IF you write documentation, THEN try a decision table

DEBORAH GETTYS

Abstract—**A decision table is a powerful documentation tool based on a simple principle: sets of responses for sets of conditions. It is used to present a large quantity of complex information in a simple, straightforward manner. Since the decision table requires no special symbols nor shapes, it is understood by nonprogrammers and programmers alike. Because of this, it can serve as a common denominator for a project team, a quick-reference card, and a step-by-step procedural guide. Invest time in explaining how to use a decision table, and both you and the user will profit from a valuable documentation tool.**

AS A DOCUMENTATION SPECIALIST for a major agribusiness, I am interested in clear and concise ways of explaining the logic and operations of complex programs. Because of this, I began studying decision tables as a documentation tool and found them very adaptable. This article defines a decision table, explains its construction, and presents a method for teaching its use.

WHAT IS A DECISION TABLE?

A decision table is a collection of *If* → *then* statements. Most people have had experience with a simple form of a decision table, a tax table. *If* you make X amount of money, *then* you pay Y amount of taxes. Based on the condition, or *If* statement, a decision is made—in this case, how much you must pay in taxes.

Because computer programs usually have more than one condition or *If*, the resulting decision table is more complex. The principle, however, is the same: sets of responses for sets of conditions. And although the principle is simple, the table can relay vast amounts of complex information, and do so in easy-to-understand terms. In other words, it is a powerful documentation tool.

WHAT ARE THE ADVANTAGES OF USING A DECISION TABLE?

The first and most important advantage of a decision table is that it can summarize several pages of detailed written documentation in one page; it is thus ideal for review or quick referral. Recently, I used three decision tables on a reference card to summarize how-to procedures from an 80-page detailed user manual.

Deborah Gettys writes user documentation, including manuals, online help, and reference materials, for research systems of Pioneer Hi-Bred International, Inc. in Des Moines, Iowa. Prior to joining Pioneer in 1985, she taught in the public schools and also wrote and designed five computer-based training courses for a national insurance company. She received an M.S. degree in teaching and curriculum development from Drake University in Des Moines.

Second, unlike other graphic tools, such as the flow chart, a decision table requires no arrows, symbolic shapes, or paths to follow. Detailed planning and logic are provided in a nonthreatening manner, and communication between programmers and nonprogrammers can be improved. As a common denominator, or "translator" for the user, manager, and programmer, the decision table benefits each in any discussion of system logic and conditions.

Third, a decision table allows easy comparison of the options and responses available to the user. By reviewing a table that gives a system or screen summary, the user is able to make a better choice. In fact, its most effective use is at the point where the user must select one alternative from a set of many possibilities. Specific applications of decision tables include the following:

- Screen handling
- Routes through a menu-driven system
- Troubleshooting guide for software or hardware
- Command functions
- Valid entries for data items

WHAT ARE THE DISADVANTAGES OF USING A DECISION TABLE?

Although a decision table has numerous advantages, it is not a documentation panacea. First, the need for detailed how-to documentation is not met with this tool, nor does it do well in demonstrating information or process flow. As with any tool, it fits some situations better than others.

Second, if you use a decision table, you must be willing to teach the user how to read it. This can be done quickly and easily through a variety of media. One approach I have used is explained in detail at the end of this article.

HOW IS A DECISION TABLE CONSTRUCTED?

The process of constructing a decision table consists of four steps. In the following paragraphs, I have demonstrated each step and developed a decision table for the hypothetical task of adding several records to a system. When you begin the process of developing your own decision table, I recommend using word processing or paper

0361-1434/86/1200-0061$01.00

Part I	Part II
Part III	Part IV

Figure 1. The quadrants of a decision table.

Are records in sequential file?	
Are historical data stored?	Part II
Want calculated values stored?	
Part III	Part IV

Figure 2. The list of conditions.

and pencil—not ink. It makes arranging and rearranging information in the initial stage much easier.

Step 1:

Divide a piece of paper into quadrants (Fig. 1). Parts I and II are in the upper half, parts III and IV in the lower half. In part I, list all the conditions the user must consider when deciding what to do. Conditions are usually written as *IF* statements or YES/NO questions, such as:

- IF the red light is blinking
- IF the value is between 0 and 1
- Do you want to update the data base?
- Have you plugged the machine in?
- Are the records in a sequential file?

Are records in sequential file?								
Are historical data stored?								
Want calculated values stored?								
Part III	Pa	rt	IV					

Figure 3. The number of columns equals 2^n.

Are records in sequential file?	Y	N	Y	N	Y	N	Y	N
Are historical data stored?	Y	Y	N	N	Y	Y	N	N
Want calculated values stored?	Y	Y	Y	Y	N	N	N	N
Part III	Pa	rt	IV					

Figure 4. Possible combinations of answers.

Try to arrange the list in a logical order: chronologically, alphabetically, or in the order in which options display on a menu. For the example of adding records to a system, I have listed three YES/NO questions as the conditions (Fig. 2).

Step 2:

Once the conditions are listed, move to part II. This quadrant will contain possible combinations of YES/NO answers to the conditions listed in part I. Divide parts II and IV into columns; the number of columns needed is determined by the formula 2^n, where n equals the number of conditions listed:

$$n = 2 \text{ conditions} \rightarrow 2^2 = 4 \rightarrow 4 \text{ columns}$$

$$n = 3 \text{ conditions} \rightarrow 2^3 = 8 \rightarrow 8 \text{ columns}$$

$$n = 4 \text{ conditions} \rightarrow 2^4 = 16 \rightarrow 16 \text{ columns}$$

$$n = 5 \text{ conditions} \rightarrow 2^5 = 32 \rightarrow 32 \text{ columns}$$

Since this example includes 3 conditions, 8 columns are needed (Fig. 3).

To be sure that all possible combinations are listed, begin with the first condition and alternately fill in *Y* and *N* all across the top row of part II. Now, move down to the second condition. Once again, fill in the columns of part II, only this time repeat *Y Y N N*. For the third condition, fill in *Y Y Y Y N N N N*. In other words, double the pattern of answers with each condition (Fig. 4). In some cases, where there are several columns and the user is going to be looking specifically for "Yes" answers, I omit the Ns from the table (or vice versa). It makes it easier for the user to pick out answers.

Step 3:

Now, in part III, write a summary list of responses to all the conditions. These responses can be explanatory or directive, written as *THEN* statements or as commands. Some appropriate responses for the conditions listed earlier include:

- If the red light is blinking, *then a power failure has occurred.*
- If the value is less than 0, *then add it to 1.*
- Do you want to update the data base? *Type STORE and press <RETURN>.*
- Have you plugged the machine in? *Plug the cord into a heavy-duty outlet.*
- Are the records in a sequential file? *Type the file name.*

Keep in mind the purpose of the decision table when writing these responses. If the purpose is to guide someone through a system, the responses need to be in a command form, such as "Select ADD from the menu," "Press <*RETURN*>," or "Move to the Command Line." If the purpose is for troubleshooting, the responses may need to be more explanatory, such as "Evidence of power surge," or "Paper tray is empty".

As with the conditions, try to arrange the list of responses in a logical order. Most of the screens that I document have two work areas: the main body of the screen where data are typed, and a command line at the bottom of the screen where specific actions are listed. Because of this, I often divide part III into two sections: actions to take in the main body of the screen, and commands to type at the command line (Fig. 5). It makes it easier for the user to see at a glance what needs to be done at each location.

Step 4:

Finally, look at the first column in Part II and decide which responses are appropriate for that condition set. Follow the column down to part IV, and at the point(s) of intersection between the column and each appropriate response row, fill in an asterisk. One condition set may require three responses while another requires only one. If the order in which the responses are made is important, use numbers instead of asterisks. Work through each column in this manner (Fig. 6).

Notice that the last column contains all *N*s in part II and no responses have been identified in part IV. Sometimes, a certain response to one or more of the conditions determines that the procedure cannot be done. In this example, if all the conditions are answered "No," the records cannot be added to the system. This information can be communicated in one of two ways:

- By directing the user to a source such as a detailed reference manual or another decision table that gives the required instructions
- By deleting the entire last column from the table

The decision depends on the level of sophistication of the

Are records in sequential file?	Y	N	Y	N	Y	N	Y	N
Are historical data stored?	Y	Y	N	N	Y	Y	N	N
Want calculated values stored?	Y	Y	Y	Y	N	N	N	N
Do this on the screen:								
Type file name								
Type list of records								
Do this at the Command Line:								
Type RETRIEVE								
Type CONTINUE								
Type EXIT								

Figure 5. The list of responses.

Are records in sequential file?	Y	N	Y	N	Y	N	Y	N
Are historical data stored?	Y	Y	N	N	Y	Y	N	N
Want calculated values stored?	Y	Y	Y	Y	N	N	N	N
Do this on the screen:								
Type file name	1		1		1		1	
Type list of records		1		1		1		
Do this at the Command Line:								
Type RETRIEVE	2	2			2	2		
Type CONTINUE	3	3	2	2				
Type EXIT					3	3	2	

Figure 6. Appropriate responses for each set of conditions.

Customer Information System: Sequential Add Screen
Adding multiple records

Are records in sequential file?	Y	N	Y	N	Y	N	Y
Are historical data stored?	Y	Y	N	N	Y	Y	N
Want calculated values stored?	Y	Y	Y	Y	N	N	N
Do this on the screen:							
Type file name	1		1		1		1
Type list of records		1		1		1	
Do this at the Command Line:							
Type RETRIEVE	2	2			2	2	
Type CONTINUE	3	3	2	2			
Type EXIT					3	3	2

Figure 7. The completed decision table.

How to Use Decision Tables

A decision table summarizes sets of responses for sets of conditions. Think of it as several "IF ⟶ THEN" statements, like a tax table: if you make this much money, then you pay this amount in taxes.

A decision table is divided into four parts:

1. Part I is a set of conditions or questions.
2. Part II includes all possible combinations of responses to the conditions in Part I.
3. Part III lists possible actions needed to answer the conditions in Part I.
4. Part IV identifies specific actions needed to answer the conditions selected in Part I. These are numbered if the order is important.

I	II
III	IV

Here's a decision table that might be used by a gardener.

Is the ground dry?	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N
Are the weeds winning?	Y	Y	N	N	Y	Y	N	N	Y	Y	N	N	Y	Y	N	N
Any signs of insects?	Y	Y	Y	Y	N	N	N	N	Y	Y	Y	Y	N	N	N	N
Any sign of critters?	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N
Relax and watch it grow																1
Water the garden			1		1		1		1		1		1			
Pull weeds		1			2	1			2	1			2	1		
Apply insecticide		2	2	1					3	2	2	1				
Check/replace fencing		3	3	2	3	2	2	1								
Buy produce at store	1															

Let's say the answer to all of the conditions is No. The gardener looks down the column of all N's and sees "1" after "Relax and watch it grow." That's the action for that set of conditions. Or, if the answer to both "Is the ground dry?" and "Any sign of insects?" is Yes, the gardener looks down the column where these two conditions are answered Y and sees "Water the garden" is the first action and "Apply insecticide" is the second. However, if the answer to all of the questions is Yes, the gardener can see that it's time to admit defeat and buy produce at the store.

Figure 8. Guide to use of decision tables.

user and on what other forms of documentation are available. I usually delete any column like this, and have done so for the final form of the example (Fig. 7).

At this point, the decision table is constructed and needs only a few final touches:

- A title identifying the system, screen, and/or procedure
- A legend defining any symbols, abbreviations, or conventions used in the conditions or responses if they are different from your other documentation standards
- Testing by your standard quality-assurance procedure (Fig. 7).

HOW DO I TEACH THE USER ABOUT DECISION TABLES?

In teaching the user about decision tables, start with a nonwork topic that neither questions nor threatens the user's knowledge of the job or system. Because I seldom meet our users face-to-face, I needed to develop a stand-alone guide. The example in Fig. 8. is the result.

The two most engaging powers of an author are to make new things familiar, familiar things new.

—William Makepeace Thackeray