Configuring and Manufacturing highly complex and extra long BOM using Oracle Configurator: A Customer Case Study

Eashwaran Pallipuram & Tarun Sharma

Keste

Introduction

New York Blower, a leading industrial fan-blower maker was looking to use OM and Configurator to design, configure and manufacture their products. The manufacturing bills and options were huge and complexity of streamlining the process appeared very cumbersome to accomplish. Keste partnered with NYB in solving this problem and helped realize a very feasible solution using the Oracle modules including Configurator. This case study will review the key approaches and implemented optimization techniques. Audience ROI: Insights into streamlining your complex MFG by leveraging Oracle OM and Configurator.

Challenges/Requirements

Various Challenges were faced for implementing an effective Configurator solution for New York Blower. The major ones are:

Segregating the order entry process from the engineering or manufacturing process: During an order placement process there could be situations when based upon various user selections in configurator, the engineering would decide which part number to use. It could also end up creating a new part number. Various techniques were used for capturing such configurations.

Placeholder Items were created in Item Master and Bill of Materials in the required Organization and Operating Unit. These placeholder items were then imported in Configurator and during a runtime session they would get driven under special conditions. These special conditions were nothing but the user selections requiring a part which can only be decided by engineering. So after the placeholder item gets driven in configurator and captured in order management, engineering would work towards replacing this item with an actual part number.

Engineering would then start working on creating a new part number for it. A batch program was developed which would based upon the feedback from Engineering go about creating new part numbers in the required organization and operating unit. If a new part number was not required because an existing part number could satisfy the requirements, this program would not generate a new part number. This program communicates with the legacy system on which engineering works and goes about creating new part numbers in Oracle Applications.

Another batch program would use the new part numbers created and replace it in place of the placeholder item. If a new part number was not created and an existing part number was to be used instead, it would do so. This replacement of part numbers would happen after the order has been booked and progressed.

Enable configuration of a Huge and Complex Bill Of Material: If a Bill is not created effectively no matter how efficient the configurator's performance is, the overall system gets affected. Thus we had to make sure that Bill of Material for the items to be implemented was created properly. The Bill Of Material for all the items was more or less created already. But after studying the Bill in detail we identified a few places where Bill was required to be modified. The techniques followed were:

The Bill should be as abstract as possible rather than being explicit. There were a few option classes which were placed underneath each of the three main ATO models. This is not only bad from maintenance point of view but from configurator point of view also. The size of BOM increases and rules need to be written 3 times for driving the same item. So such option classes were redefined underneath the PTO model directly.

A few option classes had required items underneath them, so their optional flag in BOM was unchecked thus making them required option classes. By doing so they did not flow in configurator reducing the size

of BOM Model instantiated in Configurator in runtime session. Also it did not require any rules to be written for driving them, thus reducing the number of rules.

In case of New York Blower the routing items were not defined at Option Class level. The reason for that being, items underneath the same option class could have different routings depending upon user selections. Thus separate routing items were created in the Item Master. These routing items were driven based upon user selections in configurator. Thus one of their major roadblocks of driving user selection based routings was achieved successfully.

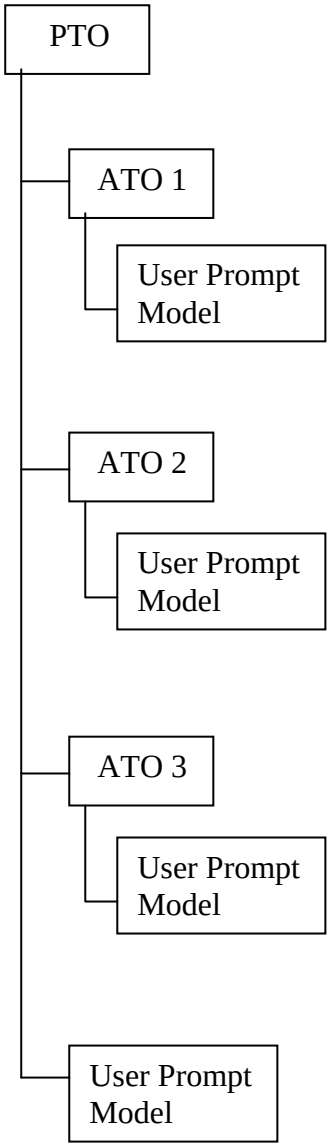


Figure: 1.0

As shown in the figure 1.0 above only one of the three ATO models should be instantiated. These ATO models have on an average 7 more ATO models underneath them and around 80 option classes underneath the main ATO models and 10 underneath each of the remaining 7 ATO models. Also some of the option classes could have around 700 items underneath them. Thus if all the three ATO models were instantiated at the runtime session, It majorly affected the performance of configurator. Thus we introduced configurator extensions to instantiate only one of the three ATO models. The model which gets instantiated would depend upon the user selections in User Prompt or Non – BOM Model. After making all the selections the user would press “Configure Fan” button which would trigger an ‘On Command’ event extension, which would finally instantiate the required ATO Model.

During prototype it was realized that if rules were written directly between the User Prompt Model and the ATO model, it would create situations wherein certain selections were unavailable because of BOM Data

issues. Basically incorrect BOM data would prevent the rule from making a unique selection thus disabling or excluding options from option feature which could drive that BOM Item. Thus by creating a design wherein User Prompt Model was made independent of the ATO models eliminated this issue. Also after user made all the selections, it was then copied to the NON BOM Model underneath the instantiated ATO. This NON BOM Model was nothing but the same ATO Model referenced underneath all the three ATO's shown in figure above. An extension would trigger at the same time when the required ATO model gets instantiated, which would copy all the user selections from the ATO independent NON BOM Model to the referenced NON BOM Model underneath the instantiated ATO. The rules were then written in between these user selections and the BOM Model driving the BOM Items.

The number of BOM Items and the validations required at BOM Item level and User Prompt level were huge. The best way of selecting BOM Items is by using compatibility rules, but these rules need to be mapped at each node level. Thus time taken for all the BOM Option classes increases substantially. Thus to reduce this time taken, rules were written in a statement format. These statements were written using various word editors like 'Notepad', 'Notepad++' or 'WordPad'. By using this method various rules could be copied across option classes and uploaded in a single file. This not only reduced the time frame required to write the rules but also reduced the time to upload them in configurator. This reduction in development time ended up in cost savings for the customer.

Pass Additional Information Downstream without Increasing the BOM/Order Size: New York Blower had a requirement of capturing various user inputs, which could then be used for numerous downstream processes. Majorly attribute mapping was the technique used to capture all these user selections required for the downstream processes.

User Inputs captured through attribute mapping technique was used for determining the price of various items. For doing this, pricing context and its segments was defined in application developer. These segments were nothing but various user selections whose unique combination would decide the future price of the item. The items were defined in a Price list and various pricing context along with attributes was attached to these items. Depending upon various attributes a specific price was defined in the price list. After user makes selection in the configurator and hits the finish button, a extension would then get trigerred which would then capture the attributes which have been mapped to the cz schema table. An SQL pricing routine would then compare these attributes with the attributes in price list and come up with a price for it which would get displayed in the sales order lines.

These attributes which would get captured could also be used to modify the description of various Star BOM items in a given configuration. These descriptions were required to be modified, so that they could be well understood on the shopfloor. For carrying out a description update, the items whose descriptions were required to be modified were attached a catalog and descriptive element indicating Description Update was attached to them. At the end of configuration runtime session when this item would get captured in CZ Schema a custom SQL Routine would then modify its description based upon the user selections. This description would consist of a concatenated string of user selections made in configurator session. A batch program was defined for executing this routine. Thus a business user from customer's end could carry out description update without having any knowledge of SQL.

Leveraging Customers Configurator Experience: Current Configurator Design ensures that customers are able to place orders for valid configurations. There are several possible scenarios which were taken into account to ensure the above.

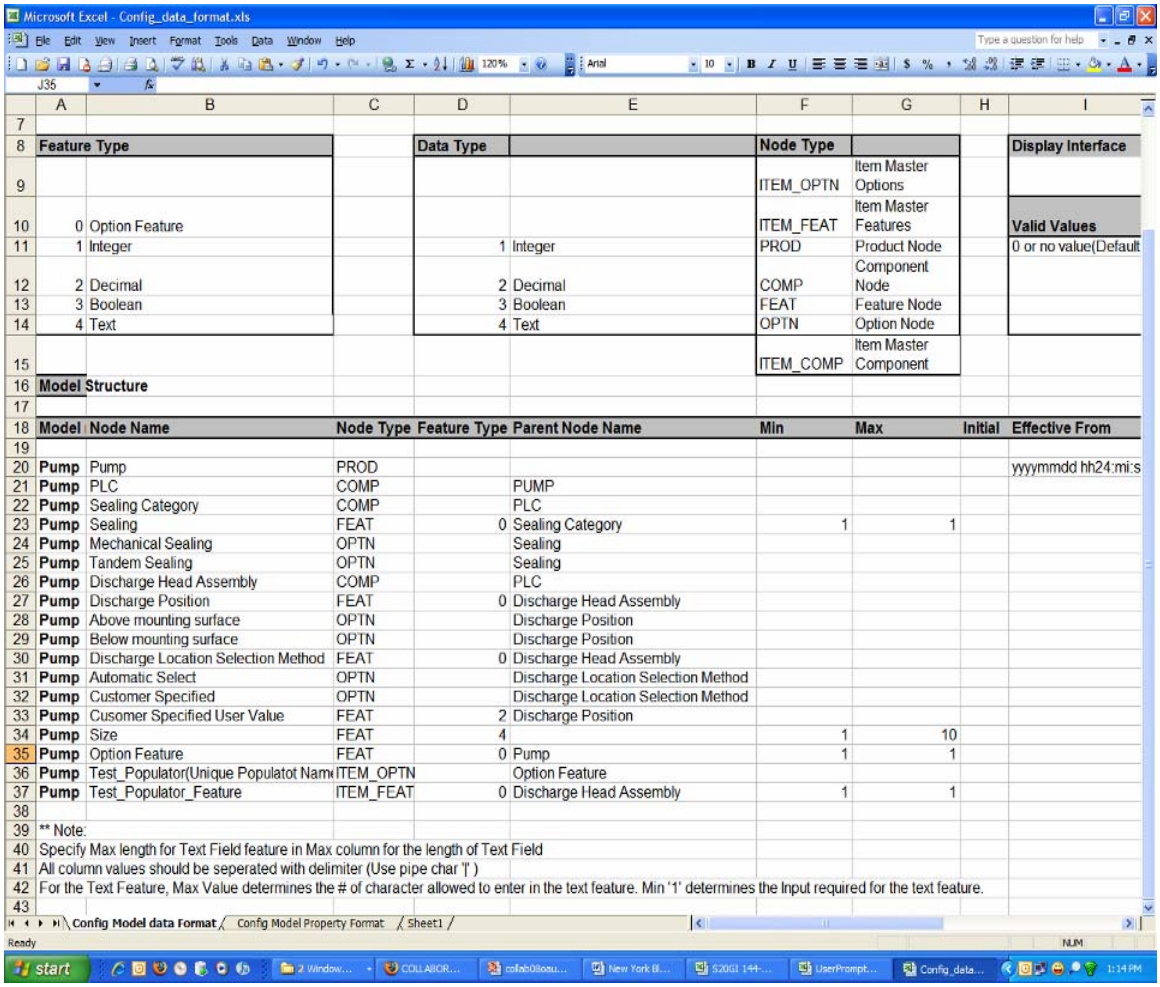
There are situations when BOM items have either incorrect Descriptive Element values, missing descriptive elements or two or more items have the same descriptive element values. In such cases one or more options get excluded from option features or valid user selection might indicate contradiction messages. To prevent any such situation NON BOM Models were created independently of ATO Models. i.e. there were no rules linking the NON BOM Model directly with any of the three ATO Items. The method adopted to achieve this functionality has been described earlier in the paper.

Runtime sessions when a rule is not able to select a unique item underneath an option class have been mentioned above. To handle such a scenario custom SQL scripts have been provided to the customer. These scripts are able to find out the items with duplicate descriptive element values, null descriptive element values or missing descriptive elements. Thus customer can then correct the data issues.

Configurator with a standard design approach across models reaffirms the reduction in time taken for a NYB user with minimum configurator experience to start using the application effectively. The user interface templates across various product families are almost the same thus order entry user does not need separate training for using the application. The design of BOM and validation rules is simplistic and comparable across models thus NYB's IT team would not have problem understanding multiple models. The templates used to capture requirements and test cases are also generic, thus they can be used for future implementations as well reducing the development time and effort.

Lessons Learned from Prototype: Prototype which was developed for NYB before going ahead with the implementation of various product families turned out to be a good learning exercise for both NYB and Keste. The lessons learned can be summarized in the points mentioned below:

User Prompt Model consisted of 10 components with on an average of 30 option features and 5 options underneath each of these option features. All the options underneath these option features had at least one property attached to them. Thus creating all these options and properties manually was going to be a very time consuming job. Thus a SQL script was developed which used the values entered in a standard pre-decided format in excel sheet and uploaded them in the configurator developer. This script was able to create hierarchical structure and create the entire NON BOM Model structure the way it was intended to.



Microsoft Excel - Config_data_format.xls

Feature Type		Data Type		Node Type		Display Interface
				ITEM_OPTN	Item Master Options	
0	Option Feature			ITEM_FEAT	Item Master Features	Valid Values
1	Integer	1	Integer	PROD	Product Node	0 or no value(Default
2	Decimal	2	Decimal	COMP	Component Node	
3	Boolean	3	Boolean	FEAT	Feature Node	
4	Text	4	Text	OPTN	Option Node	
				ITEM_COMP	Item Master Component	

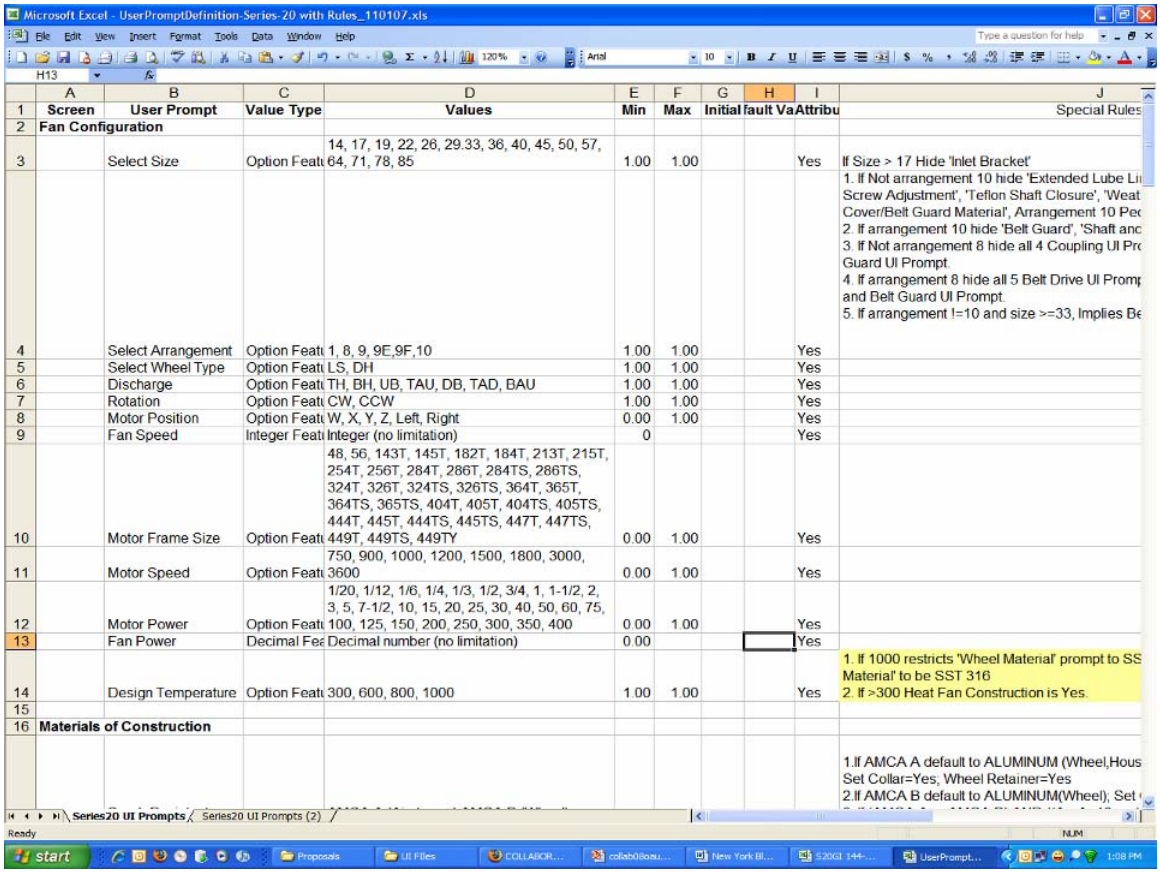
Model Structure

Model	Node Name	Node Type	Feature Type	Parent Node Name	Min	Max	Initial	Effective From
Pump	Pump	PROD						yyyymmdd hh24:mi:s
Pump	PLC	COMP		PUMP				
Pump	Sealing Category	COMP		PLC				
Pump	Sealing	FEAT	0	Sealing Category	1	1		
Pump	Mechanical Sealing	OPTN		Sealing				
Pump	Tandem Sealing	OPTN		Sealing				
Pump	Discharge Head Assembly	COMP		PLC				
Pump	Discharge Position	FEAT	0	Discharge Head Assembly				
Pump	Above mounting surface	OPTN		Discharge Position				
Pump	Below mounting surface	OPTN		Discharge Position				
Pump	Discharge Location Selection Method	FEAT	0	Discharge Head Assembly				
Pump	Automatic Select	OPTN		Discharge Location Selection Method				
Pump	Customer Specified	OPTN		Discharge Location Selection Method				
Pump	Cusomer Specified User Value	FEAT	2	Discharge Position				
Pump	Size	FEAT	4		1	10		
Pump	Option Feature	FEAT	0	Pump	1	1		
Pump	Test_Populator(Unique Populatot Nam	ITEM_OPTN		Option Feature				
Pump	Test_Populator_Feature	ITEM_FEAT	0	Discharge Head Assembly	1	1		

** Note:
Specify Max length for Text Field feature in Max column for the length of Text Field
All column values should be seperated with delimiter (Use pipe char '|')
For the Text Feature, Max Value determines the # of character allowed to enter in the text feature. Min '1' determines the Input required for the text feature.

Figure 1.1

The figure 1.1 shows a template created to capture the various details such as Model Name, Component, Option Feature and finally options. This template was used to create NON BOM Model. A similar template was used to create properties underneath these options.



	Screen	User Prompt	Value Type	Values	Min	Max	Initial	fault Va	Attribu	Special Rules
2	Fan Configuration									
3		Select Size	Option Feat	14, 17, 19, 22, 26, 29.33, 36, 40, 45, 50, 57, 64, 71, 78, 85	1.00	1.00			Yes	If Size > 17 Hide 'Inlet Bracket'
4		Select Arrangement	Option Feat	1, 8, 9, 9E,9F,10	1.00	1.00			Yes	1. If Not arrangement 10 hide 'Extended Lube Li Screw Adjustment', 'Teflon Shaft Closure', 'Weat Cover/Belt Guard Material', Arrangement 10 Pe 2. If arrangement 10 hide 'Belt Guard', 'Shaft anc 3. If Not arrangement 8 hide all 4 Coupling UI Pr Guard UI Prompt. 4. If arrangement 8 hide all 5 Belt Drive UI Promp and Belt Guard UI Prompt. 5. If arrangement !=10 and size >=33, Implies Be
5		Select Wheel Type	Option Feat	LS, DH	1.00	1.00			Yes	
6		Discharge	Option Feat	TH, BH, UB, TAU, DB, TAD, BAU	1.00	1.00			Yes	
7		Rotation	Option Feat	CW, CCW	1.00	1.00			Yes	
8		Motor Position	Option Feat	W, X, Y, Z, Left, Right	0.00	1.00			Yes	
9		Fan Speed	Integer Feat	Integer (no limitation)	0				Yes	
10		Motor Frame Size	Option Feat	48, 56, 143T, 145T, 182T, 184T, 213T, 215T, 254T, 256T, 284T, 286T, 284TS, 286TS, 324T, 326T, 324TS, 326TS, 364T, 365T, 364TS, 365TS, 404T, 405T, 404TS, 405TS, 444T, 445T, 444TS, 445TS, 447T, 447TS, 449T, 449TS, 449TY	0.00	1.00			Yes	
11		Motor Speed	Option Feat	750, 900, 1000, 1200, 1500, 1800, 3000, 3600	0.00	1.00			Yes	
12		Motor Power	Option Feat	1/20, 1/12, 1/6, 1/4, 1/3, 1/2, 3/4, 1, 1-1/2, 2, 3, 5, 7-1/2, 10, 15, 20, 25, 30, 40, 50, 60, 75, 100, 125, 150, 200, 250, 300, 350, 400	0.00	1.00			Yes	
13		Fan Power	Decimal Fea	Decimal number (no limitation)	0.00				Yes	
14		Design Temperature	Option Feat	300, 600, 800, 1000	1.00	1.00			Yes	1. If 1000 restricts 'Wheel Material' prompt to SS Material' to be SST 316 2. If >300 Heat Fan Construction is Yes.
15										
16	Materials of Construction									
										1.If AMCA A default to ALUMINUM (Wheel,Hous Set Collar=Yes, Wheel Retainer=Yes 2.If AMCA B default to ALUMINUM(Wheel); Set

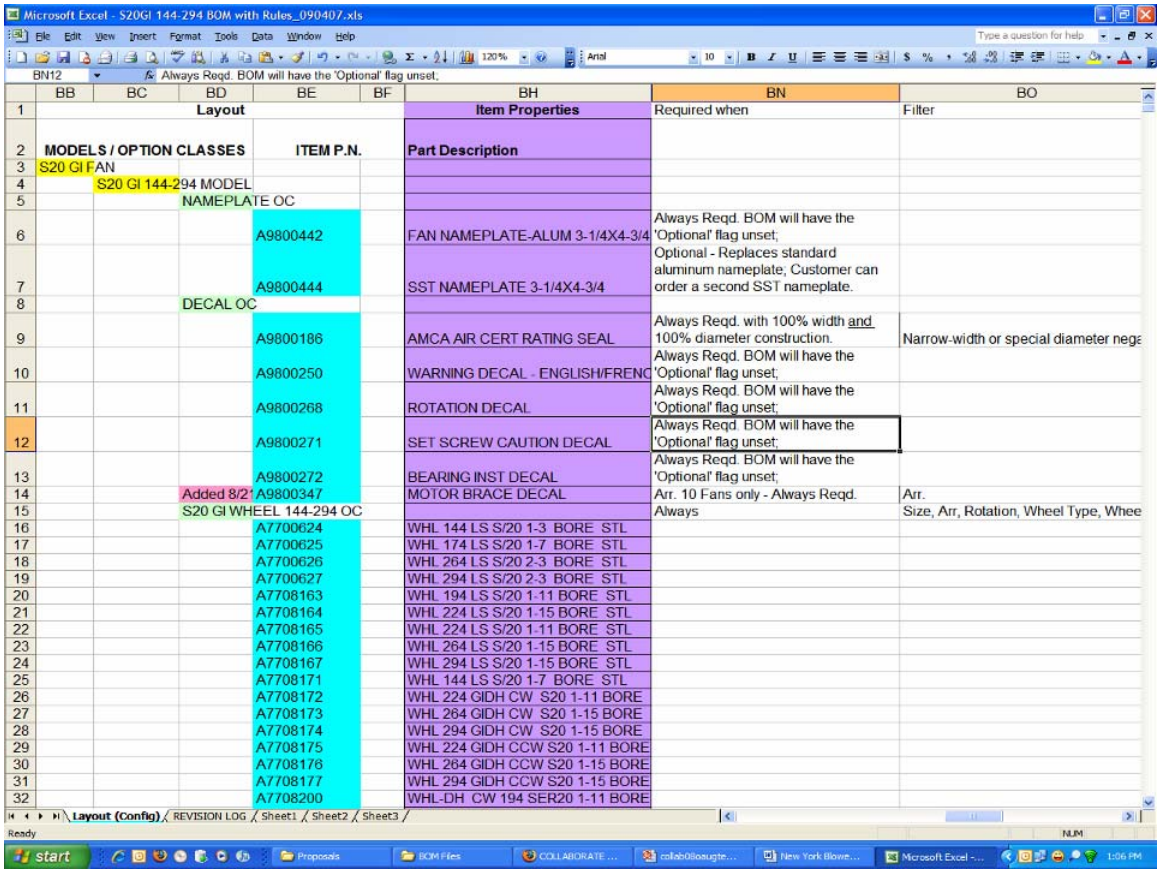
Figure 1.2

After understanding the business process and various product families of NYB, a standardized template was developed (shown in figure 1.2) to gather the requirements for various Components, Option Features and Options underneath User Input Prompt Model. This template being simple in nature was understood clearly by the customer. Thus the time taken for requirement gathering process could be reduced to only a couple of days from 10 days. As the customer got accustomed to the template requirement gathering turned out to be more fruitful, exhaustive and successful. These templates also made the customer understand the product more, thus improving the quality of communication between the customer and Keste.

The BOM Items and Routing Items were also captured in an excel spreadsheet. They were captured in a manner which showed their hierarchical structure in Oracle Applications as well as criteria's or rules which can be used to drive them. These templates made sure that all the BOM Items were considered and perfect criteria could be chosen to select a unique item. Figure 1.3 shows the template used for capturing the BOM Model structure with rules.

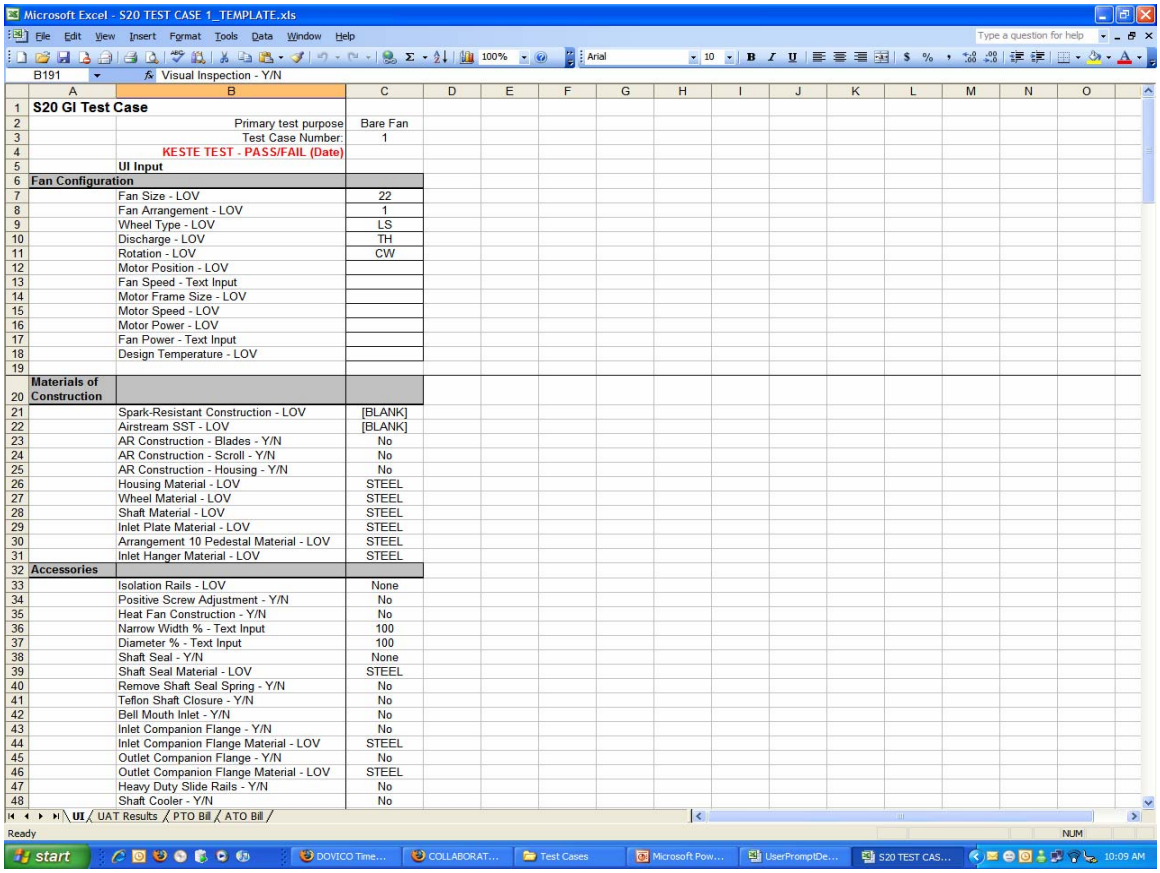
Test Cases with end user selections and expected Manufacturing BOM structure enabled the developers to weed out cases where configurations were not captured exactly as per expectations. The templates were not only simple but also enlisted the outcome very effectively. Figure 1.4 and 1.5 shows the templates.

Thus on the whole capturing of requirements as well as test cases in standardized templates helped the customer in increasing their understanding of the product. For Keste it helped to understand the requirements totally and clearly thus developing a quality solution for NYB.



	BB	BC	BD	BE	BF	BH	BN	BO
1			Layout			Item Properties	Required when	Filter
2		MODELS / OPTION CLASSES		ITEM P.N.		Part Description		
3	S20 GI FAN							
4		S20 GI 144-294 MODEL						
5			NAMEPLATE OC					
6				A9800442		FAN NAMEPLATE-ALUM 3-1/4X4-3/4	Always Reqd. BOM will have the 'Optional' flag unset;	
7				A9800444		SST NAMEPLATE 3-1/4X4-3/4	Optional - Replaces standard aluminum nameplate; Customer can order a second SST nameplate.	
8			DECAL OC					
9				A9800186		AMCA AIR CERT RATING SEAL	Always Reqd. with 100% width and 100% diameter construction.	Narrow-width or special diameter nega
10				A9800250		WARNING DECAL - ENGLISH/FRENC	Always Reqd. BOM will have the 'Optional' flag unset;	
11				A9800268		ROTATION DECAL	Always Reqd. BOM will have the 'Optional' flag unset;	
12				A9800271		SET SCREW CAUTION DECAL	Always Reqd. BOM will have the 'Optional' flag unset;	
13				A9800272		BEARING INST DECAL	Always Reqd. BOM will have the 'Optional' flag unset;	
14			Added 8/2	A9800347		MOTOR BRACE DECAL	Arr. 10 Fans only - Always Reqd.	Arr.
15			S20 GI WHEEL 144-294 OC				Always	Size, Arr, Rotation, Wheel Type, Whee
16				A7700624		WHL 144 LS S/20 1-3 BORE STL		
17				A7700625		WHL 174 LS S/20 1-7 BORE STL		
18				A7700626		WHL 264 LS S/20 2-3 BORE STL		
19				A7700627		WHL 294 LS S/20 2-3 BORE STL		
20				A7708163		WHL 194 LS S/20 1-11 BORE STL		
21				A7708164		WHL 224 LS S/20 1-15 BORE STL		
22				A7708165		WHL 224 LS S/20 1-11 BORE STL		
23				A7708166		WHL 264 LS S/20 1-15 BORE STL		
24				A7708167		WHL 294 LS S/20 1-15 BORE STL		
25				A7708171		WHL 144 LS S/20 1-7 BORE STL		
26				A7708172		WHL 224 GIDH CW S20 1-11 BORE		
27				A7708173		WHL 264 GIDH CW S20 1-15 BORE		
28				A7708174		WHL 294 GIDH CW S20 1-15 BORE		
29				A7708175		WHL 224 GIDH CCW S20 1-11 BORE		
30				A7708176		WHL 264 GIDH CCW S20 1-15 BORE		
31				A7708177		WHL 294 GIDH CCW S20 1-15 BORE		
32				A7708200		WHL-DH CW 194 SER20 1-11 BORE		

Figure 1.3



	A	B	C
1	S20 GI Test Case		
2		Primary test purpose	Bare Fan
3		Test Case Number:	1
4		KESTE TEST - PASS/FAIL (Date)	
5		UI Input	
6	Fan Configuration		
7		Fan Size - LOV	22
8		Fan Arrangement - LOV	1
9		Wheel Type - LOV	LS
10		Discharge - LOV	TH
11		Rotation - LOV	CW
12		Motor Position - LOV	
13		Fan Speed - Text Input	
14		Motor Frame Size - LOV	
15		Motor Speed - LOV	
16		Motor Power - LOV	
17		Fan Power - Text Input	
18		Design Temperature - LOV	
19			
20	Materials of Construction		
21		Spark-Resistant Construction - LOV	[BLANK]
22		Airstream SST - LOV	[BLANK]
23		AR Construction - Blades - Y/N	No
24		AR Construction - Scroll - Y/N	No
25		AR Construction - Housing - Y/N	No
26		Housing Material - LOV	STEEL
27		Wheel Material - LOV	STEEL
28		Shaft Material - LOV	STEEL
29		Inlet Plate Material - LOV	STEEL
30		Arrangement 10 Pedestal Material - LOV	STEEL
31		Inlet Hanger Material - LOV	STEEL
32	Accessories		
33		Isolation Rails - LOV	None
34		Positive Screw Adjustment - Y/N	No
35		Heat Fan Construction - Y/N	No
36		Narrow Width % - Text Input	100
37		Diameter % - Text Input	100
38		Shaft Seal - Y/N	None
39		Shaft Seal Material - LOV	STEEL
40		Remove Shaft Seal Spring - Y/N	No
41		Teflon Shaft Closure - Y/N	No
42		Bell Mouth Inlet - Y/N	No
43		Inlet Companion Flange - Y/N	No
44		Inlet Companion Flange Material - LOV	STEEL
45		Outlet Companion Flange - Y/N	No
46		Outlet Companion Flange Material - LOV	STEEL
47		Heavy Duty Slide Rails - Y/N	No
48		Shaft Cooler - Y/N	No

Figure 1.4

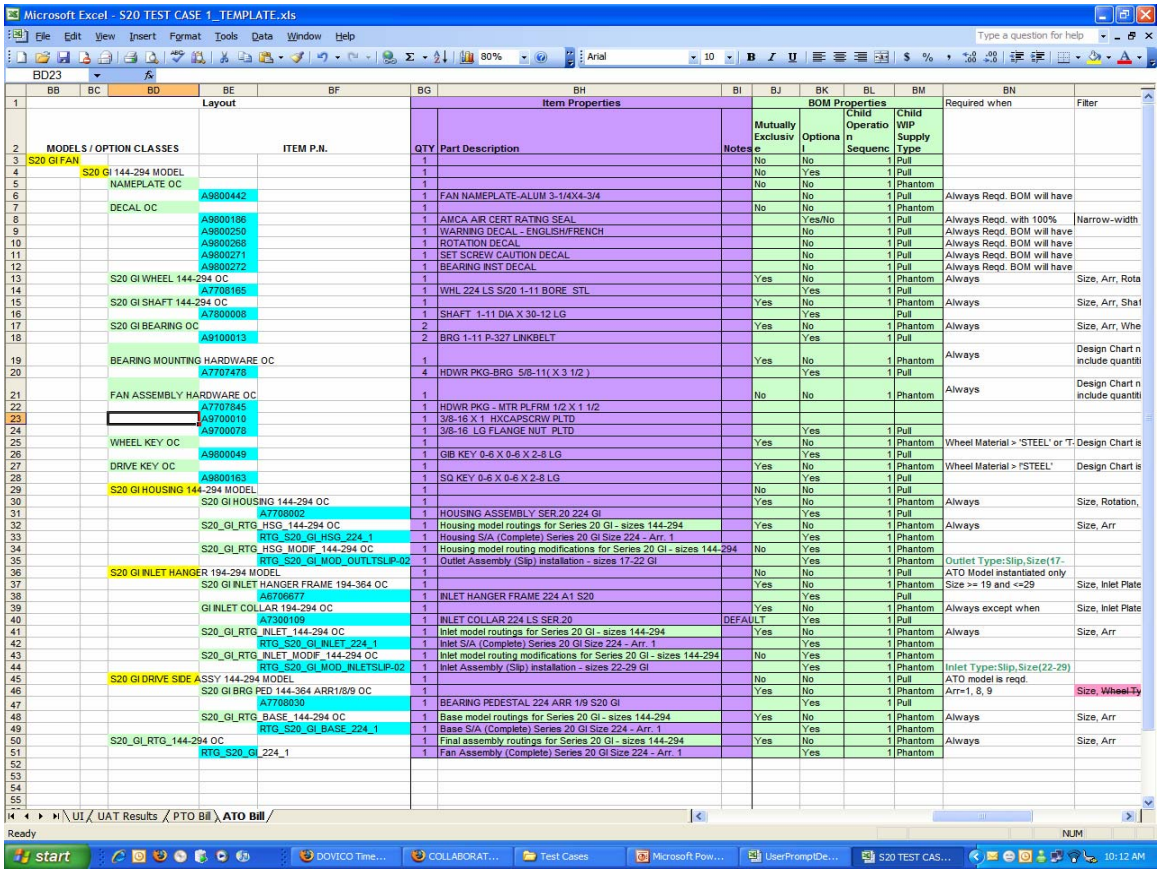


Figure 1.5

Some of the type of rules used during prototype was modified during actual implementations. The reason being that by modifying the rule type a noticeable improvement in performance was seen. For example keywords like CONSTRAIN was used instead of COMPTABILE for reducing the time taken for configurator in selecting items underneath a BOM Option Class with around 1000 items. Populators were used to populated BOM items of a specific item type underneath an Option Feature and use them for BOM Item selection. But Populators did not prove to be effective, as rules written in between populators and Options in Option Features would at times disable options or option features due to BOM data issues. Thus it was decided to do away with populators altogether. This led to developing a new technique of referencing the NON BOM Model underneath the instantiated ATO Model discussed in the paper earlier. This technique was much more effective and efficient and ensured capture of valid configurations.

Thus prototype proved to be a good test bed for trying out various techniques before actually rolling out the configurator solution for various product families.

Positive Impact of Configurator Solution on NYB's overall Enterprise Solution

The following tangible benefits were derived post implementation:

- Accurate Bill of Material
- Better Management of Inventory
- Better Scheduling and Forecasting
- Better Costing of items
- Various Reports on Sales across product families
- Intelligence Reports

Conclusion

We obtained almost all the benefits we planned for at the outset. The implementation methodology and the approach used did reduce the total implementation cost. The implementation time was also reduced due clarity of solution mapping, proper training and ownership taken by the implementation team. This implementation also led to improvement of the existing process of the organization. We have also learnt a great deal from this implementation, which can only benefit our own clients in their implementations.

