Chapter 3 - Structuring System Requirements: Logic Modeling

Logic Modeling

Data flow diagrams do not show the logic inside the processes. Logic modeling involves representing internal structure and functionality of processes depicted on a DFD. Logic modeling can also be used to show when processes on a DFD occur.

Modeling Logic with Structured English

Structured English are modified form of English used to specify the logic of information processes. It might look familiar to programming students because it resembles pseudocode. It uses a subset of English. It has no specific standards. It includes Action verbs, Noun phrases, but doesn't include adjectives or adverbs. Structured English uses three basic types of statements to describe a process. They are:

(1) Sequence

A sequence structure is a single step or action included in a process. It does not depend on the existence of any condition, and when encountered, is always taken. Sequence requires no special structure but can be represented with one sequential statement following another.

Example: A block of action where no branching occurs.
Action #1
Action #2
Action #3

(2) Conditional statements

The action sequences are often included within condition structures that identify conditions. Conditional statements occur when two or more actions can be taken, depending on the value for a certain condition.

Example 1: If … endif

```
If
  {
      Quantity-in-stock is less than minimum-order-quantity
      Then generate new order
  }

else
   {
      do nothing!
   }
```

Example 2: Case

```
READ quantity-in-stock
SELECT CASE

CASE 1 (quantity-in-stock is greater than minimum-order-quantity)
        Do nothing
CASE 2 (quantity-in-stock equals minimum-order-quantity)
        Do nothing
CASE 3 (Quantity-in-stock is less than minimum-order-quantity)
        GENERATE new order
CASE 4 (stock out)
        INITIATE emergency reorder routine

END CASE
```

(3) Repetition

In routine operating activities, it is common to find that certain activities are repeated while a certain condition exists or until a condition occurs. Repetition permits analysts to describe these cases.

Example 1: Do-until

```
DO
        READ inventory_records
        BEGIN IF
                IF quantity-in-stock is less than minimum-order-quantity
                THEN generate neworder
        ELSE do nothing
        END IF
UNTIL end-of-file
```

Example 2: Do-while

```
READ inventory_records
WHILE NOT end-of-file DO


                BEGIN IF
                        IF quantity-in-stock is less than minimum-order-quantity
                        THEN generate neworder
                ELSE do nothing
```

```
                    END IF
                END DO
```

Example 3: Iteration

```
for (int i = 0; i < someValue(); i++)
        {
        Count_the_inventory();
        }
```

Five conventions to follow when using Structured English:

- ✓ Express all logic in terms of sequence, conditional statements, or repetitions.
- ✓ Use and capitalize accepted keywords such as: IF, THEN, ELSE, DO, DO WHILE, DO UNTIL, and PERFORM.
- ✓ Indent blocks of statements to show their hierarchy (nesting) clearly.
- ✓ When words or phrases have been defined in the Data Dictionary, underline those words or phrases to indicate that they have a specialized, reserved meaning.

Example

This example shows how a logical DFD maps to Structured English. Each of the four processes on the below DFD are represented here.

Figure Current logical DFD for inventory control system

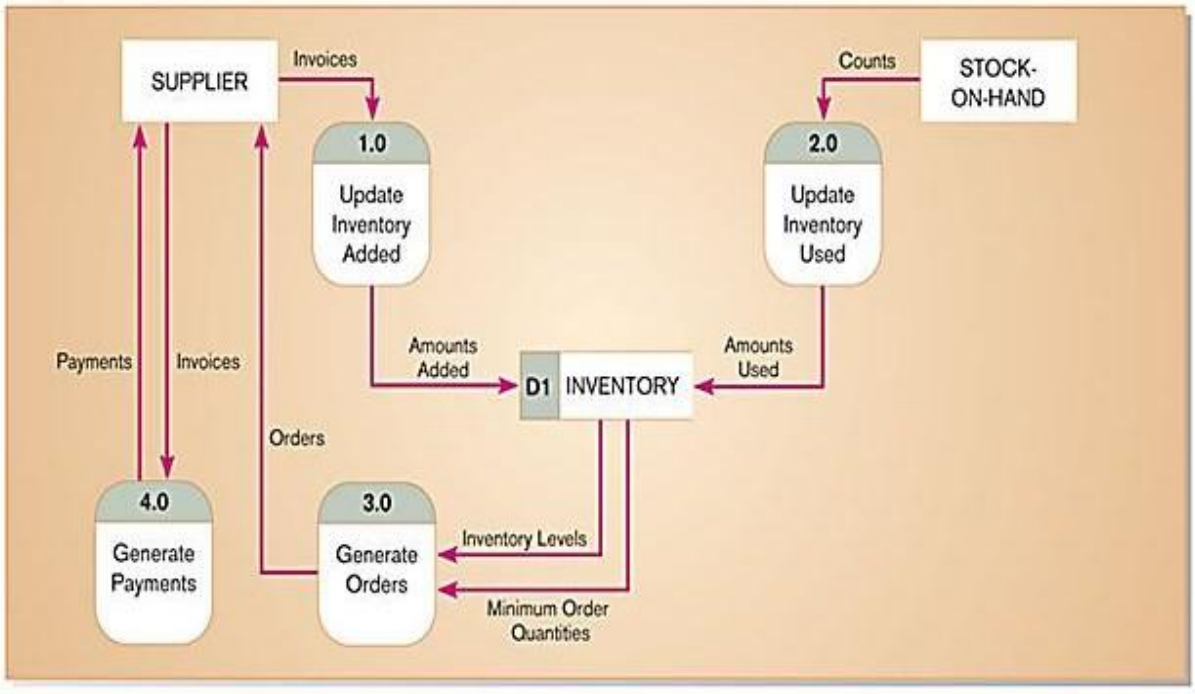


Figure Structured English representations of the four processes

```
Process 1.0: Update Inventory Added
DO
    READ next Invoice-item-record
    FIND matching Inventory-record
    ADD Quantity-added from Invoice-item-record to Quantity-in-stock on
      Inventory-record
UNTIL End-of-file
```

```
Process 2.0: Update Inventory Used
DO
    READ next Stock-item-record
    FIND matching Inventory-record
    SUBTRACT Quantity-used on Stock-item-record from Quantity-in-stock on
      Inventory-record
UNTIL End-of-file
```

```
Process 3.0: Generate Orders
DO
    READ next Inventory-record
    BEGIN IF
      If Quantity-in-stock is less than Minimum-order-quantity
      THEN GENERATE Order
    END IF
UNTIL End-of-file
```

```
Process 4.0: Generate Payments
READ Today's-date
DO
    SORT Invoice-records by Date
    READ next Invoice-record
    BEGIN IF
      IF Date is 30 days or greater than Today's-date
      THEN GENERATE Payments
    END IF
UNTIL End-of-file
```

Another Example of Structured English

SAMPLE OF A SALES PROMOTION POLICY:

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

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
```

Modeling Logic with Decision Tables

Decision tables are a precise yet compact way to model complicated logic. Decision tables, like if-then-else and switch-case statements, associate conditions with actions to perform. But, unlike the control structures found in traditional programming languages, decision tables can associate many independent conditions with several actions in an elegant way.

Decision table is a matrix representation of the logic of a decision. It 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. It is best used for complicated decision logic. You can have more than two possible outcomes. They are the best way to describe a complex set of conditions.

Procedure for Creating Decision Tables

- ✓ Name the condition and values each condition can assume
- ✓ Name all possible actions that can occur
- ✓ List all rules
- ✓ Define the actions for each rule
- ✓ Simplify the table

A Decision table has three parts:

1. Condition stubs
   - ✓ Lists condition relevant to decision
2. Action stubs
   - ✓ Actions that result from a given set of conditions
3. Rules
   - ✓ Specify which actions are to be followed for a given set of conditions

Indifferent Condition

It is the condition whose value does not affect which action is taken for two or more rules.

	Conditions/ Courses of Action	Rules					
		1	2	3	4	5	6
Condition Stubs	Employee type	S	H	S	H	S	H
	Hours worked	< 40	< 40	40	40	> 40	> 40
Action Stubs	Pay base salary	X		X		X	
	Calculate hourly wage		X		X		X
	Calculate overtime						X
	Produce Absence Report		X				

Figure: Complete Decision Table for Payroll System

Note: For salaried employees the action stub chosen will always be the same…therefore hours worked is an indifferent condition.

Conditions/ Courses of Action	Rules			
	1	2	3	4
Employee type	S	H	H	H
Hours worked	–	< 40	40	> 40
Pay base salary	X			
Calculate hourly wage		X	X	X
Calculate overtime				X
Produce Absence Report		X		

Figure: Reduced Decision Table for Payroll System

Because of indifferent condition, the complete decision table can be reduced to one with fewer rules.

Another Example of Decision Table

	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

	1	2	3	4	5	6	7	8
Preferred customer	Y	Y	Y	Y	N	N	N	N
Ordered more than $1,000	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					
$5 bonus coupon				X	X	X	X	X

Another Example of Decision Table

Conditions	Printer does not print	Y	Y	Y	Y	N	N	N	N
	A red light is flashing	Y	Y	N	N	Y	Y	N	N
	Printer is unrecognized	Y	N	Y	N	Y	N	Y	N
Actions	Check the power cable			X					
	Check the printer-computer cable	X		X					
	Ensure printer software is installed	X		X		X		X	
	Check/replace ink	X	X			X	X		
	Check for paper jam		X		X				

Modeling Logic with Decision Trees

In operations research, specifically in decision analysis, a decision tree is a decision support tool that uses a graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. A decision tree is used to identify the strategy most likely to reach a goal. Another use of trees is as a descriptive means for calculating conditional probabilities.

A decision tree is a graphical representation of a decision situation in which decision points (nodes) are connected together by arcs (one for each alternative on a decision) and terminate in ovals (the action which is the result of all of the decisions made on the path that leads to that Oval). It is a graphical representation of the conditions, actions, and rules found in a decision table.

Main components of Decision Tree are:

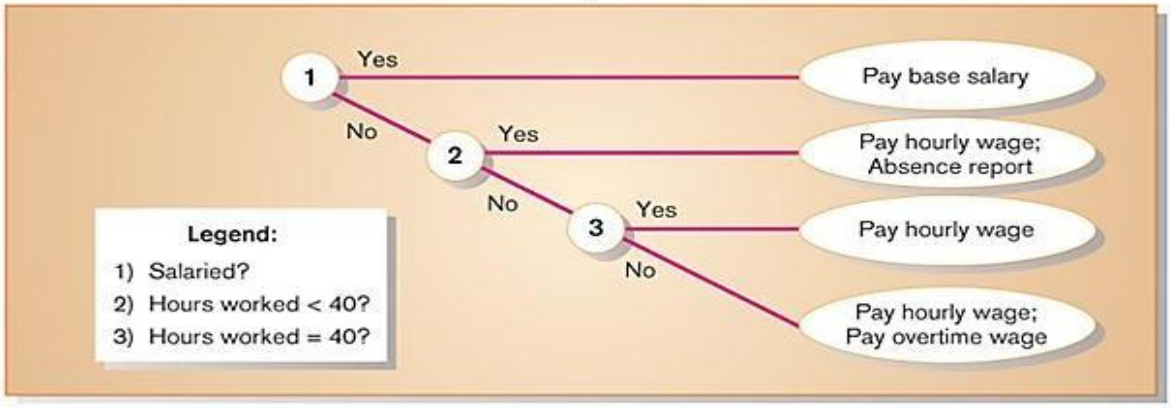
- ✓ Decision points represented by nodes
- ✓ Actions represented by ovals
- ✓ Particular choices from a decision point represented by arcs

Decision trees are organized in a hierarchical fashion, starting with the root node on the far left, and proceeding to subsequent decision nodes. All possible actions are listed in leaf nodes on the far right. Decision trees are read from left to right. Each node corresponds to a numbered choice on a legend. All possible actions are listed on the far right. Whether to use a decision table or tree often is a matter of personal preference

Four major steps in building Decision Trees:

1. Identify the conditions
2. Identify the outcomes (condition alternatives) for each decision
3. Identify the actions
4. Identify the rules.

Figure Decision tree representation of the decision logic in the decision tables with only two choices per decision point



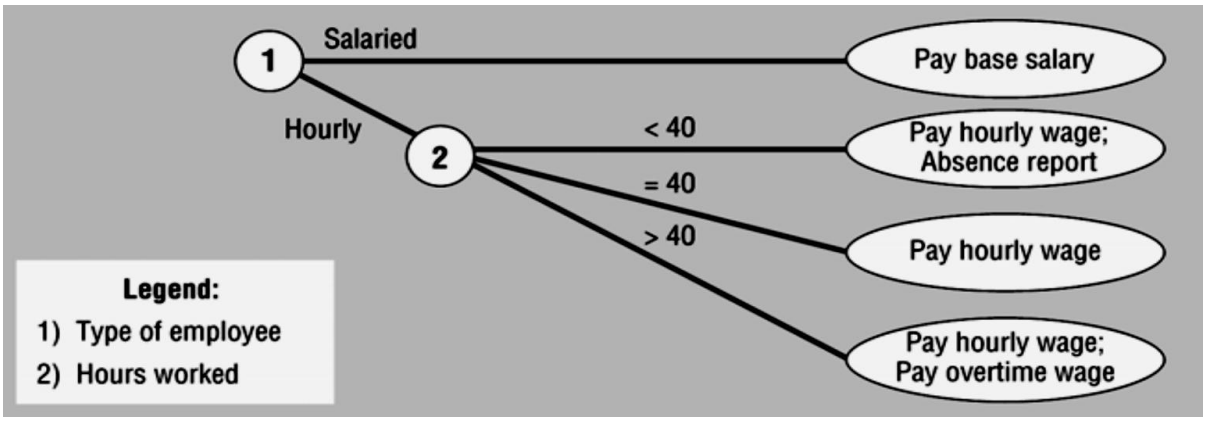
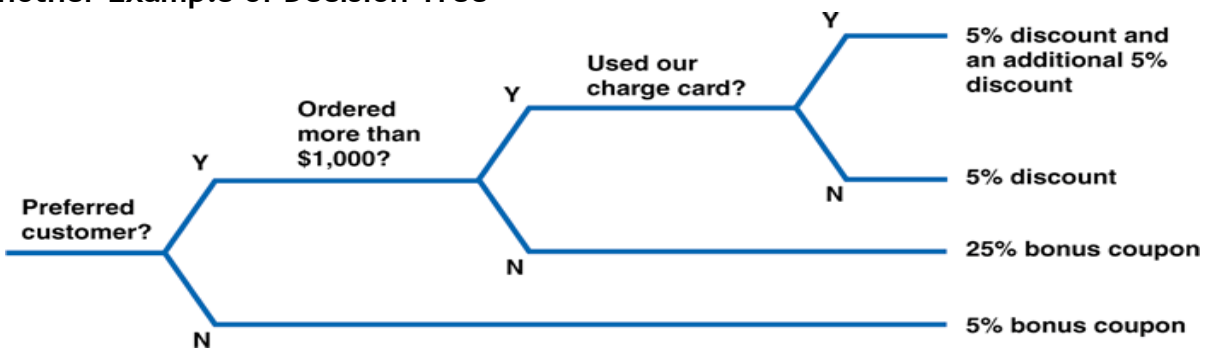


Figure: Decision Tree with Multiple Choices per Decision Point

Another Example of Decision Tree



Deciding Among Structured English, Decision Tables, and Decision Trees

Each technique has strengths & weaknesses. Since there is no best technique for all problems, Use technique that is best matched to problem. Analyst may want to use technique that he prefer or understand best.

Choose Structured English when

- ✓ There are many repetitious actions.
- ✓ Communication to end users is important.

Choose Decision Tables when

- ✓ Complex combinations of conditions, actions, and rules are found.
- ✓ You require a method that effectively avoids impossible situations, redundancies, and contradictions.

Choose Decision Trees when

- ✓ The sequence of conditions and actions is critical.
- ✓ When not every condition is relevant to every action (the branches are different).

Criteria	Structured English	Decision Tables	Decision Trees
Determining Conditions and Actions	Second Best	Third Best	Best
Transforming Conditions and Actions into Sequence	Best	Third Best	Best
Checking Consistency and Completeness	Third Best	Best	Best

Table: Comparison of 3 techniques

Criteria	Decision Tables	Decision Trees
Portraying complex logic	Best	Worst
Portraying simple rules	Worst	Best
Making decisions	Worst	Best
More compact	Best	Worst
Easier to manipulate	Best	Worst

Table: Compares Decision Tables & Decision Trees

References:

- ✓ Hoffer, J.A., George, J.F. and Valacich J.S., "Modern Systems Analysis and Design", 3rd Edition, Pearson Education, 2003.
- ✓ K.E.Kendall and J.E.Kendall, "Systems Analysis and Design", 5th Edition, Pearson Eduation, 2003.
- ✓ V.Rajaraman, "Analysis and Design of Information Systems", 2nd Edition, Prentice Hall of India, New Delhi, 2002
- ✓ E.Yourdon "Modern Structured Analysis", Prentice Hall of India, 1996.
- ✓ Elias M Awad, "Systems Analysis and Design", Galgotia.
- ✓ Igor Hawryszkiewycz, "Systems Analysis and Design", PHI.
- ✓ R.Schultheis and Mary Summer, "Management Information Systems", Tata McGraw Hill, 1999.

Assignments:

(1) Define logic modeling. Explain decision tree logic modeling technique with example.
(2) Model the logic of the following situation using Structured English, Decision Table (also simplify the table) and Decision Tree.

A student who has not missed more than 5 classes is issued an Exam Slip. For the students who have missed more than 5 classes, the personnel checks whether the Leave letter is Submitted or not. The Exam slip is then issued to the student only when the Leave is approved.

Gentle Advice:

Please go through your text books and reference books for detail study!!! Thank you all.