Development of Product Configurator for a Pressure Booster System

Rajesh R. Sampathkumar

1Senior Design Engineer, Industrial Products team, Larsen and Toubro Limited, Vadodara, India
Email: rajesh.sampathkumar@lnties.com

Abstract—This paper outlines the process of developing a product configurator for a pressure booster system. The ideas presented here were developed when developing a configurator in the CS-Enterprise package for a pressure booster system. This paper chronicles the design aspects, the configurator development planning, rule-based configurator development and testing and validation, including best practices developed during the process. Further work possible is also discussed.

Index Terms—product configuration, configuration management, product development

I. INTRODUCTION

Product configurators are becoming increasingly relevant in a world of complex product ontologies and differentiation. A rule-based product configurator was developed for a pressure booster system. The product's engineering definition was converted to a configuration definition using the CS Enterprise configuration tool. The features and options for this product came from a commercial specification which outlined the main features and options and differentiators of the product. This paper documents the best practices developed in the course of the project and can be applied to similar product development initiatives, where a variegated, fragmented product has to be designed and developed.

II. DESIGN PROCESS SUMMARY

The commercial specification for the product in question clearly defined the features and options of the product that would lead to its differentiation. The system itself was meant to handle flow rates between 160 GPM and 1600 GPM and was meant to handle pressures up to 275 PSI. The features of the product that caused product differentiation are as below:

- Station flow rate
- Suction and discharge pressures
- Operating voltage of the station
- Types of pumps used
- Number of pumps used
- Pump duty - and the number of pumps on standby
- Pump options - the pump used in the system can be selected using an algorithm for pump selection based on condition point
- Control system type - constant speed or variable speed controllers are available
- Pressure Maintenance pumps and controls
- Package options

Miscellaneous Options – control enclosure options, pressure switches, alarms, etc.

The design process followed during the project aimed at a modular approach to design, with a focus on reducing redundant structural weight and improving part commonality and interoperability. The piping and structure were together treated as a mechanical module, the control panel, its enclosures, wires and conduits were treated as a control module and finally, the pumps themselves were treated as parts (although in other designs they could be referred to as a pump module).

III. PRODUCT NOMENCLATURE

A product nomenclature was devised to indicate the features and options of the product configurations. The nomenclature is essentially alphanumeric code, with position-based feature definition and character based option definition on the nomenclature string. The mechanical module picked depends on the branch and manifold size feature options on the nomenclature. Similarly, the control module picked depends on control type selection, station operating voltage and other factors. The pump module similarly depends on the pump selection algorithm and its outputs. A features-based nomenclature for the booster system was developed as per the image shown below:

![Figure 1: Nomenclature System for Booster](image)

The nomenclature system is usually features-based but could, in cases of other products, be parameter-based or brand-based. In all cases, the objective of having a nomenclature for the configured product should be a unique designation, and the nomenclature should hence be sufficiently intuitive or otherwise uniform in its definition across all products in the organization. In the specific case of a booster package, the nomenclature will look similar to the scheme shown above, with modifications.

IV. DESIGN CONSIDERATIONS

Design considerations for the system and the pump are typically different. The pumps are designed as per the Hydraulic Institute (HI) codes and standards and pass...
standard tests for potability (NSF-61). However, the system has a separate set of engineering requirements as per the HI codes and standards. The design studies of the system primarily concerned the integrity of the structure and piping in steady state pumping conditions under different conditions of suction pressure and demand delivery pressure and flow rate. In order to address this design problem, the modular design included different branch sizes and manifold sizes as the primary differentiators between different modules.

Using velocity recommendations from the requirements on manufacturer specifications of check valves, the module manifold sizes and design were determined. Because of the different flow rates, several different branch sizes and a few different manifold sizes were used in the mechanical modules. The control module design was made of design considerations as per NEC 2008 using a variable speed controller, as the commercial specification called for this. The control module was also varied across the different configurations, and was influenced by the system type, the pump type, operating voltage, frequency and phase and other criteria. The design tables developed for mechanical and control modules evolved into design tables that were eventually used in the configurator logic.

V. CONFIGURATOR DEVELOPMENT

The development of the configurator started parallel to the process of developing the design drawings and the detail design of the product. The configurator was developed in a product configurator package called CS Enterprise (popularly known as eLogia). The configurator is written in the Java language and is web-based with a client-server model. It interfaces with SQL databases, JDBC and ODBC linked databases and is a rule-based configuration platform. In addition to pulling parts and assemblies based on features and options, the configurator has also to pull module or assembly numbers from the design tables. The eLogia framework allows for this by enabling communication with SQL tables.

A. Planning the Configurator Development Process

The configurator development process is typically done in conjunction with the design process. Several deliverables from the design process are necessary in order to streamline the configurator development, develop a solid architecture for the configurator which is unchanging, and which eventually has to be tested and validated little. Accordingly, a data mapping process was carried out to map the features and options of the product.

![Figure 2: Features Map of Booster System](image.png)

The product nomenclature was utilized to develop this data map, starting from the basic feature descriptions (as illustrated above) and the logical connections between materials and features, working through the configurator logic and arriving at the feature, infusion and material block and detail definition of the configurator. A flowchart of the logic of the configurator was planned in Microsoft Visio.

B. Developing the Configurator

The configurator development itself took place largely in a week-long project. The use of a Kaizen event to develop a configurator helped recruit a cross-functional team to perform rule writing in eLogia. The cross functional team consisted of engineers, supply chain specialists, manufacturing engineers and IT specialists. This cross functional team started at the project charter, worked through features definition, infusion rules to link blocks and tables, module-based table lookup rule writing, formulas, and other aspects of the product configurator. The primary assignments to the team during the project were:

- Features definition and feature rules: The product features are converted to features in the configurator
- Infusion rules: The product’s features and design tables can communicate with each other through infusion rules. The infusion rules was also used to write table look-up functionality into the configurator
- User Interface: The product configurator’s UI is where the features and options are displayed to the end user of the configurator.
- Edit and Compatibility rules: These rules are used to display errors, guidance and violations on the configurator user interface

C. Configurator testing

Configurator testing began with the development of a test plan. The development of a test plan depends on several aspects, chiefly the following:

- Overall complexity, features and options
- Valid configurations of the booster system
- Inputs entry sequence and methodology
- Configurator logic that aids input selection
- Table lookup information and the associated testing
- Validation of design data vis-à-vis configuration data
- Bill of materials testing process established at the site
- Meta data for parts and assemblies pulled in the process
- Configuration costing (done using feature-based logic)
- Connection prevalent from the configurator and the business system

These factors considered, the configurator testing process began in earnest after much of the rule writing for the rule based configurator was complete and table lookup logic was in place. The overall process of configurator testing involved the use of test sets, which were used to compare the engineering and manufacturing bills of materials. The configurator testing leveraged the offshore team who ensured
that issues were logged as per test sets. The use of a standard error log for the configurator enabled case-wise solutions to the configurator errors. The following flow chart explains the process of planning and developing the product configurator and testing it until implementation.

![Figure 3: Configurator Development Flowchart](image)

The final implementation of the configurator was arrived at by means of testing, validation and corrections to the rule sets, especially the infusion rules and table lookup functionality which was used to pull table information.

CONCLUSIONS AND FURTHER WORK

A web based product configurator was successfully developed using the CS Enterprise framework and tested for a booster system. The configurator’s logic was designed concurrently with the design of the booster system and the features and options were translated into a configured product definition starting from a commercial specification.

Best practices developed in this work can potentially be applied to other product configuration management frameworks. Further work in this area could include

- Integrating test plan development along with configurator system design
- Automated configurator testing methods
- Integrating data in the PLM environment used for the design process and the configuration management principles applied here.

REFERENCES

The author wishes to thank his team and the management of Larsen and Toubro who made this work possible.

[3] Tomi Männistö, Hannu Peltonen and Reijo Sulonen, View to Product Configuration Knowledge Modelling and Evolution, Helsinki University of Technology (IIA Research Centre and Laboratory of Information Processing Science)
[4] NSF, ANSI, NEC and HI standards references and manuals
[5] CS Enterprise documentation