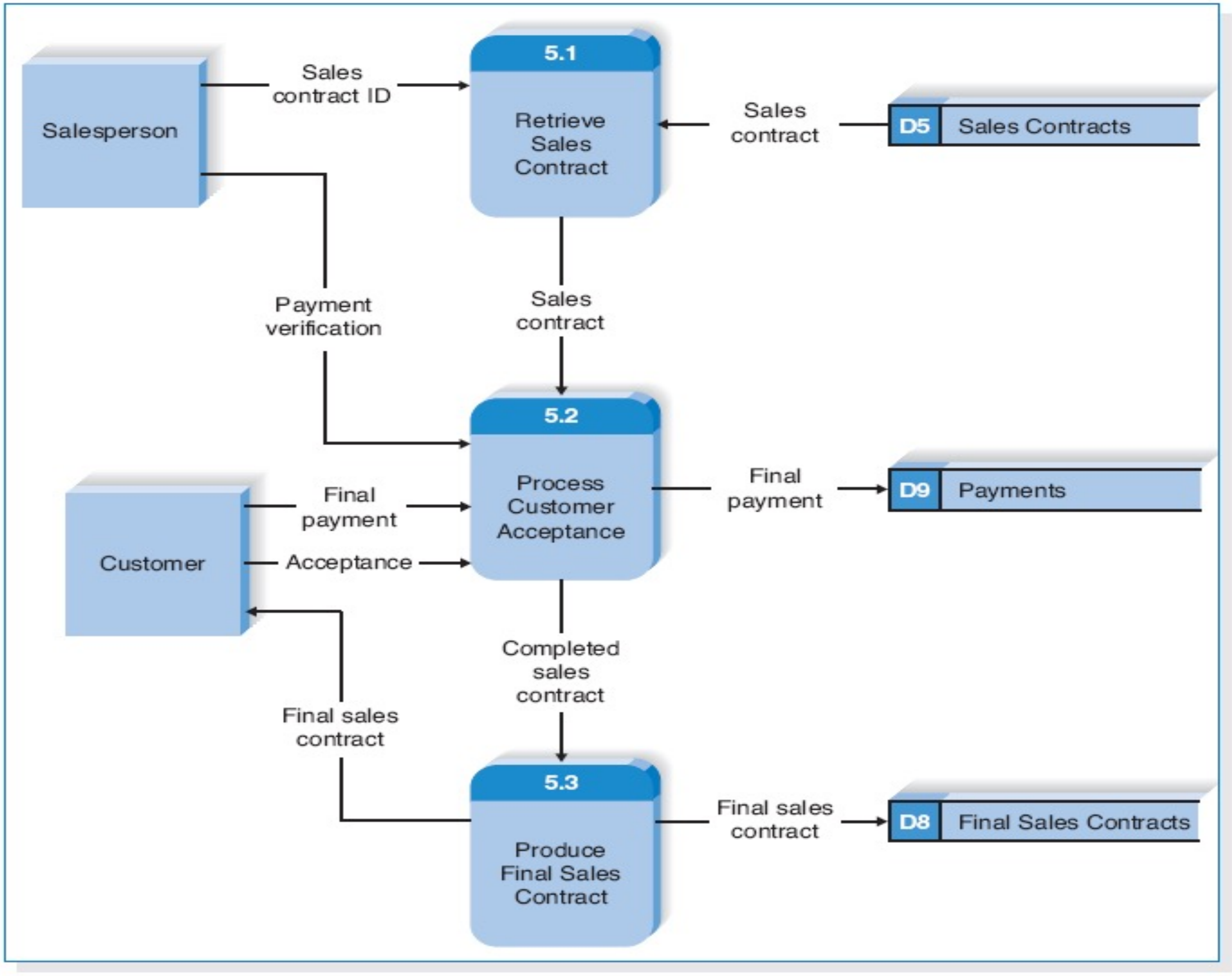


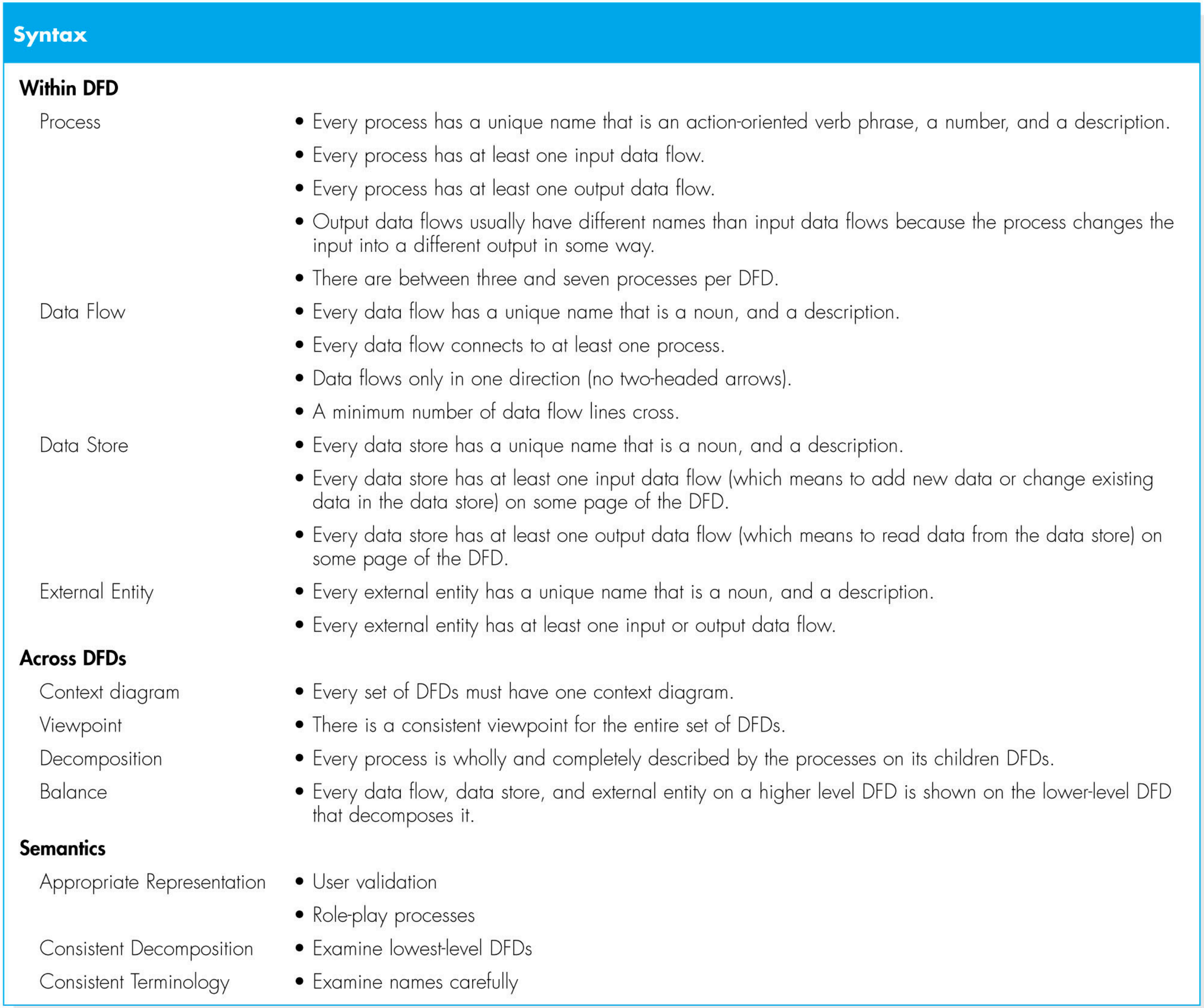
Owner/ Manager
Pending offer ID
Vehicle ID
Offer decision
4.1
Accept Offer Decision
Vehicle details
D1
New Vehicles
Pending offer
D2
Pending Offers
Pending offer to delete
Rejected offer details
Offer rejection reason
4.2
Prepare Offer Rejection
Salesperson
Offer rejection notice
New rejected offer
D3
Rejected Offers
Accepted offer details
New sales contract
4.3
Process Accepted Offers
New accepted offer
D4
Accepted Offers
Customer
New sales contract
Deposit
New sales contract
D5
Sales Contracts
Purchase deposit
D7
Deposits
Shop work order
D6
Shop Work Orders
Shop work order notice
Shop Manager

(cont’d)


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

(cont'd)


Salesperson
Sales contract ID
5.1
Retrieve Sales Contract
Sales contract
D5
Sales Contracts
Payment verification
Sales contract
5.2
Process Customer Acceptance
Final payment
D9
Payments
Customer
Final payment
Acceptance
Completed sales contract
Final sales contract
5.3
Produce Final Sales Contract
Final sales contract
D8
Final Sales Contracts



Syntax

Within DFD	
Process	• Every process has a unique name that is an action-oriented verb phrase, a number, and a description.
	• Every process has at least one input data flow.
	• Every process has at least one output data flow.
	• Output data flows usually have different names than input data flows because the process changes the input into a different output in some way.
	• There are between three and seven processes per DFD.
Data Flow	• Every data flow has a unique name that is a noun, and a description.
	• Every data flow connects to at least one process.
	• Data flows only in one direction (no two-headed arrows).
	• A minimum number of data flow lines cross.
Data Store	• Every data store has a unique name that is a noun, and a description.
	• Every data store has at least one input data flow (which means to add new data or change existing data in the data store) on some page of the DFD.
	• Every data store has at least one output data flow (which means to read data from the data store) on some page of the DFD.
External Entity	• Every external entity has a unique name that is a noun, and a description.
	• Every external entity has 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.
Semantics	
Appropriate Representation	• User validation
	• Role-play processes
Consistent Decomposition	• Examine lowest-level DFDs
Consistent Terminology	• Examine names carefully

Validating the DFD (cont'd)

- There two fundamental types of errors in DFDs:

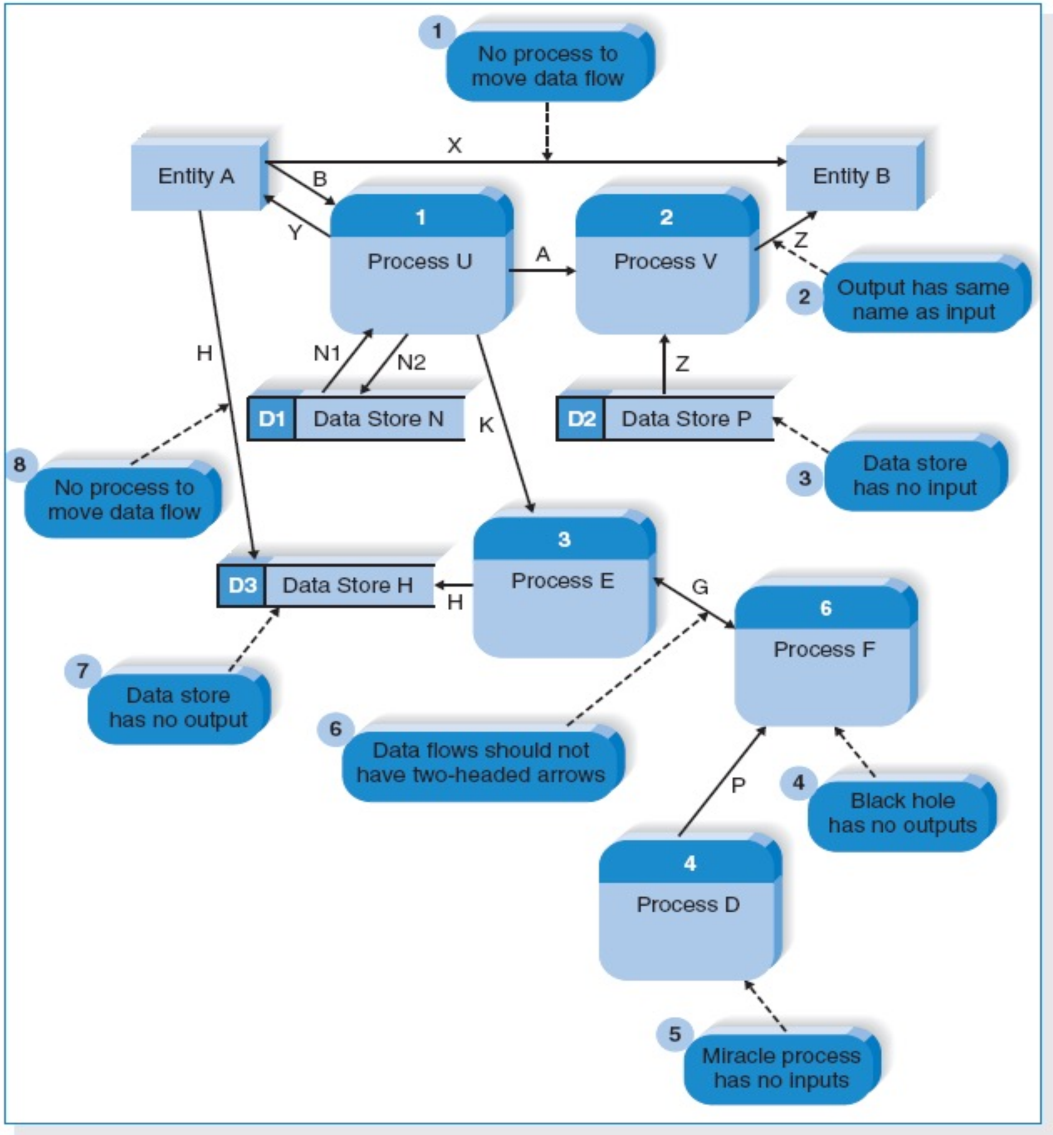
1. ***Syntax errors*** – can be thought of as grammatical errors that violate the rules of the DFD language.
2. ***Semantics errors*** – can be thought of as misunderstandings by the analyst in collecting, analyzing, and reporting information about the system.

(cont’d)

- Syntax errors are easier to find and fix than are semantics errors because there are clear rules that can be used to identify them.
- Most **CASE tools** have syntax checkers that will detect syntax errors.

(cont'd)

Common Syntax Errors


1
No process to move data flow
X
Entity A
Entity B
B
Y
1
Process U
A
2
Process V
Z
2
Output has same name as input
H
N1
N2
Z
D1
Data Store N
K
D2
Data Store P
3
Data store has no input
8
No process to move data flow
3
Process E
G
6
Process F
D3
Data Store H
H
7
Data store has no output
6
Data flows should not have two-headed arrows
P
4
Black hole has no outputs
4
Process D
5
Miracle process has no inputs

(cont'd)

- **Semantics errors** cause the most problems in system development.
- Three useful checks to help ensure that models are semantically correct:

1. to ensure that the model is an appropriate representation by asking the users to validate the model in a **walk-through**
2. to ensure consistent decomposition
3. to ensure that the terminology is consistent throughout the model

SUMMARY

- **Data Flow Diagram Syntax** – four symbols are used on data flow diagrams (processes, data flows, data stores, and external entities).
- **Creating Data Flow Diagrams**

- The DFDs are created from use cases.

- Every set of DFDs starts with a context diagram.

- DFDs segments are created for each use case, and are then organized into a level 0 DFD.

- Level 1 DFDs are developed on the basis of the steps within each use case.

- The set of DFDs are validated to make sure that they are complete and correct and contain no syntax or semantics errors.