Software Engineering

By Sanjay Shamsundar Manvatkar
Analysis & Design Tools

- Decision tree & decision Table
- DFD (Data Flow Diagram)
- DD (Data Dictionary)
  1) Element of DD
  2) Advantages & Disadvantages of DD
- Input & Output Design.
- Pseudo Code
- Case Studies on Above Topic.
We know that software analysis and design is the intermediate stage which helps human-readable requirements to transformed into actual code.

1. Decision Tree:

Introduction: Decision making is an integral part of any organization no matter how small, simple or big and complex it may be.
• Decision tree is a decision support analysis and design tools that uses a tree like graph or model of decision and their possible consequences.
• Decision tree shows which condition to consider first, which is second and so on.
• Decision tree also shows the relationship of each condition and its permissible actions.
• Decision sequence starts from the root of the tree that is usually on the left of the diagram.
• Decision Tree can be defined as “a graphical representation of specific decision situations in a structured decision process.”
- Decision tree:
  - Sample Format:

  ![Decision Tree Diagram]

  **Fig. 5.1: Basic Format of Decision Tree**
Example: Discount policy of book for Individuals & Bookstores:

Decision tree shown in Fig. 5.3.

- **Type of customer**
  - **Discount policy**
    - **Bookstore**
      - 6 or More: 25%
      - Less than 6: NIL
    - **Libraries or Individuals**
      - 50 or More: 15%
      - 20 - 49: 10%
      - 6 - 19: 5%
      - Less than 6: NIL

Fig. 5.3
Example 2:-
Discount policy for Saree manufacturer:-

Fig. 5.2
## Advantages & Disadvantages of Decision Tree

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Decision tree are simple to understand and interpret.</td>
<td>1. A Large number branches with many path will confuse rather than help in analysis.</td>
</tr>
<tr>
<td>2. It helps the analyst to identify the actual decision to be made.</td>
<td>2. Large decision trees can be hard to interpret.</td>
</tr>
<tr>
<td>3. Decision trees can be combined with other decision techniques.</td>
<td>3. Decision tree are uses a step function that can have large errors.</td>
</tr>
</tbody>
</table>
2. DT (Decision Table)

- **Introduction:**
  1. Decision table is a graphical method for **Explaining the logic of making decision in tabular format.**
  2. In DT set of **Condition** and **Actions** are available.

    Ex. Calculate Discount rates.
1. **Condition stub**: will show the condition that determines which action will result.

2. **Condition Entries**: combination of conditions expressed as rules.

3. **Action Stub**: which contains Possible Actions which can occur as a result of the different condition combinations.

4. **Action Entries**: Which contains the action to be taken.

- Right part of a Decision Table shows, each column is named by a rule number or rule identifier.
Sample Format Of DT:-

<table>
<thead>
<tr>
<th>Conditions stub</th>
<th>Actions stub</th>
<th>Rule identifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of Table</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Conditions entry</th>
<th>Actions entry</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


Example of Decision Table

**Basic Idea behind Decision Tables:**
- Fig. 5.6 shows a decision table. In a nominal limited entry decision table a decision indicator is just a Y (Yes) or N (No) to indicate that the given condition must be fulfilled or not. An indicator of an action is an “X” for those actions to execute and “-” for those actions to drop.

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Car is in good condition</td>
<td>Y</td>
</tr>
<tr>
<td>Its price is under $7500</td>
<td>Y</td>
</tr>
<tr>
<td>Its registration is current</td>
<td>N</td>
</tr>
<tr>
<td>Actions</td>
<td></td>
</tr>
<tr>
<td>Purchase the car</td>
<td>X</td>
</tr>
<tr>
<td>Reject the car</td>
<td>X</td>
</tr>
</tbody>
</table>

Possible outcomes

Yes/No question

Y = Condition is true
N = Condition is false

X = This action matches the given rules
Advantages & Disadvantages of Decision Table (DT)

**Advantages**

1. Decision Table Can be changed according to situation.
2. DT may be easier to construct than flowchart.
3. It Provides Compact Representation of making process.

**Disadvantages**

1. DT does not show the flow of logic for the solution to a given problem.
2. If there are too many alternatives it is difficult to list in DT.
3. It may be difficult for the programmer to translate decision table in to prog.
3. DFD (Data Flow Diagram)

- DFD is a graphical representation of the flow of data through a information system.
- DFD represents System Data in a “Hierarchical Manner” with required levels of details.

For **DFD** Creation We use various types of symbols are used.
1. Entry

2. Process

3. Data Flow

4. Data Store

Product/ Data Store

Customer

Verify Order
Types of DFD

- There are two types of DFD both of which supports a top-down approach to system analysis.
  1. Logical DFDs: Specify the various logical processes performed on data i.e. type of operations performed.
  2. Physical DFDs: shows how system will be implemented. Work done in manually or computerized.
Example of DFDs

- DFD For Online Shopping System: Level 0
Online Shopping System: Level 1 & Level 2
Advantages & Disadvantages of DFDs (Data Flow Diagram)

**Advantages**

1. Simple graphical step & easy to understand for technical & non-technical user.
2. It is useful for communicating current system knowledge to user.
3. DFD can provide a detailed representation of system components.

**Disadvantages**

1. DFD undergoes lot of alteration before going to users, so makes the process slow.
2. DFD takes long time to create.
3. Different DFD models have different symbols since it is complicated.
DD (Data Dictionary)

• Introduction: DD is an organized collection of relational information.

• DD is centralized repository of information about system and its elements such as meaning, relationships, origin, usage and format.

• DD Consist of following items
  
  Name, Alias, use, Content description, Additional information etc.
Sample of DD (Data Dictionary)

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Use</th>
<th>Content Description</th>
<th>Supplement/ Additional Information</th>
</tr>
</thead>
</table>

Fig. 5.17
Example of DD (Data Dictionary)

Build the requirements dictionary:

<table>
<thead>
<tr>
<th>Name:</th>
<th>Telephone number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aliases:</td>
<td>Phone number, number</td>
</tr>
<tr>
<td>Where/How used:</td>
<td>Read-phone-number (input) display-phone-number (output) analyze-long-distance-calls (input)</td>
</tr>
<tr>
<td>Description:</td>
<td>Telephone no. = [local extension</td>
</tr>
<tr>
<td>Format:</td>
<td>Alphanumeric data</td>
</tr>
</tbody>
</table>
# Advantages & Disadvantages of DD (Data Dictionary)

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. DD manage the details in large system and DD improves Consistency.</td>
<td>1. For large organization a DD is grows rapidly in size and complexity.</td>
</tr>
<tr>
<td>2. DD is used to communicate common meaning for all system element.</td>
<td>2. DD is difficult to maintain manually.</td>
</tr>
</tbody>
</table>
Input & output Design

- **Input Design:**

  Input design means “Designing the screen used to enter the information, as well as any forms on which users write or type information like time card”.

  The goal of input design is to capture accurate information for the system simply and easily.

  The fundamental principles for input design reflect the nature of the inputs & ways to simplify their collection.
The system analyst should work out the following input design details:

1. What data is to be input?
2. What medium is to be used?
3. How should the data be arranged or coded?
4. The interface to be provided to the users for providing input.
5. Entities & transactions requiring validation to detect errors.
6. Ways to perform input validation & measures to take in case error occurs.
Thank You