System Analysis and Design

CSE 307
Instructor: Sabrina Alam

Lecture 09

Process Specifications and Structured Decisions

Learning Objectives

- Understand the purpose of process specifications.
- Recognize the difference between structured and semistructured decisions.
- Use structured English, decision tables, and decision trees to analyze, describe, and document structured decisions.
- Choose an appropriate decision analysis method for analyzing structured decisions and creating process specifications.

Logic of Decisions

- Documenting and analyzing logic:
  - Structured English
  - Decision tables
  - Decision trees
- Logic and structured decisions are distinguishable from semistructured decisions
- Structured decision analysis methods promote completeness, accuracy, and communication

Major Topics

- Process specifications
- Business rules
- Structured English
- Decision tables
- Decision trees
- Horizontal balancing

Process Specifications

- Sometimes called minispecs
- Created for primitive processes as well as for some higher level processes on a data flow diagram
- Created for class methods in object-oriented design and for the steps in a use case

Goals of Producing Process Specifications

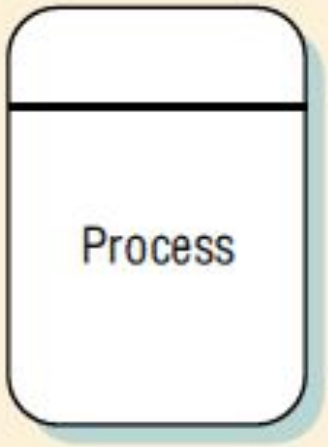
- Reduce process ambiguity
- Obtain a precise description of what is accomplished
- Validate the system design

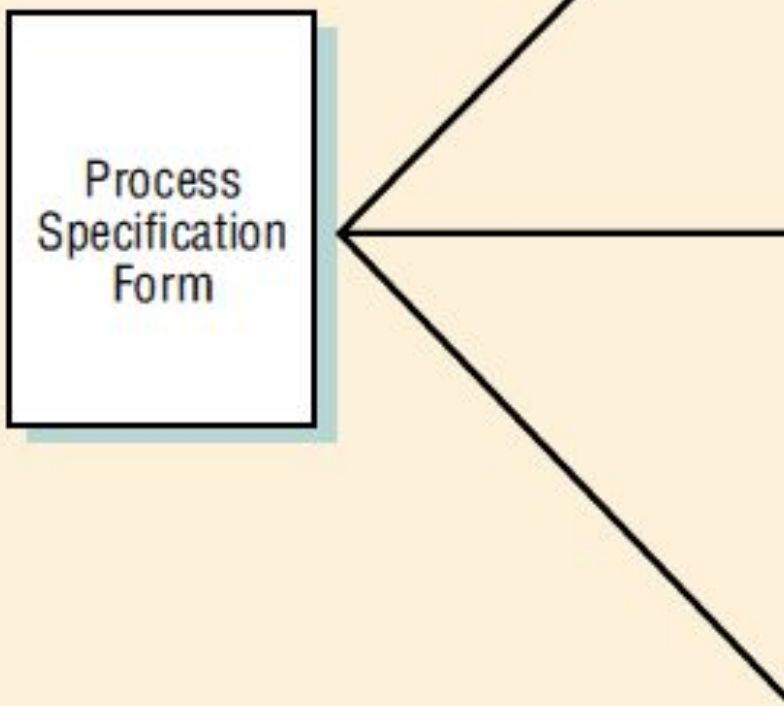
Goals of Producing Process Specifications

- Reduce process ambiguity—compels the analyst to learn details about how the process works.
- Obtain a precise description of what is accomplished—usually included in a packet of specifications for the programmer.
- Validate the system design—ensures that a process has all the input data flow necessary for producing the output.

Process Specifications Are Not Created for

- Processes that represent physical input and/or output
- Processes that represent simple data validation
- Processes that use prewritten code


Process


Process
Specification
Form

Process Specification Format Information

- The process number
- The process name
- Description of what the process accomplishes
- A list of input data flow
- Output data flows
- Type of process
- Uses prewritten code
- Process logic description
- Logic method reference
- List any unresolved issues

The Process Number

- Must match the process ID on the data flow diagram
- Allows the analyst to work on or review any process, and to locate the data flow diagram containing the process easily

The Process Name

- The same as displays within the process symbol on the DFD

Description of What the Process Accomplishes

- Example:

  Determine if an item is available for sale. If it is not available, create a backordered item record. Determine the quantity available.

List of Input Data Flow

- Uses the names found on the data flow diagram
- Data names used in the formula or logic should match the data dictionary, for consistency and good communication

Output Data Flows

- Uses data flow diagram and data dictionary names

Type of Process

- Batch
- Online
  - Require screen design or web pages
- Manual
  - Should have well-defined procedures for employees performing the process tasks

Uses Prewritten Code

- Include the name of the subprogram or function containing the code

Process Logic Description

- This should state policy and business rules, not computer language pseudo-code
- Business rules are the procedures that allow a corporation to run its business

Common Business Rule Formats

- Definitions of business terms
- Business conditions and actions
- Data integrity constraints
- Mathematical and functional derivations
- Logical inferences
- Processing sequences
- Relationships among facts about the business

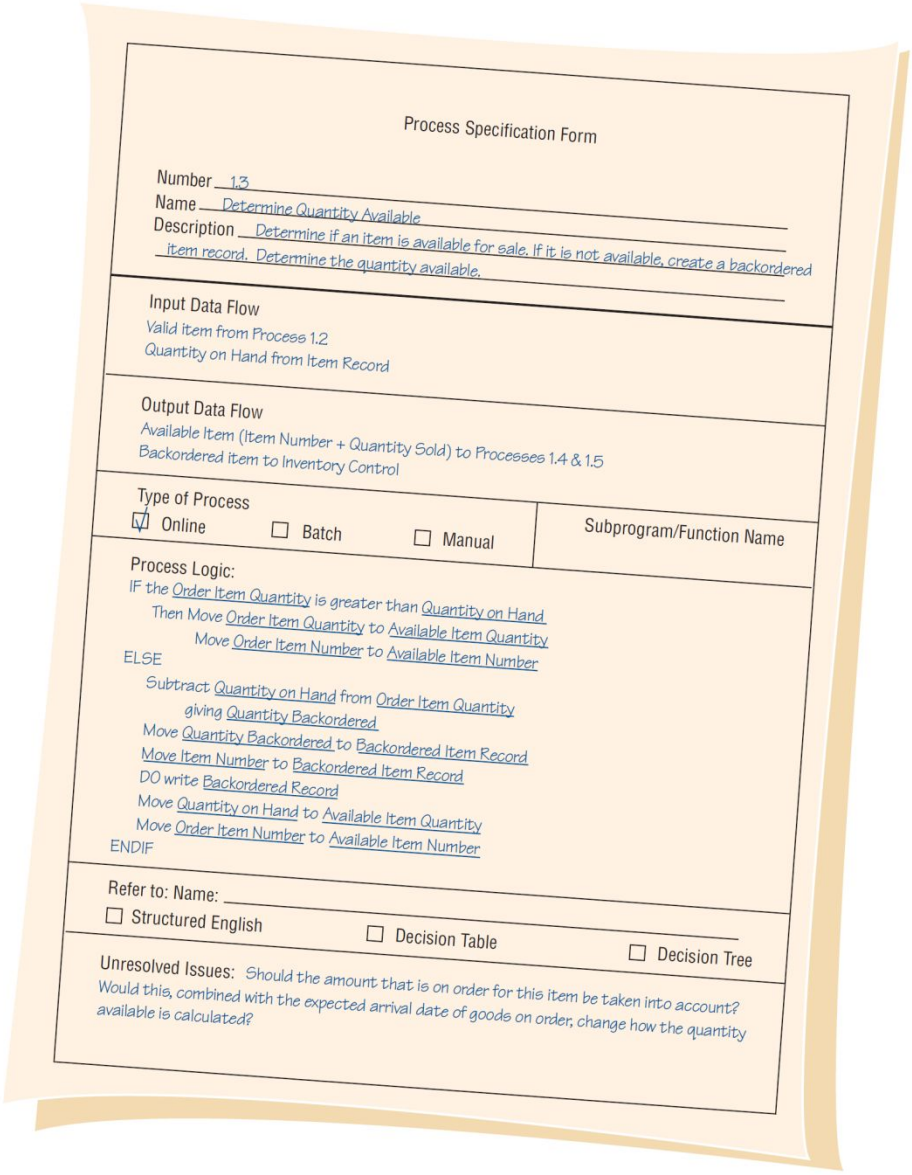
Logic Method Reference

- If there is not enough room for a complete structured English description include a reference to the structured English description, decision table, or tree depicting the logic

List Any Unresolved Issues

- Incomplete portions of logic
- These issues form the basis of the questions used for follow-up interviews with users or business experts you have added to your project team

An Example of a Completed Process Specification Form (Figure 9.2)



Process Specification Form

Number 1.3
Name Determine Quantity Available
Description Determine if an item is available for sale. If it is not available, create a backordered item record. Determine the quantity available.

Input Data Flow
Valid item from Process 1.2
Quantity on Hand from Item Record

Output Data Flow
Available Item (Item Number + Quantity Sold) to Processes 1.4 & 1.5
Backordered item to Inventory Control

Type of Process
☑ Online ☐ Batch ☐ Manual

Subprogram/Function Name

Process Logic:
IF the Order Item Quantity is greater than Quantity on Hand
Then Move Order Item Quantity to Available Item Quantity
Move Order Item Number to Available Item Number
ELSE
Subtract Quantity on Hand from Order Item Quantity
giving Quantity Backordered
Move Quantity Backordered to Backordered Item Record
Move Item Number to Backordered Item Record
DO write Backordered Record
Move Quantity on Hand to Available Item Quantity
Move Order Item Number to Available Item Number
ENDIF

Refer to: Name:
☐ Structured English ☐ Decision Table ☐ Decision Tree

Unresolved Issues: Should the amount that is on order for this item be taken into account? Would this, combined with the expected arrival date of goods on order, change how the quantity available is calculated?

Structured English

- Used when the process logic involves formulas or iteration, or when structured decisions are not complex
- Based on structured logic and simple English statements such as add, multiply, and move

Writing Structured English

- Express all logic in terms of sequential structures, decision structures, case structures, or iterations
- Use and capitalize accepted keywords such as IF, THEN, ELSE, DO, and PERFORM
- Indent blocks of statements to show their hierarchy (nesting) clearly
- Underline words or phrases that have been defined in a data dictionary
- Clarify the logical statements

Structured English Type	Example
Sequential Structure A block of instructions in which no branching occurs	Action #1 Action #2 Action #3
Decision Structure Only IF a condition is true, complete the following statements; otherwise, jump to the ELSE	IF Condition A is True THEN implement Action A ELSE implement Action B ENDIF
Case Structure A special type of decision structure in which the cases are mutually exclusive (if one occurs, the others cannot)	IF Case #1 implement Action #1 ELSE IF Case #2 Implement Action #2 ELSE IF Case #3 Implement Action #3 ELSE IF Case #4 Implement Action #4 ELSE print error ENDIF

Advantages of Structured English

- Clarifying the logic and relationships found in human languages
- An effective communication tool, it can be taught to and understood by users in the organization

Data Dictionary and Process Specification

- The data dictionary is a starting point for creating structured English:
  - Sequence—a simple sequence of statements MOVE, ADD, and SUBTRACT
  - Selection—[] entries become IF…THEN...ELSE statements
  - Iteration { } entries become DO WHILE, DO UNTIL, or PERFORM UNTIL

Decision Tables

- A table of rows and columns, separated into four quadrants:
  - Conditions
  - Condition alternatives
  - Actions to be taken
  - Rules for executing the actions

Standard Format Used for Presenting a Decision Table (Figure 9.7)

Conditions and Actions	Rules
Conditions	Condition Alternatives
Actions	Action Entries

Customer Checkout Decision Table (Figure 9.8)

Conditions and Actions	Rules 1	2	3	4
Under $50	Y	Y	N	N
Pays by check with two forms of ID	Y	N	Y	N
Uses credit card	N	Y	N	Y
Complete the sale after verifying signature.	X			
Complete the sale. No signature needed.		X		
Call supervisor for approval.			X	
Communicate electronically with bank for credit card authorization.				X

Constructing a Decision Table for Deciding Which Catalog to Send to Customers Who Order Only from Selected Catalogs (Figure 9.9)

Conditions and Actions	Rules 1	2	3	4	5	6	7	8
Customer ordered from Fall catalog.	Y	Y	Y	Y	N	N	N	N
Customer ordered from Christmas catalog.	Y	Y	N	N	Y	Y	N	N
Customer ordered from specialty catalog.	Y	N	Y	N	Y	N	Y	N
Send out this year's Christmas catalog.		X		X		X		X
Send out specialty catalog.			X				X	
Send out both catalogs.	X				X			

Developing Decision Tables

- Determine conditions that affect the decision
- Determine possible actions that can be taken
- Determine condition alternatives for each condition
- Calculate the maximum number of columns in the decision table

Developing Decision Tables

- Fill in the condition alternatives
- Complete table by inserting an X where rules suggest actions
- Combine rules where it is apparent
- Check for impossible situations
- Rearrange to make more understandable

Developing Decision Tables Step 1

- Determine the number of conditions that may affect the decision
- Combine rows that overlap, such as conditions that are mutually exclusive
- The number of conditions becomes the number of rows in the top half of the decision table

Developing Decision Tables Step 2

- Determine the number of possible actions that can be taken
- That number becomes the number of rows in the lower half of the decision table

Developing Decision Tables Step 3

- Determine the number of condition alternatives for each condition
- In the simplest form of decision table, there would be two alternatives (Y or N) for each condition
- An extended entry table may have many alternatives for each condition
- Make sure that all possible values for the condition are included

Developing Decision Tables Step 4

- Calculate the maximum number of columns in the decision table by multiplying the number of alternatives for each condition
- If there were four conditions and two alternatives (Y or N) for each of the conditions, there would be 16 possibilities

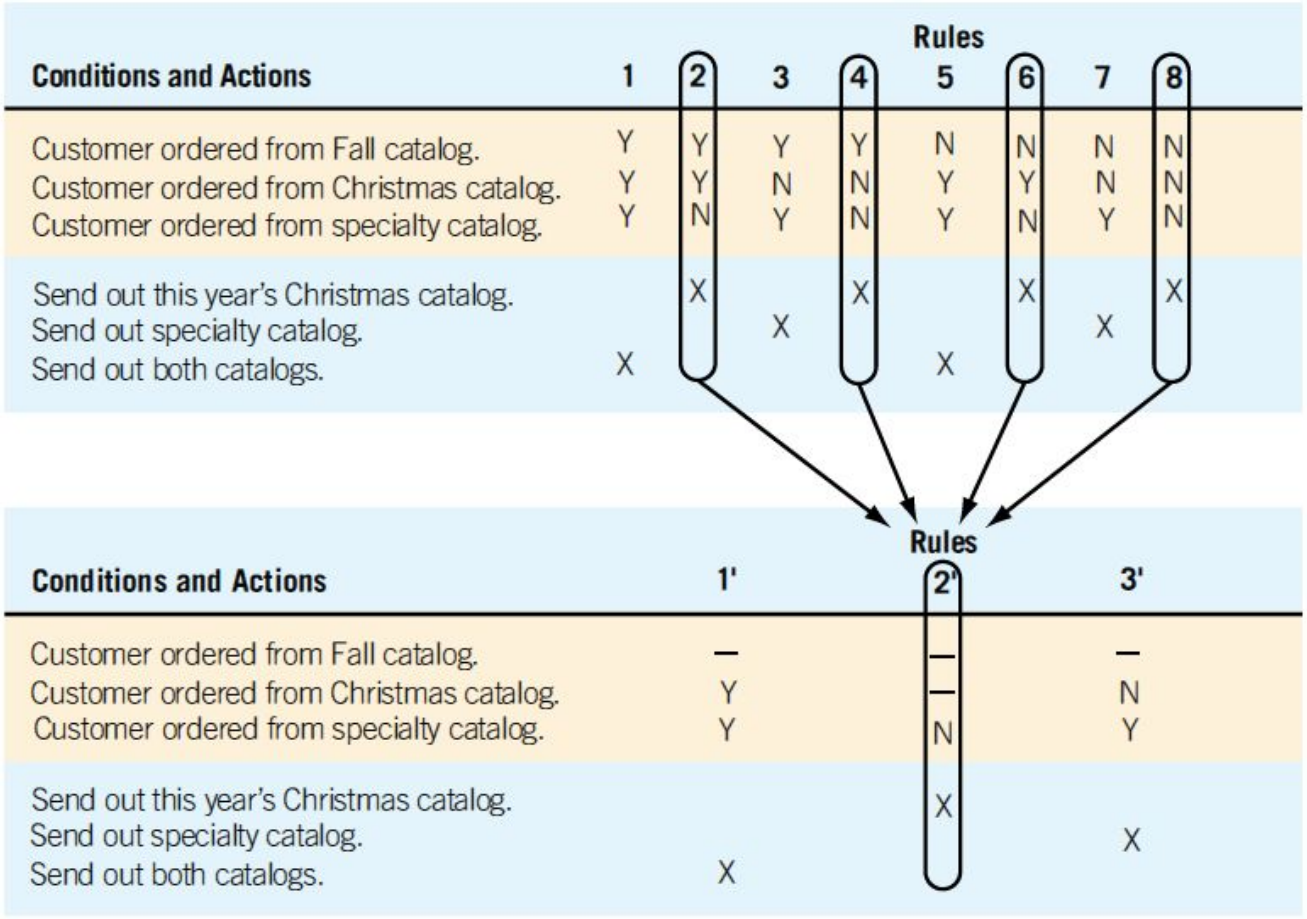
Developing Decision Tables Step 5

- Fill in the condition alternatives
- Start with the first condition and divide the number of columns by the number of alternatives for that condition
- If there are 16 columns and two alternatives (Y or N), then 16 divided by 2 is 8
- Choose one of the alternatives, say Y, and write it in the first eight columns
- Finish by writing N in the remaining eight columns

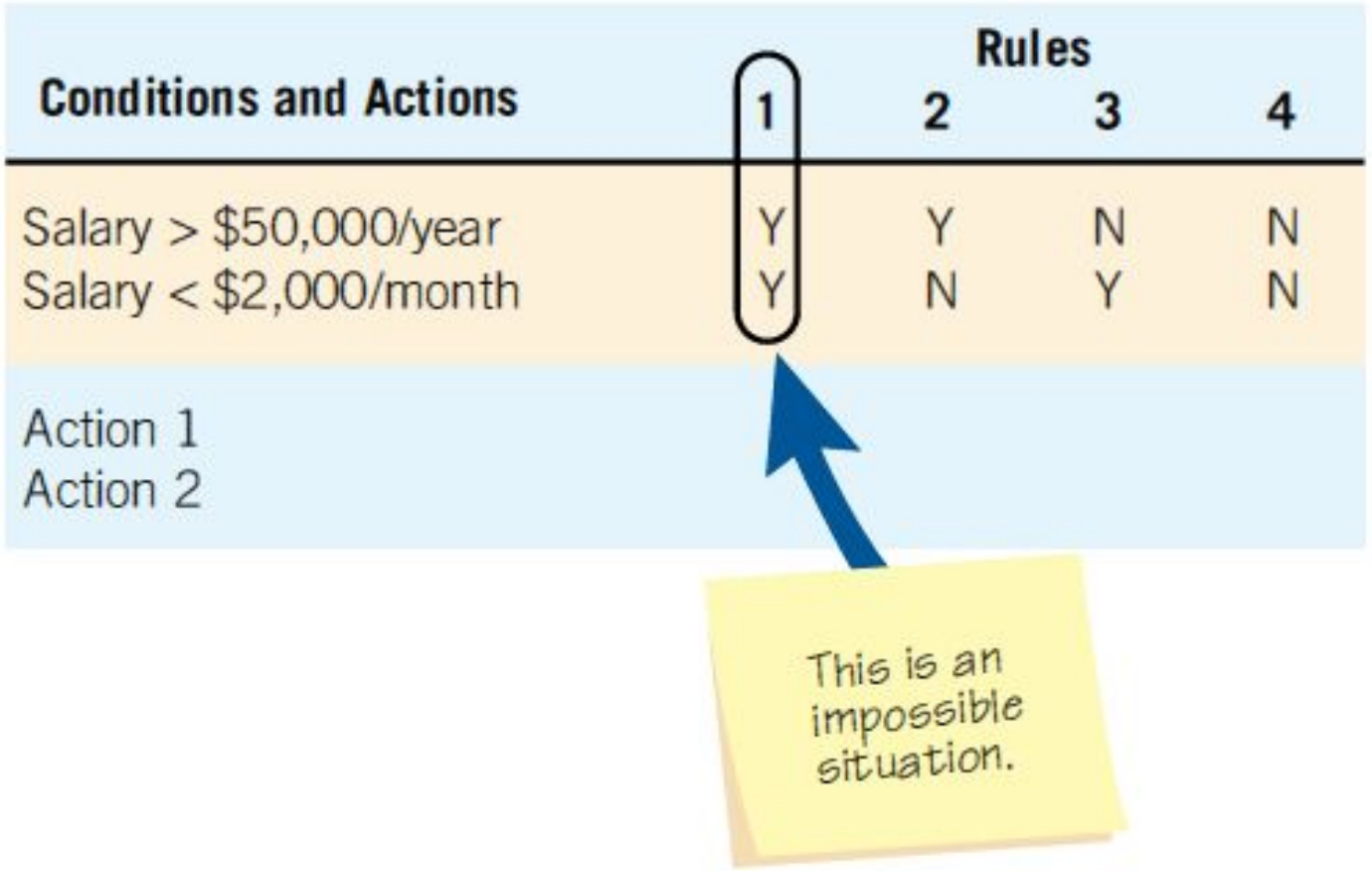
Checking for Completeness and Accuracy

- Four main problems:
  - Incompleteness
  - Impossible situations
  - Contradictions
  - Redundancy

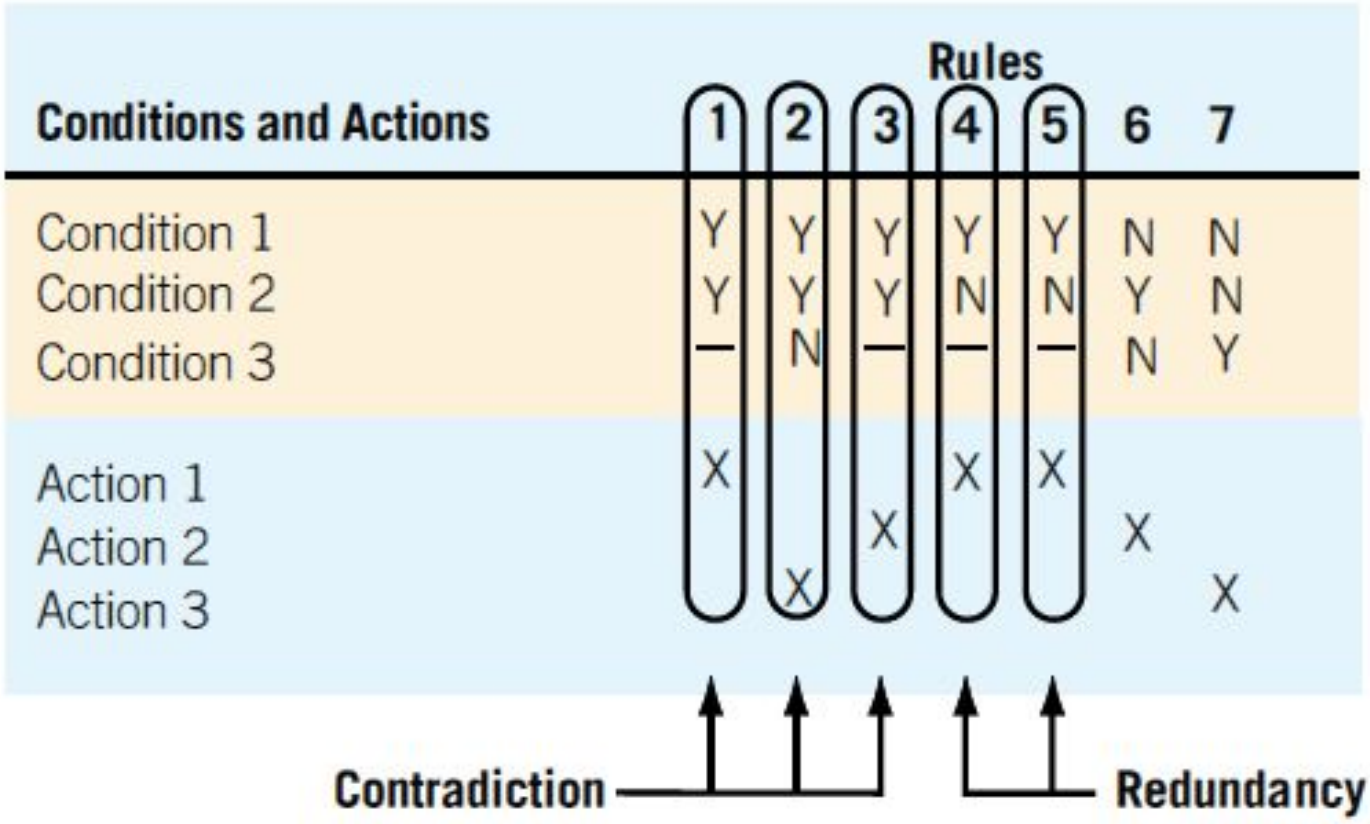
Combining Rules to Simplify the Decision Table (Figure 9.10)



Checking the Decision Table for Impossible Situations (Figure 9.12)



Checking for Contradictions and Redundancy (Figure 9.13)



Decision Table Advantages

- Help the analysis ensure completeness
- Easy to check for possible errors
  - Impossible situations
  - Contradictions
  - Redundancy

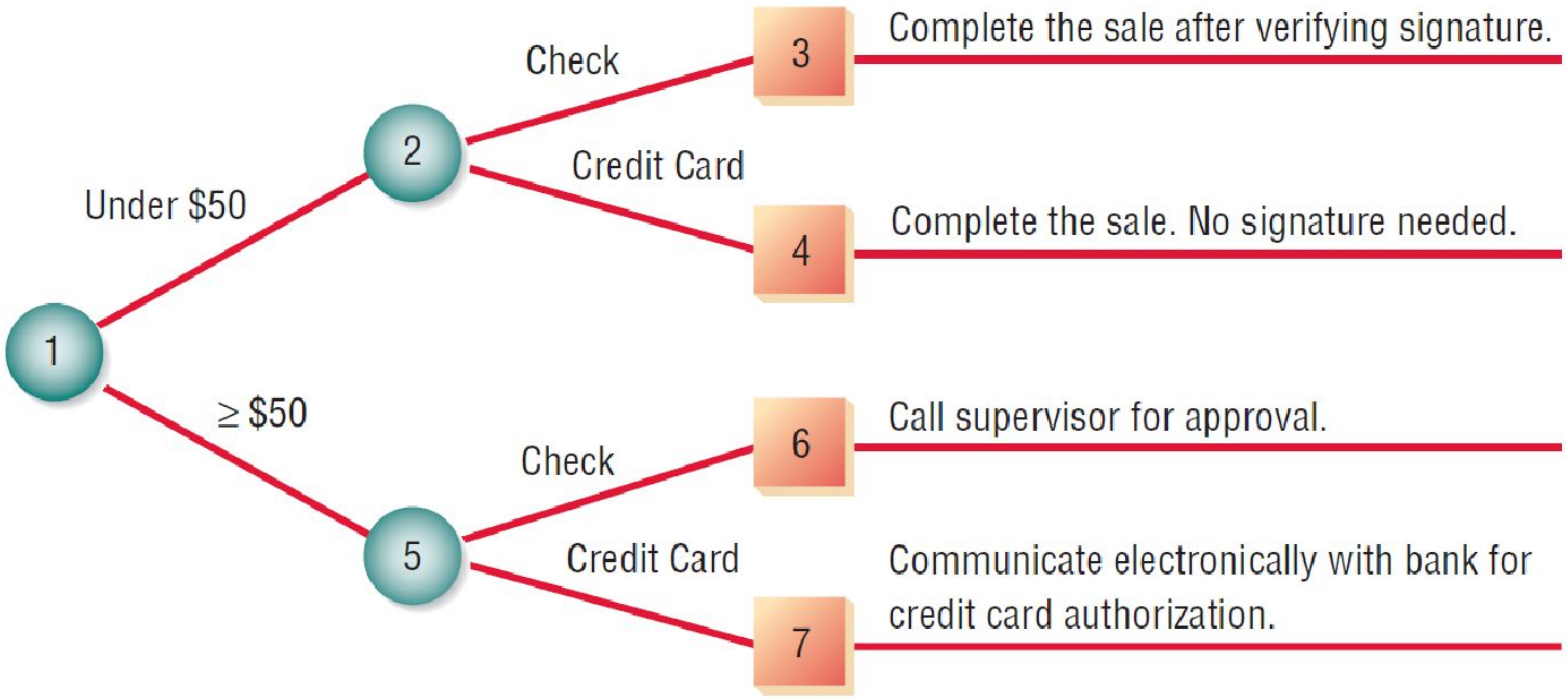
Decision Trees

- Decision trees are used when complex branching occurs in a structured decision process
- Trees are also useful when it is essential to keep a string of decisions in a particular sequence

Drawing Decision Trees

- Identify all conditions and actions and their order and timing (if they are critical)
- Begin building the tree from left to right, making sure you list all possible alternatives before moving to the right

Drawing a Decision Tree to Show the Noncash Purchase Approval Actions for a Department Store (Figure 9.14)



Decision Tree Advantages

- The order of checking conditions and executing actions is immediately noticeable
- Conditions and actions of decision trees are found on some branches but not on others
- Compared to decision tables, decision trees are more readily understood by others in the organization

Selecting a Structured Decision Analysis Technique

- Use structured English when there are many repetitious actions or when communication to end users is important
- Use decision tables when a complex combination of conditions, actions, and rules are found or you require a method that effectively avoids impossible situations, redundancies, and contradictions
- Use decision trees when the sequence of conditions and actions is critical or when not every condition is relevant to every action (the branches are different)

Summary

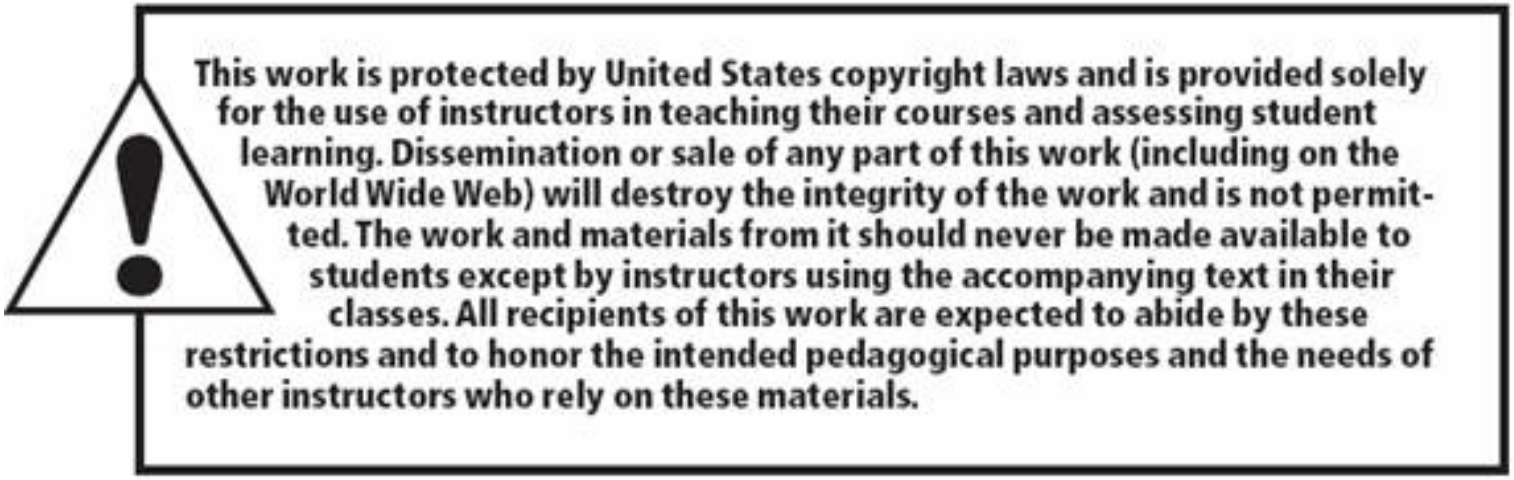
- Process specifications
- Decision analysis
  - Structured English
    - Logic is expressed in sequential structures, decision structures, case structures, or iterations

Summary (continued)

- Decision tables
  - Four quadrants are used to:
    - Describe the conditions
    - Identify possible decision alternatives
    - Indicate which actions should be performed
    - Describe the actions
- Decision trees
  - Consist of nodes and branches

Summary (continued)

- Decision analysis advantages
  - Structured English is useful when many actions are repeated and when communicating with others is important
  - Decision tables provide complete analysis of complex situations while limiting the need for change attributable to impossible situations, redundancies, or contradictions
  - Decision trees are important when proper sequencing of conditions and actions is critical and when each condition is not relevant to each action



END OF
Lecture 09