Daffodil Polytechnic Institute

System Analysis Design

Chapter 5

Understand the tools of structured analysis

(Decision trees, Decision table, Structured english and Data dictionary)

Daffodil Polytechnic Institute

Learning Goals

In this module we will learn:

1. How to use structured English to precisely specify processes
2. The terminology used in structured English
3. Terminology of decision tables and how it is used to specify complex logic
4. How to detect errors in decision table specifications
5. Terminology and use of decision trees
6. Comparison of structured English, decision tables and decision trees

Process Specification

- Once a DFD is obtained the next step is to precisely specify the process.
- Structured English, Decision tables and Decision Trees are used to describe process.
- Decision tables are used when the process is logically complex involving large number of conditions and alternate solutions
- Decision Trees are used when conditions to be tested must follow a strict time sequence.

Structured English

- Structured English is similar to a programming language such as Pascal
- It does not have strict syntax rules as programming language
- Intention is to give precise description of a process
- The structured English description should be understandable to the user

Example: Structured English

```
if customer pays advance
  then
      Give 5% Discount  else
     if purchase amount >=10,000
      then
          if the customer is a regular customer
           then Give 5% Discount  else
             No Discount
          end if
      else No Discount  end if
end if
```

Daffodil Polytechnic Institute

Decision Table-Example

- **Same structured English procedure given as decision table**

CONDITIONS	RULE1	RULE2	RULE3	RULE4
Advance payment made	Y	N	N	N
Purchase amt >=10,000	-	Y	Y	N
Regular Customer?	-	Y	N	-
ACTIONS				
Give 5% Discount	X	X	-	-
Give No Discount	-	-	X	X

Decision Table-Explanation

- Conditions are questions to be asked
- 'Y' is yes,'N' is no & '-' is irrelevant
- A 'X' against the action says the action must be taken
- A '-' against the action says the action need not be taken

- Rule 2 in decision table DISCOUNT states:

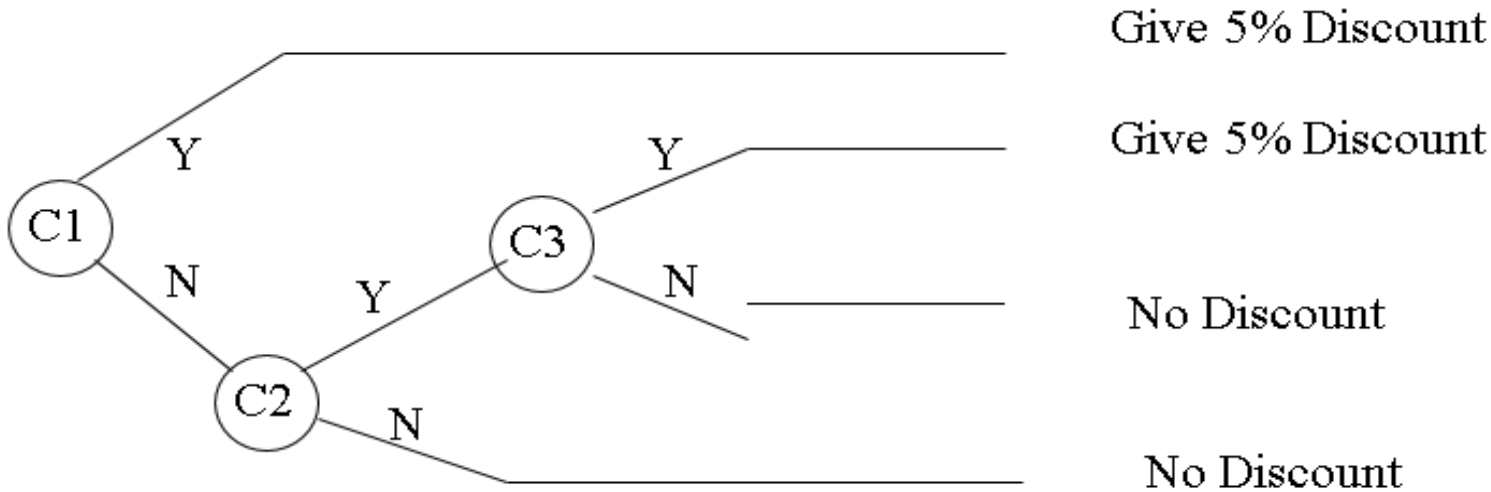
- if no advance payment and purchase amount >=10000 and regular customer then give 5% discount

Structured English

- **Imperative sentences-** Actions to be performed should be precise and quantified
- **Good Example:** Give discount of 20%
- **Bad Example**: Give substantial discount
- **Operators -**Arithmetic : +, -, /, *
  - Relational : >, >=, <, <=, =, !=
  - Logical : and, or, not
  - Keywords : if, then, else, repeat, until, while, do, case, until, while, do, case, for, search, retrieve, read, write
- **Delimiters** – {, }, end, end if, end for

Decision Tree: Example

- **The structured English procedure is expressed as a Decision tree below:**



C1: Advance payment made
C2: Purchase amount >=10,000
C3: Regular Customer

Y = Yes
N = No

Structured English-Decision Structures

```
If condition
        then
            { Group of statements
        }  else
            { Group of statements
    }  end if
```

Example: if(balance in account >= min.balance)

```
            then    honor request
            else    reject request
    end if
```

Daffodil Polytechnic Institute

Structured English- Case Statement

Case (variable)

Variable = P: { statements for alternative P} Variable = Q: { statements for alternative Q} Variable = R: { statements for alternative R} None of the above: { statements for default case}

end case

Example : Case(product code)

product code =1 : discount= 5%

product code =2 : discount =7%

None of the above : discount=0 end case

Daffodil Polytechnic Institute

Structured English-Repetition Structure

```
for index = initial to final do
        { statements in loop }
end for
```

Example : Total =0

```
for subject =1 to subject =5 do
   total marks=total marks +marks(subject)
   write roll no,total marks
end for
```

Structured English-while Loop

while condition do
{ statements in loop
} end while

Example : while there are student records left to do

read student record
compute total
marks find class
write total marks, class, roll no end while

Example

Update inventory file

```
for each item accepted record do
    { search inventory file using item code  if
      successful
        then { update retrieved inventory record;
               write updated record in inventory file using accepted record}
        else { create new record in inventory file;
               enter accepted record in inventory file}
      end if
end for
```

Decision Table-Motivation

- A procedural language tells how data is processed
- Structured English is procedural
- Most managers and users are not concerned how data is processed- they want to know what rules are used to process data.
- Specification of what a system does is non-procedural.
- Decision Tables are non-procedural specification of rules used in processing data

Advantages of Decision Table

- Easy to understand by non-computer literate users and managers
- Good documentation of rules used in data processing.
- Simple representation of complex decision rules .
- Tabular representation allows systematic validation of specification
- detection of redundancy,incompleteness & inconsistency of rules
- Algorithms exist to automatically convert decision tables to equivalent computer programs.
- Allows systematic creation of test data

Method of obtaining decision table from word statement of rules

EXAMPLE

A bank uses the following rules to classify new accounts. If depositor's age is 21 or above and if the deposit is Rs 100 or more, classify the account type as A If the depositor is under 21 and the deposit is Rs 100 or more, classify it as type B If the depositor is 21 or over and deposit is below Rs 100 classify it as C If the depositor is under 21 and deposit is below Rs 100 do-not open account

Identify Conditions: Age >= 21 Cl
Deposits >= Rs 100: C2

Identify Actions : Classify account as A, B or C
Do not open account

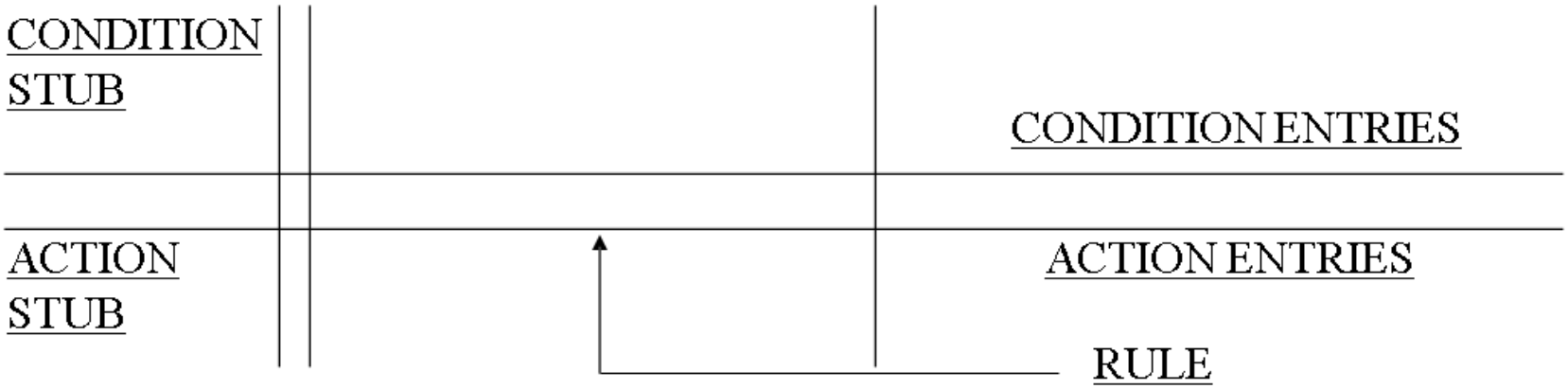
Daffodil Polytechnic Institute

Decision table from word statement

Condition Stub

CODITIONS	Rule 1	Rule 2	Rule 3	Rule 4
C1 : Age >=21	Y	N	Y	N
C2: Deposit >=100	Y	Y	N	N
Action Stub				
ACTIONS				
A1: Classify as A	X	-	-	-
A2: Classify as B	-	X	-	-
A3: Classify as C	-	-	X	-
A4: Do not open Account	-	-	-	X

Decision Table Notation Explained



- 4 Quadrants-demarcated by two double lines
- CONDITION STUB LISTS ALL CONDITIONS TO BE CHECKED
- ACTION STUB LISTS ALL ACTIONS TO BE CARRIED OUT
- LIMITED ENTRY DECISION TABLE:ENTRIES ARE Y or N or -.Y-YES,N- NO,-IRRELEVANT(DON'T CARE)
- X against action states it is to be carried out.
- -against action states it is to be ignored.
- Entries on a vertical column specifies a rule

Decision Table Notation (Contd…)

- Order of listing conditions irrelevant
  - i.e. Conditions may be checked in any order
- Order of listing actions important
- Actions listed first carried out first

Sequential execution of actions

- Rules may be listed in any order

Interpreting Decision Table-Else Rule

	R1	R2	
C1: Is applicant sponsored	Y	Y	ELSE
C2: Does he have min qualification	Y	Y	
C3: Is fee paid?	Y	N	
A1: Admit letter	X	-	-
A2: Provisional Admit letter	-	X	-
A3: Regret letter	-	-	X

Interpretation

- **R1: If applicant sponsored and he has minimum qualifications and his fee is paid –Send Admit letter**
- **R2: If applicant sponsored and has minimum qualifications and his fee not paid send provisional admit letter**

- **ELSE: In all cases send regret letter. The else rule makes a decision table complete**

Daffodil Polytechnic Institute

Decision Table For Shipping Rules

	R1	R2	R3	R4
C1: Qty ordered <= Quantity in stock?	Y	Y	N	N
C2: (Qty in stock-Qty ordered)<=reorder level	N	Y	-	-
C3: Is the partial shipment ok?	-	-	Y	N
A1:Qty shipped=Qty ordered	X	X	-	-
A2:Qty shipped=Qty in stock	-	-	X	-
A3:Qty shipped=0	-	-	-	X
A4:Qty in stock=0	-	-	X	-
A5:Back order=qty ordered-qty shipped	-	-	X	X
A6:Initiative reorder procedure	-	X	X	X
A7: Qty in stock€Qty in stock -Qty shipped	X	X	-	-

Daffodil Polytechnic Institute

Extended Entry Decision Table

- Condition Entries not necessarily Y or N
- Action entries not necessarily X or –
- Extended Entry Decision Tables(EEDT) more concise
- EEDT can always be expanded to LEDT

Example	R1	R2	R3	R4	R5	R6
C1 : Product code	1	1	1	1	1	2
C2 : Customer code	A	B	A	B	C	-
C3 : Order amount	<=500	<=500	>500	>500	-	-
Discount =	5%	7.5%	7.5	% 10%	6%	5%

Mixed Entry Decision Table

Can mix up Yes, No answers with codes

	Rl	R2	R3	R4	R5	R6
Cl : Product code = 1?		Y	Y	Y	Y	N
C2: Customer code =		B	A	B	C	-
C3: Order amount < 500?	A Y	Y	N	N	-	-
Discount =	5%	7 5%	7 5%	10%	6%	5%

Choice of LEDT, EEDT, MEDT depends on ease of communication with user, software available to translate DTs to programs, ease of checking etc.

Linked Decision Table

Decision table 1

Salary point=6	N	e
Conduct OK?	Y	l
Diligence OK?	Y	s
Efficiency OK?	Y	e
Go to table 2	X	-
No promotion	-	X

Decision table 2

Salary point>2	N	N	N	Y
1 yr as class 1 officer	Y	N	-	-
Departmental test Passed?	Y	-	N	-
Advance to next salary point	X	- X	-	-
No promotion	-	-	X	-
Go to Table3	-		-	X

Decision table3

Complete departmental Course	Y	else
1 yr since last increment	Y	
Advance to next salary point	X	-
No promotion	-	X

1.Observe that one can branch between tables

2.Whenever complex rules are given it is a good idea to break them up into manageable parts

Logical Correctness of Decision Table

Consider decision table DT-1:

	Rl	R2
Cl: x>60	Y	-
C2:x<40	-	Y
Al	X	-
A2 :	-	X

We can expand decision table by replacing each –by Y & N

DT2:	R11	R12	R21	R22
Cl: x>60	Y	Y	N	Y
C2:x<40	Y	N	Y	Y
Al	X	X	-	-
A2 :	-	-	X	X

A rule which has no – is an Elementary rule

DT2 is an Elementary Rule Decision Table (ERDT)

Logical Correctness of Decision Table (Cont….)

- A decision table with 1 condition should have 2 elementary rules
- Each elementary rule must be distinct
- Each elementary rule must have distinct action
- If a decision table with k conditions does not have 2^k rules specified it is said to be incomplete
  - For example : DT2 does not have the elementary rule C1:N, C2:N.
- It is thus incomplete.
- ▪If the decision table has the same elementary rule occurring more than once it is said to have multiplicity of specifications
  - For Example:In DT2 The rule C1:Y,C2:Y occurs twice.Thus it has multiplicity of specification

Logical correctness of decision table (cont..)

- If action specified for multiple identical rules are different then it is called **ambiguous specifications**
  DT2 has an ambiguity.Rules R_{11} and R_{21} are identical but have different actions

- Ambiguity may be apparent or real

- It is said to be apparent if the rule leading to the ambiguity is logically impossible

- For example,(x>60)=Y and (x<40)=Y cannot occur simultaneously. Thus in DT2 rules R11 and R22 are apparently ambiguous rules

- Apparently ambiguous rules is **not** an error

Logical correctness of decision table (cont..)

- If an apparently ambiguous specification is real then it is a contradiction
- For example : If C1:(X > 60) = Y and C2:(X > 40) = Y then X = 70 will satisfy both inequalities.
- As two actions are specified for (Cl = Y, C2 = Y) and they are different the rule is really ambiguous and is called Contradictory Specification.

Logical Correctness Of Decision Table (Cont..)

- If all 2^k elementary rules are not present in a k condition decision table is said to be **incomplete.**
- DT2 (PPT 6.3.1) is incomplete as rule C1:N, C2:N is missing
- Rule C1=N, C2:=N is logically possible as C1=N is X<=60
- and C2=N is X >= 40. A value of X = 50 will make C1=N,C2=N Thus DT2 has a **real** incomplete specification
- A decision table which has no real ambiguities or real **incompleteness** is said to be logically correct
- A decision table with logical errors should be corrected

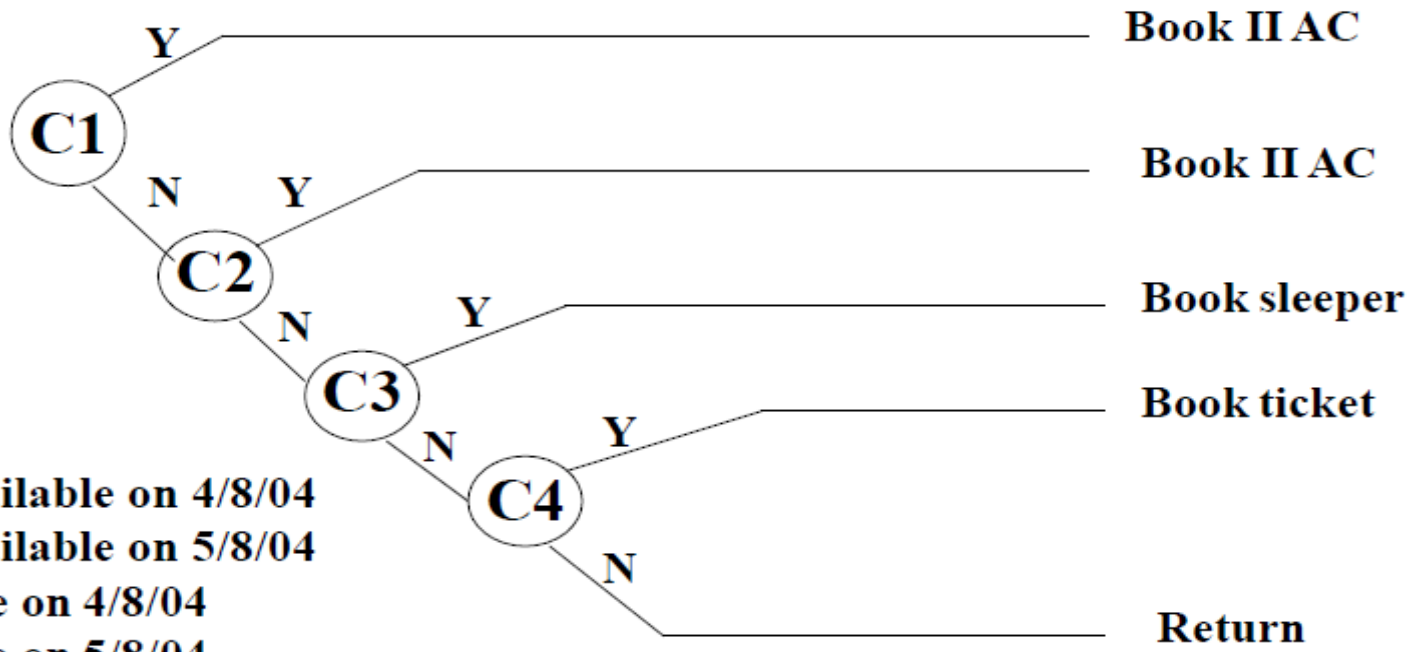
Decision Trees

- Used when sequence of testing condition is important
- It is more procedural compared to Decision tables

Daffodil Polytechnic Institute

Example – Decision Tree to Book Train Ticket

Book by II AC on 4/8/04 if available else book by II AC on 5/8/04.If both not available book by sleeper on 4/8/04 if available else book on 5/8/04 by sleeper.If none available return.



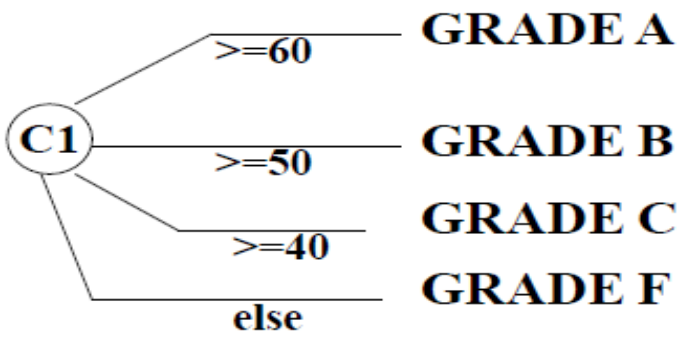
C1: Is II AC ticket available on 4/8/04
C2: Is II AC ticket available on 5/8/04
C3: Is sleeper available on 4/8/04
C4: Is sleeper available on 5/8/04

Observe in the tree sequencing of conditions which is important in this example

Decision Trees

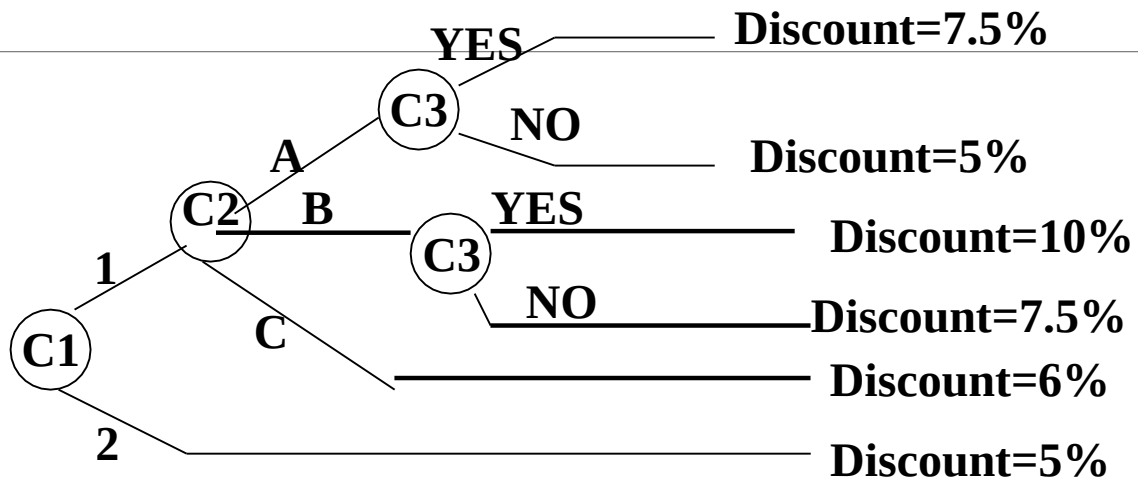
- Decision trees are drawn left to right
- Circles used for conditions
- Conditions labelled and annotation below tree
- Conditions need not be binary

For example:



- Sometimes Decision trees are more appropriate to explain to a user how decisions are taken

Decision Trees



C1: PRODUCT CODE
C2 : CUSTOMER CODE
C3: ORDER AMOUNT >500?

•Observe that the 3 alternatives for connection C2 shown as three branching lines

SOME PEOPLE FIND DECISION TREE EASIER TO UNDERSTAND

Comparison of Structured English, Decision Tables and Decision Trees

CRITERION FOR COMPARISON	STRUCTURED ENGLISH	DECISION TABLES	DECISION TREES
ISOLATING CONDITIONS & ACTIONS	NOT GOOD	BEST	GOOD
SEQUENCING CONDITIONS BY PRIORITY	GOOD	NOT GOOD	BEST
CHECKING FOR COMPLETENESS, CONTRADICTION & AMBIGUITIES	NOT GOOD	BEST	GOOD

When To Use Structured English, Decision Tables and Decision Trees

- Use Structured English if there are many loops and actions are complex
- Use Decision tables when there are a large number of conditions to check and logic is complex
- Use Decision trees when sequencing of conditions is important and if there are not many conditions to be tested