Decision Tables Make for Better Decisions

By Burton Grad

Written: June, 2016

CHM Reference number: R0361.2016

Introduction

In 1957, I was working in the General Electric Production Control Services staff in New York City and we were deeply immersed in the then early use of digital computers. They were being used to automate a wide range of manufacturing management activities including inventory control, factory scheduling, cost accounting and parts and materials purchasing. At the same time, there was extensive activity in Manufacturing Engineering Services on how to automate the physical manufacturing processes for job shop (e.g. custom) production in addition to the ubiquitous assembly lines for standardized products.

Decision Tables Make for Better Decisions

General Electric, with its wide range of electrical and related products, was in the forefront of seeing how to use these new technologies. But the efforts were scattered among dozens of factory locations and were being separately pursued by each of the operational functions: manufacturing; engineering; marketing and sales; accounting; human relations. Based on discussions with colleagues in both the Corporate Staff and in some of the product divisions, we came up with the idea that an integrated effort to design a comprehensive business information system in tandem with an effort to use the newest job shop factory automation technology could give GE a leg up on its competitors and provide significant marketing and sales advantages for certain products.

Production Control Services proposed to initiate an Integrated Systems Project in cooperation with the other Corporate Staff organizations and with selected representatives from some of the product divisions. The proposal was accepted and the initial work was started in the 39 Frame Motor department located in Fort Wayne, Indiana. The team size was relatively small with about 6-10 participants, and at least one person from each of the organizational functions.

The 39 Frame Motor department produced relatively small motors that were used for a variety of applications like room air conditioners, furnace blowers, fans and blowers, unit coolers, unit heaters, and other air-moving applications Each motor was unique and each product had different requirements depending on the exact function that these motors were to perform and the environment in which it would be used. In addition, there were certain physical (size and power) boundaries for the entire 39 Frame Motor line that had to be incorporated.

We felt that the key to automating any of the information processes was to fully understand the current decision logic used in carrying out the function. For example, when an engineer designed a product, he needed to identify (and record) many factors such as:

- What requirements had to be considered in determining shape and size of each component?
- What elements were critical in selecting the parts to be used?
- How should each individual part look so it could perform its function effectively within cost and weight constraints?

In addition to the many decisions to be made by the engineers, other departments had numerous items that needed to be identified when a product was to be manufactured and priced:

- Manufacturing engineers had to determine how the parts would be made and then assembled, what kind of tools should be used, what templates and jigs would be needed, and how much time each individual operation would take since payment to many of the factory employees was on a piecework basis.

- The production control specialists were concerned about the specific parts to be purchased in what quantities and when, and how to schedule the manufacturing operations to deliver the product in a timely fashion while effectively controlling the costs.

- Cost accounting had its own factors to consider in determining the costs for different lot sizes and Marketing had come up with a competitive price which would still insure adequate profitability.

The list of decisions went on and on. Because of the range and breadth of these factors, it became of interest to GE management to come up with a way to make the entire process faster and less expensive, with fewer errors.

I was the team manager of this very talented group of functional specialists. We all knew that we were pushing the envelope in terms of both computer and factory automation technology and in terms of our ability to translate the words from everyone involved into a form that could be programmed for computer operation.

I personally started working with the product engineers to understand exactly how they decided on the design of the specific parts that were needed for the end product and how they determined the dimensions for each of these parts. The engineers had a variety of blueprints (drawings) which had been developed over many years for individual parts for previous 39 Frame motors and they used these wherever possible to avoid having to make new blueprints. Each drawing not only had a to-scale picture of the part, but also had the detailed dimensions of every aspect of it. These drawings needed to be in complete detail so that the factory workers who had to actually make the parts would know exactly what had to be done. There were thousands of these drawings.

In addition, the engineers prepared Bills of Materials which spelled out each of the sub-assemblies and assemblies that were needed up to and including the final product itself. These Bills of Materials stated things like the exact number of screws and nuts and bolts needed, the color and type of paint to be used, etc. The set of Bills of Material for a customer's order included every part or sub-assembly that was either purchased or had to be made in the plant. This set identified the quantity of every individual part and was supported by a blueprint for each of those parts and assemblies. It was a tedious process to produce a detailed and accurate set of Bills of Material for every unique order.

We started talking with the 3 or 4 engineers who were responsible for identifying the detailed specifications for particular products. The key to these interviews was that we selected some products that they had designed in the last year or so. We asked them to start with the customer order and explain to us just what criteria in the order led them to decide what parts were needed and the dimension by dimension selections for each part. We were surprised to see that each engineer tended to have their own style and that in many cases there was no particular logic behind why the shaft in the motor was of exactly a certain length or why the chamfer on the end of shaft was at a different angle or why the lubricating grooves were different in depth or width. There seemed to be inconsistencies about how many bolts were needed to anchor the frame to the base and what size bolts should be used not just from one engineer to the other but also even at times for a single engineer.

So, how were we going to record the current engineering design logic and then simplify it so that it could be programmed to enable the computer to actually create the design automatically for a new customer order for a product that was different from any that had been built before? It became clear that if we constructed flow charts to review with the engineers, that this wouldn't work; if we tried to use Boolean algebra, that would be a losing cause. And besides, it wasn't just the communication with the engineers that would be difficult, we ourselves would not be sure that we have covered all of the possible (even reasonable) combinations of conditions that might occur in a future customer order.

And then came the AHA! Moment. At 3 AM one night while at a motel in Fort Wayne, I woke up with visions of tables dancing in my head. I had always found tables an effective way to organize information and I had used lists to ensure that I kept track of the tasks to be done. A tabular format (like accounting documents or a bill of materials) provides a structure and framework to both record information and, often more important, to understand it and to analyze and identify possible errors and inconsistencies. I pictured a two-level table with what I later called "conditions" in the upper half and "actions" in the lower half. I pictured a "stub" on the left which listed the conditions or actions to be taken and to the right, a series of columns each of which represented a particular combination of conditions and their corresponding actions.

This is an overly simplified example of a decision table that could be used to determine when to grant credit to a customer:

TABLE: CREDIT	**Rule 1**	**Rule 2**	**Rule 3**	**Rule 4**
Credit limit is OK	Y	N	N	N
Pay experience is favorable		Y	N	N
Special clearance has been obtained			Y	N

Approve order	X	X	X	
Return order to Sales				X

The top of the “Stub” on the left lists possible decision criteria or “Conditions” (Credit limit is OK, Pay experience is favorable, etc.). Below that on the left are possible decisions or “Actions” (Approve order or Return order to Sales). A “Y” or “N” in a Condition Rule in the top right identifies whether the condition is met or not; an “X” in the Action area in the bottom left identifies which actions to take. This is obviously an oversimplified illustration, but it does give a sense of the format and concept.

And that was the start of my infatuation with what I later called “Decision Tables.” When I got to the factory the next morning, I brought the team together and shared with them what I believed to be a possible solution to what had not only been bugging us with the engineering logic but also with the logic in all of the other functions: manufacturing engineering, production control, cost accounting, etc. After some serious questions regarding what I might have been drinking the previous evening, they agreed that this might be worth a trial to see if it really worked. We decided that I would try it out on the engineering design of the motor shaft. And so, I went through all of my notes on shaft design and started writing tables to express the logic. I did some simpler tables first like the chamfer on the end of the shaft and the shape and number of grooves and the length of the shaft. Each of these was a separate table with the appropriate conditions from the customer order in the top portion of the table and the various dimensions for the particular aspect of the part in the lower portion.

I then showed these tables to the product engineers and asked them if they understood the logic and then, if they were able to follow the logic, would they come up with a satisfactory design decision. With some adjustments in the values of some of the conditions and adding some additional conditions that I had not considered, it turned out that for all of the shaft dimensions they felt that the tables would work and that the resulting designs would be suitable.

With this initial success, we then had the other team members try the Decision Tables out on their portions of the project. And again, with some adjustments in how things were expressed in the stub portion of the table and in the individual columns, the approach seemed to work both for those designing the new automated system and for those whose logic we were trying to capture.

Unfortunately, because of a serious short term business issue at the 39 Frame Motor department, we had to terminate the project at this point. To continue, we had to seek a new location in the company to move forward with our research project.

The good news was that we had made enough progress that our various managers on the Corporate Staff were sufficiently impressed that they agreed that we could look for a new department to work with.

In addition, they recommended that we substantially expand the size of the team and try to design a comprehensive “cradle to grave” information system starting from the receipt of the customer’s order through to the packaging and shipping of the product and the billing to the customer. They also said that we should specifically see what we could do to incorporate numerically controlled machine tools then being made by Kearney & Trecker and others to try to more fully automate the manufacturing process.

With a team of almost 40 people (this was 1958 and it was a high priority project), we were able to find a great location: the Meter department in Lynn, Massachusetts. Lynn is just about 30 minutes north of Boston; GE was a major employer in Lynn, and there were reasonable hotels and some good restaurants there although the town itself was somewhat depressed and very blue collar (and unionized).

Without going into much detail (the Integrated System Project is another story), the product we worked with was a meter which measured an electrical value related either to speed, strain, load, etc., on almost any type of equipment. Because the uses were so varied each order had a number of key factors which would affect the design of the meter. Since they were custom designed and custom built they tended to be relatively expensive and took many months from the time an order came in until the product could be delivered. This became a real full-scale testing ground for Decision Tables. Fortunately, they met the challenge and enabled us to come up with a comprehensive plan on how to automate the entire information process and even introduce two numerically controlled machines, one to drill the base for the meter and the other to produce the meter dial photographically. The project was completed in 1959. If implemented, we believed that the cost of production could be reduced by close to half and the time to produce and ship the meter from the date when an order was received to just a week

While this project was underway, Production Control Services had independently proceeded with the construction of Decision Table processing programs which were produced by CUC, then one of the few professional services companies in the world. This was called Tabsol for Tabular Solution and initially ran on an IBM 650. Tom Kananagh was the lead on that project. Tabsol would produce operating code to run on the IBM 650 based on the Condition and Action entries in the Stub and Columns of the Decision Table as long as those entries were in a program language.

The major problem that we had in using Decision Tables was the difficulty of creating the tables with pencil and paper. At the time there were no terminals, no time-sharing, and no way to either construct or modify the tables in any kind of automated fashion. Instead, any time we had to add a condition or an action or change some of the variables or even add a new column to recognize a new combination of conditions, we had to do it with pencil and paper. And as we went to even more complex logical decisions, particularly where they were concatenated (one table led to another and then to another), it became more and more difficult to keep track of and manage what we were doing without the direct aid of a computer.

The other major problem was that we found that while our team members had been comfortable thinking in table format (remember, these people were not programmers), when we brought this approach to programmers, we got a very cold reception; they were used to flow charts and that was the way that they preferred to represent logical decision processes.

It turns out that in parallel with what we were doing at GE in developing Decision Tables, Orren Y. Evans who was working for Sutherland & Co. for the US Air Force and then Hunt Foods and Industries independently came up with a similar approach in the late 1950s. This was a somewhat more elegant approach since he essentially put the entire conditional logic statement for each factor in the stub and therefore was able to just use Y, N or – in the cells of the table to indicate the value to be tested for. Similarly, in the Action section he only had to put an X into those cells where the action was to be carried out (as in the example that we used earlier in this Personal Account).

During the next 10 years or so, a number of organizations used Decision Tables. Additional Decision Table processing programs were produced for different computers, books were written on the subject and it was a topic of discussion at various computer conferences. But the end result has been that Decision Tables never became a standard process for the design of programs, except in some limited situations (more in system programs than in application programs) and has gradually completely lost the interest of the computer design community. There are still a few companies who offer Decision Table programs for those who want to use the technique, but they are not significant players. Nevertheless, I recently Googled “Decision Tables” and found that there were quite a few entries (including a Wikipedia article) with some ongoing interest in using them for complex decision processes. Maybe Decision tables will rise again.

I’ve always wondered if we had the PC capabilities of the late 1980s with the ability to build the tables on a screen, using some of the sophisticated capabilities introduced with VisiCalc and its successors like Lotus with 1-2-3 and Microsoft with Excel, whether there might have been a different story. I still occasionally use a Decision Table to help me completely analyze a complex problem with many input factors and a variety of outcomes or where I want to show someone the choices that they have when considering the combination of factors which could influence their decision.

But at the time that we created Decision Tables and started to use them it seemed that we had discovered the Rosetta Stone. We expected that Decision Tables would enable us to dramatically improve the quality and accuracy of program design and implementation. In addition, we hoped to finally be able to eliminate or drastically reduce the scourge of program bugs which were usually caused by logical errors or omissions. Oh well, it was a great dream.

