

Mastering the Configuration of Variant-Rich Products by Modularization

Part 2: Approach and Methodology

Translation of Complexity Management Journal
Issue 05/2008, Pages 10 to 13
ISSN 1613-8155

Abstract:

After introducing principles and effects of product modularization in the first part of the article, we will now discuss the approach and necessary tools. The consequent use of simple function, modularization and interface matrices leads to a continuous optimization. As a result the configuration logic will remain transparent and lean. The ease of use is the key to success.

Authors of this Article

[Reinhard Große Wienker, Ph.D.](#)

[Hans R. Tanner, Ph.D., CEO, Schuh Complexity Management, Inc.](#)

Phone: +1 678 810 0885

hans.tanner@schuh-group.com

Schuh & Company
Complexity Management

Mastering the Configuration of Variant-Rich Products by Modularization

Part 2: Approach and Methodology

Reinhard Große Wienker Ph.D. / Hans R. Tanner, Ph.D.

After presenting basic methodologies and advantages of product modularization in order to develop a lean configuration logic in the first part of this article, we will discuss practical approaches and some simple tools in this second part. The deduction of product functionalities from goals and requirements will not be discussed any further, as it falls into the area of strategic product planning, which was already covered in CMJ 4/2008. The focus of this article is to identify the important steps towards a well structured lean configuration logic and to show how to deal with critical issues along the path.

Creating Visibility on a Functional Level

In the first part we pointed out the significance of the functional orientation. Looking at product functions is essential for product structuring. This neutral layer is well suited for the communication with the customer. Technical details the customer is not interested in knowing are hidden behind these product functions. The customer buys a product function and not so much the mechanical parts used to build and implement the product functions, therefore the functional layer can be used as a communication tool between sales and customer. By doing so, the major part of the configuration logic is already present in the product functions. In a first step, sales and product management need to define the market offering such as what should be offered or what are meaningful functional packages or sets. Allowed and forbidden functional combinations

therefore form the first level of the configuration logic. To visualize this step easily it is best to use a function matrix. The function matrix shows product functions and their dependencies and facilitates a basis for discussion about what rules will be necessary in the configuration logic. As an example, a convertible car cannot be sold in combination with a moon roof.

The Architecture Is Defined by Mapping Product Functions to Physical Components

The true challenge of product structuring is mapping product functions to their respective physical components. The key to success is to encapsulate the defined product functions or function packages into as many independent physical units as possible. To illustrate, we chose an example from the crane technology (see figure 1): The left side displays a list of product functions and the top reflects the modules and respective physical components including their available variants. In reality this means that the modules listed in the top row represent components of a product kit, each corresponding to a parts list.

Quite often such parts lists are used to define pre-assembly modules or supplied sub-systems. For the configuration it is important that alternative configuration options are modeled as well. For example, load ranges 3 and 4 in the shown example are implemented using one single head frame but two different cable winches. The modules include all variants. These variants are also available in the form of the module parts lists with the disadvantage that these parts lists carry partial redundancies. Cable and control switch kits 1 and 2 for example are available with different cable lengths but are identical otherwise. This is acceptable in the example because the cable and control switch kit is a subassembly provided by a supplier and always ordered as a complete kit. Therefore, it makes sense to keep two different modules. It might be necessary to introduce additional structural layers to the parts list logic in order to bundle all parts needed for the implementation of a specific function. Doing so helps to greatly reduce the complexity of configuration rules.

Modules	Frame		Cable Winch			Switch set				Carriage				...
	Head frame 1	Head frame 2	Cable Winch 1	Cable Winch 2	Cable Winch 3	Cable/Switch Set 1	Cable/Switch Set 2	Unlimited Radio Control	Limited Radio Control	Rigid Assembly Set	Railbound traveling	Smooth Rotor Disc	Coarse Rotor Disc	
Functions														
Lift load														
Load range 1	█		█											
Load range 2		█		█										
Load range 3					█									
Load range 4						█								
Move vertically														
Lift 1			█			█								
Lift 2				█			█							
Lift 3					█			█						
Move horizontally														
Stationary										█				
on Rails											█			
on Leveled floor												█		
on unlevelled floor													█	
Steering														
Wired						█	█							
Wireless								█						
Wireless w. limits									█					

Figure 1: Easy function allocation - Module to summarize the configuration logic

For the configuration logic this means that based on the configuration rules each selected product function will lead to the selection of exactly one variant of a specific module and associated module parts list. It is essential that the different functions are each represented by a unique parts list. The definition of these module parts lists is the responsibility of the engineer. He is in charge of the respective module and its variants as well as the technical optimization of all module variants. The configuration logic itself does not need to be detailed down to the parts level, it is sufficient to make use of the defined parts lists on the module and building block level.

As experience shows, the number of variants on the module layer can be kept relatively simple and it can be managed by a single person. The degree of modularization and the extent of the configuration logic represent a trade off (figure 2). There is no absolute optimum but an optimal range.

The matrix also helps to visualize the assignment of product functions and technical components. Since all module variants are listed, the matrix provides an easy to understand overview to all involved persons. It becomes easy to recognize dependencies that have an impact on the configuration logic without having to

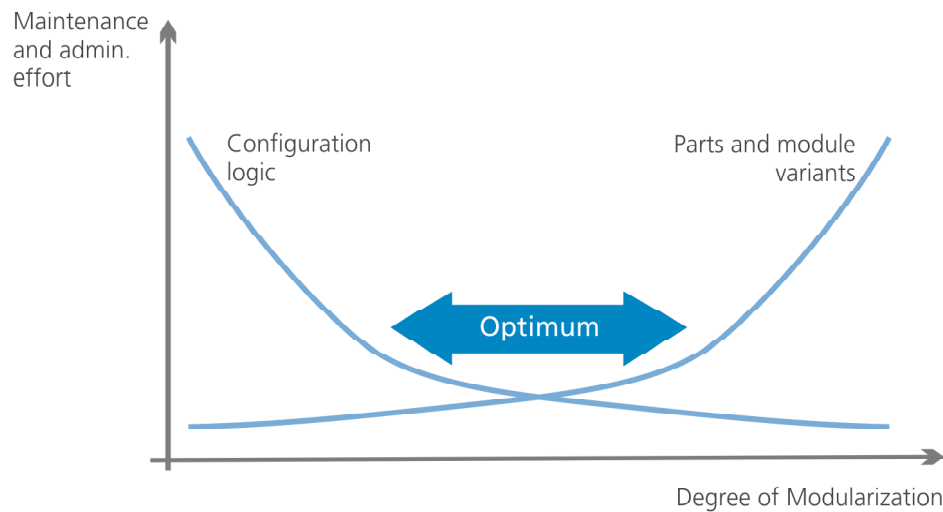


Figure 2: Efforts and Advantages of Modularization

dig into the code of the configurator software. If the product is developed further, it needs to be clarified whether new variants are to be added to the list or the change takes place within an existing variant only. Ideally, there is a 1:1 relationship between functions and module variants, represented by a diagonal marking on the matrix. In reality it should be the goal to come as close as possible to the theoretical ideal situation. The closer we come to the ideal situation, the simpler the resulting configuration logic will be.

This shows the importance of product structuring and architecture definition for the complexity of the resulting configurator logic. An easy to understand and shared basis therefore helps the discussion between sales and technical development very much.

Interfaces – Dependencies between Technical Components

The product as a whole consists of different physical modules which of course have interfaces to each other. Thereby, different types of interfaces can be distinguished:

- Physical interfaces, e.g. connectors or adaptor flanges
- Geometrical interfaces representing an available area
- Logical interfaces e.g. between two control units, that communicate using electrical signals to implement functionality

On the physical component level the interfaces between modules have a huge impact on the configuration logic and times and times again lead to the development and as a result maintenance of extensive logical rule sets. Therefore it is of greatest importance to identify dependencies and to keep the interfaces as stable as possible. While physical interfaces are obvious, it is much more difficult to not overlook geometrical or especially logical interfaces, particular when it comes to describing them in a configurator system. Functional requirements normally have an impact on the positioning of interfaces – and vice versa.

For example, the control function in our example (figure 1) may include several completely different variants: wired control switches or a much more sophisticated unit using radio communication. Both of these very different solutions require an interface to the crane, whereby the receipt of the control command is implemented through a radio receiver in one case, and with a simple cable connector in the other. After adjusting the signals, the technical implementation on the crane unit itself should be identical in order to ensure independence of the unit from the control system, thereby allowing for the development of more control alternatives in the future. Thus, the interface to the crane unit must lie behind the respective receiver unit and both solutions must serve the same interface on the crane unit. In case of changes to the interface, both control systems must be adapted at the same time. Achieving this is even more complicated as two

different technologies and therefore usually also two different suppliers are involved. Consequently, deviating versions of the same interface must be avoided.

Consequently, interfaces are to be properly managed. They must be treated as integral part of the product and be part of the product specific release process. The person responsible for a module must consider the impacts on the interfaces when developing the module. Changes must be communicated and functional tests of new modules must include both sides of the concerned interfaces. Furthermore, the impact of changes on configuration options must be visualized. The more unified and constant an interface is, the more stable the configuration logic will be.

As mentioned earlier, it is not always possible to design physical interfaces that are optimized for product structuring since not all characteristics of physical components follow logical principles. Therefore, it is possible that the visualization of interfaces in the form of an interface matrix will lead to reviewing the module structure itself. This is a valid result of the process and must be allowed as the definition of the product structure logic solely based on the functional layer is only a starting point for product structuring.

Modeling of Product Logic in the Configuration System

As explained in the above paragraphs, product structuring is indispensable for the development of a lean configuration logic. Simple tools can provide a valuable contribution to achieve that goal:

- Examinations from market and customer view on a functional level provide a certain level of abstraction that helps to avoid getting lost in technical details too early in the process
- Mapping of functions or function packages to technical components is a precondition for a simple configuration logic
- Clearly defined interfaces help to identify the impact of changes and need for clarification

- The use of simple tools like matrices help to identify the real need for discussion. Everybody involved in the process is able to use these tools and can contribute to finding solutions.

If these principles are observed and consequently pursued in an enterprise, product logic and related dependencies will remain known and logical. Product innovations and their impact will be faster analyzed and evaluated with regards to their impact on the overall product architecture. This provides for more secure developments and improved time to market because of enhanced knowledge about the results of changes. The process becomes more efficient and more time is available to plan the steps ahead.

Contact

Hans R. Tanner, Ph.D.,
Phone: +1 678 810 0885
hans.tanner@schuh-group.com