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.
Process Specifications Are Not Created

- Processes that represent physical input and/or output
- Processes that represent simple data validation
- Processes that use prewritten code
How Process Specifications Relate to the Data Flow Diagram (Figure 9.1)
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 designs
- 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 pseudocode.
- 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 for Determining Whether an Item Is Available (Figure 9.2)
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.
Examples of Logic Expressed in a Sequential Structure, a Decision Structure, a Case Structure, and an Iteration (Figure 9.3)

<table>
<thead>
<tr>
<th>Structured English Type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential Structure</td>
<td>Action #1</td>
</tr>
<tr>
<td>A block of instructions in which no branching occurs</td>
<td>Action #2</td>
</tr>
<tr>
<td></td>
<td>Action #3</td>
</tr>
<tr>
<td>Decision Structure</td>
<td>IF Condition A is True</td>
</tr>
<tr>
<td>Only IF a condition is true, complete the following statements; otherwise, jump to the ELSE</td>
<td>THEN implement Action A</td>
</tr>
<tr>
<td></td>
<td>ELSE implement Action B</td>
</tr>
<tr>
<td></td>
<td>ENDIF</td>
</tr>
<tr>
<td>Case Structure</td>
<td>IF Case #1 implement Action #1</td>
</tr>
<tr>
<td>A special type of decision structure in which the cases are mutually exclusive (if one occurs, the others cannot)</td>
<td>ELSE IF Case #2</td>
</tr>
<tr>
<td></td>
<td>Implement Action #2</td>
</tr>
<tr>
<td></td>
<td>ELSE IF Case #3</td>
</tr>
<tr>
<td></td>
<td>Implement Action #3</td>
</tr>
<tr>
<td></td>
<td>ELSE IF Case #4</td>
</tr>
<tr>
<td></td>
<td>Implement Action #4</td>
</tr>
<tr>
<td></td>
<td>ELSE print error</td>
</tr>
<tr>
<td></td>
<td>ENDIF</td>
</tr>
<tr>
<td>Iteration</td>
<td>DO WHILE there are customers.</td>
</tr>
<tr>
<td>Blocks of statements that are repeated until done</td>
<td>Action #1</td>
</tr>
<tr>
<td></td>
<td>ENDDO</td>
</tr>
</tbody>
</table>
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)

<table>
<thead>
<tr>
<th>Conditions and Actions</th>
<th>Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conditions</td>
<td>Condition Alternatives</td>
</tr>
<tr>
<td>Actions</td>
<td>Action Entries</td>
</tr>
</tbody>
</table>

Kendall & Kendall

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall
Constructing a Decision Table for Deciding Which Catalog to Send to Customers Who Order Only from Selected Catalogs (Figure 9.9)

<table>
<thead>
<tr>
<th>Conditions and Actions</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer ordered from Fall catalog.</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Customer ordered from Christmas catalog.</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Customer ordered from specialty catalog.</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Send out this year’s Christmas catalog.</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Send out specialty catalog.</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Send out both catalogs.</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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.
- 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.
Checking for Completeness and Accuracy

• Four main problems:
  • Incompleteness
  • Impossible situations
  • Contradictions
  • Redundancy
Checking the Decision Table for Inadvertent Contradictions and Redundancy Is Important (Figure 9.13)

<table>
<thead>
<tr>
<th>Conditions and Actions</th>
<th>Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condition 1</td>
<td>Y</td>
</tr>
<tr>
<td>Condition 2</td>
<td>Y</td>
</tr>
<tr>
<td>Condition 3</td>
<td>Y</td>
</tr>
<tr>
<td>Action 1</td>
<td>X</td>
</tr>
<tr>
<td>Action 2</td>
<td>X</td>
</tr>
<tr>
<td>Action 3</td>
<td>X</td>
</tr>
</tbody>
</table>

Contradiction  
Redundancy
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)

1. Under $50
   - 2. Check
       - 3. Complete the sale after verifying signature.
   - 2. Credit Card
       - 4. Complete the sale. No signature needed.

2. ≥ $50
   - 5. Check
       - 6. Call supervisor for approval.
   - 5. Credit Card
       - 7. Communicate electronically with bank for credit card authorization.
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.
• 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.