Systems Analysis and Design 10th Edition

- Chapter 5
  Data and Process Modeling

Chapter Objectives

- Describe data and process modeling concepts and tools, including data flow diagrams, a data dictionary, and process descriptions
- Describe the symbols used in data flow diagrams and explain the rules for their use
- Draw data flow diagrams in a sequence, from general to specific

Chapter Objectives (Cont.)

- Explain how to level and balance a set of data flow diagrams
- Describe how a data dictionary is used and what it contains
- Use process description tools, including structured English, decision tables, and decision trees
- Describe the relationship between logical and physical models

Overview of Data and Process Modeling Tools

- Systems analysts use many graphical techniques to describe an information system
- A data flow diagram (DFD) uses various symbols to show how the system transforms input data into useful information

Data Flow Diagrams

- A data flow diagram (DFD) shows how data moves through an information system but does not show program logic or processing steps
- A set of DFDs provides a logical model that shows what the system does, not how it does it

Data Flow Diagrams (Cont.)

- DFD Symbols
  - Four basic symbols
  - Gane & Sarson used in text
  - Yourdon also popular

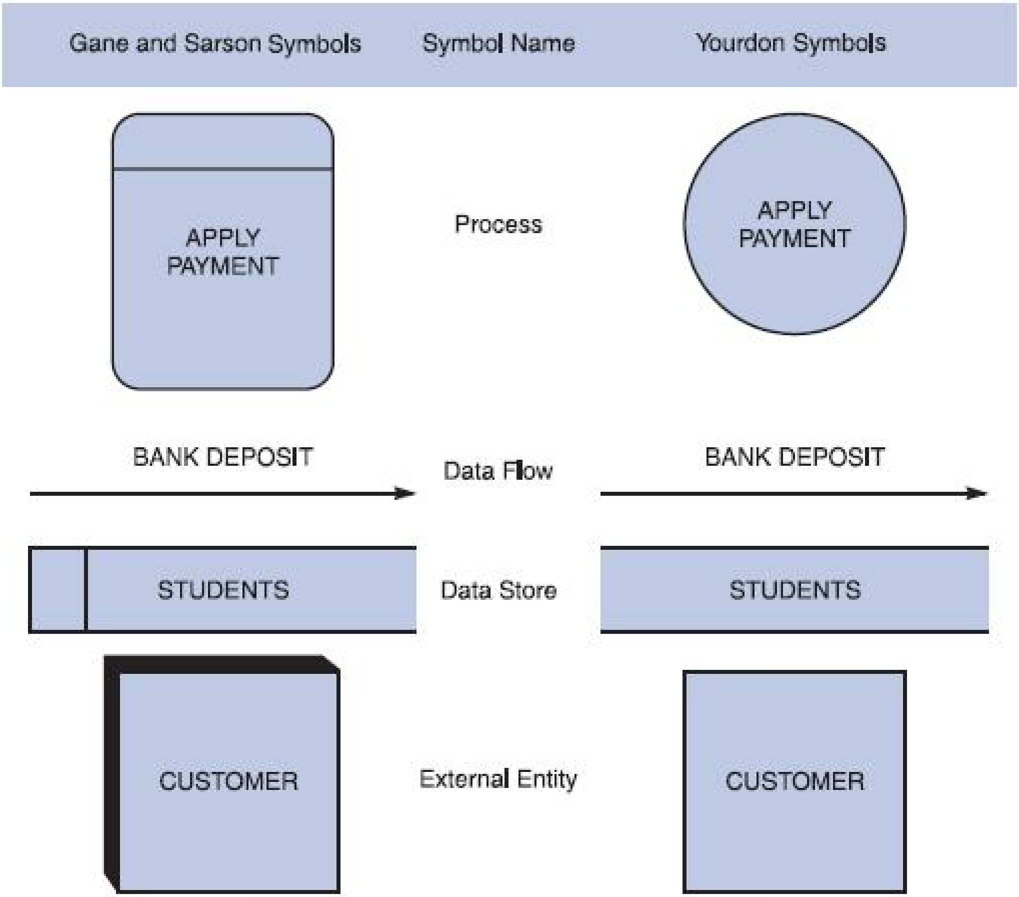


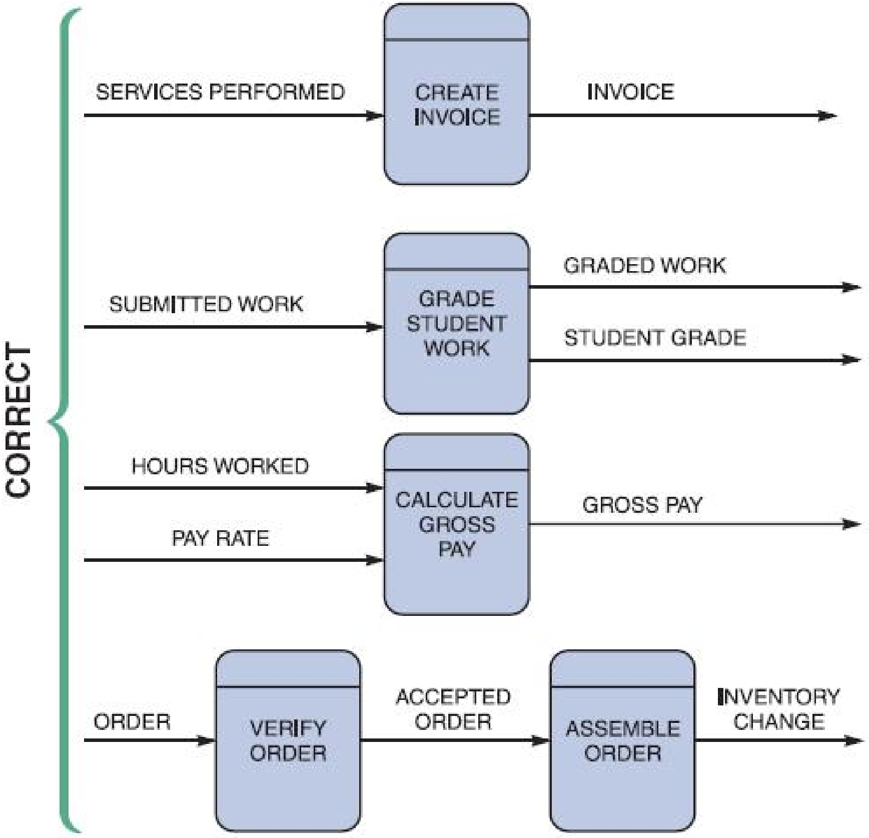
FIGURE 5-3 Data flow diagram symbols, symbol names, and examples of the Gane and Sarson and Yourdon symbol sets

Data Flow Diagrams (Cont.)

Process Symbol

- Must have at least one input and at least one output
- Contains business logic that transforms the data
- Process name identifies its function (verb)
- Process number does not signify precedence
- Examples: “print bill” or “add customer”

Data Flow Diagrams (Cont.)



[redacted] correct combinations of data flow and process symbols

- Data flow symbol
  - Represents one or more data items
  - The symbol for a data flow is a line with a single or double arrowhead

Data Flow Diagrams (Cont.)

- Data flow symbol
  - Spontaneous generation (Process must act on input)
  - Black holes
  - Gray holes

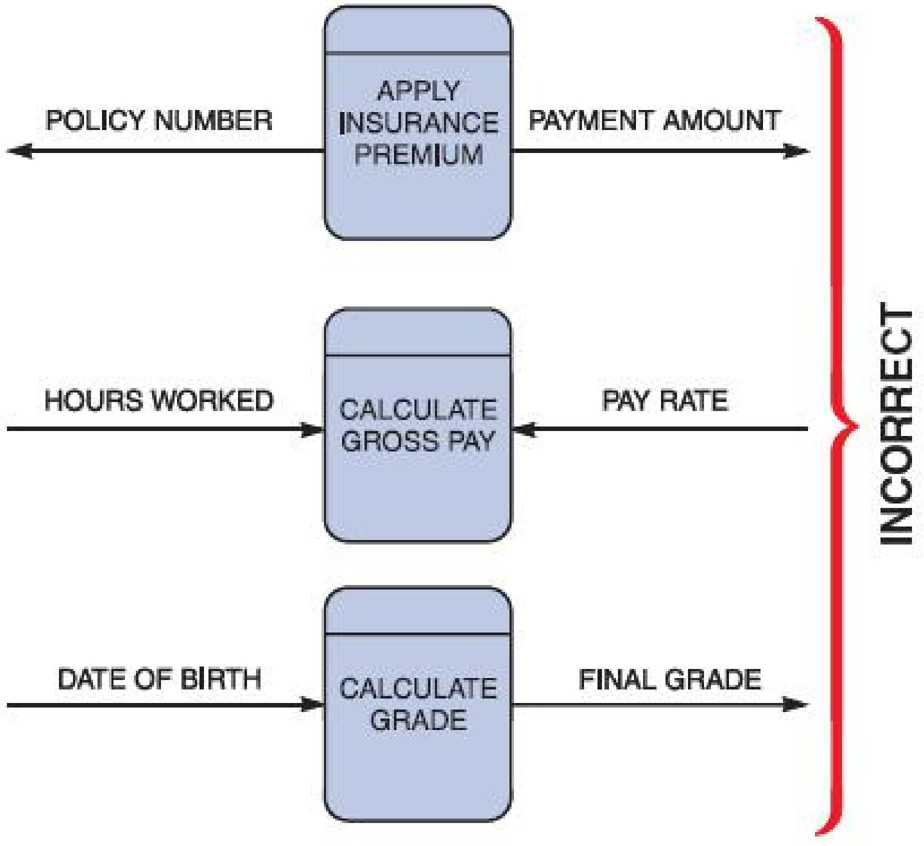
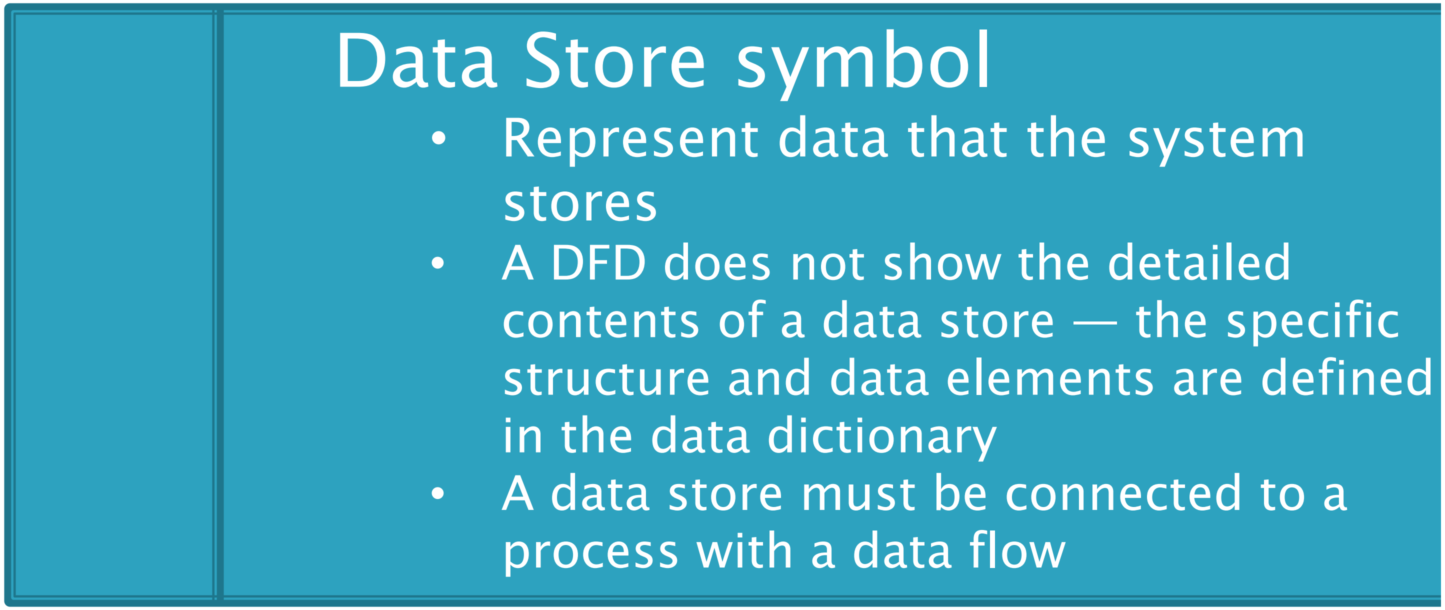


FIGURE 5-6 Examples of incorrect combinations of data flow and process symbols. APPLY INSURANCE PREMIUM has no input and is called a
generation process. CALCULATE GROSS PAY has no outputs and
ack hole process. CALCULATE GRADE has an input that is
able to produce the output. This process is called a gray hole

Data Flow Diagrams (Cont.)



Data Store symbol

- Represent data that the system stores
- A DFD does not show the detailed contents of a data store — the specific structure and data elements are defined in the data dictionary
- A data store must be connected to a process with a data flow

Data Flow Diagrams (Cont.)

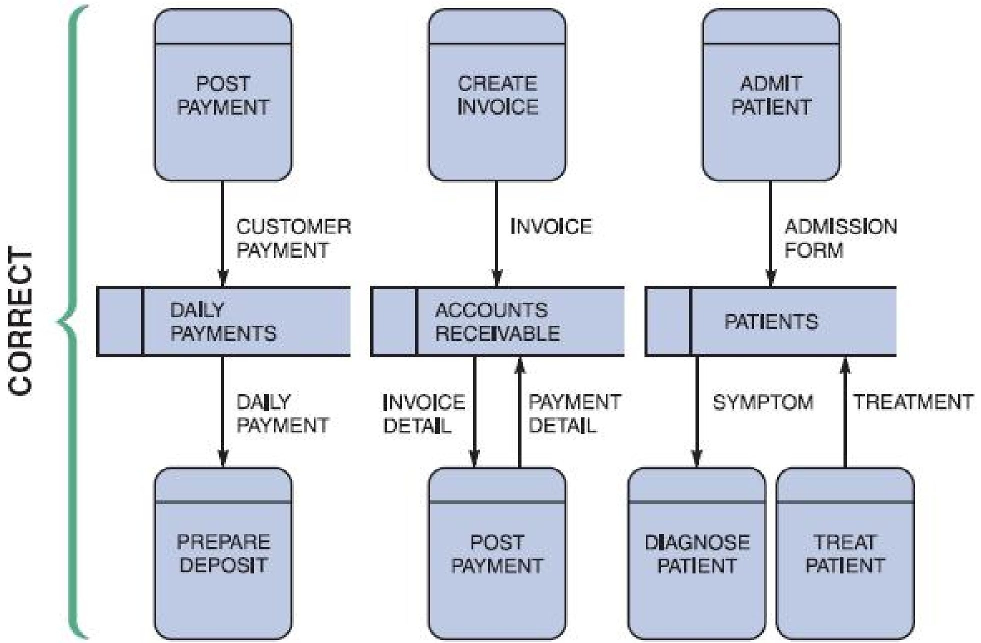


FIGURE 5-7 Examples of correct uses of data store symbols in a data flow diagram

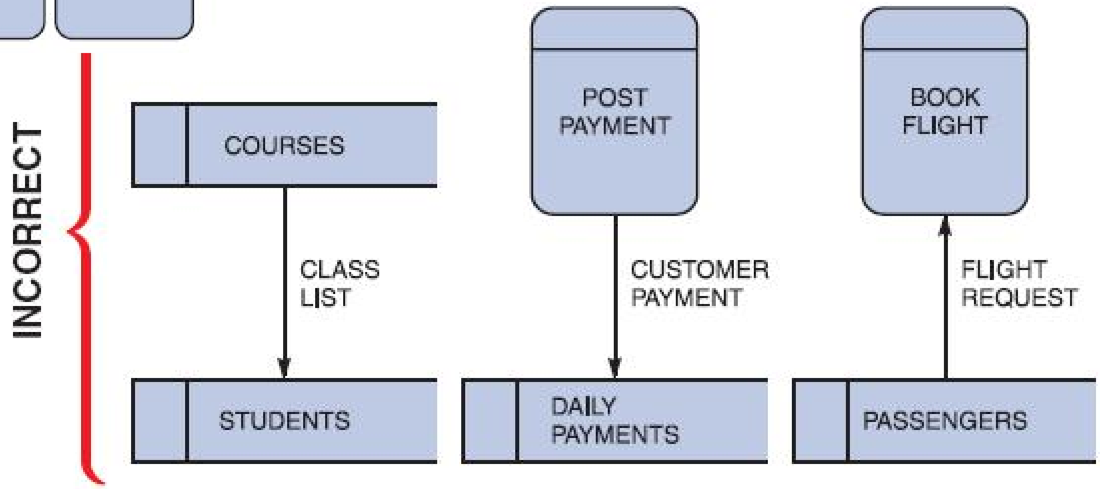


FIGURE 5-8 Examples of incorrect uses of data store symbols: Two data stores cannot be connected by a data flow without an intervening process, and each data store should have an outgoing and incoming data flow

Data Flow Diagrams (Cont.)

- Shows how the system interfaces with the outside world
- A DFD shows only external entities that provide data to the system or receive output from the system
- DFD entities also are called **terminators** because they are data origins or final destinations
- Each entity must be connected to a process by a data flow

Data Flow Diagrams (Cont.)

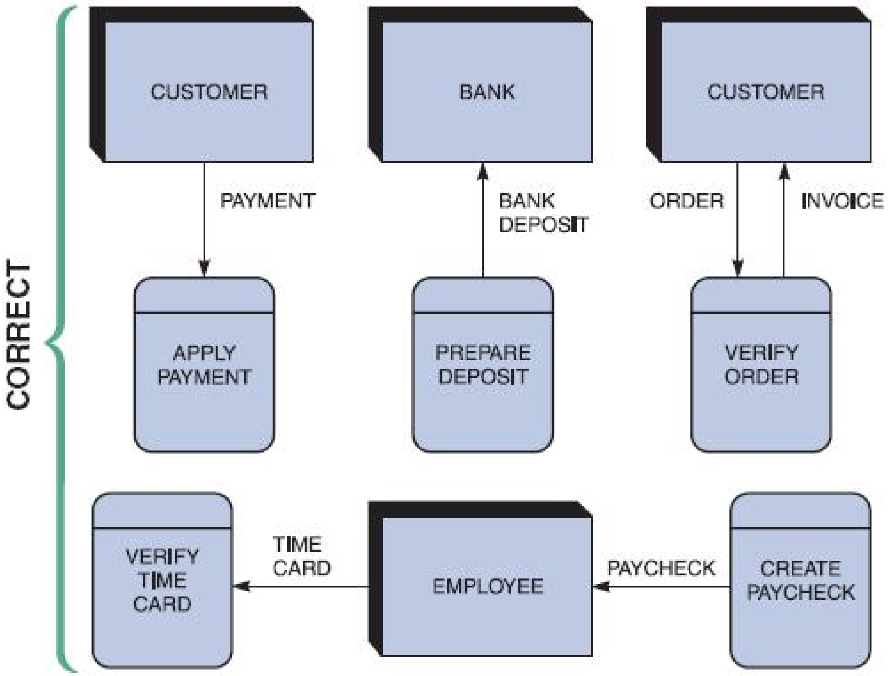
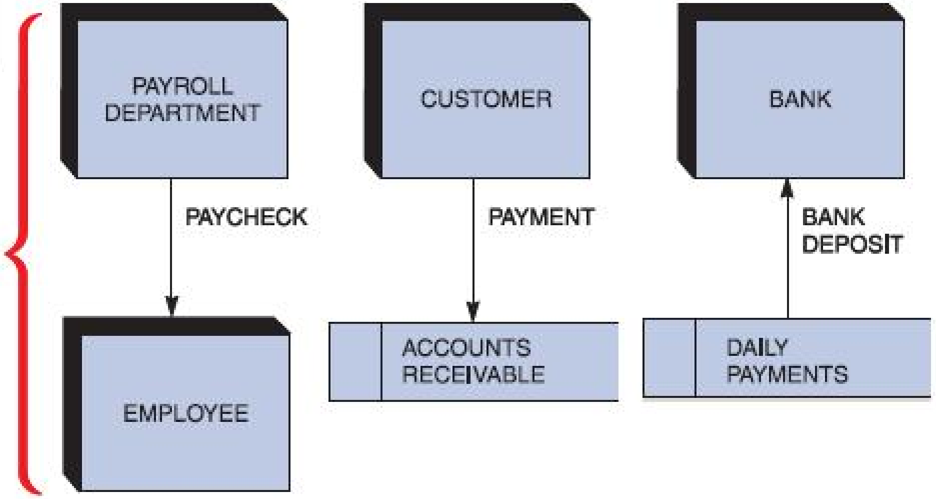


FIGURE 5-9 Examples of correct uses of external entities in a data flow diagram

FIGURE 5-10 Examples of incorrect uses of external entities. An external entity must be connected by a data flow to a process, and not directly to a data store or to another external entity



Creating a Set of DFDs

- Create a graphical model of the information system based on your fact-finding results
  - First, you will review a set of guidelines for drawing DFDs
  - Then you will learn how to apply these guidelines and create a set of DFDs using a three-step process

Creating a Set of DFDs (Cont.)

- Keep in mind:
  - All flow lines must be labeled
  - Large processes can be broken down into smaller components

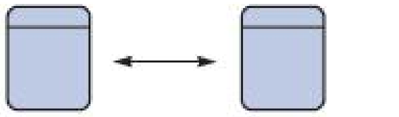
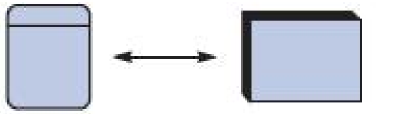
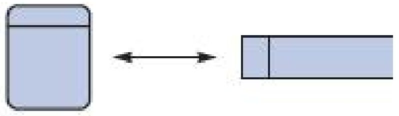
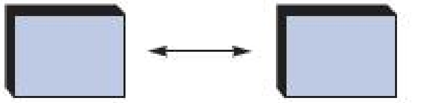
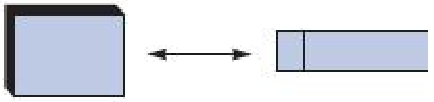
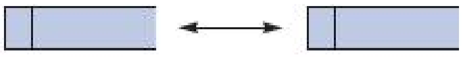
Correct and Incorrect Examples of Data Flows		
	Process to Process	✓
	Process to External Entity	✓
	Process to Data Store	✓
	External Entity to External Entity	✗
	External Entity to Data Store	✗
	Data Store to Data Store	✗

FIGURE 5-11 Examples of correct
[redacted]s

Creating a Set of DFDs (Cont.)

- Guidelines for Drawing DFDs
  - Draw the context diagram so that it fits on one page
  - Use the name of the information system as the process name in the context diagram
  - Use unique names within each set of symbols
  - Do not cross lines
  - Provide a unique name and reference number for each process
  - Ensure that the model is accurate, easy to understand, and meets the needs of its users

Creating a Set of DFDs (Cont.)

- Step 1: Draw a Context Diagram

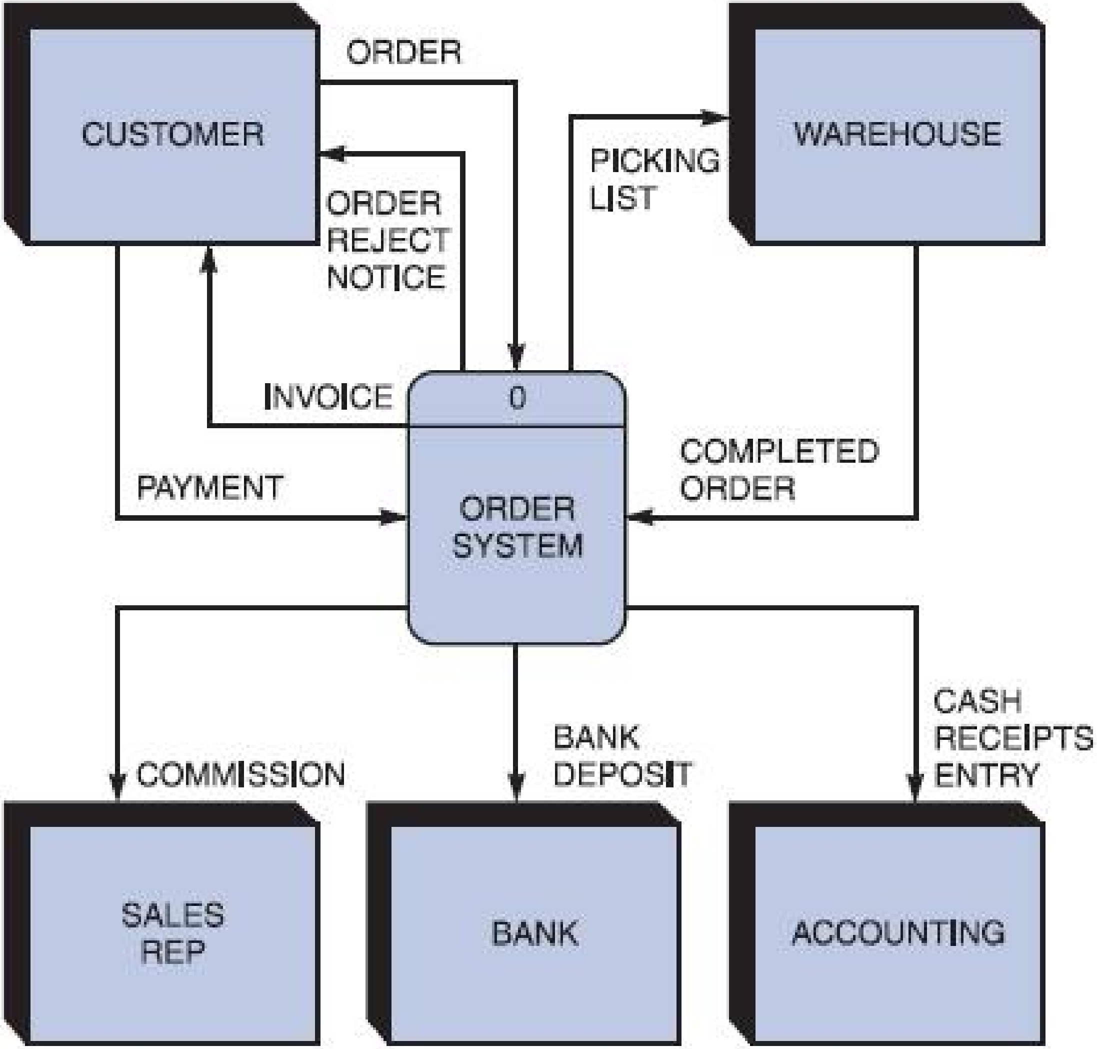


FIGURE 5-13 Context diagram DFD for an order system

Creating a Set of DFDs (Cont.)

- Step 2: Draw a Diagram 0 DFD
  - If same data flows in both directions, you can use a double-headed arrow
  - Diagram 0 is an exploded view of process 0
  - Parent diagram
  - Child diagram
  - Functional primitive

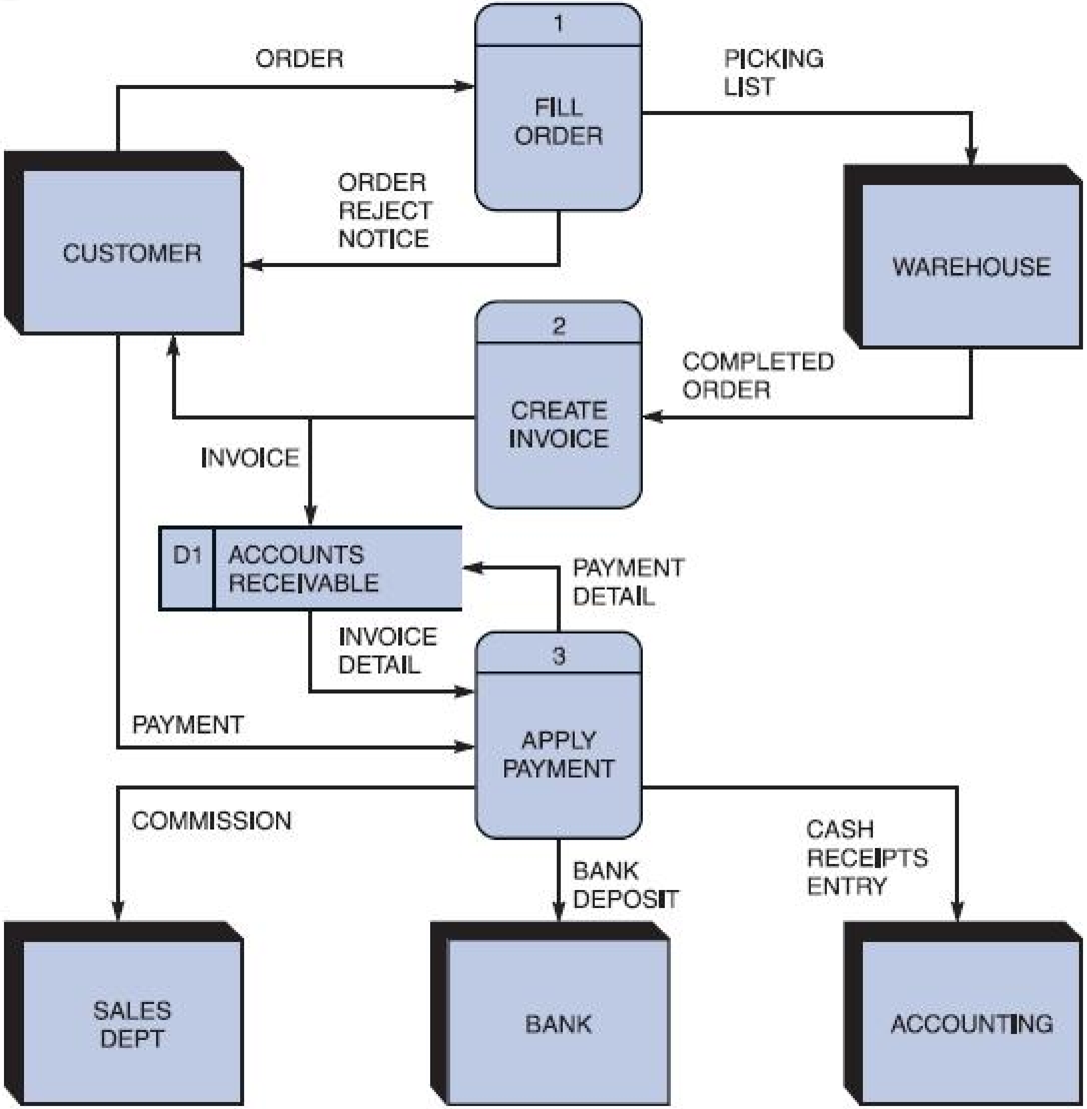


FIGURE 5-16 Diagram 0 DFD for the order system

Creating a Set of DFDs (Cont.)

- Step 3: Draw the Lower Level Diagrams

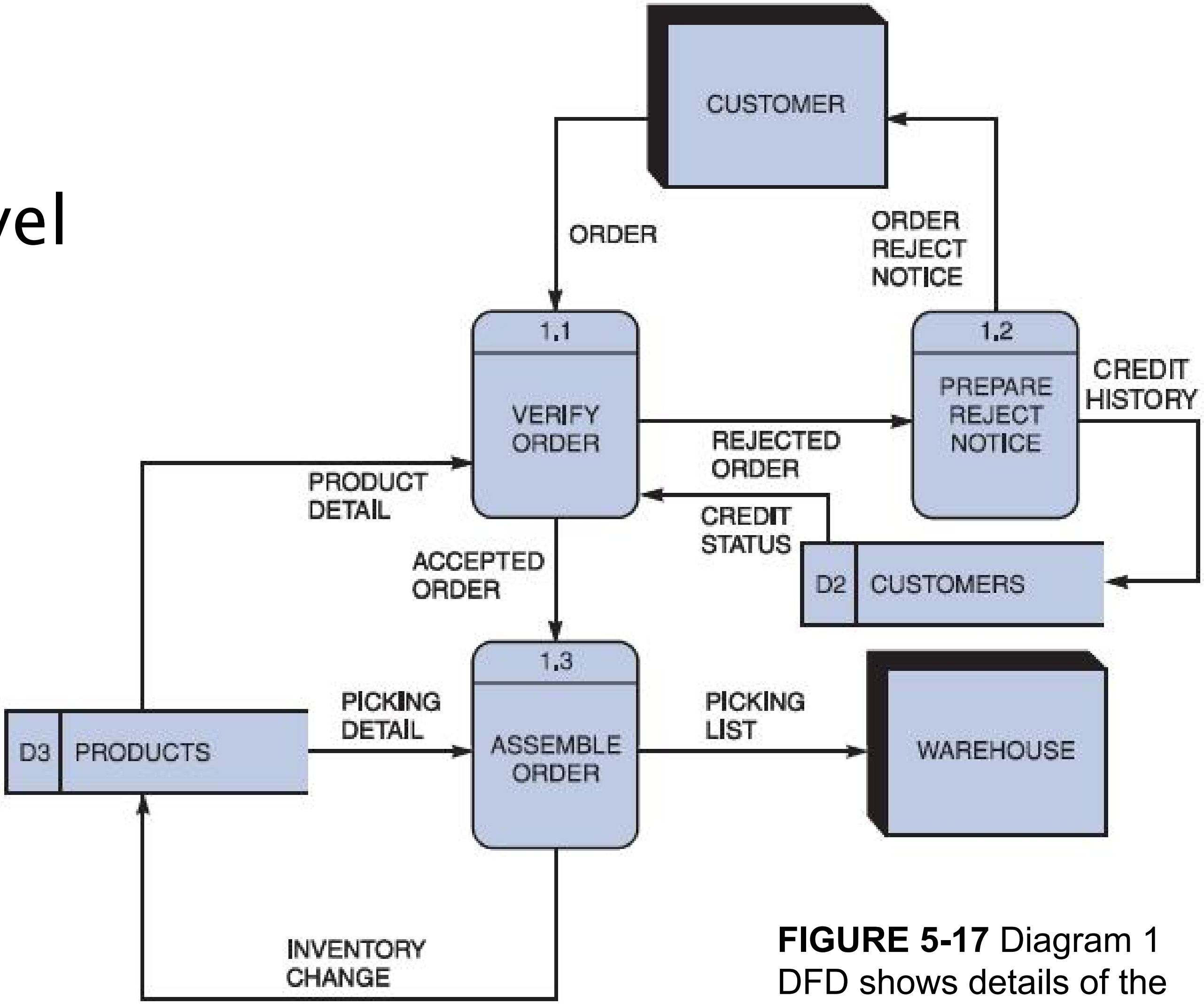


FIGURE 5-17 Diagram 1 DFD shows details of the FILLORDER process in the order system

Creating a Set of DFDs (Cont.)

- Must use leveling and balancing techniques
- Leveling examples
  - Uses a series of increasingly detailed DFDs to describe an information system
  - Exploding, partitioning, or decomposing

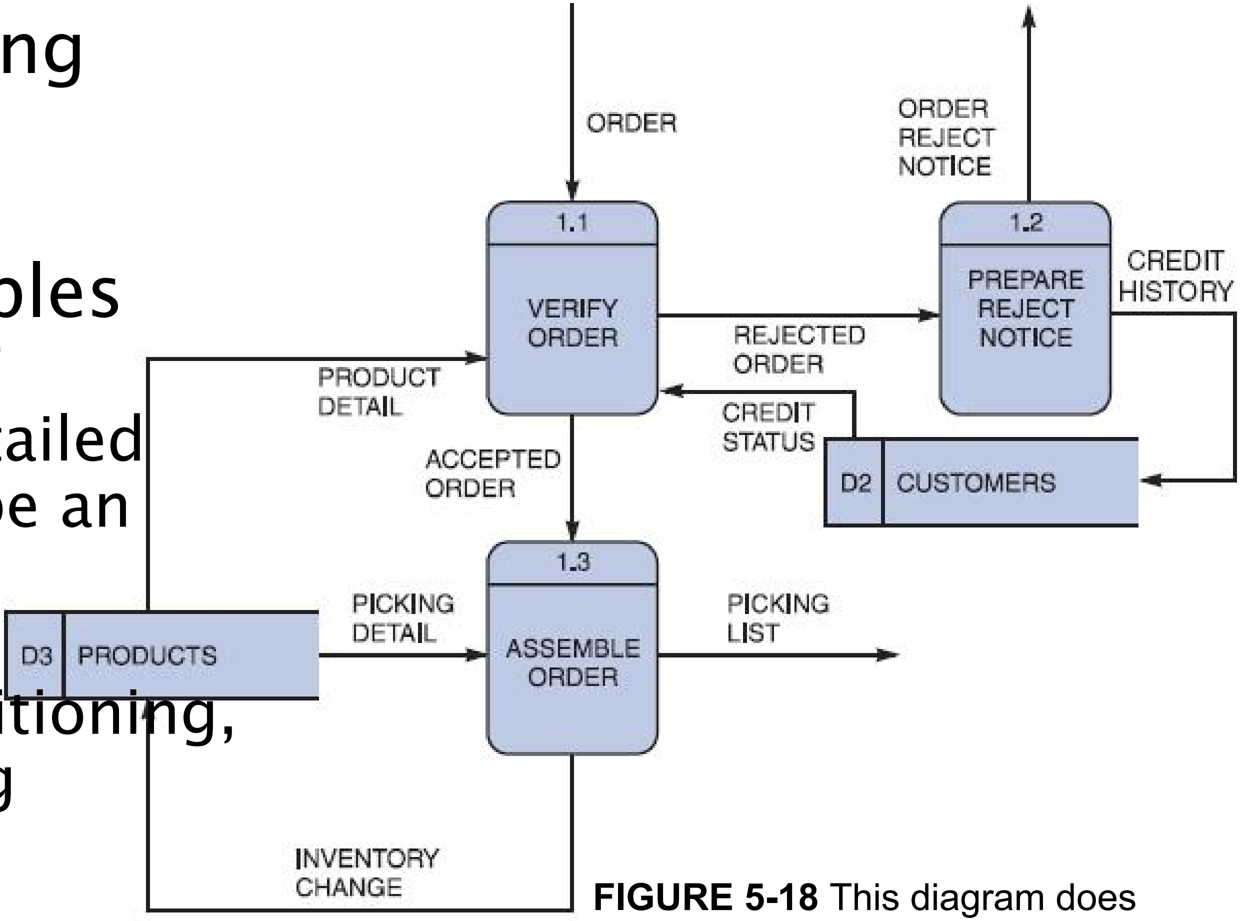


FIGURE 5-18 This diagram does not show the symbols that connect to data flows entering or leaving FILL ORDER on the context diagram

Creating a Set of DFDs (Cont.)

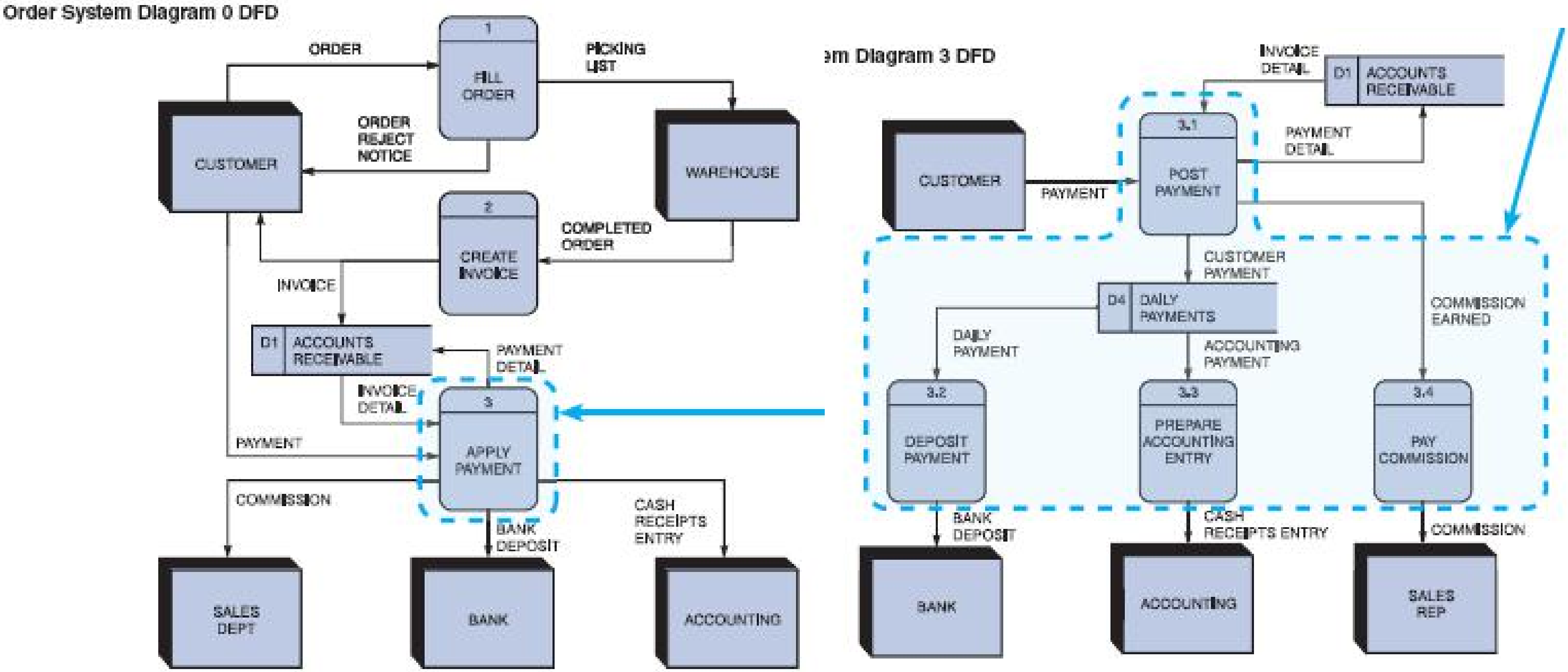


FIGURE 5-19 The order system diagram 0 is shown at the top of the figure, and exploded diagram 3 DFD (for the APPLY PAYMENT process) is shown at the bottom. The two DFDs are balanced because the child diagram at the bottom has the same input and output flows as the parent process 3 shown at the top

Creating a Set of DFDs (Cont.)

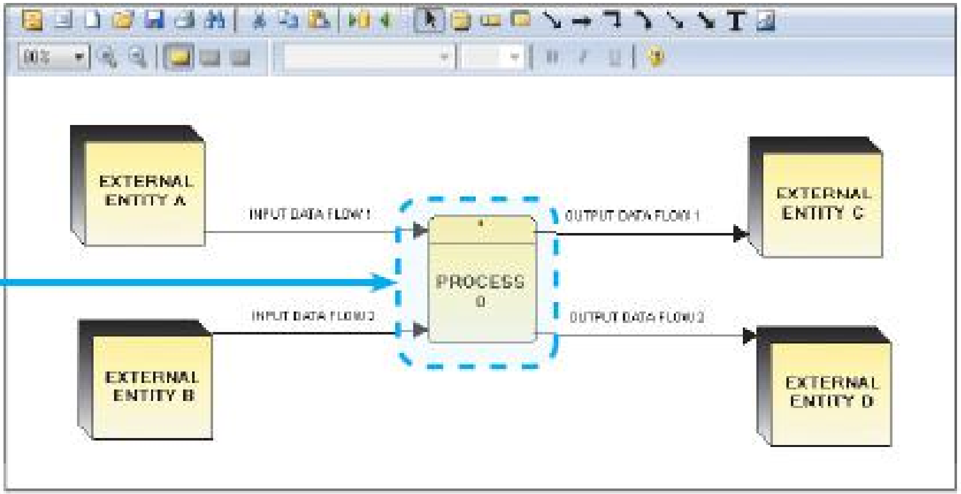


FIGURE 5-20 Example of a parent DFD diagram, showing process 0 as a black box

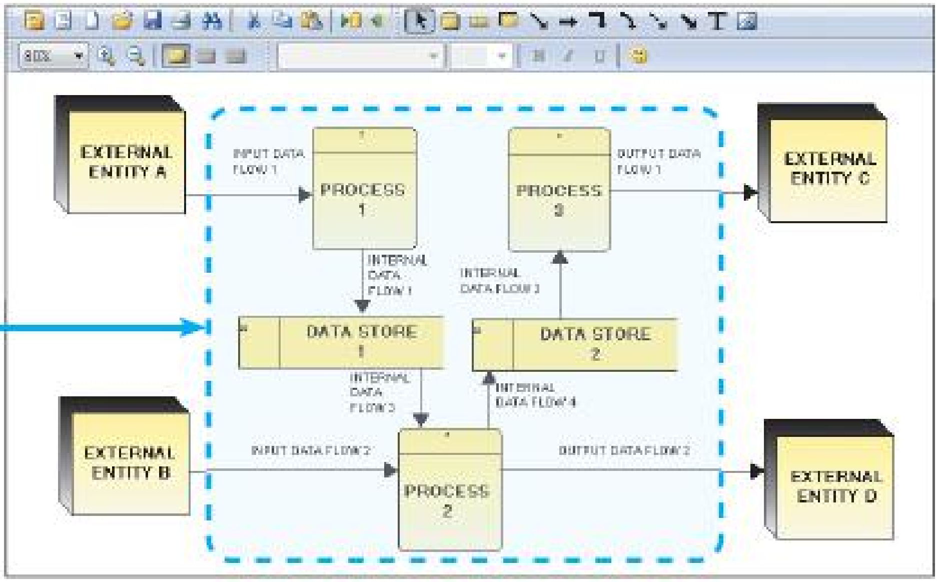


FIGURE 5-21 In the next level of detail, the process 0 black box reveals three processes, two data stores, and four internal data flows — all of which are shown inside the dashed line

Data Dictionary

- A data dictionary, or data repository, is a central storehouse of information about the system's data
- An analyst uses the data dictionary to collect, document, and organize specific facts about the system
- Also defines and describes all data elements and meaningful combinations of data elements

Data Dictionary (Cont.)

- A data element, also called a data item or field, is the smallest piece of data that has meaning
- Data elements are combined into records, also called data structures
- A record is a meaningful combination of related data elements that is included in a data flow or retained in a data store

Data Dictionary (Cont.)

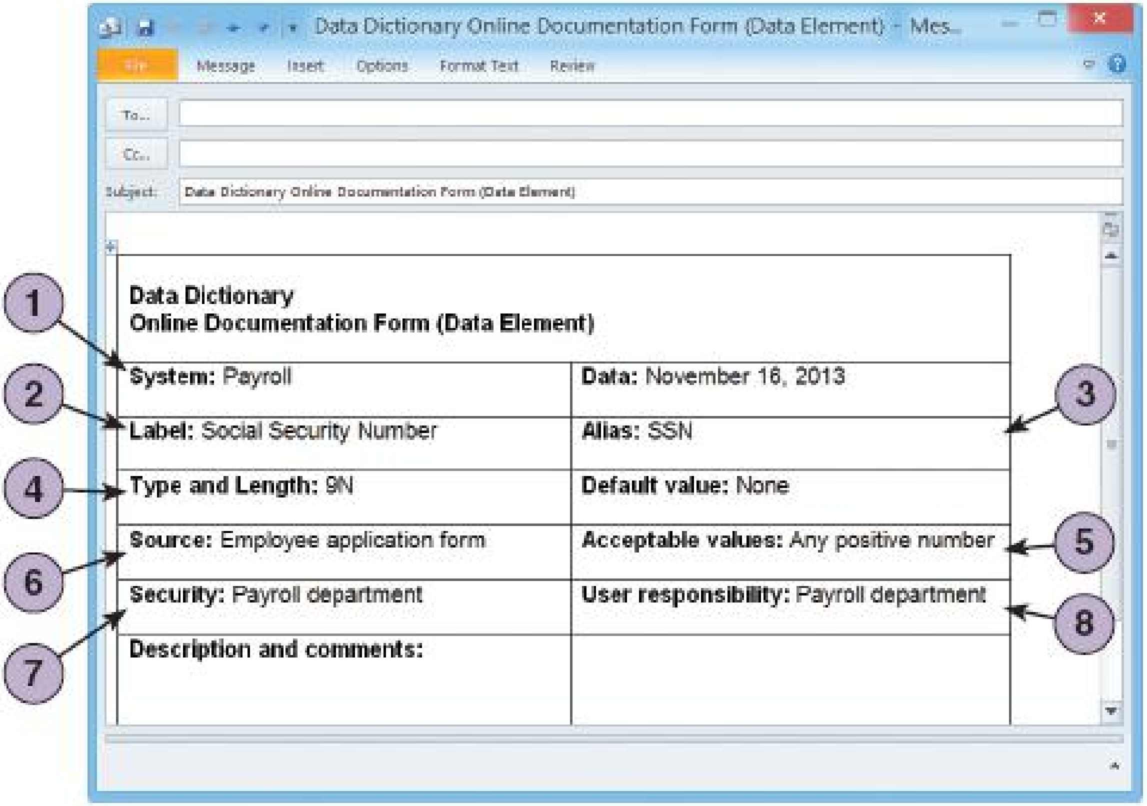
- Using CASE Tools for Documentation
  - The more complex the system, the more difficult it is to maintain full and accurate documentation
  - Modern CASE tools simplify the task
  - A CASE repository ensures data consistency
  - The CASE tools in Part B of the Systems Analyst's Toolkit can help you document business functions and processes
    - To learn more about these tools, turn to Part B of the four-part Toolkit that follows Chapter 12

Data Dictionary (Cont.)

- Documenting the Data Elements
  - You must document every data element in the data dictionary
  - The objective is the same: to provide clear, comprehensive information about the data and processes that make up the system

Data Dictionary (Cont.)

1. Online or manual documentation entries often indicate which system is involved. This is not necessary with a CASE tool because all information is stored in one file that is named for the system.
2. The data element has a standard label that provides consistency throughout the data dictionary.
3. The data element can have an alternative name, or alias.
4. This entry indicates that the data element consists of nine numeric characters.
5. Depending on the data element, strict limits might be placed on acceptable values.
6. The data comes from the employee's job application.
7. This entry indicates that only the payroll department has authority to update or change this data.
8. This entry indicates the individual or department responsible for entering and changing data.


Data Dictionary Online Documentation Form (Data Element) - Mes...
File
Message
Insert
Options
Format Text
Review
To...
Cc...
1
2
3
4
5
6
7
8

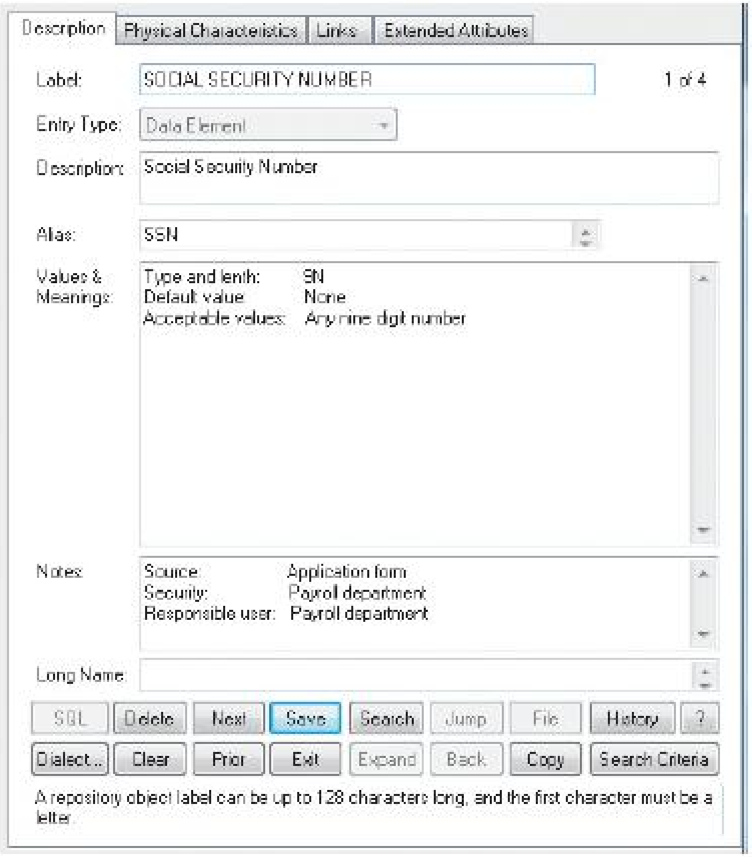
Subject: Data Dictionary Online Documentation Form (Data Element)

Data Dictionary **Online Documentation Form (Data Element)**	
System: Payroll	**Data:** November 16, 2013
Label: Social Security Number	**Alias:** SSN
Type and Length: 9N	**Default value:** None
Source: Employee application form	**Acceptable values:** Any positive number
Security: Payroll department	**User responsibility:** Payroll department
Description and comments:	

FIGURE 5-23 Using an online documentation form, the analyst has recorded information for a data element named SOCIAL SECURITY NUMBER. Later, the analyst will create a data dictionary entry using a CASE tool

Data Dictionary (Cont.)

- Documenting the Data Elements
  - Data element name and label
  - Alias
  - Type and length
  - Default value
  - Acceptable values - Domain and validity rules
  - Source
  - Security
  - Responsible user(s)
  - Description and comments



Label: SOCIAL SECURITY NUMBER 1 of 4

Entry Type: Data Element

Description: Social Security Number

Alias: SSN

Values & Meanings:
Type and lenth: SN
Default value None
Acceptable values: Any nine digit number

Notes:
Source: Application form
Security: Payroll department
Responsible user: Payroll department

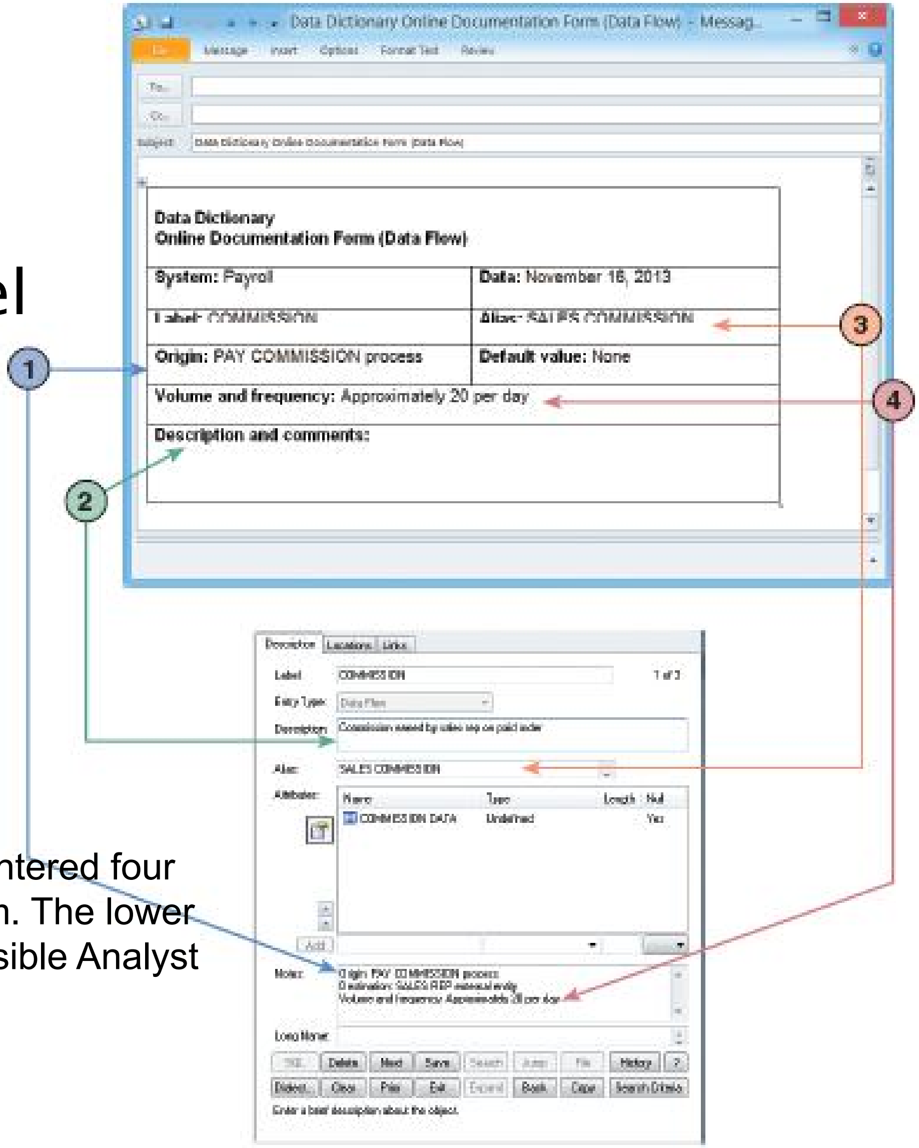
Long Name:

A repository object label can be up to 128 characters long, and the first character must be a letter

FIGURE 5-24 A Visible Analyst screen describes the data element named SOCIAL SECURITY NUMBER. Notice that many of the items were entered from the online form shown in Figure 5-23

Data Dictionary (Cont.)

- Documenting the Data Flows
  - Data flow name or label
  - Description
  - Alternate name(s)
  - Origin
  - Destination
  - Record
  - Volume and frequency



Data Dictionary Online Documentation Form (Data Flow) - Messag...

Subject: Data Dictionary Online Documentation Form (Data Flow)

Data Dictionary
Online Documentation Form (Data Flow)

System: Payroll	**Data:** November 16, 2013
Label: COMMISSION	**Alias:** SALES COMMISSION
Origin: PAY COMMISSION process	**Default value:** None
Volume and frequency: Approximately 20 per day	
Description and comments:	

Label: COMMISSION 1 of 3

Entry Type: Data Flow

Description: Commission earned by sales rep on paid order

Alias: SALES COMMISSION

Attributes:

Name	Type	Length	Null
COMMISSION DATA	Undefined		Yes

Notes: Origin: PAY COMMISSION process
Destination: SALES REP external entity
Volume and frequency: Approximately 20 per day

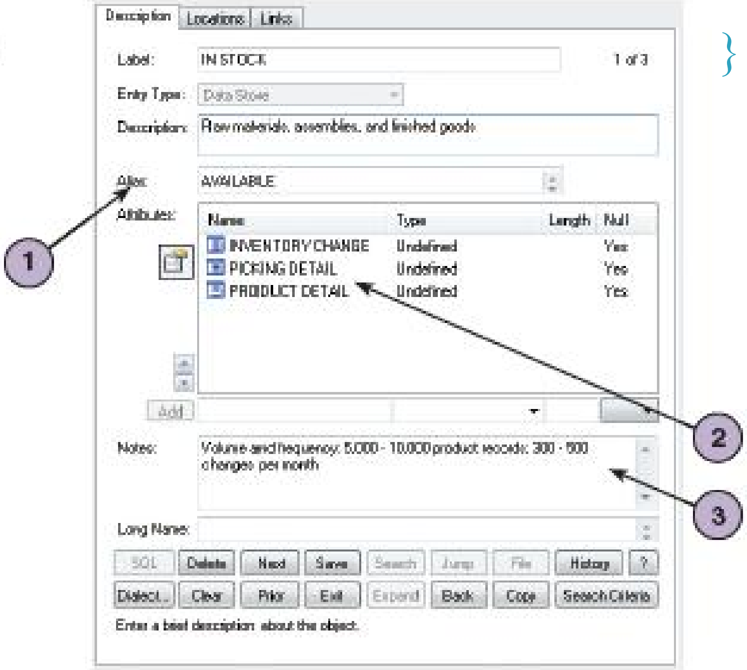
Long Name:

Enter a brief description about the object.

FIGURE 5-25 In the upper screen, an analyst has entered four items of information in an online documentation form. The lower screen shows the same four items entered into a Visible Analyst da

Data Dictionary (Cont.)

1. This data store has an alternative name, or alias.
2. For consistency, data flow names are standardized throughout the data dictionary.
3. It is important to document these estimates because they will affect design decisions in subsequent SDLC phases.



Label: IN STOCK 1 of 3

Entry Type: Data Store

Description: Raw materials, assemblies, and finished goods

Alias: AVAILABLE

Attributes:

Name	Type	Length	Null
INVENTORY CHANGE	Undefined		Yes
PICKING DETAIL	Undefined		Yes
PRODUCT DETAIL	Undefined		Yes

Notes: Volume and frequency: 5,000 - 10,000 product records; 300 - 500 changes per month

Long Name:

Enter a brief description about the object.

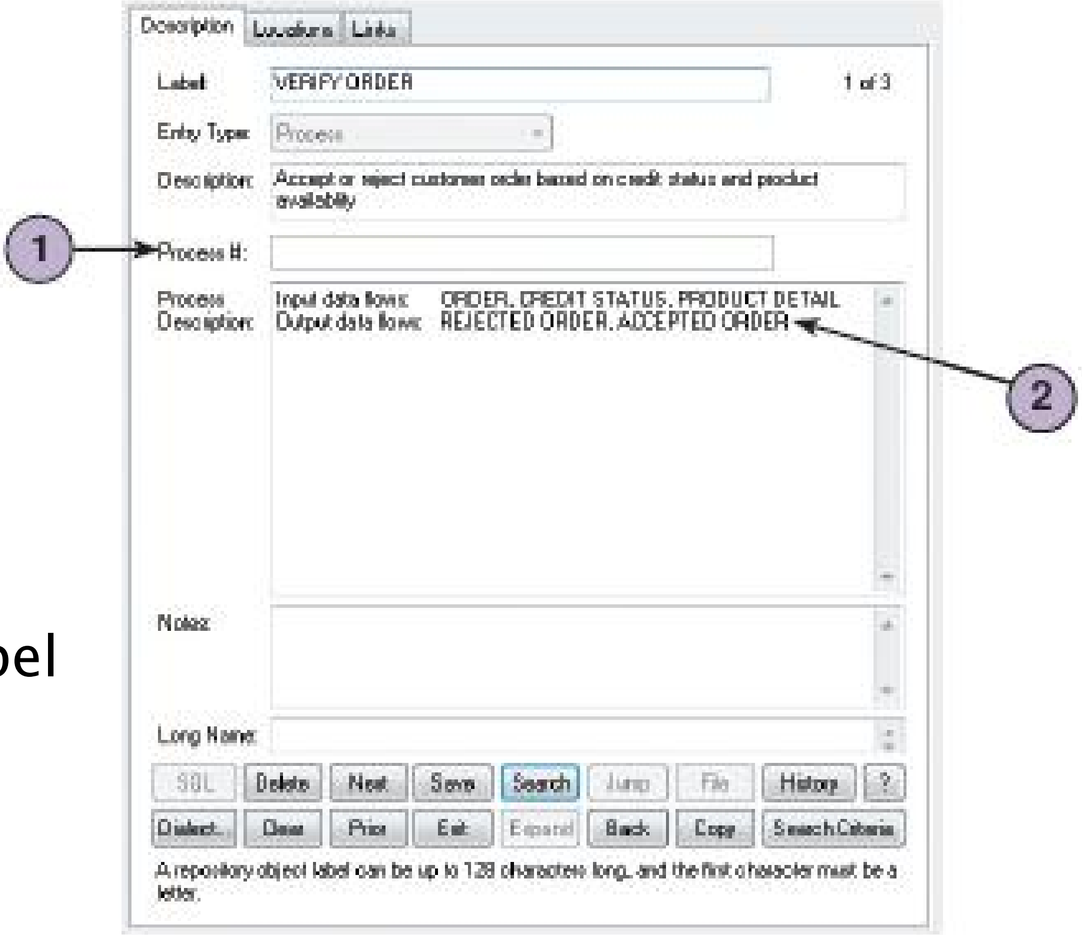
FIGURE 5-26 Visible Analyst screen that documents a data store named IN STOCK

- Documenting the Data Stores
  - Data store name or label
  - Description
  - Alternate name(s)
  - Attributes
  - Volume and frequency

Data Dictionary (Cont.)

1. The process number identifies this process. Any subprocesses are numbered 1.1, 1.2, 1.3, and so on.
2. These data flows will be described specifically elsewhere in the data dictionary.

- Documenting the Processes
  - Process name or label
  - Description
  - Process number
  - Process description



Label: VERIFY ORDER 1 of 3

Entry Type: Process

Description: Accept or reject customer order based on credit status and product availability

Process #:

Process Description: Input data flows: ORDER, CREDIT STATUS, PRODUCT DETAIL
Output data flows: REJECTED ORDER, ACCEPTED ORDER

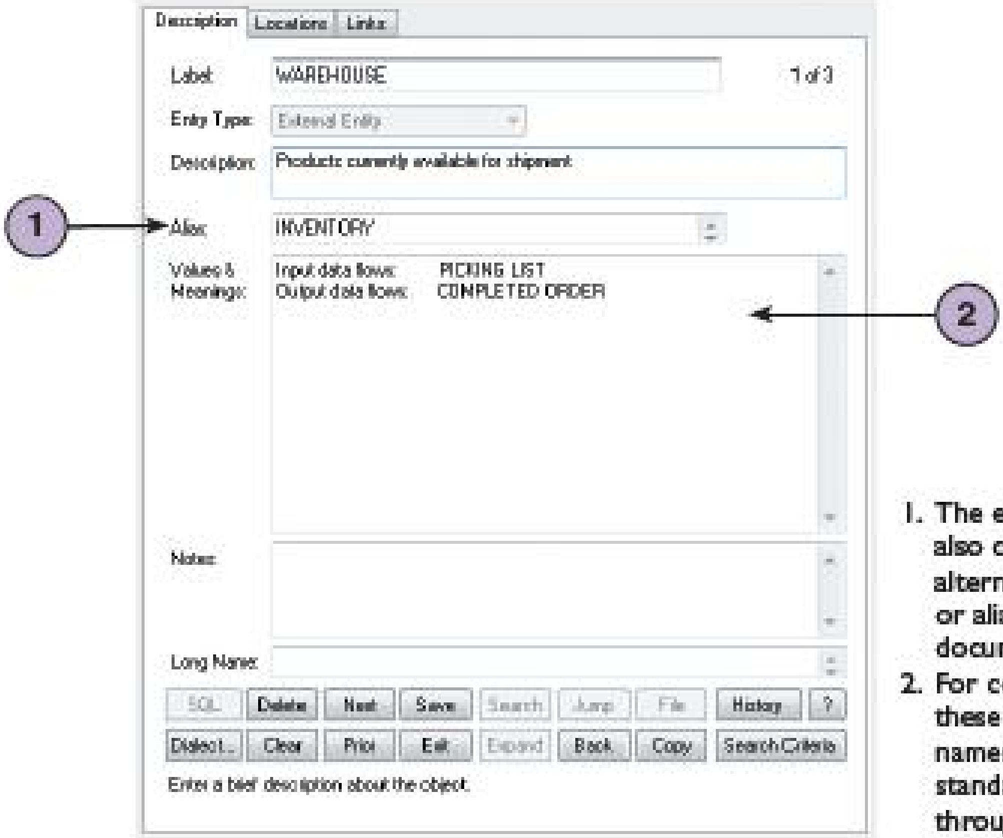
Notes:

Long Name:

A repository object label can be up to 128 characters long, and the first character must be a letter.

FIGURE 5-27 Visible Analyst screen that describes a process named VERIFY ORDER

Data Dictionary (Cont.)



Label: WAREHOUSE 1 of 3

Entry Type: External Entity

Description: Products currently available for shipment

Alias: INVENTORY

Values & Meanings: Input data flows: PICKING LIST
Output data flows: COMPLETED ORDER

Notes:

Long Name:

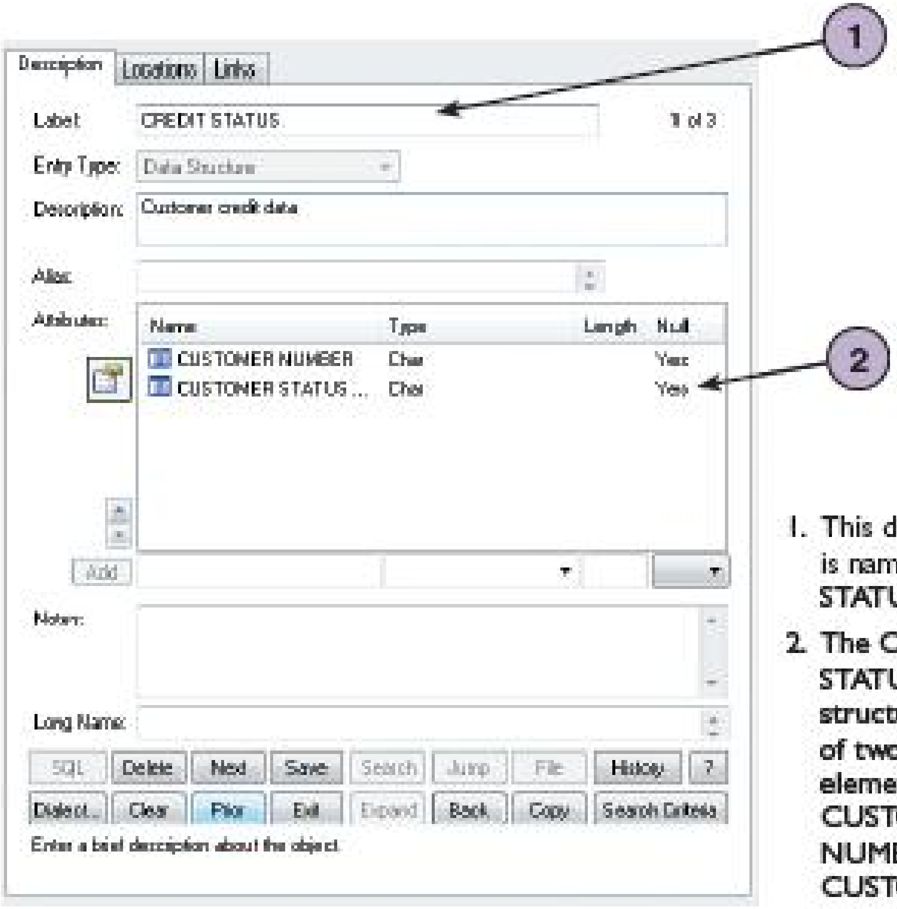
Enter a brief description about the object.

1. The external entity also can have an alternative name, or alias, if properly documented.
2. For consistency, these data flow names are standardized throughout the data dictionary.

Analyst screen that documents
med WAREHOUSE

- Documenting the Entities
  - Entity name
  - Description
  - Alternate name(s)
  - Input data flows
  - Output data flows

Data Dictionary (Cont.)



Label: CREDIT STATUS

1 of 3

Entry Type: Data Structure

Description: Customer credit data

Alias:

Attributes:

Name	Type	Length	Null
CUSTOMER NUMBER	Char		Yes
CUSTOMER STATUS ...	Char		Yes

Notes:

Long Name:

Enter a brief description about the object.

1. This data structure is named CREDIT STATUS.
2. The CREDIT STATUS data structure consists of two data elements: CUSTOMER NUMBER and CUSTOMER STATUS CODE.

- Documenting the Records
  - Record or data structure name
  - Definition or description
  - Alternate name(s)
  - Attributes

[redacted] Analyst screen that documents a [redacted]ure named CREDIT STATUS

Data Dictionary (Cont.)

- Data Dictionary Reports
  - Many valuable reports
    - An alphabetized list of all data elements by name
    - A report describing each data element and indicating the user or department that is responsible for data entry, updating, or deletion
    - A report of all data flows and data stores that use a particular data element
    - Detailed reports showing all characteristics of data elements, records, data flows, processes, or any other selected item stored in the data

Process Description Tools

- Typical process description tools include structured English, decision tables, and decision trees
- Process description tools also can be used in object-oriented development
  - O-O programmers use different terminology. They create the same kind of modular coding structures, except that the processes, or methods, are stored inside the objects, rather than as separate components

Process Description Tools (Cont.)

- Modular Design
  - Based on combinations of three logical structures, sometimes called control structures, which serve as building blocks for the process
    - Sequence
    - Selection
    - Iteration – looping

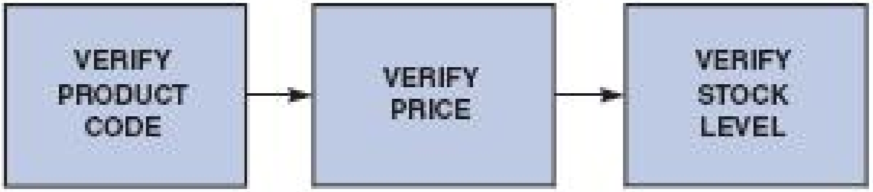


FIGURE 5-30 Sequence structure

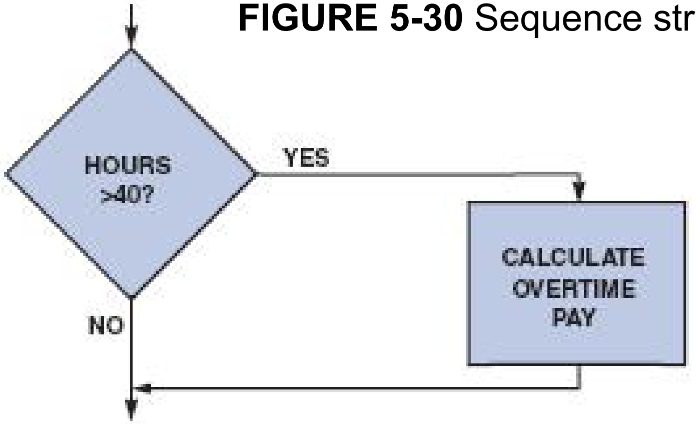


FIGURE 5-31 Selection structure

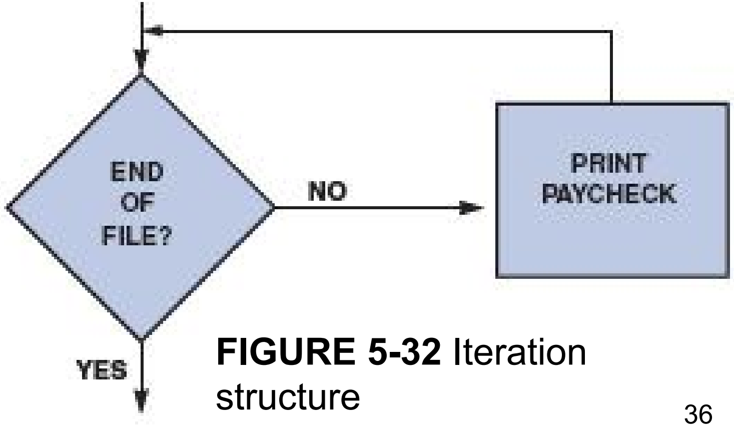
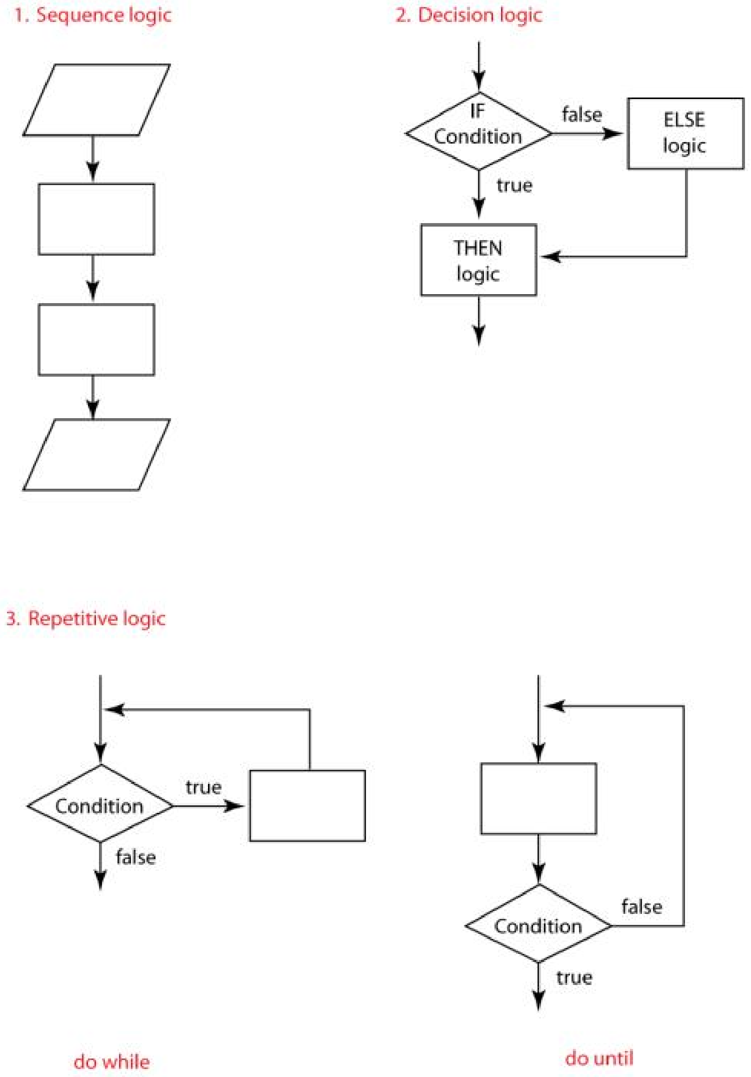


FIGURE 5-32 Iteration structure

Program logic



1. Sequence logic

2. Decision logic

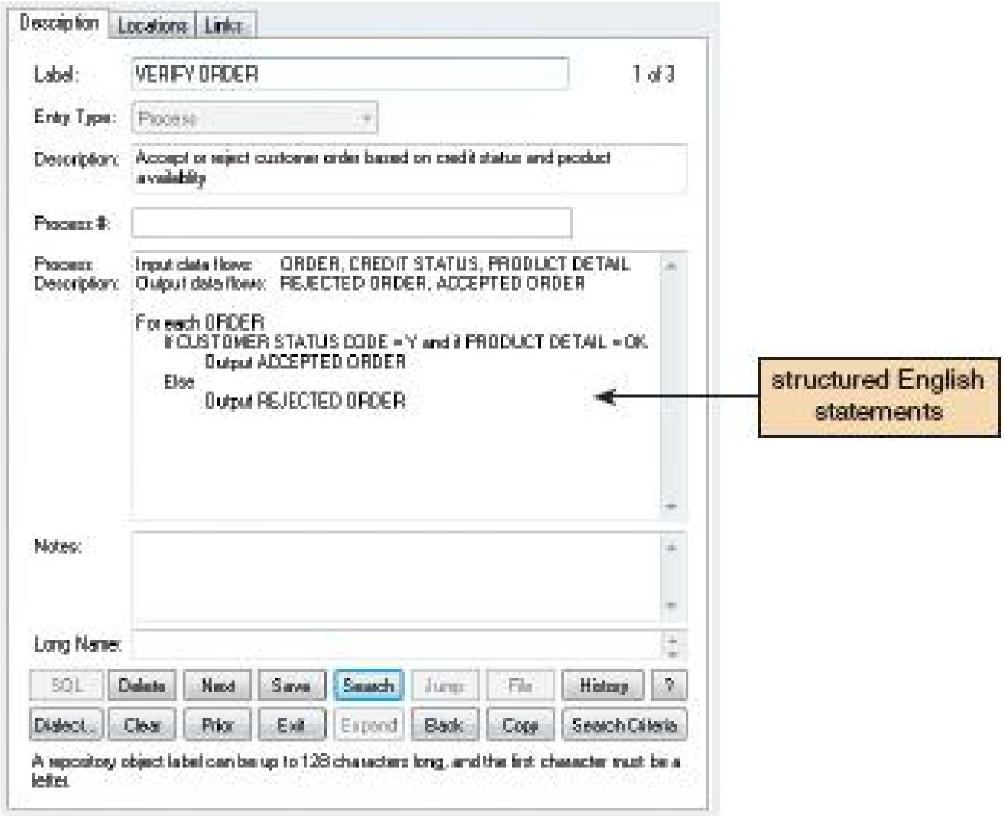
3. Repetitive logic

do while

do until

Process Description Tools (Cont.)

- Structured English
  - Must conform to the following rules
    - Use only the three building blocks of sequence, selection, and iteration
    - Use indentation for readability
    - Use a limited vocabulary, including standard terms used in the data dictionary and specific words that describe the processing rules



Label: VERIFY ORDER 1 of 3

Entry Type: Process

Description: Accept or reject customer order based on credit status and product availability

Process #:

Process Description:

```
Input data flows:   ORDER, CREDIT STATUS, PRODUCT DETAIL
Output data flows:  REJECTED ORDER, ACCEPTED ORDER

For each ORDER
    If CUSTOMER STATUS CODE = Y and if PRODUCT DETAIL = OK
        Output ACCEPTED ORDER
    Else
        Output REJECTED ORDER
```

Notes:

Long Name:

A repository object label can be up to 128 characters long, and the first character must be a letter.

structured English statements

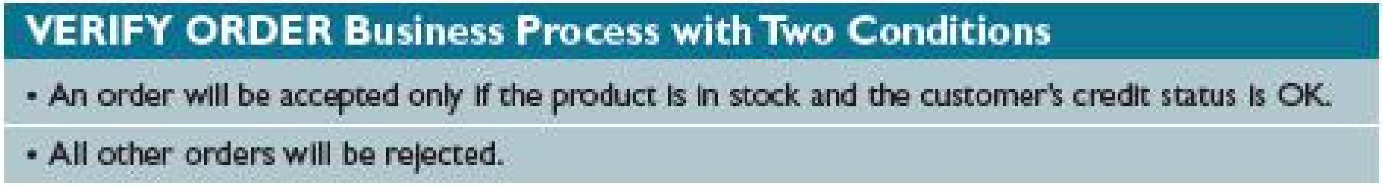
FIGURE 5-33 The VERIFY ORDER process description includes logical rules and a structured English version of the policy. Notice the alignment and indentation of the logic statements

Process Description Tools (Cont.)

- Decision Tables
  - Shows a logical structure, with all possible combinations of conditions and resulting actions
  - It is important to consider every possible outcome to ensure that you have overlooked nothing
  - The number of rules doubles each time you add a condition
  - Can have more than two possible outcomes
  - Often are the best way to describe a complex set of conditions

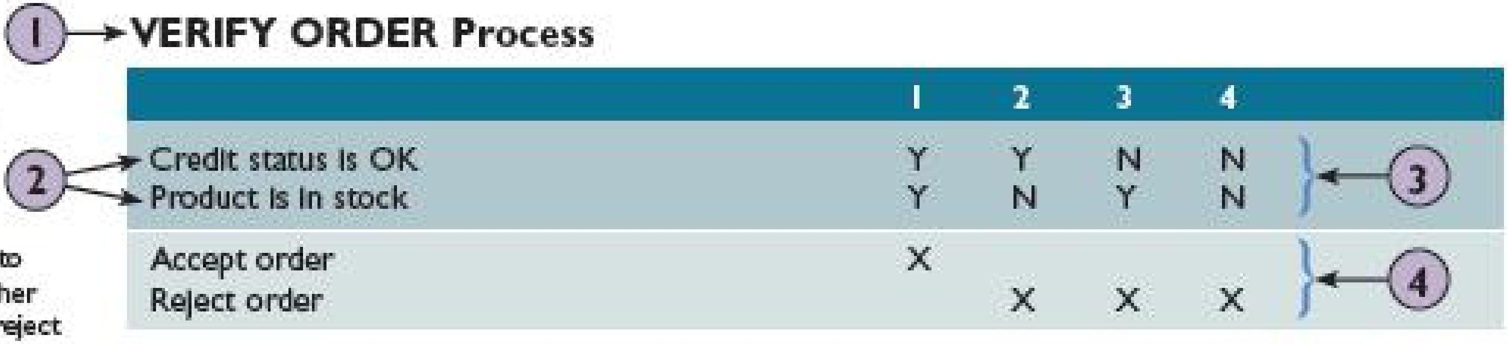
Process Description Tools (Cont.)

1. Place the name of the process in a heading at the top left.
2. Enter the conditions under the heading, with one condition per line, to represent the customer status and availability of products.
3. Enter all potential combinations of Y/N (for yes and no) for the conditions. Each column represents a numbered possibility called a rule.
4. Place an X in the action entries area for each rule to indicate whether to accept or reject the order.



VERIFY ORDER Business Process with Two Conditions
• An order will be accepted only if the product is in stock and the customer's credit status is OK.
• All other orders will be rejected.

FIGURE 5-34 The Verify Order business process has two conditions. For an order to be accepted, the product must be in stock and the customer must have an acceptable credit status

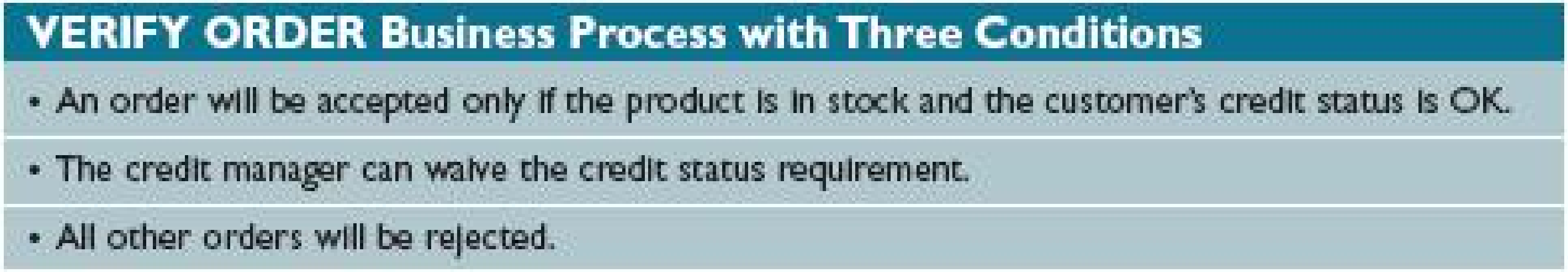


VERIFY ORDER Process

	1	2	3	4
Credit status is OK	Y	Y	N	N
Product is in stock	Y	N	Y	N
Accept order	X			
Reject order		X	X	X

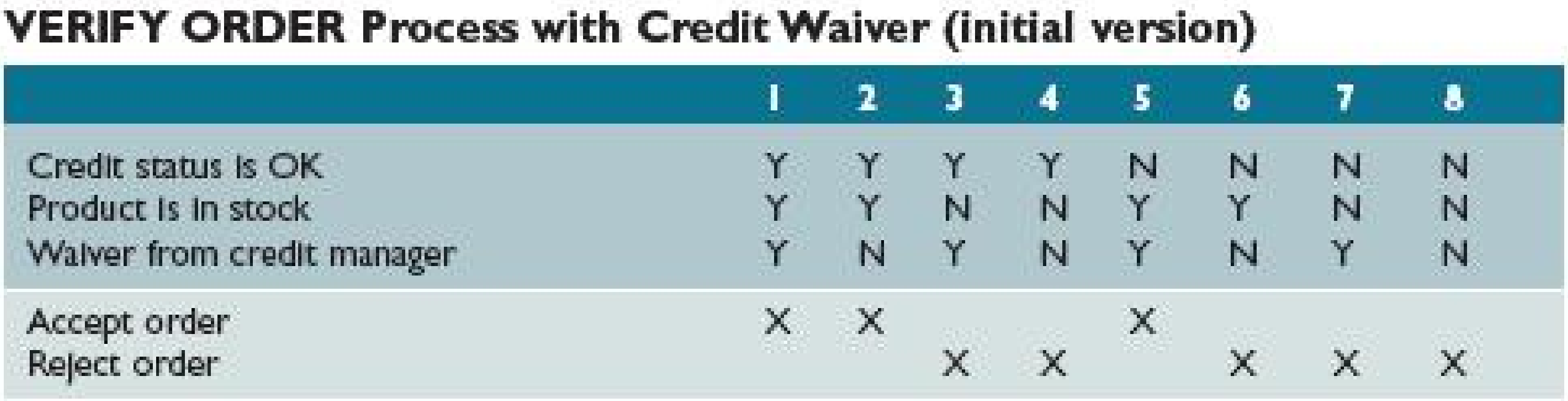
FIGURE 5-35 Example of a simple decision table showing the processing logic of the VERIFY ORDER process

Process Description Tools (Cont.)



VERIFY ORDER Business Process with Three Conditions
• An order will be accepted only if the product is in stock and the customer's credit status is OK.
• The credit manager can waive the credit status requirement.
• All other orders will be rejected.

FIGURE 5-36 A third condition has been added to the Verify Order business process. For an order to be accepted, the product must be in stock and the customer must have an acceptable credit status. However, the credit manager now has the authority to waive the credit status requirement

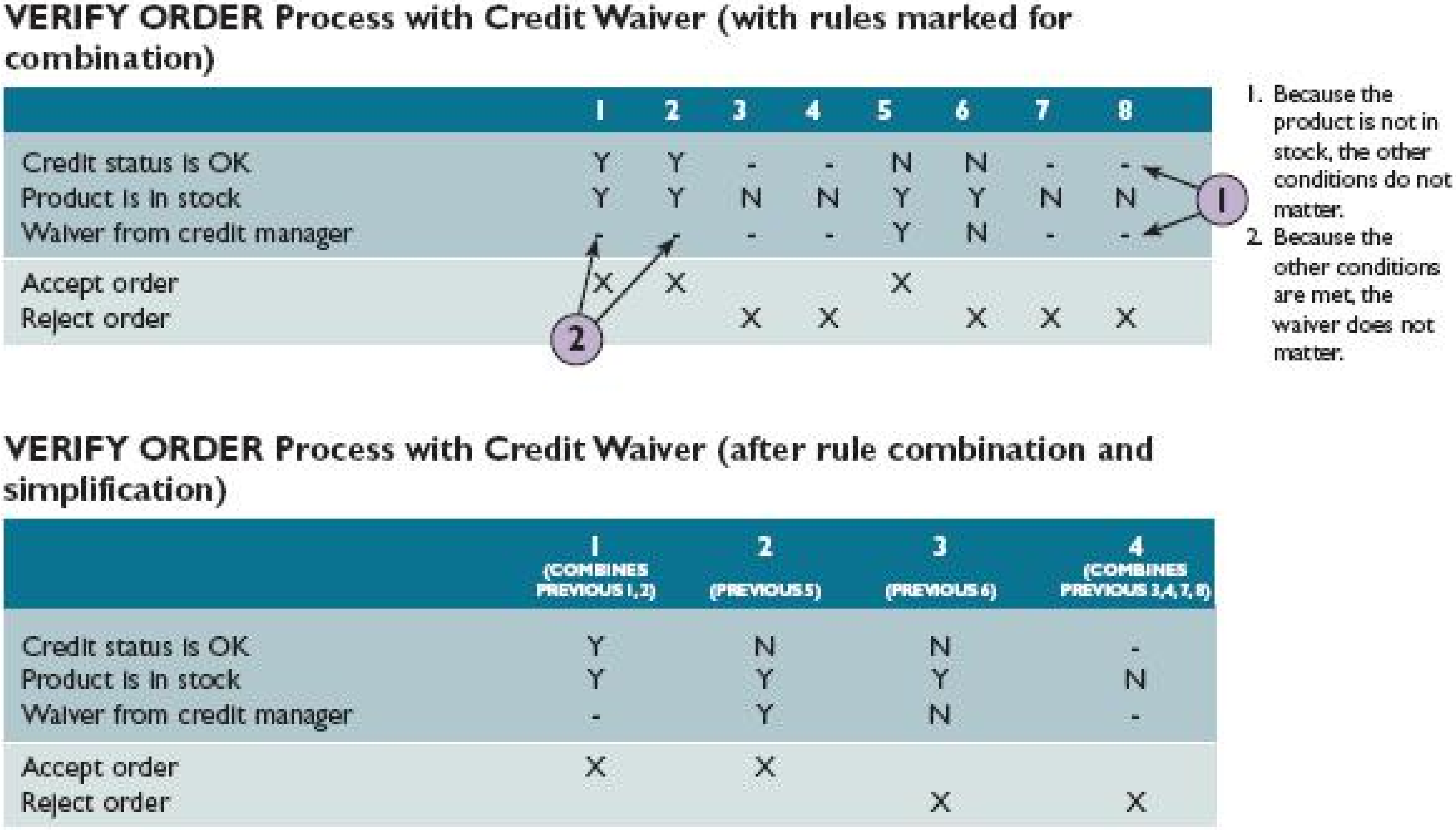


VERIFY ORDER Process with Credit Waiver (initial version)

	1	2	3	4	5	6	7	8
Credit status is OK	Y	Y	Y	Y	N	N	N	N
Product is in stock	Y	Y	N	N	Y	Y	N	N
Waiver from credit manager	Y	N	Y	N	Y	N	Y	N
Accept order	X	X			X			
Reject order			X	X		X	X	X

ble is based on the Verify Order conditions shown in Figure
itions, there are eight possible combinations, or rules

Process Description Tools (Cont.)



VERIFY ORDER Process with Credit Waiver (with rules marked for combination)

	1	2	3	4	5	6	7	8
Credit status is OK	Y	Y	-	-	N	N	-	-
Product is in stock	Y	Y	N	N	Y	Y	N	N
Waiver from credit manager	-	-	-	-	Y	N	-	-
Accept order	X	X			X			
Reject order			X	X		X	X	X

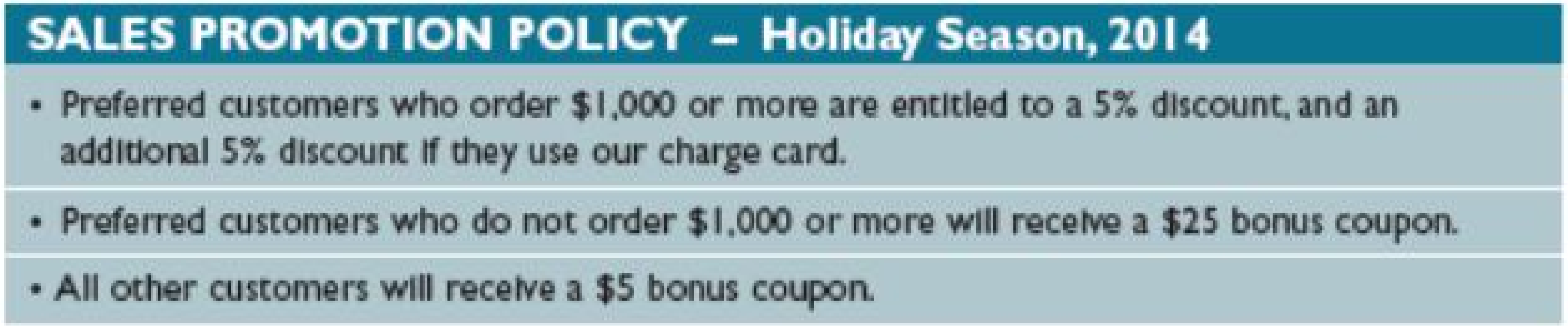
1. Because the product is not in stock, the other conditions do not matter.
2. Because the other conditions are met, the waiver does not matter.

VERIFY ORDER Process with Credit Waiver (after rule combination and simplification)

	1 (COMBINES PREVIOUS 1, 2)	2 (PREVIOUS 5)	3 (PREVIOUS 6)	4 (COMBINES PREVIOUS 3, 4, 7, 8)
Credit status is OK	Y	N	N	-
Product is in stock	Y	Y	Y	N
Waiver from credit manager	-	Y	N	-
Accept order	X	X		
Reject order			X	X

FIGURE 5-38 In the first table, dashes have been added to indicate hat a condition is not relevant. In the second version, rules have een combined. Notice that in final version, only four rules remain. hese rules document the logic, and will be transformed into rogram code when the system is developed

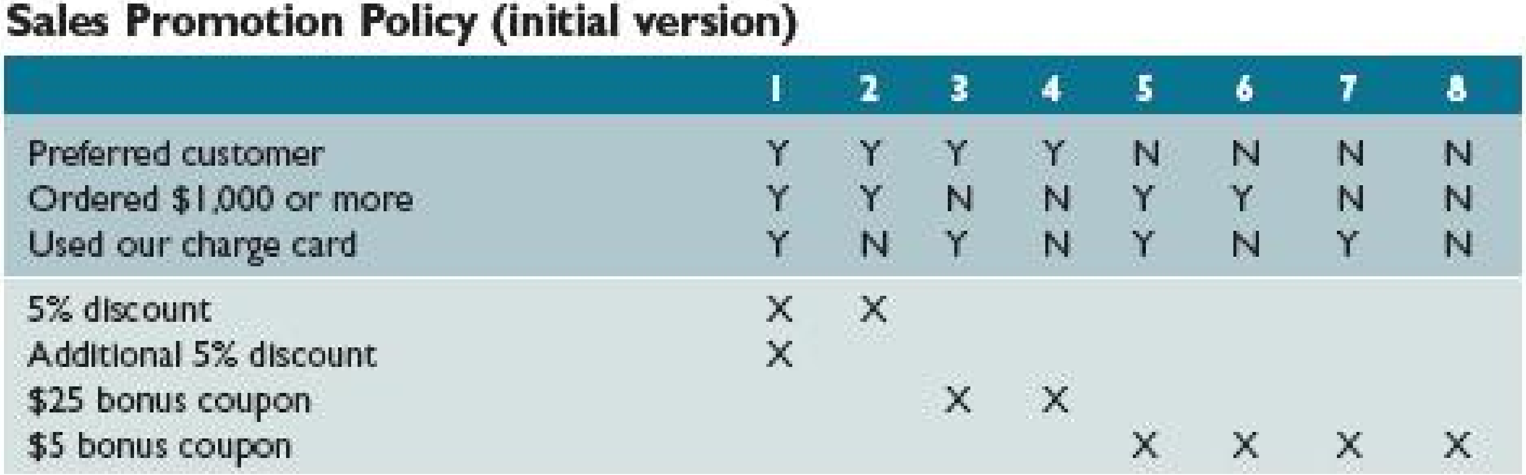
Process Description Tools (Cont.)



SALES PROMOTION POLICY – Holiday Season, 2014

- Preferred customers who order $1,000 or more are entitled to a 5% discount, and an additional 5% discount if they use our charge card.
- Preferred customers who do not order $1,000 or more will receive a $25 bonus coupon.
- All other customers will receive a $5 bonus coupon.

FIGURE 5-39 A sales promotion policy with three conditions. Notice that the first statement contains two *separate* conditions – one for the 5% discount, and another for the additional discount

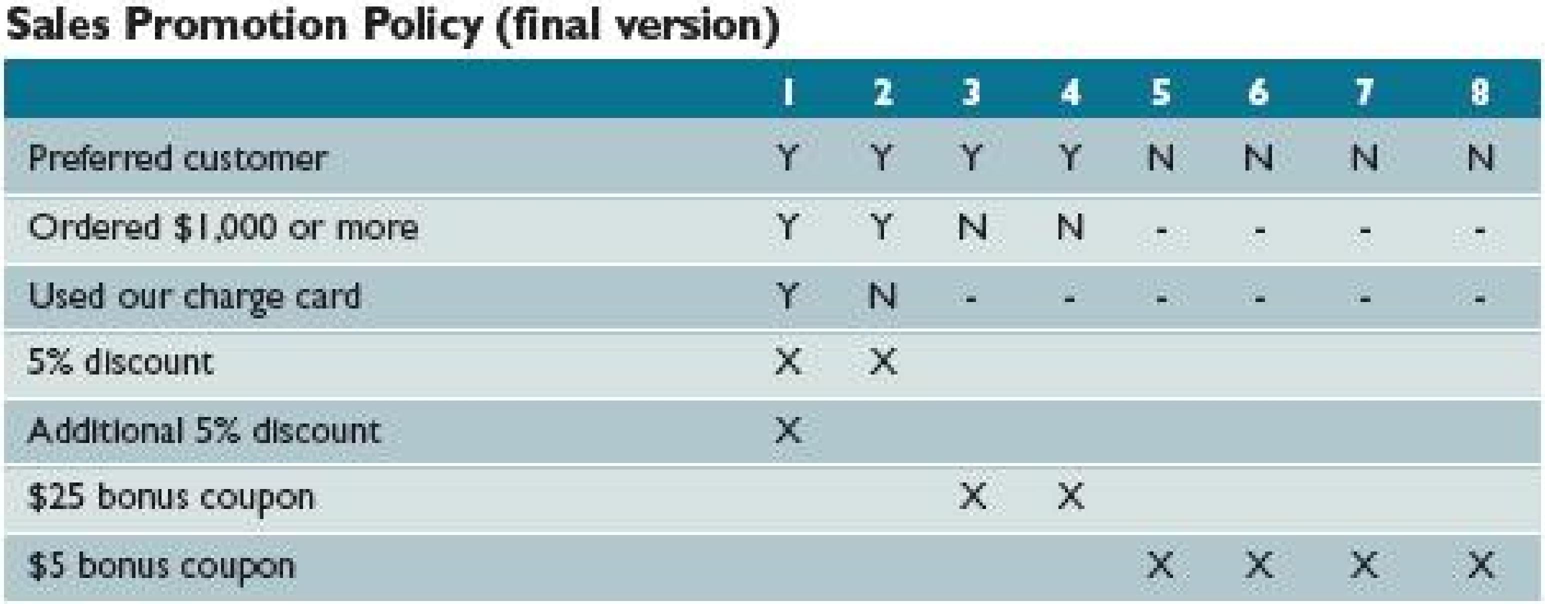


Sales Promotion Policy (initial version)

	1	2	3	4	5	6	7	8
Preferred customer	Y	Y	Y	Y	N	N	N	N
Ordered $1,000 or more	Y	Y	N	N	Y	Y	N	N
Used our charge card	Y	N	Y	N	Y	N	Y	N
5% discount	X	X						
Additional 5% discount	X							
$25 bonus coupon			X	X				
$5 bonus coupon					X	X	X	X

[redacted] **5-40** This decision table is based on the sales promotion policy in -39. This is the initial version of the table, before simplification

Process Description Tools (Cont.)



Sales Promotion Policy (final version)

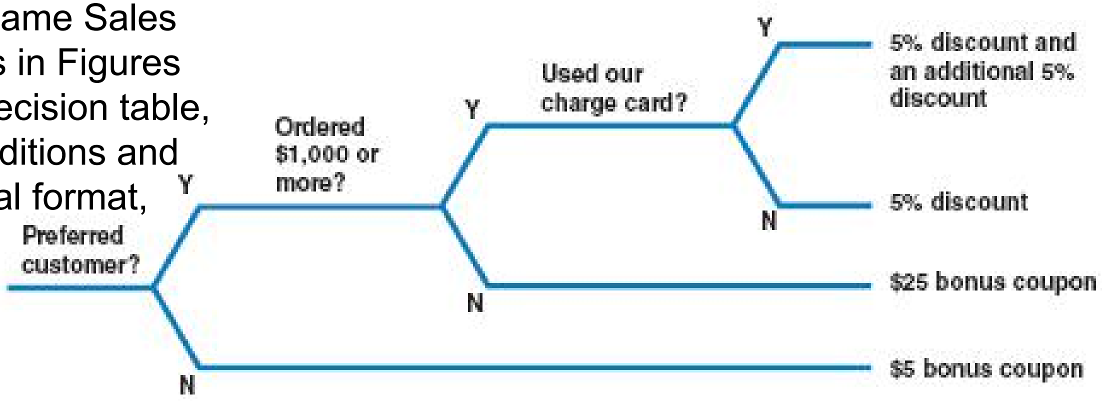
	1	2	3	4	5	6	7	8
Preferred customer	Y	Y	Y	Y	N	N	N	N
Ordered $1,000 or more	Y	Y	N	N	-	-	-	-
Used our charge card	Y	N	-	-	-	-	-	-
5% discount	X	X						
Additional 5% discount	X							
$25 bonus coupon			X	X				
$5 bonus coupon					X	X	X	X

FIGURE 5-41 In this version, dashes have been added to indicate that a condition is not relevant. At this point, it appears that several rules can be combined

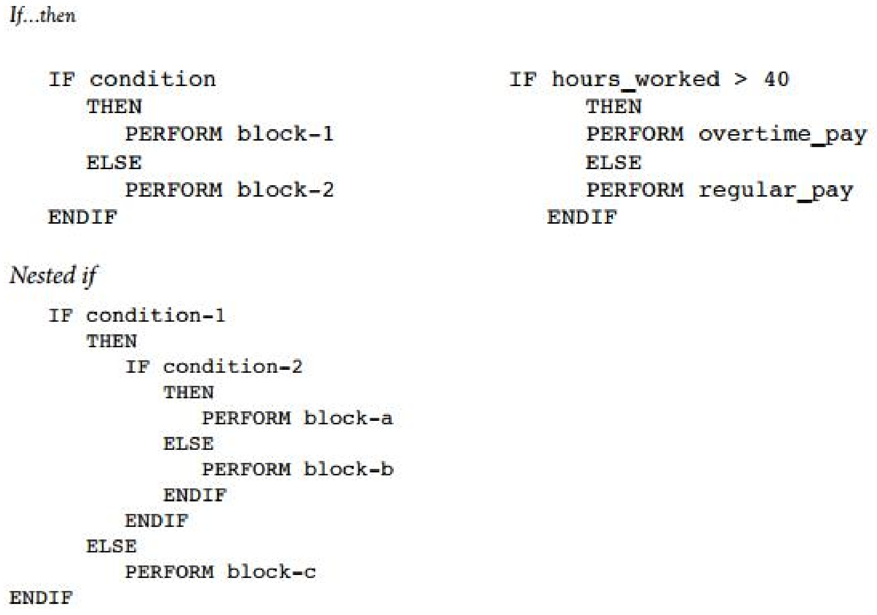
Process Description Tools (Cont.)

- Decision Trees
  - Graphical representation of the conditions, actions, and rules found in a decision table
  - Show the logic structure in a horizontal form that resembles a tree with the roots at the left and the branches to the right
  - Decision trees and decision tables provide the same results, but in different forms

FIGURE 5-42 This example is based on the same Sales Promotion Policy shown in the decision tables in Figures 5-40 and 5-41 on the previous page. Like a decision table, a decision tree shows all combinations of conditions and outcomes. The main difference is the graphical format, which many viewers find easier to interpret



Pseudocode



If...then

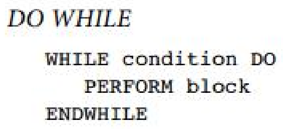
```
   IF condition
      THEN
         PERFORM block-1
      ELSE
         PERFORM block-2
   ENDIF
```

```
IF hours_worked > 40
      THEN
      PERFORM overtime_pay
      ELSE
      PERFORM regular_pay
   ENDIF
```

Nested if

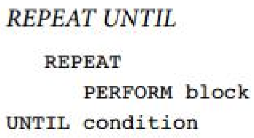
```
   IF condition-1
      THEN
         IF condition-2
            THEN
               PERFORM block-a
            ELSE
               PERFORM block-b
            ENDIF
         ENDIF
      ELSE
         PERFORM block-c
ENDIF
```

Pseudocode



DO WHILE

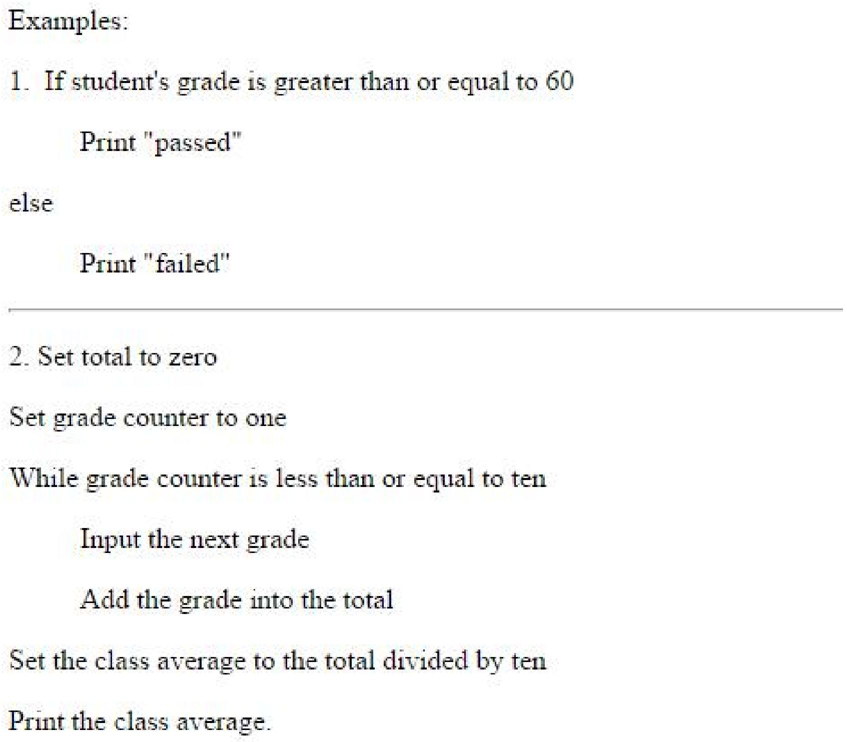
```
  WHILE condition DO
    PERFORM block
  ENDWHILE
```



REPEAT UNTIL

```
  REPEAT
    PERFORM block
UNTIL condition
```

Pseudocode



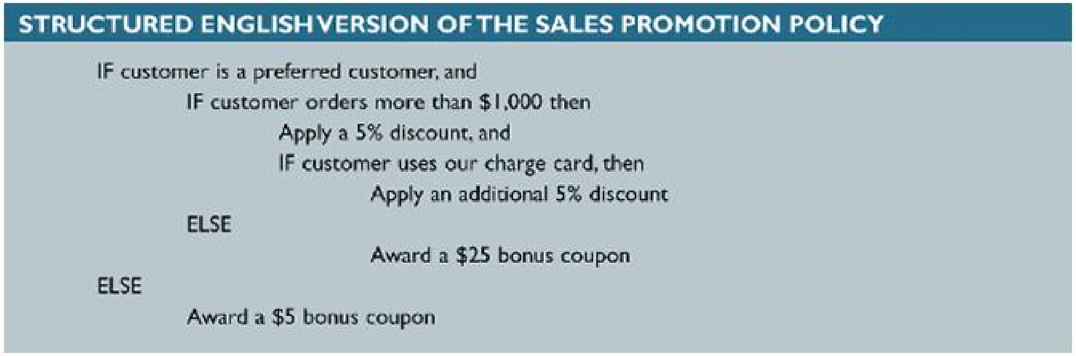
Examples:

```
1.  If student's grade is greater than or equal to 60
        Print "passed"
else
        Print "failed"
```

```
2. Set total to zero
Set grade counter to one
While grade counter is less than or equal to ten
        Input the next grade
        Add the grade into the total
Set the class average to the total divided by ten
Print the class average.
```

Pseudocode

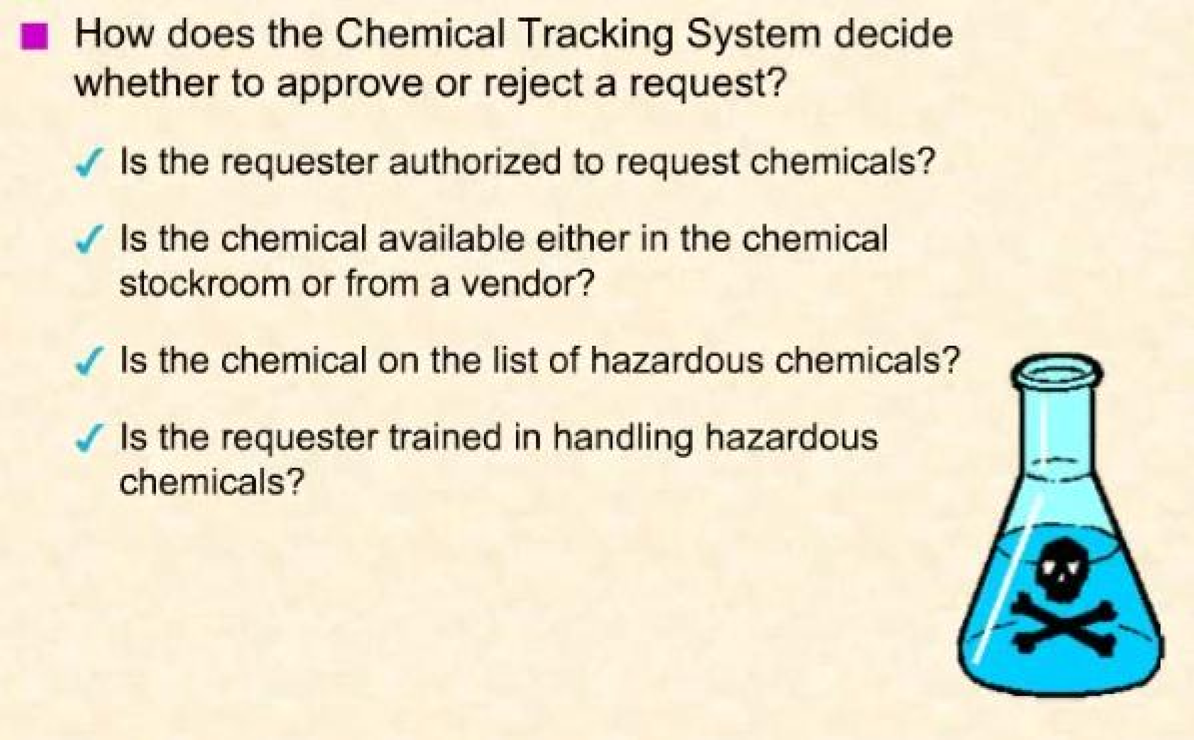
- Structured English
  - – Might look familiar to programming students because it resembles pseudocode
  - – The primary purpose of structured English is to describe the underlying business logic



STRUCTURED ENGLISH VERSION OF THE SALES PROMOTION POLICY

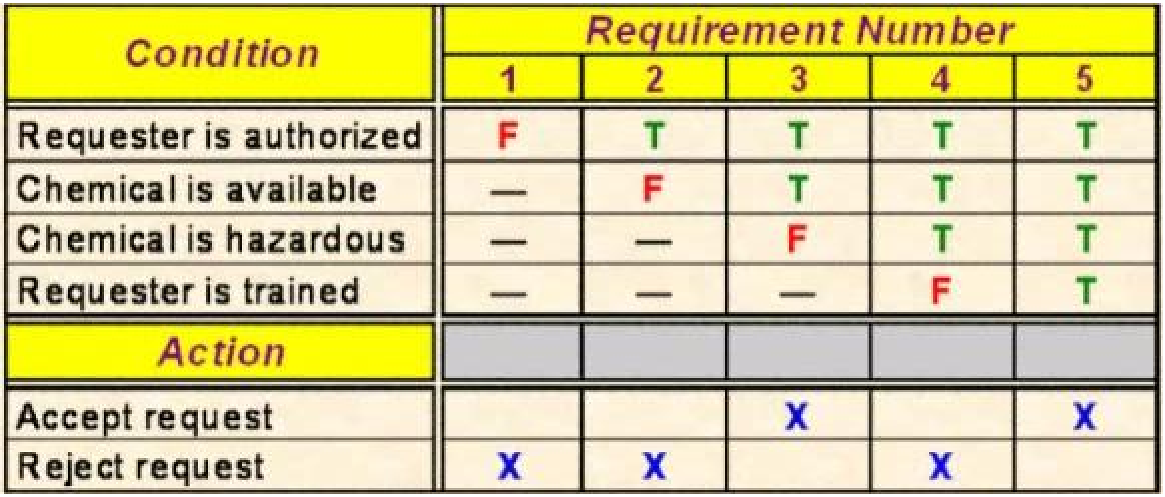
```
IF customer is a preferred customer, and
        IF customer orders more than $1,000 then
                Apply a 5% discount, and
                IF customer uses our charge card, then
                        Apply an additional 5% discount
        ELSE
                        Award a $25 bonus coupon
ELSE
        Award a $5 bonus coupon
```

Decision Table & Decision Tree



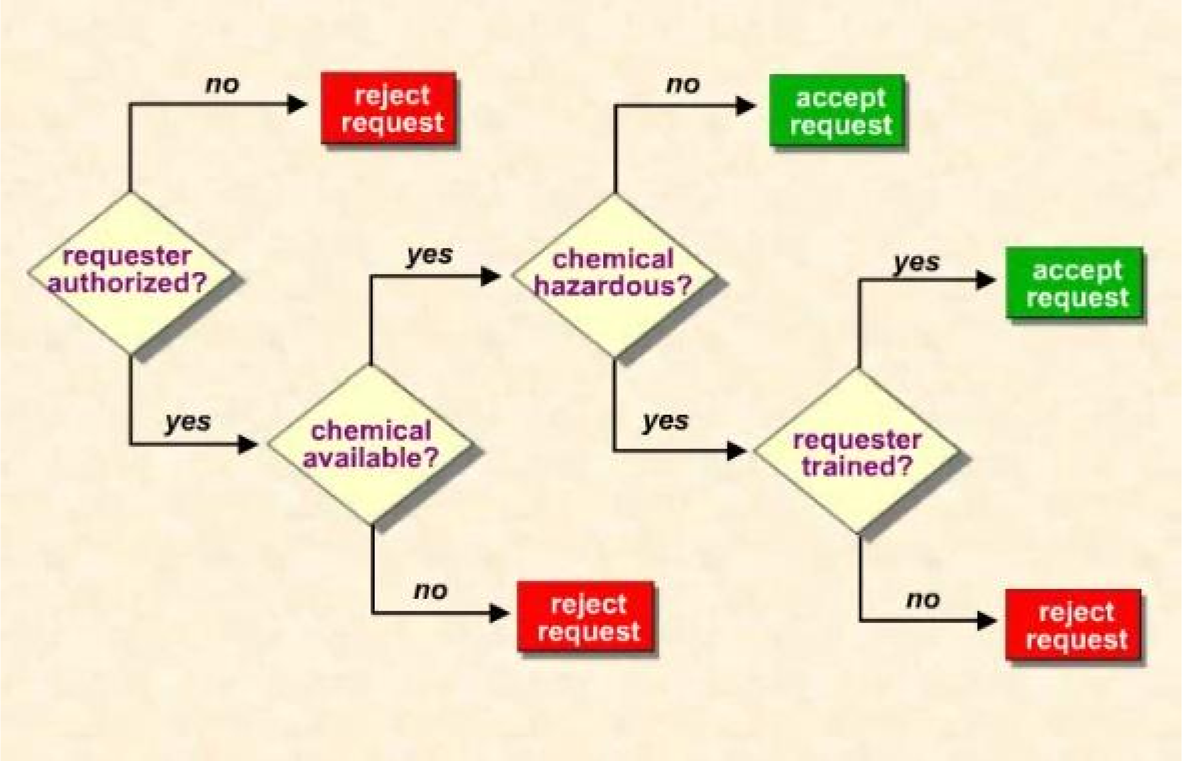
- How does the Chemical Tracking System decide whether to approve or reject a request?
  - ✓ Is the requester authorized to request chemicals?
  - ✓ Is the chemical available either in the chemical stockroom or from a vendor?
  - ✓ Is the chemical on the list of hazardous chemicals?
  - ✓ Is the requester trained in handling hazardous chemicals?

A Sample Decision Table



Condition	Requirement Number				
	1	2	3	4	5
Requester is authorized	F	T	T	T	T
Chemical is available	—	F	T	T	T
Chemical is hazardous	—	—	F	T	T
Requester is trained	—	—	—	F	T
Action					
Accept request			X		X
Reject request	X	X		X	

A Sample Decision Tree



Logical versus Physical Models

- While structured analysis tools are used to develop a logical model for a new information system, such tools also can be used to develop physical models of an information system
- A physical model shows how the system's requirements are implemented

Logical versus Physical Models
(Cont.)

- Sequence of Models
  - Many systems analysts create a physical model of the current system and then develop a logical model of the current system before tackling a logical model of the new system
  - Performing that extra step allows them to understand the current system better

Logical versus Physical Models

(Cont.)

- Four-Model Approach
  - Develop
    - A physical model of the current system
    - A logical model of the current system
    - A logical model of the new system
    - A physical model of the new system
  - The only disadvantage of the four-model approach is the added time and cost

Chapter Summary

- During data and process modeling, a systems analyst develops graphical models to show how the system transforms data into useful information
- The end product of data and process modeling is a logical model that will support business operations and meet user needs
- Data and process modeling involves three main tools: data flow diagrams, a data dictionary, and process descriptions

Chapter Summary (Cont.)

- Data flow diagrams (DFDs) graphically show the movement and transformation of data in the information system
- DFDs use four symbols
- A set of DFDs is like a pyramid with the context diagram at the top
- The data dictionary is the central documentation tool for structured analysis

Chapter Summary (Cont.)

- Each functional primitive process is documented using structured English, decision tables, and decision trees
- Structured analysis tools can be used to develop a logical model during one systems analysis phase, and a physical model during the systems design phase