COMP1010 – Week 6
Data and Process Modelling

Dr. Raymond Chiong, Dr. Mira Park, Dr. Mark Wallis
COMP1010 – Introduction to Computing
University of Newcastle

Learning 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

Learning 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

Introduction

- **Logical Model**: Shows *what* the system must do, regardless of how it will be implemented physically
- **Physical Model**: Describes *how* the system will be constructed

Overview of Data and Process Modeling Tools

- Systems analysts use graphical techniques to describe an information system
- 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. 1)

- DFD Symbols
  - Four basic symbols represent processes, data flows, data stores, and entities
  - **Gane and Sarson**: Used in data flow diagrams
    - Processes, data flows, data stores, and external entities all have a unique symbol
  - **Yourdon**: Used in data flow diagrams
    - Processes, data flows, data stores, and external entities each have a unique symbol

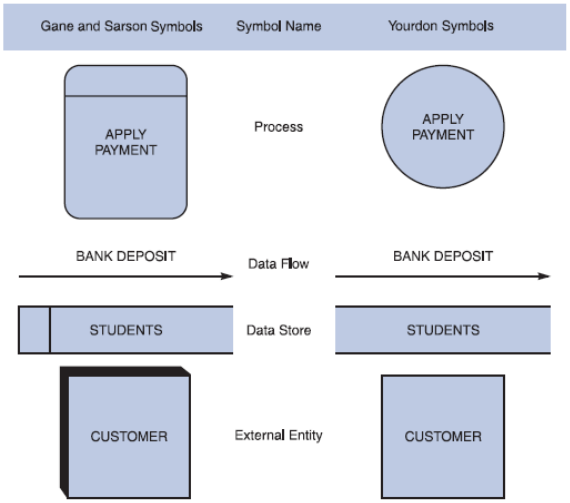


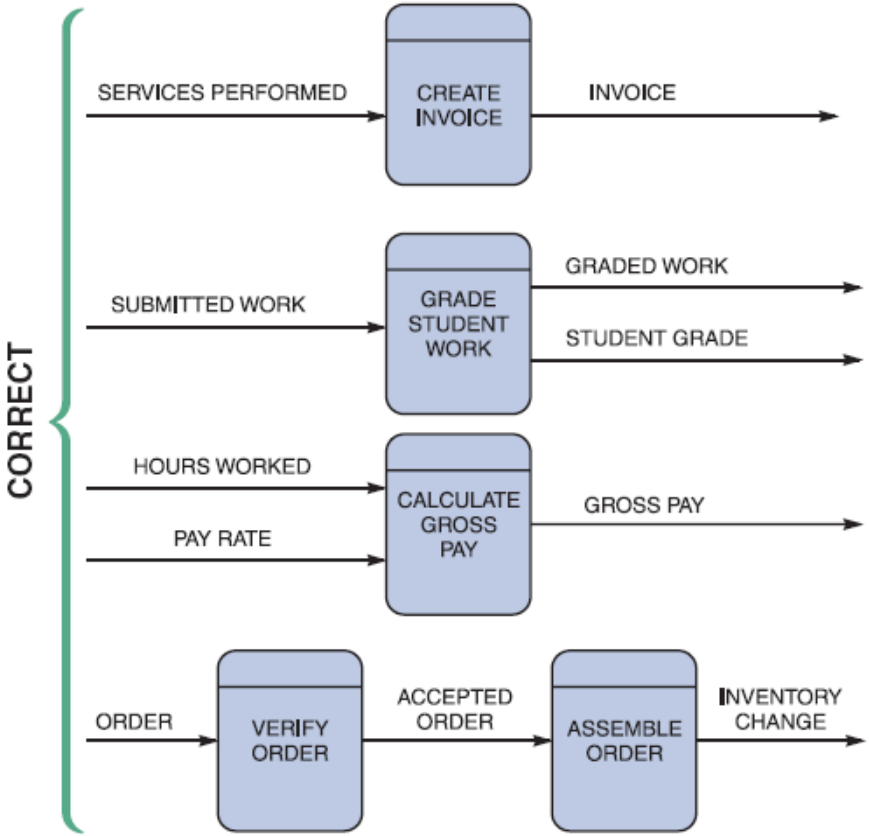
FIGURE 5-1 Data flow diagram symbols, symbol names, and examples of the Gane and Sarson and Yourdon symbol sets

Data Flow Diagrams (Cont. 2)

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)
- Examples" : "apply rent payment" or "calculate commission
- In DFDs, a process symbol can be referred to as a **black box**

Data Flow Diagrams (Cont. 3)



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

FIGURE 5-3 Examples of correct combinations of data flow and process symbols

Data Flow Diagrams (Cont. 4)

- **Data Flow Symbol**
  - Following data flow and process combinations must be avoided
    - **Spontaneous generation**
    - **Black holes**
    - **Gray holes**

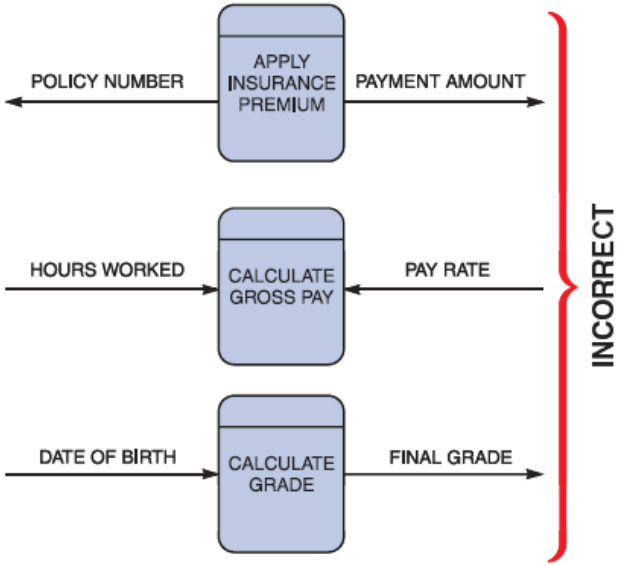


FIGURE 5-4 Examples of incorrect combinations of data flow and process symbols. APPLY INSURANCE PREMIUM has no input and is called a spontaneous generation process. CALCULATE GROSS PAY has no outputs and is called a black hole process. CALCULATE GRADE has an input that is obviously unable to produce the output. This process is called a gray hole

Data Flow Diagrams (Cont. 5)

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. 6)

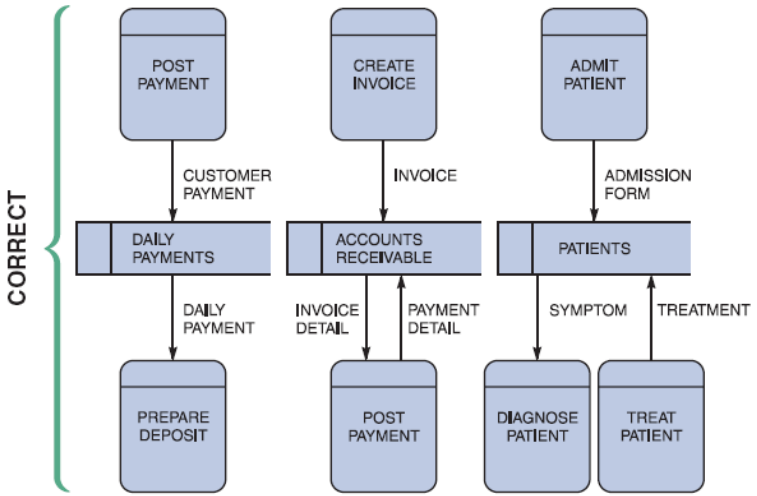
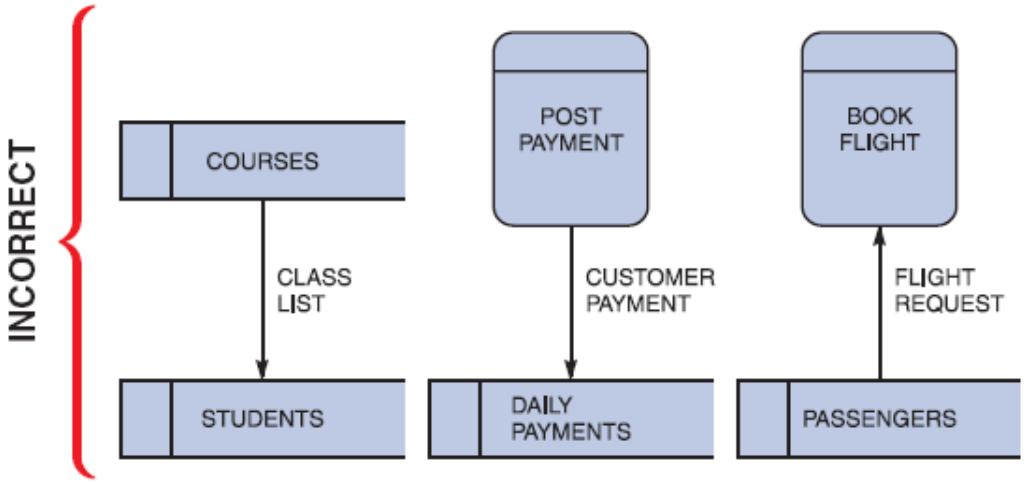


FIGURE 5-5 Examples of correct uses of data store symbols in a data flow diagram

FIGURE 5-6 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



THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Data Flow Diagrams (Cont. 7)

Entity Symbol

- 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. 8)

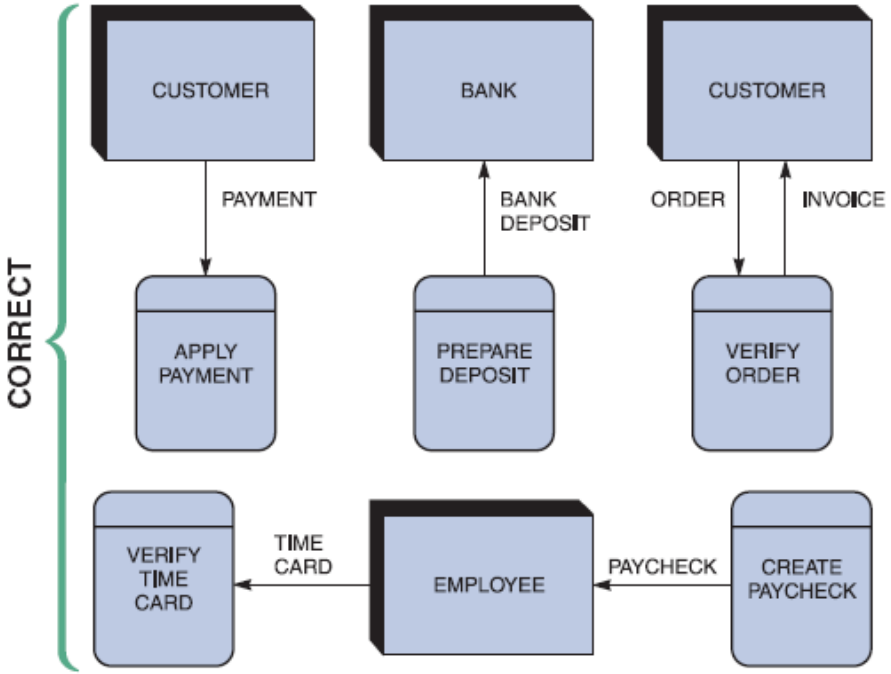


FIGURE 5-7 Examples of correct uses of external entities in a data flow diagram

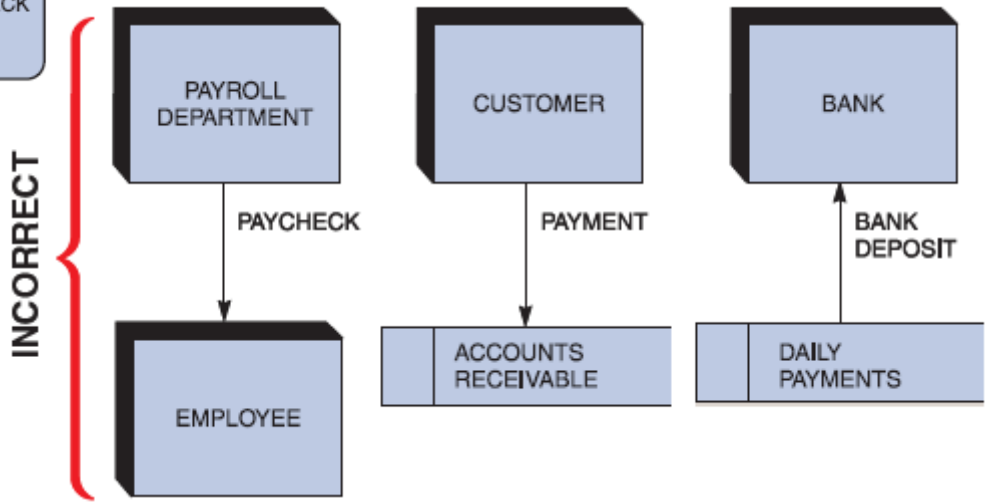


FIGURE 5-8 Examples of incorrect uses of external entities. An external entity must be connected by a data flow to a process, and not directly to a data store or to another external entity

Data Flow Diagrams (Cont. 9)

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

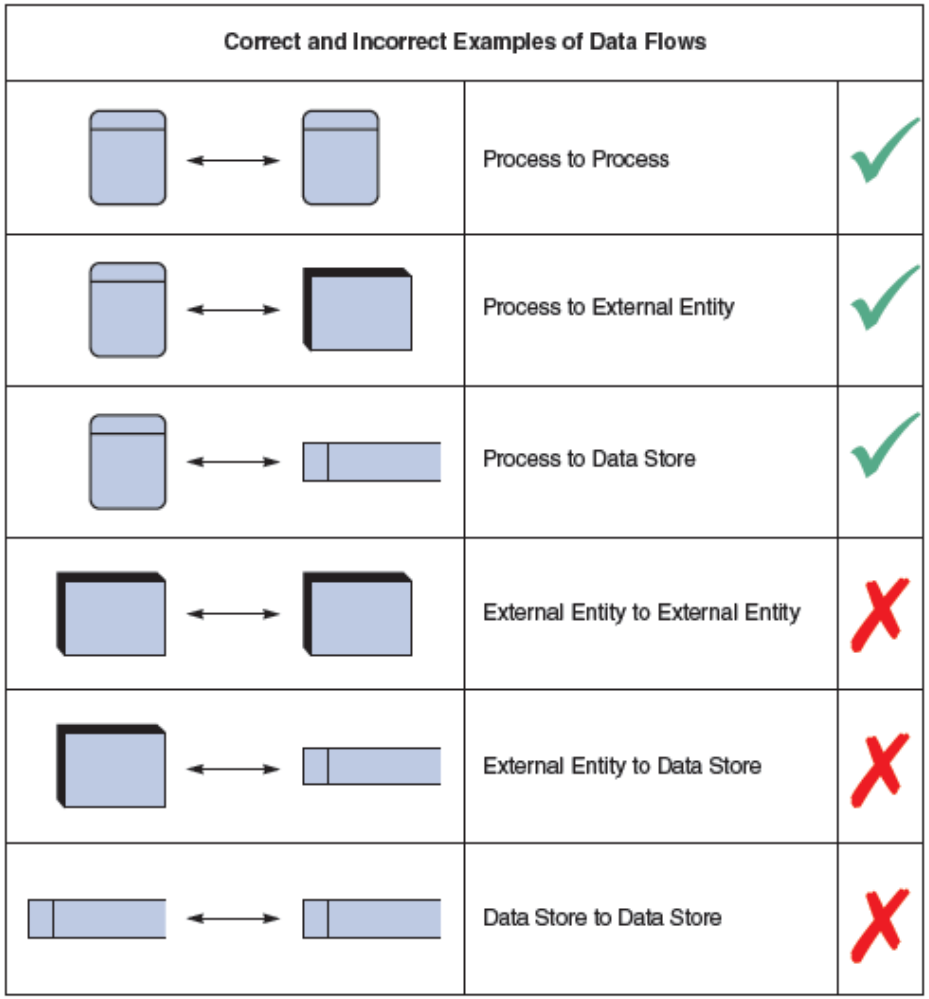


FIGURE 5-9 Examples of correct and incorrect uses of data flows

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. 1)

- 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.2)

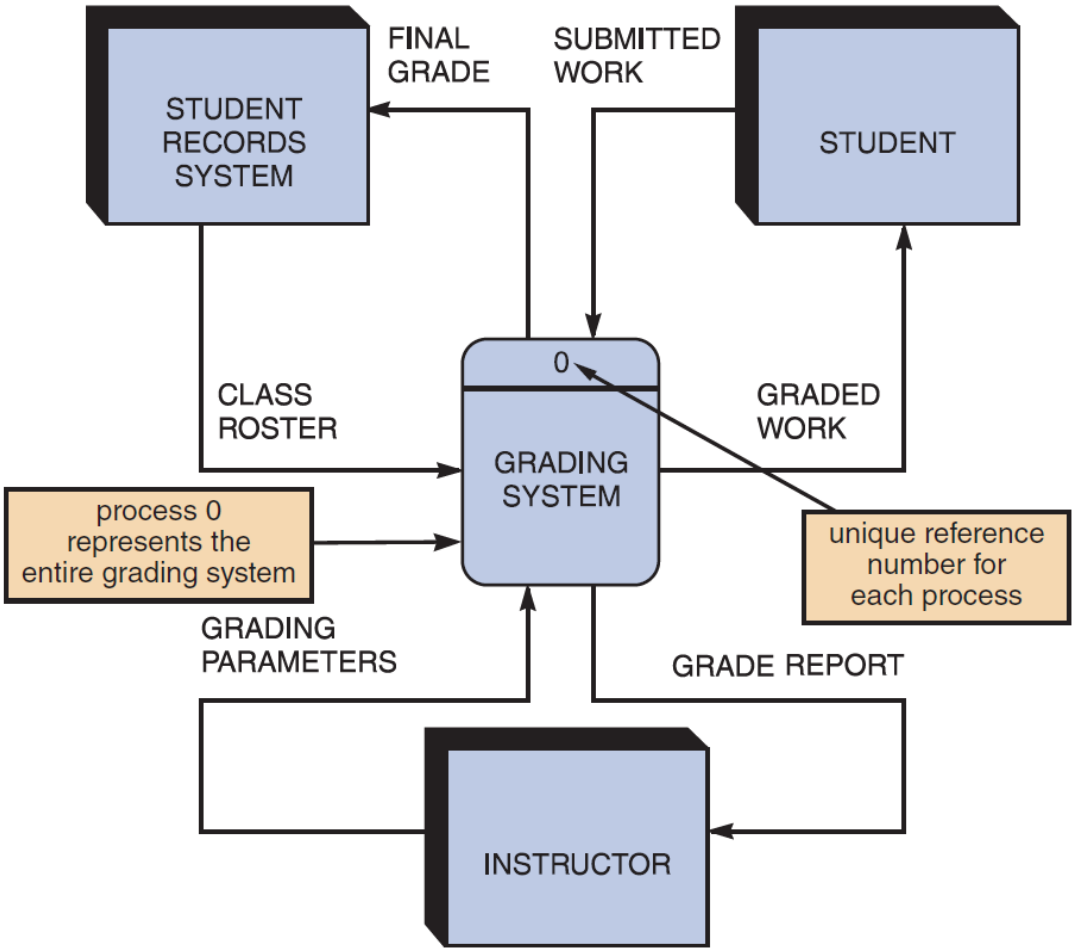


FIGURE 5-10 Context diagram DFD for grading system

Diagram 0 for Grading System

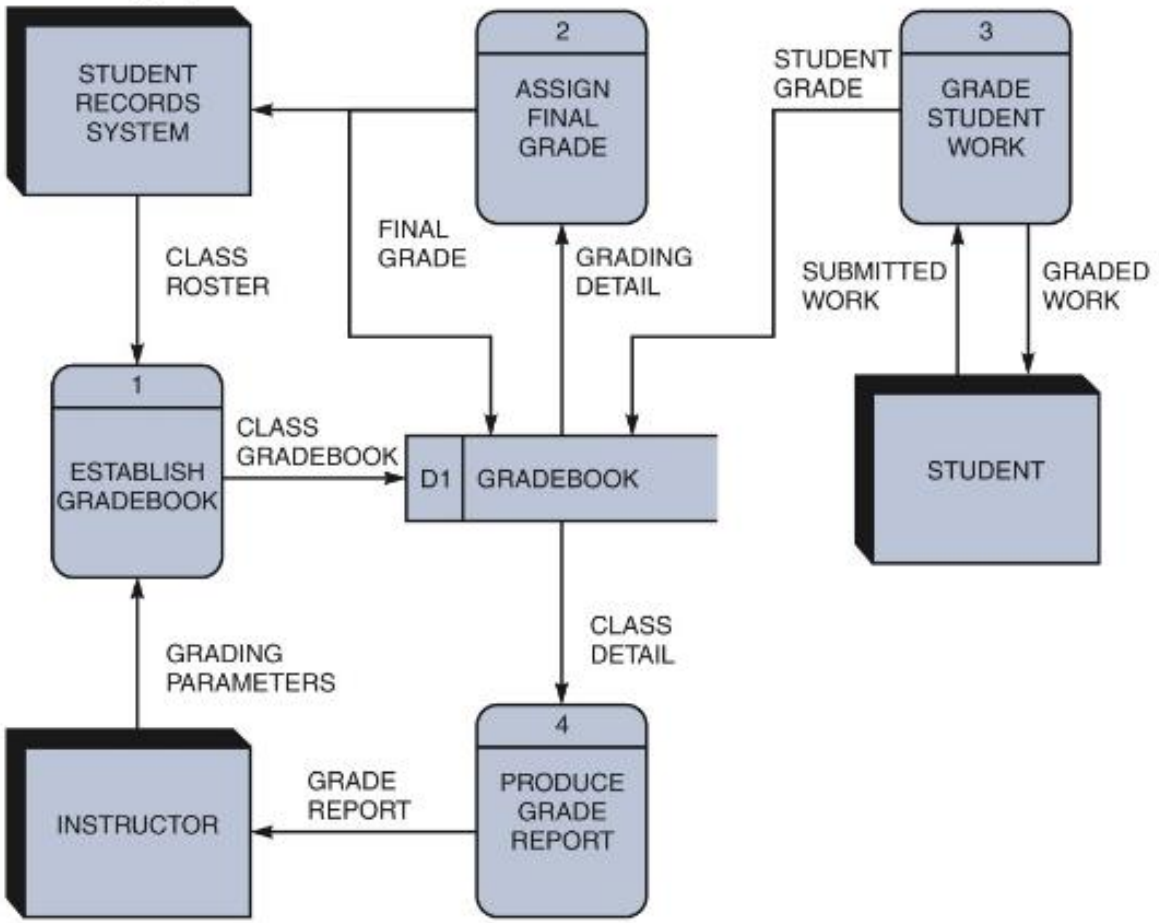


FIGURE 5-12 Context diagram and diagram 0 for the grading system.

Creating a Set of DFDs (Cont. 3)

- **Step 1: Draw a Context Diagram**

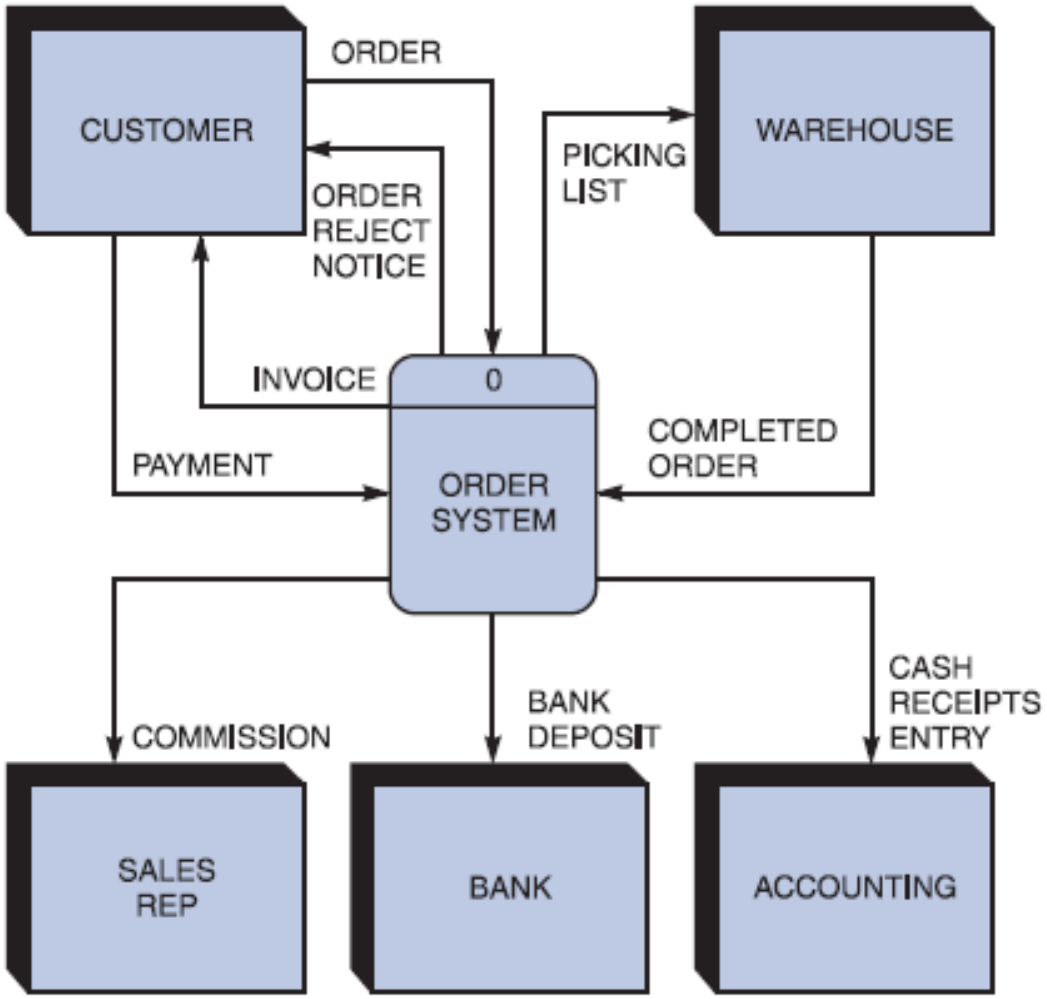


FIGURE 5-11 Context diagram DFD for an order system

Creating a Set of DFDs (Cont. 4)

- **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

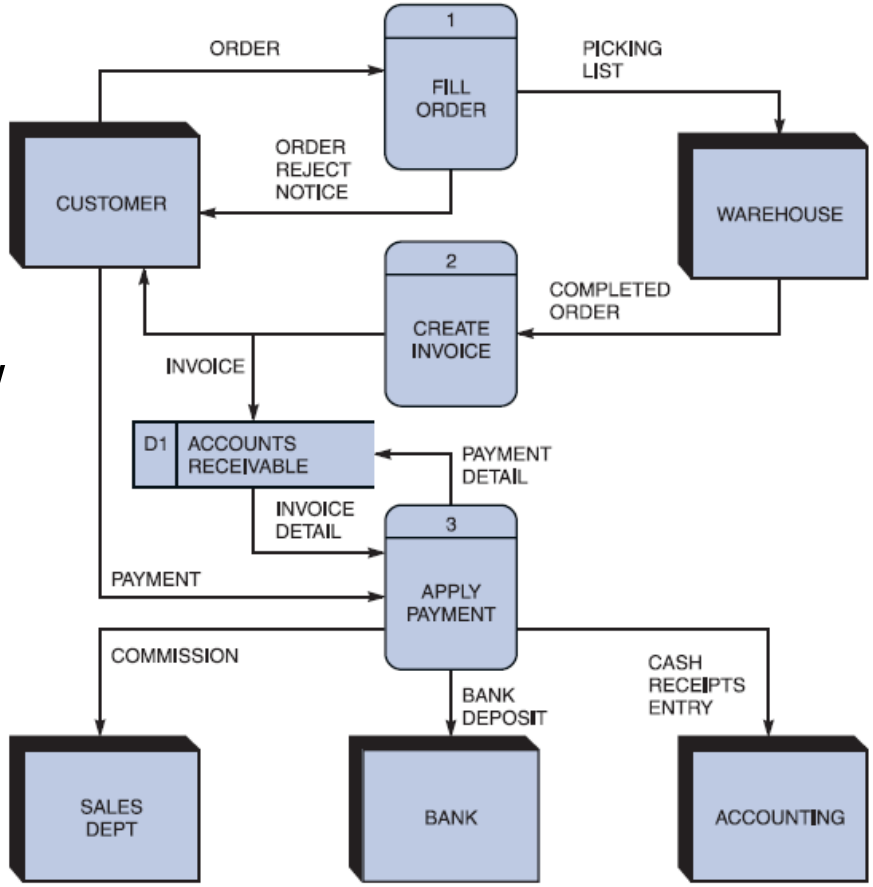


FIGURE 5-13 Diagram 0 DFD for the order system

Creating a Set of DFDs (Cont. 5)

- **Step 3: Draw the Lower Level Diagrams**

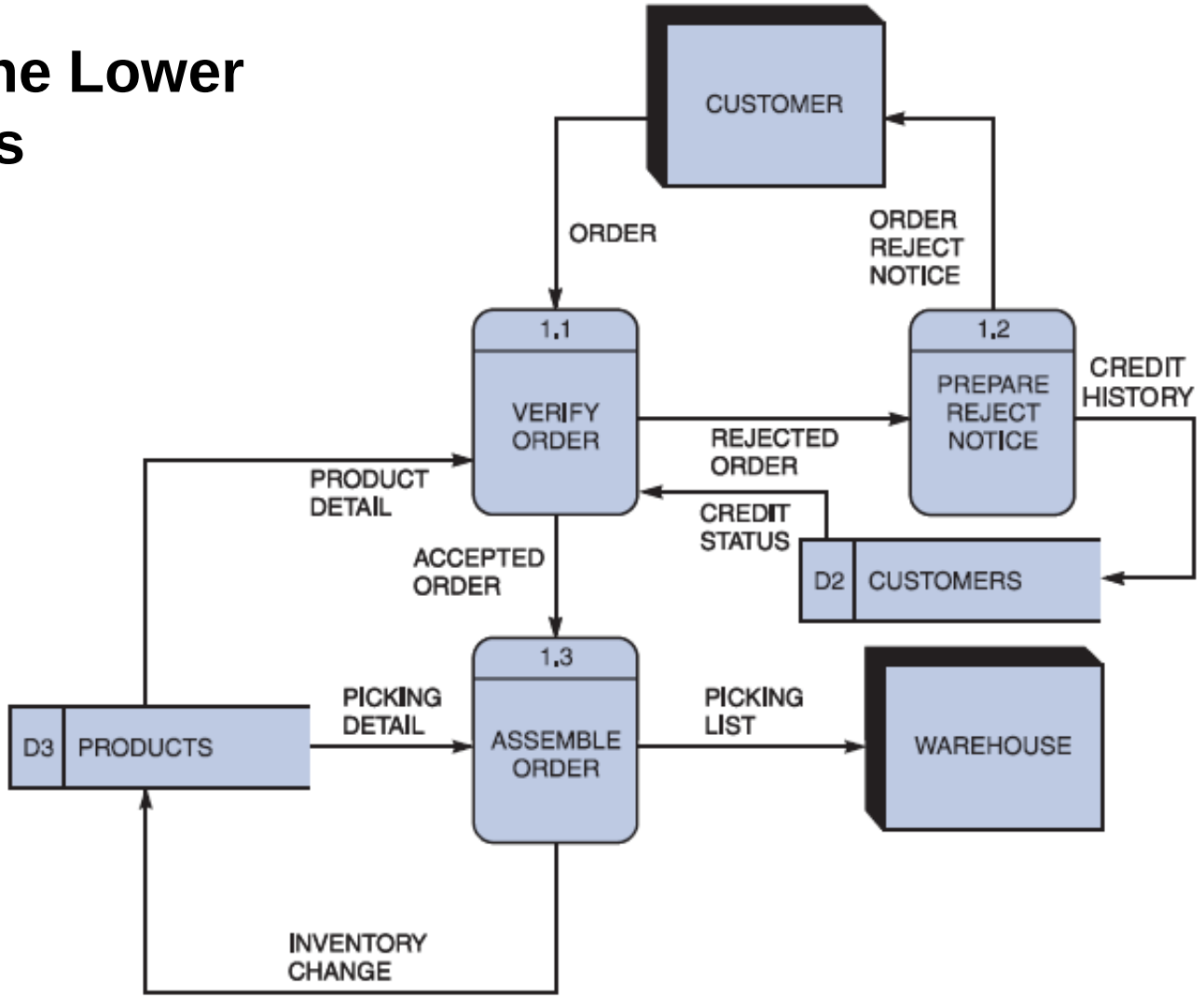


FIGURE 5-14 Diagram 1 DFD shows details of the FILLORDER process in the order system

Creating a Set of DFDs (Cont. 6)

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

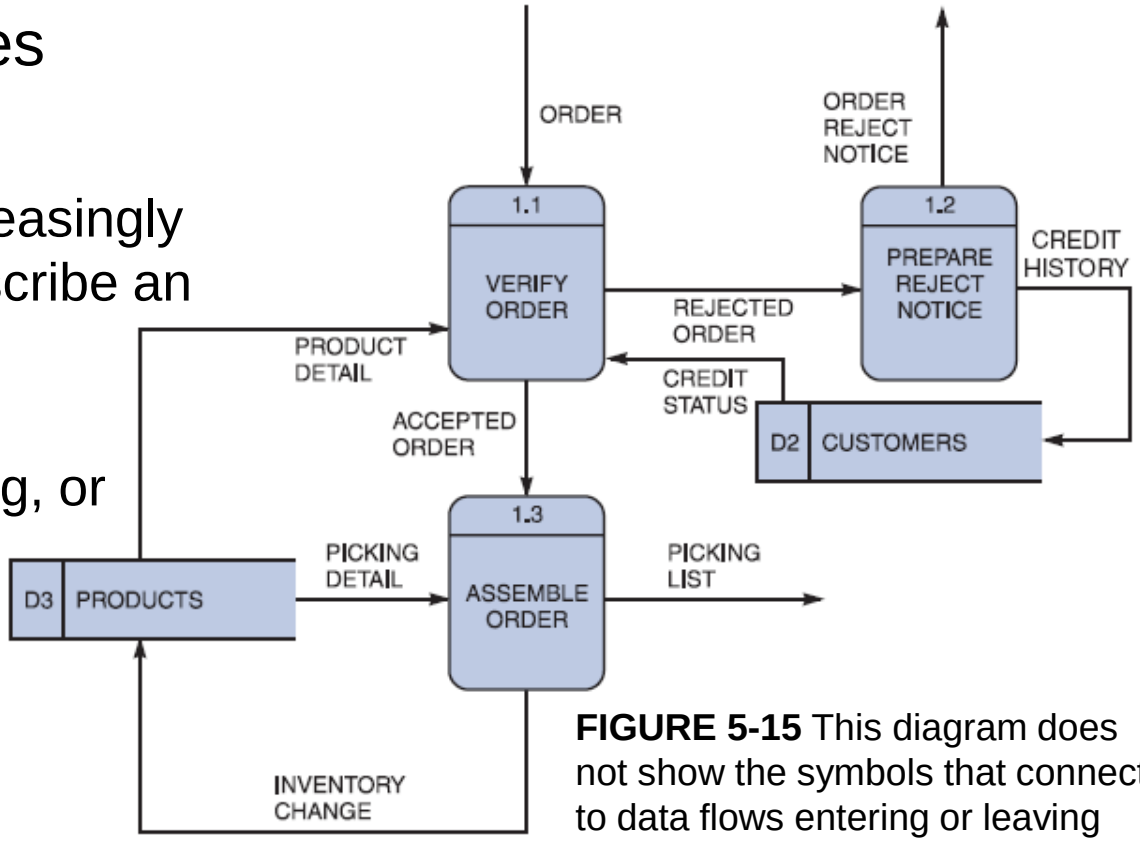


FIGURE 5-15 This diagram does not show the symbols that connect to data flows entering or leaving FILL ORDER on the context diagram

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Creating a Set of DFDs (Cont. 7)

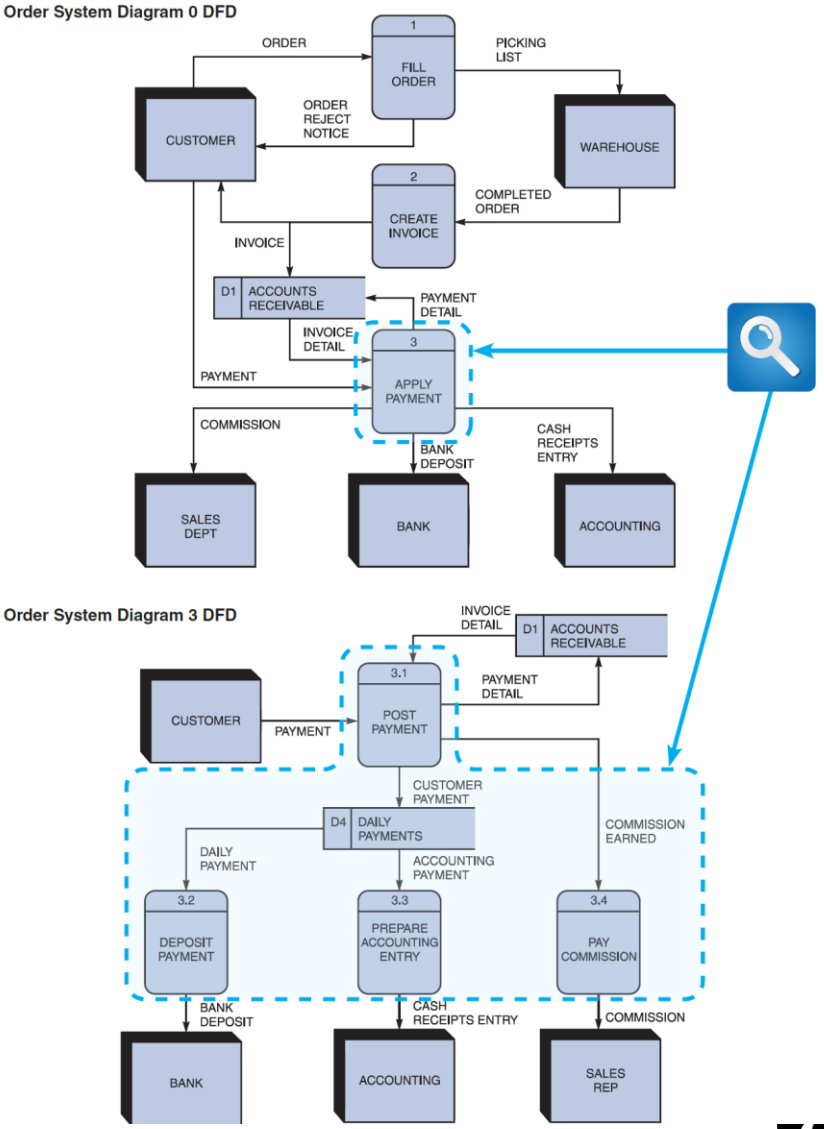


FIGURE 5-16 The order system diagram 0 is shown at the top of the figure, and exploded diagram 3 DFD (for the APPLY PAYMENT process) is shown at the bottom. The two DFDs are balanced because the child diagram at the bottom has the same input and output flows as the parent process 3 shown at the top

Creating a Set of DFDs (Cont. 8)

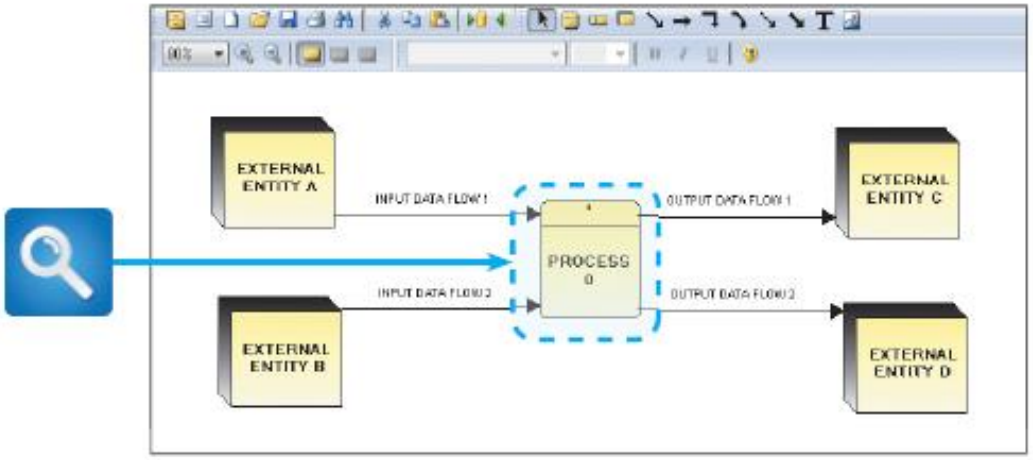


FIGURE 5-17 Example of a parent DFD diagram, showing process 0 as a black box

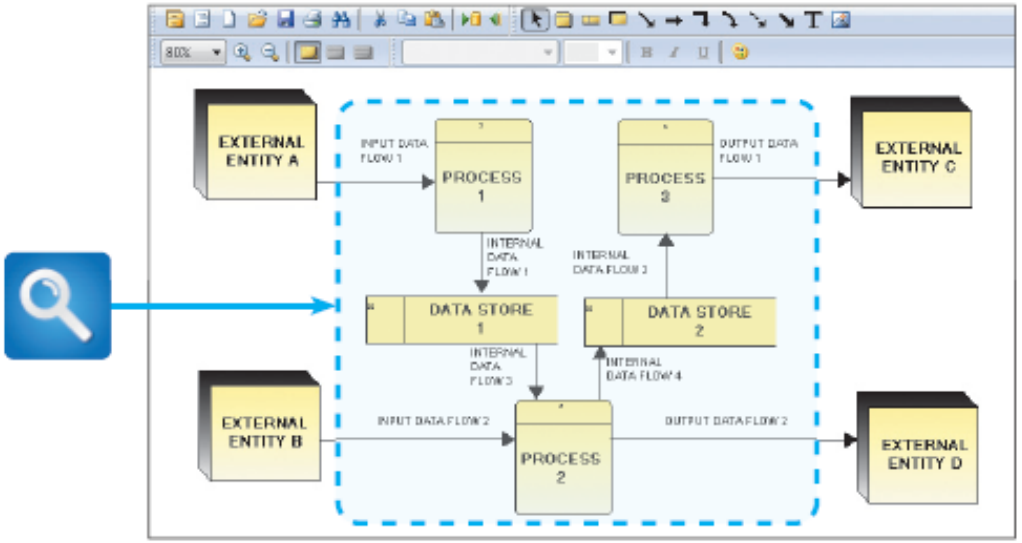


FIGURE 5-18 In the next level of detail, the process 0 black box reveals three processes, two data stores, and four internal data flows — all of which are shown inside the dashed line

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

CASE IN POINT 5.1: BIG TEN UNIVERSITY

You are the IT director at Big Ten University. As part of a training program, you decide to draw a DFD that includes some obvious mistakes to see whether your newly hired junior analysts can find them. You came up with the diagram 0 DFD shown in Figure 5-19. Based on the rules explained in this chapter, how many problems should the analysts find?

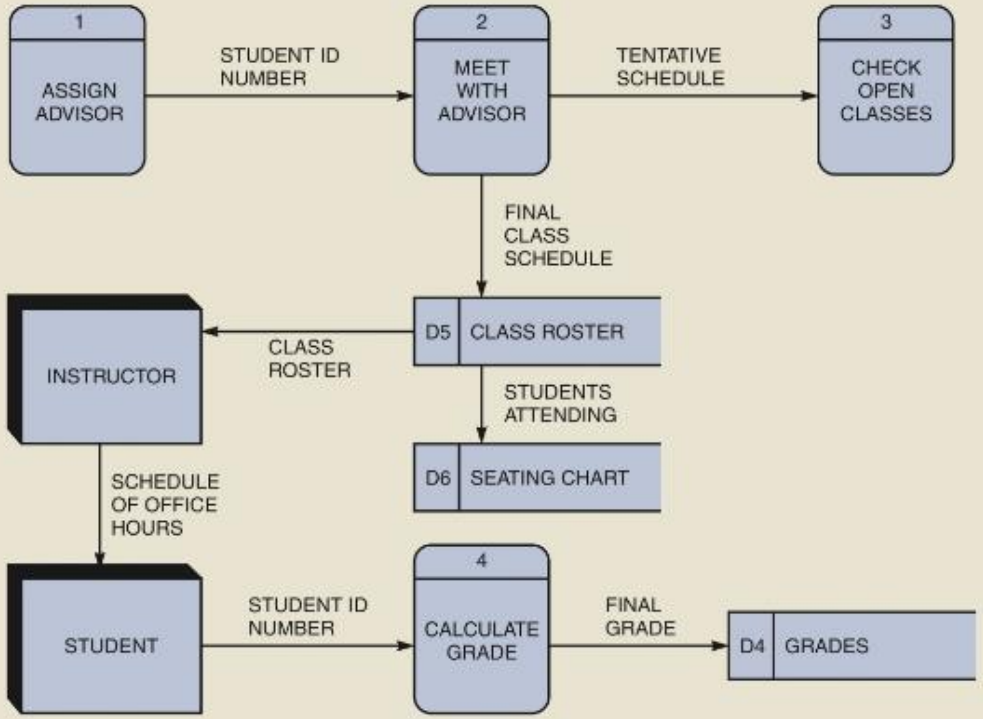


FIGURE 5-19 What are the mistakes in this diagram 0 DFD?

Data Dictionary

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

Data Dictionary (Cont. 1)

- **Data element**: Smallest piece of data that has meaning within an information system
  - Called **data item** or **field**
  - Are combined into records, also called data structures
- **Record**: Meaningful combination of related data elements that is included in a data flow or retained in a data store
  - Called **data structures**

Data Dictionary (Cont. 2)

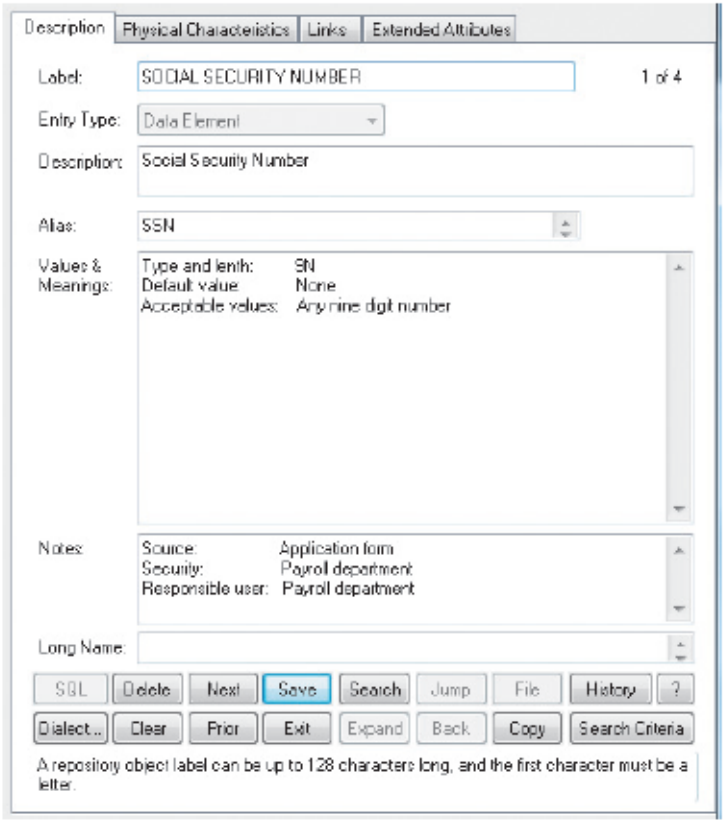
- **Using CASE Tools for Documentation**
  - More complex the system, more difficult it is to maintain full and accurate documentation
  - Modern CASE tools simplify the task
  - A CASE repository ensures data consistency

Data Dictionary (Cont. 3)

- **Documenting the Data Elements**
  - Every data element in the data dictionary should be documented
  - Objective - To provide clear, comprehensive information about the data and processes that make up a system

Data Dictionary (Cont. 4)

- **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



Source: Visible Systems Corporation.

FIGURE 5-20 A Visible Analyst screen describes the data element named SOCIAL SECURITY NUMBER.

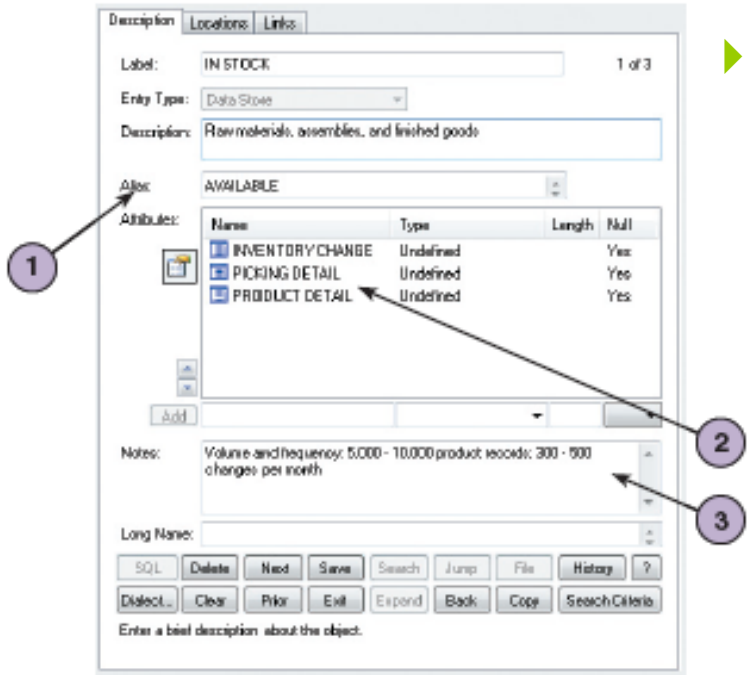
THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Data Dictionary (Cont. 5)

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

Data Dictionary (Cont. 6)

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.



Source: Visible Systems Corporation.

FIGURE 5-21 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

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Data Dictionary (Cont. 7)

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

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.

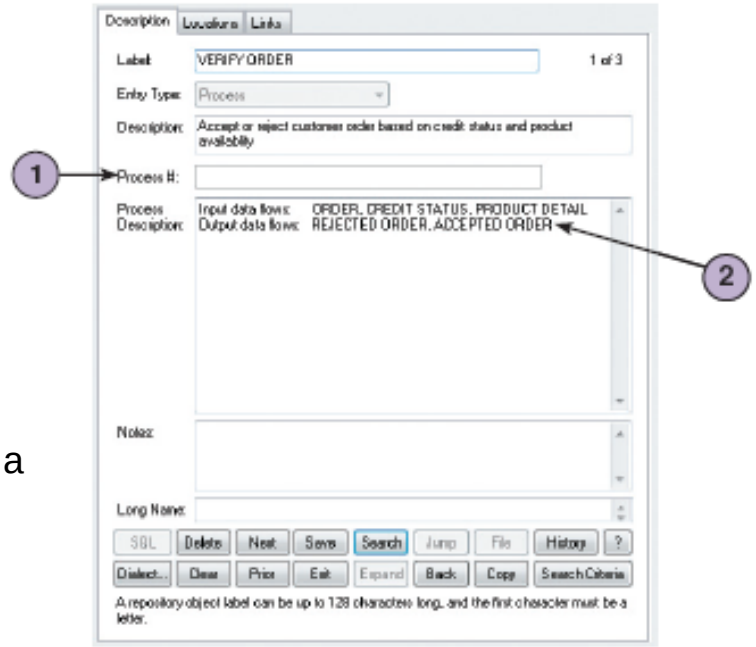


FIGURE 5-22 Visible Analyst screen that describes a process named VERIFY ORDER

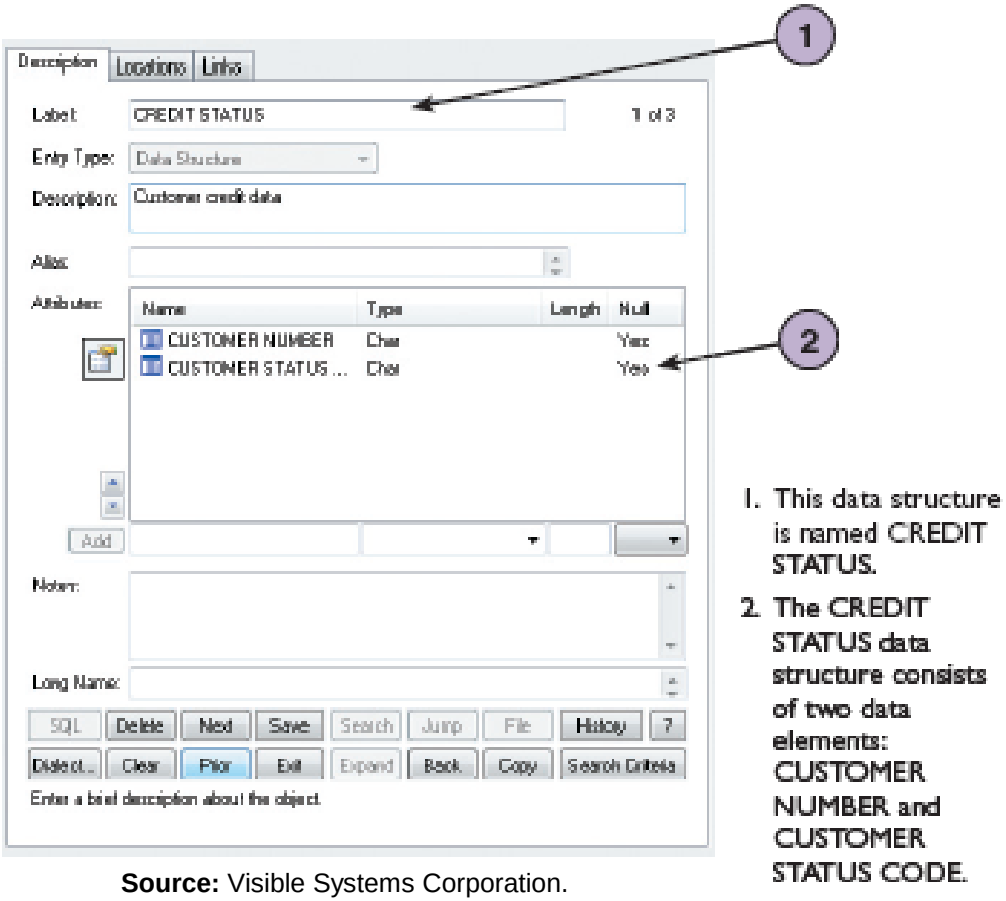
Source: Visible Systems Corporation.

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Data Dictionary (Cont. 8)

- **Documenting the Entities -** Data dictionary describes all external entities that interact with the system
  - Characteristics include
    - Entity name
    - Description
    - Alternate name(s)
    - Input data flows
    - Output data flows

Data Dictionary (Cont. 9)



Source: Visible Systems Corporation.

FIGURE 5-23 Visible Analyst screen that documents a record, or data structure named CREDIT STATUS

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

Data Dictionary (Cont. 10)

- **Data Dictionary Reports** - Following can be obtained
  - Alphabetized list of all data elements by name
  - Report describing each data element and indicating the user or department that is responsible for data entry, updating, or deletion
  - 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

- **Process description**: Documents the details of a functional primitive and represents a specific set of processing steps and business logic
- Tools - structured English, decision tables, and decision trees
- Used in object-oriented development
  - O-O analysis - combines data and the processes that act on the data into things called objects, and similar objects can be grouped together into classes
  - O-O processes are called methods

Process Description Tools (Cont. 1)

- **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 5-24 Sequence structure

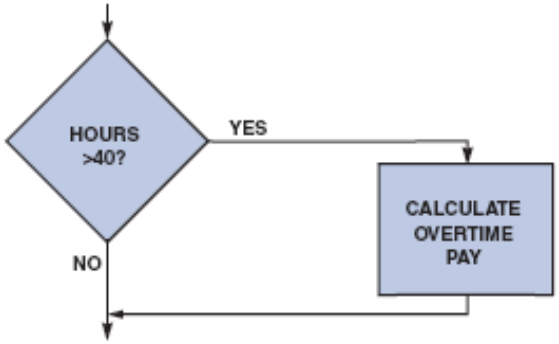


FIGURE 5-25 Selection structure

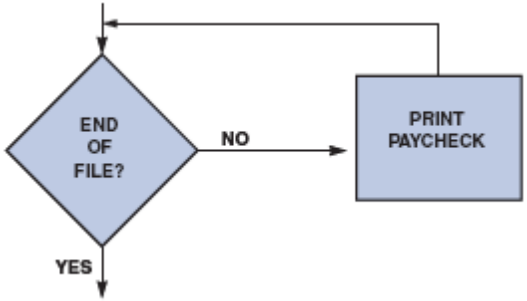
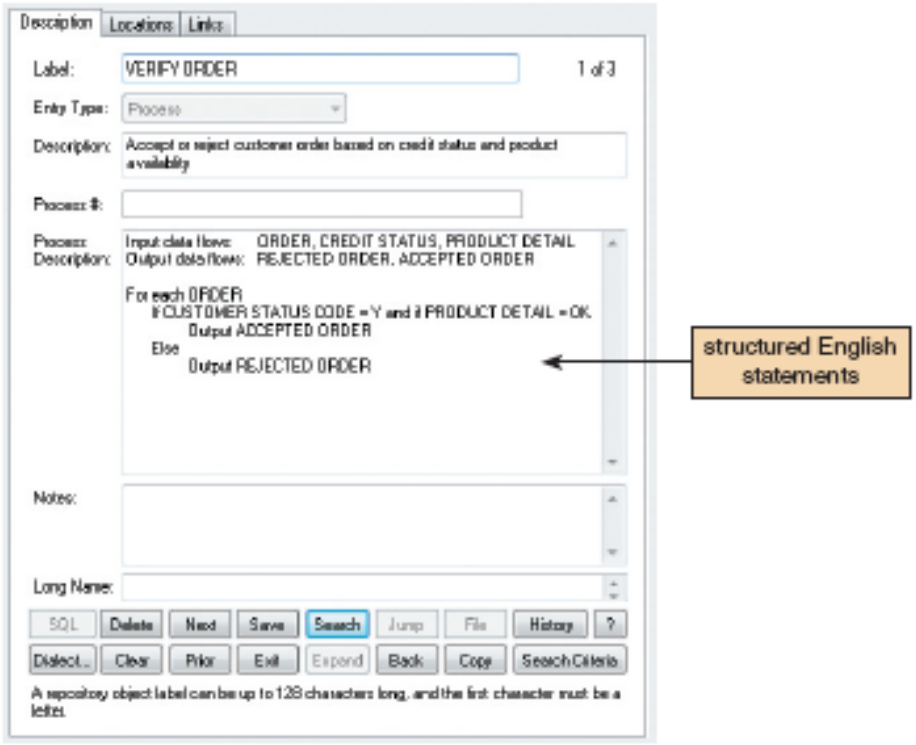


FIGURE 5-26 Iteration structure

Process Description Tools (Cont. 2)

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



Source: Visible Systems Corporation.

FIGURE 5-27 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. 3)

- **Decision Tables**
  - Show a logical structure, with all possible combinations of conditions and resulting actions
    - Every possible outcome should be considered to ensure that nothing has been overlooked
  - Number of rules doubles each time a condition is added
  - Can have more than two possible outcomes
  - Are the best way to describe a complex set of conditions

Process Description Tools (Cont. 4)

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 5-28 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	3
Product is in stock	Y	N	Y	N	
Accept order	X				4
Reject order		X	X	X	

FIGURE 5-29 Example of a simple decision table showing the processing logic of the VERIFY ORDER process

Process Description Tools (Cont. 5)

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 5-30 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

FIGURE 5-31 This table is based on the Verify Order conditions shown in Figure 5-30. With three conditions, there are eight possible combinations, or rules

Process Description Tools (Cont. 6)

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 5-32 In the first table, dashes have been added to indicate that a condition is not relevant. In the second version, rules have been combined. Notice that in final version, only four rules remain. These rules document the logic, and will be transformed into program code when the system is developed

Process Description Tools (Cont. 7)

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 5-33 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

FIGURE 5-34 This decision table is based on the sales promotion policy in Figure 5-33. This is the initial version of the table, before simplification

Process Description Tools (Cont. 8)

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 5-35 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

CASE IN POINT 5.2: ROCK SOLID OUTFITTERS (PART 1)

Leah Jones is the IT manager at Rock Solid Outfitters, a medium-sized supplier of outdoor climbing and camping gear. Steve Allen, the marketing director, has asked Leah to develop a special web-based promotion. As Steve described it to Leah, Rock Solid will provide free shipping for any customer who either completes an online survey form or signs up for the Rock Solid online newsletter. Additionally, if a customer completes the survey and signs up for the newsletter, Rock Solid will provide a $10 merchandise credit for orders of $100 or more. Leah has asked you to develop a decision table that will reflect the promotional rules that a programmer will use. She wants you to show all possibilities, and then to simplify the results to eliminate any combinations that would be unrealistic or redundant.

Process Description Tools (Cont. 9)

- **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
  - Provide the same results as decision tables, but in different forms

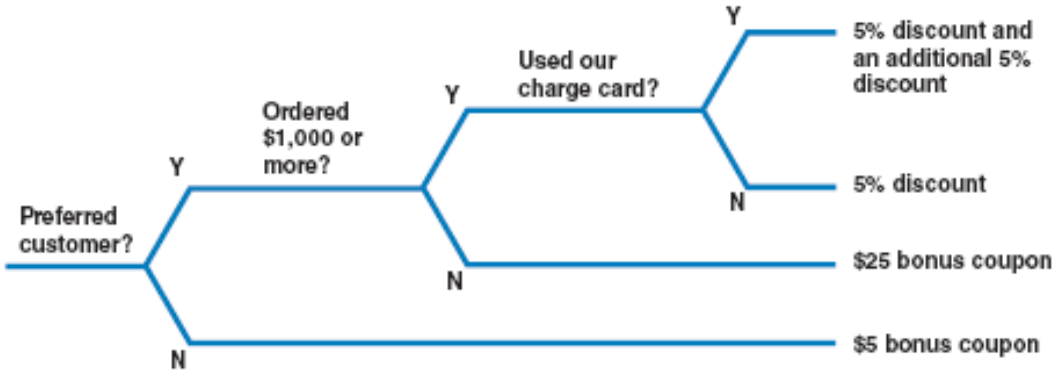


FIGURE 5-36 This example is based on the same Sales Promotion Policy shown in the decision tables in Figures 5-34 and 5-35 on the previous page. Like a decision table, a decision tree shows all combinations of conditions and outcomes. The main difference is the graphical format, which many viewers find easier to interpret

CASE IN POINT 5.3: ROCK SOLID OUTFITTERS (PART 2)

Leah Jones, the IT manager at Rock Solid Outfitters, thinks you did a good job on the decision table task she assigned to you. Now she wants you to use the same data to develop a decision tree that will show all the possibilities for the web-based promotion described in Part 1 of the case. She also wants you to discuss the pros and cons of decision tables versus decision trees.

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. 1)

- **Sequence of Models**
  - 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 extra step allows to understand the current system better

Logical versus Physical Models (Cont. 2)

- **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
  - Disadvantage - Additional time and cost

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

Summary (Cont. 1)

- 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

Summary (Cont. 2)

- 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

Summary

- We discussed data and process modelling techniques that analysts use to show the system transforms data into useful information. The deliverable (or end product) of data and process modelling is a logical model that will support business operations and meet user needs.