The Transition to a Configurator Based Design Process in an MTO+C+E Environment

Fred Ahrens
University of Cincinnati

Introduction

This case characterizes how an expert system was essentially rebuilt using a team of student interns. The configurator, an expert system that applies rules to specify component selections in a products bill of material had not been maintained for over 2 years and much of the design logic was either obsolete or incomplete. Moreover, the full time staff did not treat maintaining the application as their responsibility even though they were the primary beneficiaries. The resultant design process was only partially automated and informal at best. While redeveloping design software using student interns is not a novel concept, the scope of this effort and the way that the need for extensive domain knowledge was obviated does define a new boundary.

The dilemma of experienced worker apathy and the intense knowledge required was solved by hiring top talent interns. These interns were typically graduate students or international students whose ambition to exceed average performance levels was not an issue. In fact, getting an international assignment or into graduate school was itself a demonstration of the unusual initiative that this projects success so desperately needed.

However, attracting and retaining such talent required work of an unusually large scope. That most of these students subsequently went on to major multinational firms would be suggestive that this project did in fact live up to the value sell. This paper details, summarily, the technical elements of reverse engineering a configurator while mentioning that the majority of the result was from a team of students.

1. Bill of Material (BOM) Basics

A bill of material (BOM) is a key piece of business data as it defines the product composition and structure. The BOM is a vehicle for converting customer requirements into specific SKU's, planning materials, maintaining configuration control, documenting a design configuration and costing. It also explicitly establishes the parent-child relationships between components which has important production control implications.

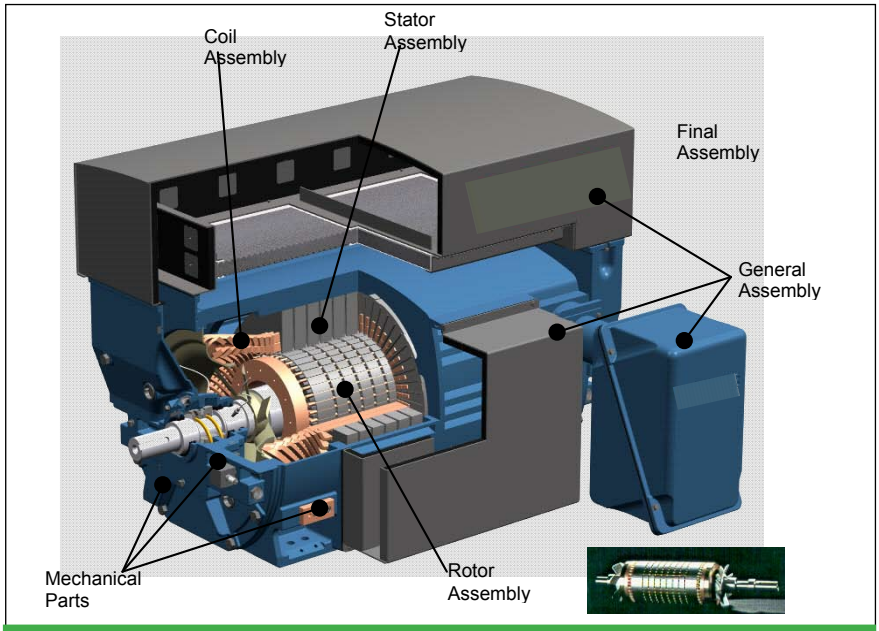


Figure 1. AC induction motor

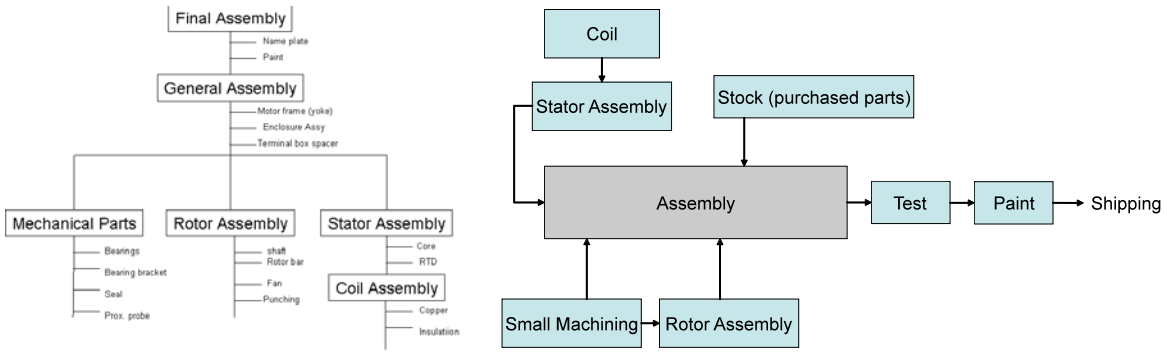
Bill of Material Structure: One Example

How a BOM is structured has important operational impacts. It is a primary determinant of how a product gets made. To examine one real case, let us consider a large electric motor (Figure 1) and its associated BOM structure at one company.

Rusk (1990) discusses the need for one BOM to serve the needs of the whole organization. The BOM is more then an itemized listing

Abstract

University student internships can be an important pre-professional experience for the student and be an immense benefit to an employer. Because of the findings of a 6-Sigma project to reduce engineering errors, a design configurator was to be rebuilt to include updated design information and expanded product coverage. Lacking available full time employees to support the project a team of mostly engineering graduate students were hired for intern positions and tasked with rebuilding this critical application A design configurator is an expert system that programmatically applies design rules to create a product configuration based on user inputs. This paper summarily reviews BOM basics and describes how a configurator was "reverse engineered" by university students by data mining BOM data previously generated in an Engineer-to-Order like (ETO-like) environment.



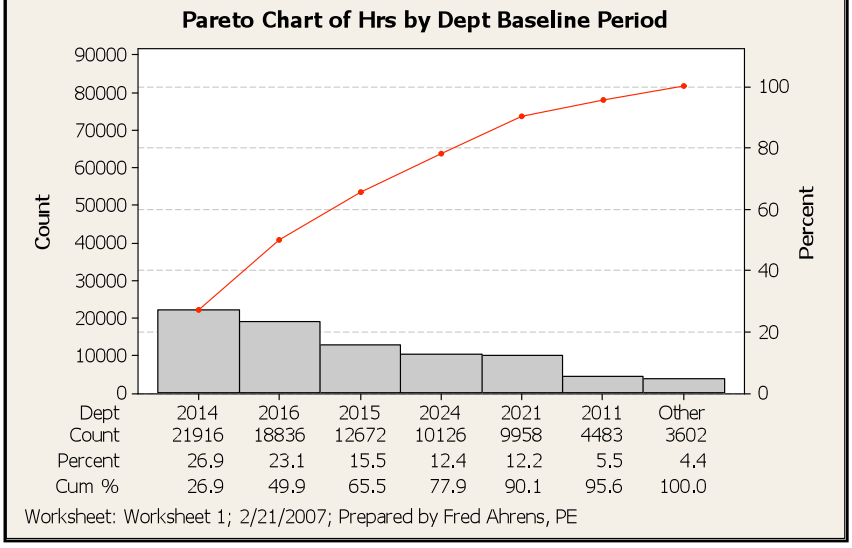
BOM	Built by Dept	Used by Dept
Final Asm	paint	shipping
General ASM	Assembly	test
Mech. Parts	pick from stock	Assembly
Stator ASM	stator	Assembly
Coil	coil	stator asm

Figure 2. BOM structure and shop floor organization

of parts, documents and text, but rather is a key piece of business data that drives manufacturing, purchasing and accounting, in other words the entire operational posture of the company. For the manufacturer of this motor, the production organization and related BOM structure is represented schematically in Figure 2.

The shop floor is organized along functional lines with assembly being the largest (Fig.3, 26.9% of hrs charged). With efforts ongoing to outsource most of the machining and a higher proportion of coils, the business is becoming more of an assembly operation. As can be seen in figure 2, the BOM roughly aligns to the production organization chart. For example, the rotor assembly BOM is produced entirely in the Rotor dept (dept. 16). As seen in fig. 1, the rotor assembly consists of the shaft, copper bars, punchings, fans and assorted hardware. If these parts were directly under the general assembly, the assembly department would receive rotor components in piece form and not be able to do anything with them (lack the tools, shaft press and machines).

The mechanical parts are a phantom and mostly come from the machining departments (dept 22 and 23) in addition to purchased parts. Phantoms are logical groupings of parts that are not kept in inventory but allow MRP to plan groupings of parts together. A phantom can be visualized as an example, a tune-up kit. One could specify all the plugs, wires and misc. items individually. But it is much more convenient to simply ask for an SKU that is a kitted collection of all the parts needed. The use of phantoms can be used to flatten the BOM structure Garwood (2001) and simplify the development of a



Dept	Desc
2011	FA and Test
2014	General Assy
2015	Coils
2016	Stator Assy
2021	Rotor Assy
2022	Large Machine
2023	Small Machine
2024	Punching
2028	Stator Cores
2041	Tool room

Figure 3. Labor hours charged

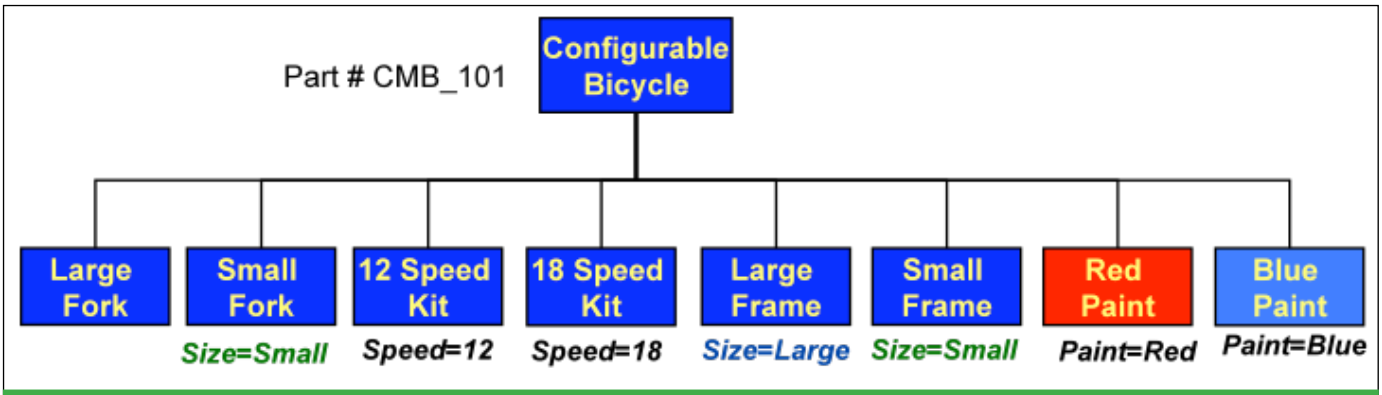


Figure 4. Single level Super BOM

configurator since only the top level part ("kit") need be specified. MRP "blows through" the phantom and plans the children level parts.

In contrast, the rotor is an assembly (not phantom), the parts mechanically interface, and the rotor can be kept in inventory and scheduled.

2. Super BOM: The Precursor to a Configurator

The traditional BOM generation method is to find a BOM on an order similar to the new one being worked and then add/delete components as required. This method is very labor intensive and prone to error. A superior alternative is to regenerate the BOM for each order using a configurator. The kernel of a configurable BOM is the Super BOM. Roughly defined, a super BOM contains all the materials required to manufacture all configurable variations of an assembly. Borrowing from database terminology, each component becomes a class object. A class (a type of part) has characteristics which have characteristic values. Figure 4 is a schematic of a configurable bicycle.

In this example, the bike will get one fork (large or small), one gear kit (12 speed or 18 speed) and one color (red or blue). A BOM can also be configured down through lower levels as well. Figure 5 is a schematic of a multi level Super BOM. In the multi level example the fork kit would comprise of one paint color (red or blue), one left fork (large or small) and one right fork (small or large). The configurator would apply design rules to the selection, however. For example, you can not have a large left fork and a small right fork. The selection logic trees would disallow the selection of infeasible options.

As an aside, it should be noted that a superbom can also be used in the role of a 'planning BOM'. That is, each possible part has a proportion assigned to it based on historical usage. For example, say the large fork was ordered 40% of the time while the small fork was ordered 60%. It would then be possible to plan at the finished goods level (the bicycle) and the demand for parts proportioned out. This planning BOM enables sales and operations planning at the aggregate level.

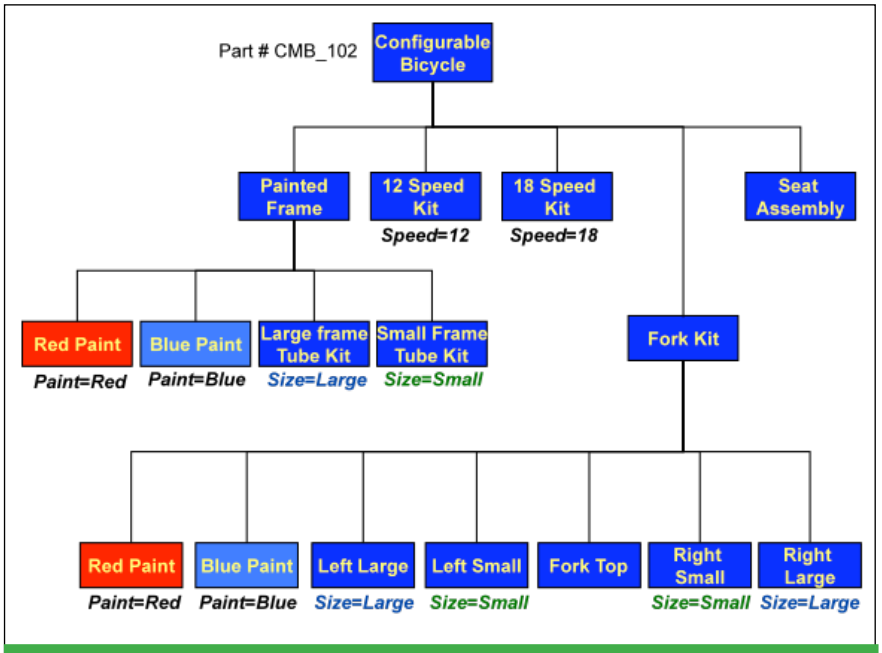


Figure 5. Multi level Super BOM

3. The Decision Tree, the Kernel of a Configurator

Each class object has characteristics and characteristic values. For example (Figure 6), a characteristic might be color and the possible value can be red, green or blue. Extending the bicycle example one could have selected a large red fork. The class object is a fork. The bike needs a fork which has the characteristics of color (possible values red, green or blue) and size (value large or small).

A decision logic tree is a set of decision points (options selected) at whose terminus lie a specific result. In a configurator, this would

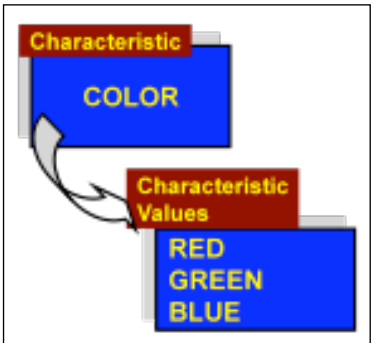


Figure 6. Characteristics and values

be the part with the desired characteristics and values to be included on the BOM.

Figure 7 is a simple decision tree for a bolt for a 6811 series motor. There are decision points for enclosure type (WPII, is shown in figure 1) and whether the bolt will be stainless steel hardware. Suppose one wanted a bolt for a 6811 WPII motor and made from stainless steel, then they would get part number CP611325375 (a specific part number) In this case, CP611325375 was an existing part number that had 'intelligence' built into the naming convention. Intelligent part numbers are not necessary in a configurator generated BOM, especially for lower level components (not directly ordered by the end user). If the enclosure were not WPI, WPII, CAZ or CAB then this bolt does not get specified. The output from the selection logic would be "Empty" because the part is not needed for that configuration of product.

With all the logic trees defined for each part, a hierarchy of trees is created. This ordering of trees is essentially the BOM structure reproduced as a series of trees that are executed in order to provide a part. A hierarchy will have all the objects that could be on that bill but contain only those that are actually output from a decision tree. Figure 8 shows the hierarchy for the Final Assembly (reference Figure 2 for details).

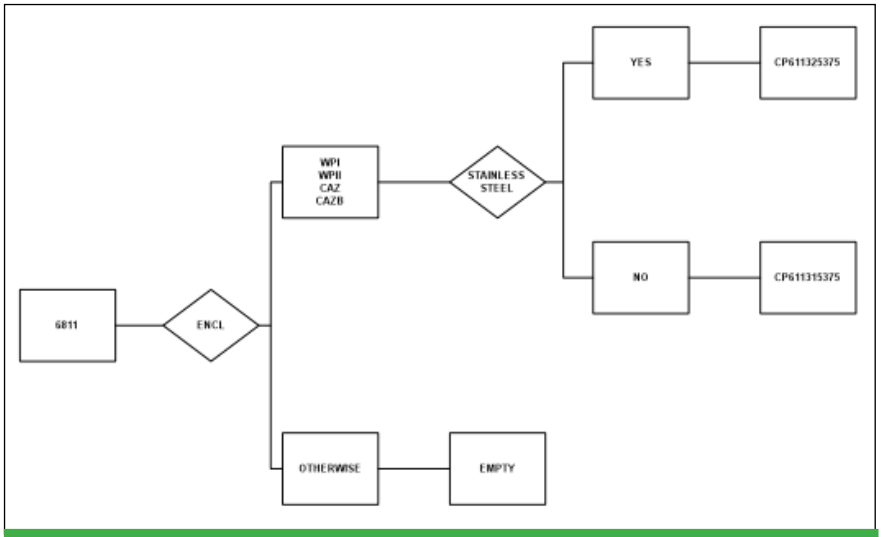


Figure 7. Decision logic tree to select a part

The configurator will execute the logic tree for each object in sequential order (top to bottom). Good logic design would have the decision variables as independent from each other. Let Y be the part output from a tree and {x} the column vector of input variables. One would try to avoid coupling, as below;

$Y_1=f(x_1,x_2,\ldots x_n)$
$Y_2=f(\{x\}, Y_1)$

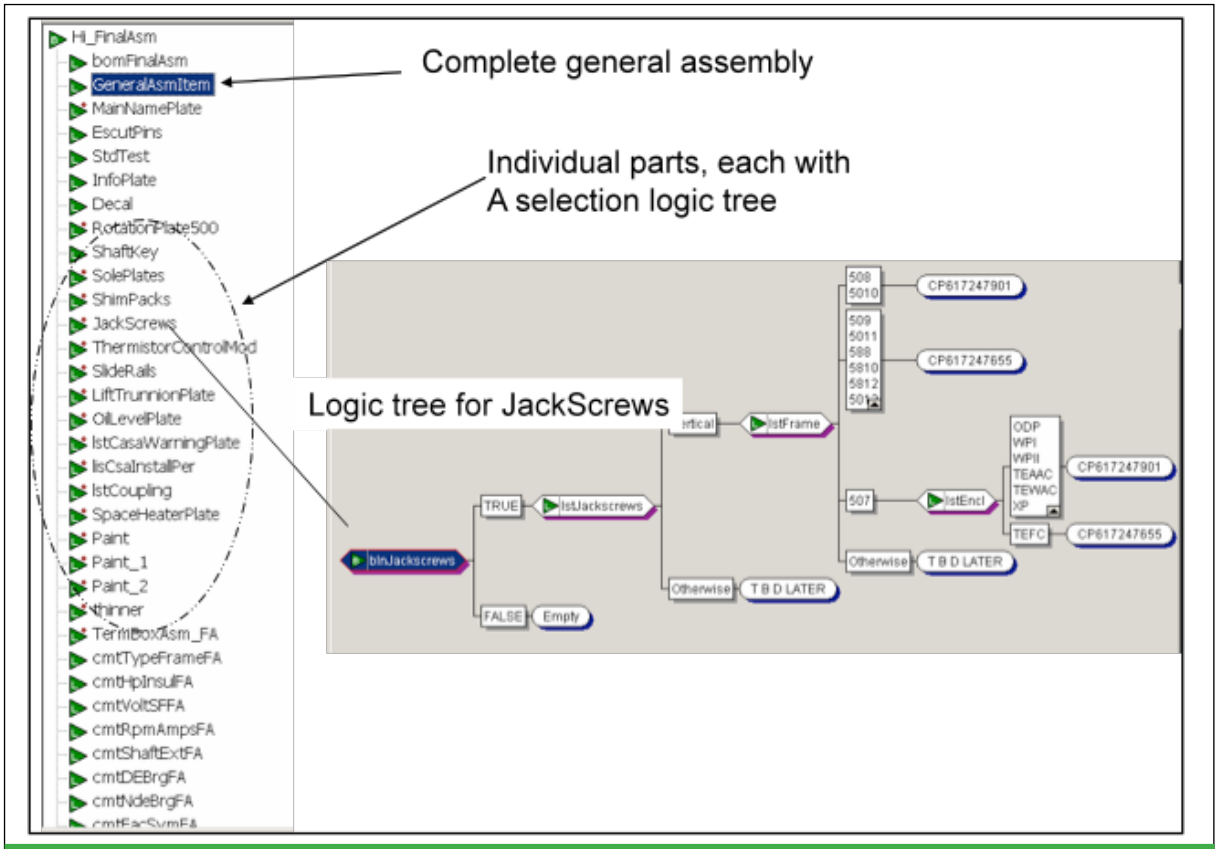


Figure 8. Hierarchy for final assembly seen in the development environment for the configurator

Here, the output Y2 is a function not only of the independent variables {x} but also the dependant variable Y1. Sometimes this coupling can not be avoided. For example, there may not be enough input variables to uniquely identify a part and, concurrently, that part may physically interfaces with another part. In the motor example, there is a bearing specified and associated with it is a specific housing that must be used. But, generally, it is usual for the inputs to be linearly independent.

4. Developing a Configurator by Reverse Engineering: A Case Study

The engineering department in the study company generated the majority of the BOM by the traditional Add/Delete model. Due to very crude data retrieval tools (in-house developed, no ongoing support) it was very often necessary to generate a bill from scratch. The design process for all practical purposes was an "ETO like" environment. There was a configurator in place that had been deployed but had not been upgraded, or even fully completed, in over three years. Consequently, there were large gaps in the logic, incorrect parts and overall obsolescence. Further, 46% percent (by sales order count) of the product had no representation in the configurator at all.

As a result of a 6-Sigma Black Belt defect reduction project Ahrens (2005), a dedicated follow on project to rebuild and expand the configurator was commissioned. The scope of the project was to complete the coverage of existing frames, update to present design offerings and add additional product offerings.

The products to be upgraded or added to the program had been fielded for years and there existed a tremendous volume of BOM data. The mandate of the project was not to re-design the product in the configurator but rather automate the design of the existing product with the maximum of available options.

Steps to Reverse Engineer a Configurator

1. Data mining

It was necessary to determine all of the components historically used and what frequency (relative to other part numbers for that part class). Prior to implementation of SAP R/3 (12/5/2005) the data was downloaded from the legacy ERP system (CINCOM) in the form of a flat text file. This was imported into MS-Access and a cross tab Query set up (Fig. 9).

DESCRIPTION	COMP PART #	Total Of Bill #	1	2	3
BRKT,BRG	55801391003	2			2
BRKT,BRG	55801391004	2		2	
BRKT,BRG	55801825024	1			1
BRKT,BRG	55802074023	1		1	
BRKT,BRG	55802074024	1			1
BRKT,BRG	55802074051	1		1	
BRKT,BRG	55802074052	1		1	
BRKT,BRG	55802627019	1			1
BRKT,BRG	58253201501	1	1		
BRKT,BRG	58253201502	1		1	
BRKT,BRG	58253202501	1	1		
BRKT,BRG	58253202502	1		1	
BRKT,BRG	58253202503	1	1		
BRKT,BRG	58453200001	4	2	2	
BRKT,BRG	58453200501	223	111	112	
BRKT,BRG	58453205501	2	1	1	
BRKT,BRG	58453209002	5	5		

Figure 9. Cross-tab Query of part usage profile

Figure 9 is a subset of the mechanical parts BOM. The object class is BRKT, BRG and the motor would get two (one on each side). The next column lists all the part numbers ever used for this object class. Next is the total count of BOM that it was used on. The column headers after (1, 2, …) are sequence numbers. Sequences numbers correspond to a bubble location on an assembly drawing, basically where the part went Mather (1982). For example, part number 58453200501 was used on 223 BOM (the next highest was only 4). Of this usage, 111 were used at sequence 1 and the remaining 112 at sequence two. For this example sequence 1 and 2 were the front and rear end of the motor respectively. So from this analysis we know what parts were historically used, where and how many times. What is not known is why one part would be used over another, the decision logic tree. This takes us to the second step, determining decision logic.

2. Identify decision variables

Data mining provided the part usage profiles and also enabled identification of paired parts (if part A then part B). Next, the high usage parts were evaluated (pulling drawings, specifications, etc) in the context of how they could be explained by possible decision variables. Figure 10 shows a schematic of a hardware item that showed up in the data analysis. This part would appear at sequence 36 whenever an air deflector (a part that directs airflow to the rotor) was specified. In this case, the analysis showed two parts could be specified; CP611325375 92.31% of the BOM or CP611315375 at 7.69%. Te part drawings showed the only difference to be ma-

terial; either stainless steel or galvanized. Thus, the logic tree had decision variables for air deflector and desired hardware material.

3. Validate proposed new logic tree

The last step was to validate the proposed new logic tree. Once the part had been identified in step 1 and a possible set of decision points determined in step 2, the drafting function was consulted on the proposed addition. These were the personnel who had been creating the BOM data set that was being reverse engineered. Basically, they were asked if this was really the part they would use given these selection criteria. For the lower level sequence numbers (higher in the bill structure) this input was important since these were the most major components of the motor. Further down would be more the hardware and ancillary components that would only show up based on what was specified higher in the bill. These parts generally required less validation. It was, however, far more difficult to data mine for them. This is because of the case of "multiple inheritance". A low level part like a bolt can have many parents and in the cross tab query would show up on multiple sequences.

5. Project Team Composition and Work Flow

The team that rebuilt the configurator comprised of one full-time employee (6-Sigma Black Belt) supported by a team of student coops. In this case, the latter were especially crucial since the project lead (author) was on the SAP implementation team and was not able to focus exclusively on this project. The coops team would typically comprise one undergraduate engineering student and two graduate level students on a 6 month rotation. The local university (Cincinnati) had a formal coop program of 3 months on then 3 months (one quarter school). This was incompatible with the needs of this project. It took 1 to 2 months just in training so the University's program was avoided and the team staffed directly by the author.

Some of the tasks associated with the project were very basic. For example, the company's part search program was wholly ineffective, it was merely a text field search program developed on the fly. To find parts to populate empty logic trees the undergrad coop was tasked with pulling thousands of drawings (1,500 copper bars and over 1,200 shaft drawings) and typing key characteristic values into a data table. For example, the copper bars would be defined by height, width, length and alloy. This provided a searchable table that the project team could use to select parts by specific characteristics. The shafts had over 30 different characteristics. This task took about 2 undergraduate coops 4 months to do. Interestingly, while these databases were developed as a project tool, the production personnel began to use it in lieu of the production system.

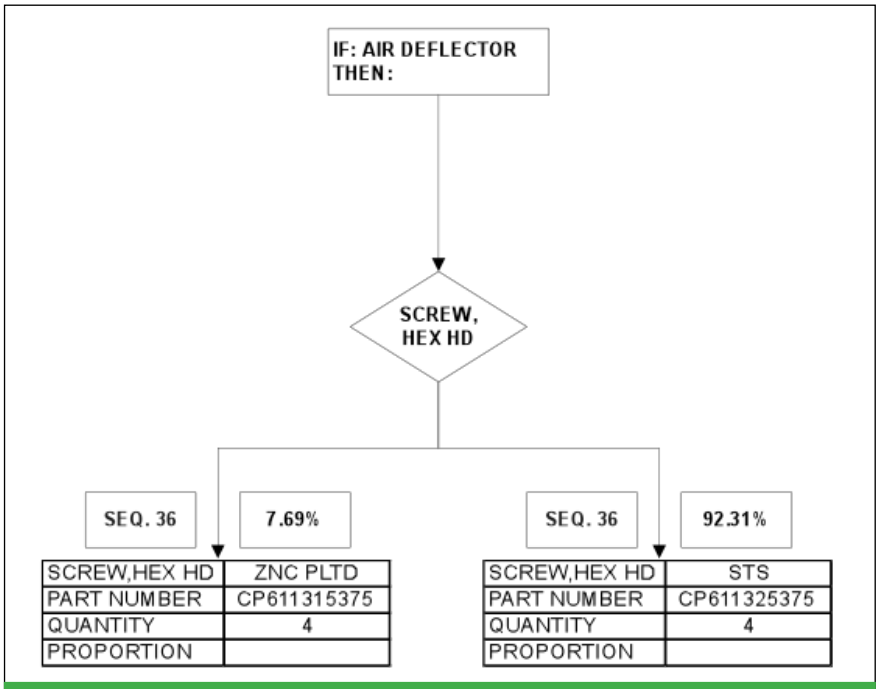


Figure 10 Characteristics of 2 parts identified at sequence 36

The other extreme were tasks that required great depth of knowledge. This included activities like data acquisition and analysis which was done mostly by the project lead and logic tree development which was done by the graduate students. It is interesting to note that 3 of the 4 new products added to the configurator were done by 1 graduate student!

6. Conclusion

This project was an novel for 2 reasons; that it was done mostly with interns and by the lack of relevant design guidelines. The configurator in a Make to Order with configuration and Engineering (MTO+C+E) company is an application of enterprise level importance; it is the primary design tool and source of automation that frees up costly resources for the "E" portion of the work (the engineering required for new products). The effort was made all the more difficult by the generic and fragmentary nature of many of the design standards. There was a strong "design as we go" mentality and thus

the configurations of product proliferated. The engineering organization was analogous to a "craft shop" were the output was the result of an individual who designed motors according to their accumulated experience and the tools at hand.

The interns were the key enabling factor for the success of this project. Figure 11 shows that the utilization of the configurator went form a baseline of around 35% of orders configurator generated to over 68%. This reflects the increasing capability of the program.

The quality of the student base, especially at the graduate level, appears to be quite high. It can be commented that graduate students at this point in their career have not lost their ambition and are still motivated to pursue challenging new assignments, in contrast to the bulk of the full time staff whose obverse characteristics resulted in the tool not being developed.

Migrating from an ETO like environment to more configuration by reverse engineering a design configurator produced tremendous value for the study company. Defects were reduced by over half and productivity was improved by multiples. This example is extensible to any organization exhibiting ETO like characteristics due to lack of design automation and data retrieval tools. However, one should note as a caution, the very reason for this project. An application of central business importance was allowed to become obsolete through lack of development. Further, an unrestrained 'customer first' philosophy allowed unprofitable proliferation of design variants. These further accelerated the obsolescence of the design automation program and were not profitable for the business. Since the time of this case, the author (project lead) and his replacement have left the company. It remains to be seen if the newly rebuilt configurator will continue to be maintained or again allowed to degrade. Ultimately, a design process is not only about the tools used in its execution but rather the business philosophy that drives it.

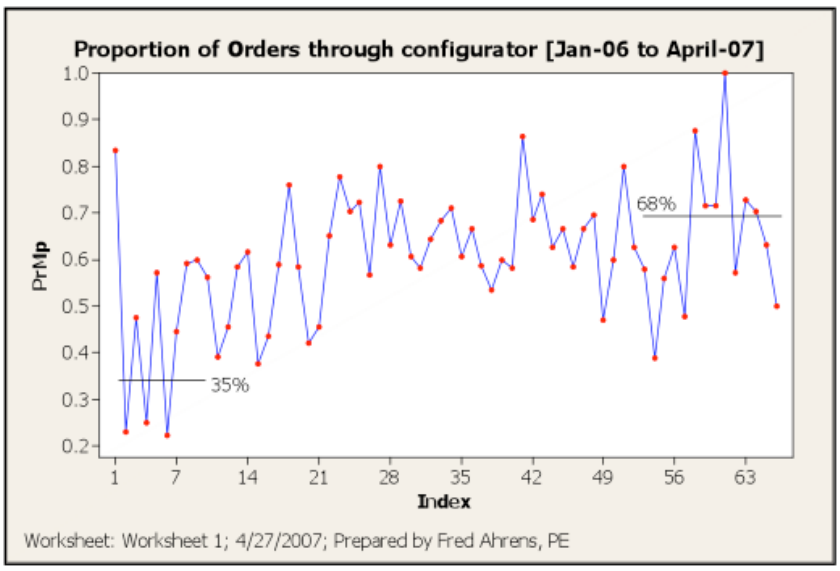


Figure 11. Configurator usage

References

Ahrens, F , "040080: PCD Norwood-Engineering Error Reduction", Siemens Energy & Automation, Inc, 2005

Garwood, D, "Bills of Material For a Lean Enterprise", Dogwood Publishing Company, Inc, 2004

Mather, H, "Bills of Materials Recipes & Formulations", Wright Publishing Company, Inc., 1982

Rusk PS, Barber, KD, "Structuring he Bills of Material for a Complex Make-to-Order Product (A Case Study), Engineering Costs and production Economics 15 (1988)

Fred Ahrens, PE is a graduate student in the Quantitative Analysis and Operations Management program at the University of Cincinnati College Of Business Administration. He will begin his PhD in the Information Systems department starting Fall Quarter 2009 at the University of Cincinnati.

His most recent industry role was as a Business Analyst with the Harris Products Group (div. of Lincoln Electric) where he was a SAP PP functional analyst and a member of the SAP implementation team for the Poland facility. Previously, at Siemens Energy & Automation , as a 6-Sigma Black Belt he led a team of student interns to reverse engineer an expert system and was also a member of the business unit SAP implementation team. Mr. Ahrens has also for several years taught courses at the technical and community college level in statistics, VB programming, thermodynamics, SPC and other engineering courses.

Mr. Ahrens also holds a Master of Science in Mechanical Engineering and an MBA.

ahrensf@uc.edu