Comparison of Configuration Rule Visualizations Methods

Anna Tidstam, Johan Malmqvist

Comparison of configuration rule visualizations methods

Anna Tidstam and Johan Malmqvist

Chalmers University of Technology, Sweden

anna.tidstam@chalmers.se

Abstract. The aim of this paper is to compare matrix- and list-based configuration rule visualization methods in order to find the effects of using a certain visualization method. Especially, the visualizations' impact on readability and how the configuration rules are authored are studied. A case study was conducted at two automotive manufacturing companies. The research method included to observe users as they were authoring configuration rules, to study the visualization methods, and to interviews users of the PDM system visualizing the configuration rules. It was found that common aim for the users is to author easy to read configuration rules, which this paper shows is highly dependent on the visualization method. The conclusion is thereby that the choice of visualization method influence how the configuration rules are authored.

Keywords: configuration rules, visualization, authoring

1 Introduction

Configurable products are specified by a set of alternative product features. Cars are often configurable, with alternative product features including engine sizes and exterior colors. To assure that the specification of a configured product belongs to the developed product offering, there are logic expressions. An example of those logic expressions is that the *exterior color red* is not offered with *engine size 1.8 liters*. Such logic expressions are in this paper called *configuration rules* following the definitions in [1, 2]. Moreover, configuration rules validate the correctness are according to [1] rules for validating the correctness of customer orders and for guiding the actual assembly of these orders. A similar explanation is used in [2], where configuration rules are the rules for how the components can be combined to form a product. One of the main difficulties with large sets of configuration rules is the maintenance, in other words modifying the configuration rules following a product offering modification [3]. To be able to maintain the configuration rules, there is a user interface with a visualization of configuration rules. A visualization of configuration rules is a representation of configuration rules for communication of configuration rules to a user. The visualization of configuration rules is a kind of *information visualization*. The aim of information visualization is to create an effective representation of the information model and the information contained therein [4]. Effectiveness means that the user as quickly as possible gets an overview of the information. The effectiveness is chal-

lenged for the configuration rule visualization as there may be hundreds of thousands of configuration rules.

Some researchers have studied different methods for visualization of configuration rules, for example matrices: K- and V-matrix [5], and trees: Product Family Master Plan [6] and the Variant Tree [7]. The mentioned previous research is however not empirical studies with configuration rules from the automotive industry. Earlier studies have shown benefits of matrix-based methods for configuration visualization [8]. For example, a matrix organizes data into a structured overview that is difficult to achieve with a list of text strings. The advantages and disadvantages specific to configuration rules have however not been fully studied. There is a list-based configuration rules visualization from an automotive company in [9], but unfortunately this visualization is not in detail described nor evaluated. If the configuration rules cannot be effectively visualized, an alternative approach is to change representation as for example mathematical equations on design parameters which was suggested in [9].

The authoring, i.e. the creation and modification, of configuration rules depends on the user: some users prefer a set of few but long (many product features) configuration rules, while others prefer a set of many but short (few product features) configuration rules [8]. The main aim with this paper is to compare configuration rule visualization methods in order to find the effects of using a certain visualization method, especially the impact on how the configuration rules are authored. A second aim is to find advantages and disadvantages for the visualization methods at the two automotive manufacturing companies. The following research questions have been addressed:

RQ1: What are the characteristics of the authoring methods for configuration rules that may be derived from the use of either matrix- or list-based visualization methods?

RQ2: Which visualization method is most suitable in the cases of:
(a) high number of configuration rules
(b) high number of product features in each configuration rule?

The remaining of the paper is organized as follows: In Section 2 the research method is described, Section 3 presents the results, Section 4 discusses the results and in Section 5 the conclusions are stated. Finally, in Section 6 future work is proposed.

2 Research Method

This paper studies the configuration rules visualization methods at two major European automotive manufacturing companies, henceforth called Alpha and Beta.

The research study was conducted during a 3 months stay at Beta in the team of configuration rules specialists, which was an adequate time-frame for revealing the process of authoring configuration rules as well as getting acquainted with the PDM backbone systems at Beta. During this time, several workshops were arranged with Alpha, which gave the possibility to compare both authoring and visualization methods. As seen in Fig. 1, the research process was structured into three phases: analysis, testing and evaluation. In the analysis phase, the authoring and visualization of con-

figuration rules were described. Then in the testing phase, the configuration rules at Beta were tested with Alpha's visualization method. This comparison was later evaluated during the evaluation phase by interviewing three configuration rules specialists with various years of experience (4-30 years) at Beta.

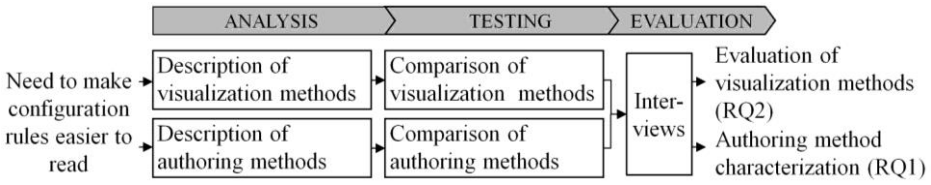

Fig. 1. Overview of applied research methods.

3 Results

The result section is divided into three sections: 3.1 Analysis phase, 3.2 Testing phase and 3.3 Evaluation phase.

3.1 Analysis phase

In this section, a use case scenario is used for describing authoring and visualization methods at Alpha and Beta. The use case describes the authoring of configuration rules for a new feature variant *a3*, e.g. *orange* exterior color. New feature variants do not have any configuration rules initially, and the use case is to author the required configuration rules. Feature variant *a3* belongs to feature family *A* (e.g. *exterior color*), which already contains the feature variants *a1* (e.g. *red*) and *a2* (e.g. *green*). Both Alpha and Beta use codes for feature families and feature variants.

Authoring and visualizing configuration rules at Alpha. The purpose of this section is to show the principles for how to author configuration rules at Alpha (using their matrix-based visualization method), which is then followed by another section for Beta (using their list-based visualization method).

The authoring process at Alpha consists of three steps: Step 1 is to visualize existing configuration rules, Step 2 is to update the feature family relationship, and Step 3 is to assign configuration rule values. The authoring of configuration rules is thereby prepared in Step 1, and actually performed in Step 2 and 3.

Step 1 is to visualize the configuration rule matrix for feature family *A,* as shown in Fig. 2. The systematic configuration rule value (S) indicates that only particular combinations of feature variants are allowed. The only allowing configuration rule for *a1* is IF *a1* THEN *(m1* AND *n1*), e.g. IF *red* THEN (*engine V6* AND *sport package*). The minus signs restrict feature variant combinations, e.g. NOT (*a1* AND *m2* AND *n1).*

			THEN	
		Feature families	**Feature variant combinations**	
		M	*m1*	*m2*
		N	*n1*	*n1*
	Feature family	**Feature variant**		
IF	*A*	*a1*	S	−
	A	*a2*	−	S
	A	*a3*	−	−

S = systematic and − = not allowed

Fig. 2. An example of Alpha's configuration rule matrix. This visualization is studied during the first step in authoring configuration rules at Alpha.

Step 2 is to update, if necessary, feature family *A*'s relation, see Table 1. Feature family relations are the source data for the generation of any configuration rule matrix. In this use case scenario, it is assumed that feature family *P* is required because of the addition of feature variant *a3*.

Table 1. Feature family *A* with its relation left) before update with *P* and right) after the update.

Before update	
IF	THEN
Feature family	**Feature families**
A	*M, N*

After update	
IF	THEN
Feature family	**Feature families**
A	*M, N, P*

Step 3 is to assign configuration rule values, see Fig. 3. The new optional configuration rule value (O) for *a3* is the feature variant combination *m2* AND *n1*. Among optional configuration rule values, there is a default configuration value (D) which is the preferred feature variant combination. The configuration rule value for *a2* also needs to be updated as it is no longer the only allowed feature variant for combination *m2* AND *n1*.

			THEN			
		Feature families	**Feature variant combinations**			
		M	*m1*	*m2*	*m1*	*m2*
		N	*n1*	*n1*	*n1*	*n1*
		P	*p1*	*p1*	*p2*	*p2*
	Feature family	**Feature variant**				
IF	*A*	*a1*	S	S	−	−
	A	*a2*	−	−	D	D
	A	*a3*	−	−	O	O

S = systematic, D = default, O = optional and − = not allowed

Fig. 3. Configuration rule matrix with updated configuration rule values for feature variant *a3*, which is done during the third and last step in authoring configuration rules at Alpha.

The authoring and visualization of configuration rules at Alpha have now been described, and now it will be shown hot the same operation is performed at Beta.

Authoring and visualizing configuration rules at Beta. The same use case scenario is used when described in the authoring method at Beta: Step 1 is to visualize the existing configuration rules, and Step 2 is to author the configuration rules.

Step 1 in the use case is to visualize configuration rules for *a1* and *a2*, see Table 2. The purpose of Step 1 is to find frequently used feature families, for example all engine sizes can have configuration rules with gear box alternatives. The configuration rules at Beta may include both positive (AND, OR) and negative operators (NOT).

Table 2. List-based visualization of configuration rules studied for finding frequently used feature families that could be suitable to use for authoring of configuration rules for *a3*.

IF	THEN
Feature variant	**Feature variant combinations**
a1	*m1* AND *n1* AND NOT(*a2*)
a2	*m2* AND *n1* AND NOT(*a1*)

Step 2 is to author the configuration rule for *a3*, see Table 3. As the customer may choose between *a2* and *a3*, also the configuration rule for *a2* has to be re-formulated. Notice that there is no difference between default and optional feature variant combinations at Beta, as this distinction is managed the separate sales system.

Table 3. List-based visualization of configuration rules with new configuration rules for *a3*, and updated configuration rule for *a2*.

IF	THEN
Feature variant	**Feature variant combinations**
a1	*m1* AND *n1* AND NOT(*a2* OR *a3*)
a2	*m2* AND *n1*
a3	*m2* AND *n1*

The analysis phase is thereby ended and the next section will describe the testing phase.

3.2 Testing phase

The purpose of the testing phase is to find relevant measures for the matrix- and the list-based visualization methods and thereby compare those using three real case examples. The real case examples were selected based on the users' at Beta interest in trying challenging examples, for instance configuration rules with extremely many configuration rules or extremely many feature variants within the configuration rules. One of the real case examples visualized with Alpha's matrix-based method is shown in Fig. 3. Two measurements have been established: a size measurement *G* in the growth direction (number of columns for the matrix, number of rows for the list), and a combinatory difficulty measurement *C* which is the number of feature variants combined in a configuration rule. The number of columns in Fig. 4 is 10, but note that the number of columns is here reduced, as for example the empty cells for the row of

feature family *P* could be filled with any of *p1* or *p2*. The combinatory difficulty measurement is equal to the number of rows with feature variants in the matrix, which is 9 in Fig. 4. The number of configuration rules, could be counted as a range between 1-10, as the configuration rule values could be combined with the OR-operator.

			Feature family	THEN										Combinatory difficulty measurement:
				Feature variant combinations										
			M	*m1*	*m1*	*m1*	*m1*	*m1*	*m1*	*m1*	*m1*	*m1*	*m1*	1 (*m1*)
			N	*n3*	*n3*	*n1*	*n1*	*n4*	*n4*	*n4*	*n2*	*n2*	*n2*	2 (*n1 – n4*)
			O	*o1*	*o3*	*o2*	*o2*	*o4*	*o4*	*o4*	*o5*	*o5*	*o6*	3 (*o1 – o6*)
			P					*p1*		*p2*	*p2*	*p1*		4 (*p1*, *p2*)
			Q					*q2*						5 (*q2*)
			R						*r1*					6 (*r1*)
			S			*s2*	*s1*							7 (*s1, s2*)
	Feature family	**Feature variant**	*T*		*t1*		*t1*							8 (*t1*)
IF	*A*	*a1*		O	O	O	O	O	O	O	O	O	*O*	9 (*a1*)
				1	2	3	4	5	6	7	8	9	10	

Growth measurement

S = systematic, D = default, O = optional and − = not allowed

Fig. 4. Alpha's configuration rule matrix with the measurements relevant for the tests (number of feature variants, number of feature variant combinations) marked out.

The same example with Beta's original authoring of configuration rule list is shown in Fig. 5. The number of rows are 5 for measurement *G,* and the combinatory difficulty measurement *C* (number of feature variants in a configuration rule) is a range 3 to 12.

IF	THEN	Growth meas-urement	Combinatory difficulty measurement
Feature variant	**Feature variant combinations**		
a1	(*n1* AND *s2*) OR (*n1* AND *s1* AND *t1*) OR (*n2* AND *o5*) OR *n4* AND *p2*	1	10 *(a1, n1, s2, n1, s1, t1, n2, o5, n4, p2)*
a1	*n4* AND *r1*	2	3 *(a1, n4, r1)*
a1	(*n1* AND *s1* AND *t1*) OR (*n4* AND *o4*)	3	6 *(a1, n1, s1, t1, n4, o4)*
a1	((*n1* AND *s2*) OR (*n1* AND *s1* AND *t1*) OR (*n2* AND (*o5* OR *o6*) OR *n4*) AND *q1*) AND *p1*	4	12 *(n1, s2, n1, s1, t1, n2, o5, o6, n4, q1, p1)*
a1	*n3* AND (*o1* OR (*o3* AND *t1*))	5	*5 (n3, o1, o3, t1)*

Fig. 5. Beta's configuration rule list with the measurements relevant for the tests (number of configuration rules and maximal number of feature variants) are marked out.

The test result summary for the three real case examples and the measurement in the growth direction (*G*) is shown in Table 4. The trend is that the number of matrix columns is higher than the list rows, which may be a disadvantage when using a ma-

trix-based visualization method. The combinatory difficulty measurement (*C*), shown in Table 5 does not show a clear trend. The matrix-based visualization can, but not necessarily, have a combinatory difficulty within the range of the list-based visualization method. It all depends on how the configuration rules are authored and which configuration rules that are visualized together.

Table 4. Test results of measurements growth direction measurement (number of columns vs. rows) for three real case examples (A-C).

Company	Alpha (matrix-based visualization)		Beta (list-based visualization)
Growth measurement	Number of columns		Number of rows
Example A	8	is more than	4
Example B	12	is more than	9
Example C	10	is more than	5

Table 5. Test results of combinatory difficulty for three real case examples (A-C).

Company	Alpha (matrix-based visualization)		Beta (list-based visualization)
Combinatory difficulty measurement	Number of feature variants in combination		
Example A	3	is less than	max 4
Example B	41	is more than	max 34
Example C	9	is less than	max 12

3.3 Evaluation phase

Three semi-structured interviews were held with configuration rules specialists from Beta, where they commented on the benefits and disadvantages of the matrix- and the list-based visualization method based on the real case example from the previous section. During the interviews it was found that the advantages of using the matrix-based visualization method were dominating. Arguments were that the list-based configuration rules at Beta were similar to programming with many symbols for Boolean operators and complicated use of brackets. The matrix-based visualization method does not contain any explicit Boolean algebra (signs such as OR (/), AND (,), NOT(-), AND-NOT (+-)). One quotation from an interviewee's comments is:

Our PDM system [with list-based visualization method] contains Boolean algebra which is for few people understandable, especially since we combine so many +, - and (). The matrix [without any explicit use of Boolean algebra] is in general always easier to overview and understand.

The non-specialists at Beta have great difficulties in reading the Boolean algebra in the configuration rules. This is in contrast to the non-specialists at Alpha who are

competent to review the configuration rule matrixes at monthly meetings, and as well as soon there is a modification need.

4 Discussion

The main aim of this paper is to compare configuration rule visualization methods in order to find effects of how configuration rules are authored from using a certain visualization method. A second aim is to find advantages and disadvantages of the visualization methods at the two automotive manufacturing companies. The following research questions have been addressed:

RQ1: What are the characteristics of the authoring methods for configuration rules that may be derived from the use of either matrix- or list-based visualization methods?

Overall, people not familiar with the authoring of configuration rules may find it difficult to even compare the use of matrix- and list-based visualization methods. For experienced users, the probably most striking difference is that, in contrast to Beta, Alpha has a visualization method generated from feature family relations. The feature family relations are extremely important for the configuration rule matrix, as reducing the number of feature family relations is the only way users can prevent the number of matrix columns from becoming high. The user wants to have few matrix columns in order to have an easy overview configuration rule matrix. Few feature family relations give short but many configuration rules.

At Beta, it is instead the number of rows in the configuration rule lists that the users are trying to prevent from growing. That is another optimization objective, causing the configuration rules to differentiate from Alpha's configuration rules: Beta authors long but few configuration rules. The combinatorial difficulty is then higher for a list than for a matrix when the users aim for an easy to overview visualization of configuration rules.

RQ2: Which visualization method is most suitable in the cases of:
(a) high number of configuration rules
(b) high number of product feature variants in each configuration rule?

Answering RQ2a: A high number of configuration rules potentially cause the visualization of configuration rules to become, independently of the visualization method, unmanageable big. The disadvantage of the matrix-based visualization method is that the number of matrix columns can potentially grow faster than the number of rows in the list-based visualization. This was shown in the tests conducted in this paper, where all three tests examples had this behavior. When there is a high number of configuration rules to be visualized, the list-based method has the advantage there are many logical operators that could be used in order to prevent the number of rows to grow rapidly. The recommendation is therefore to use a configuration rule list when a high number of configuration rules are visualized.

Answering RQ2b: A high number of feature variants in combination causes combinatorial complexity. The benefit of the matrix-based visualization is that is does not have the OR operator; there are only the operators AND and NOT. This is a very important fact, as it is the OR operator that enables the high number of features variants in combination. Without the OR operator, the configuration rule has to be visualized using several shorter configuration rules, with lower combinatorial difficulty and thereby easier to read. Based on this discussion, the conclusion is that the matrix-based visualization method is well suited for visualizing configuration rules with a high number of feature variants in combination.

5 Conclusions

The visualization method strongly influences how the configuration rules are authored, as there are different optimization goals for readability (few matrix columns vs. few list rows) and availability of logical operators (with vs. without OR operator). There are therefore obstacles when changing visualization methods, here summarized below.

From a list-based visualization method to a matrix-based visualization: the matrix will potentially get many columns.

From a matrix-based visualization to a list-based visualization: the strength of the list is not shown until the configuration rule are reformulated with the introduction of the OR operator.

The interviewed configuration rules specialists however strongly preferred the matrix-based visualization method even when admitting the strength of the list-based visualization method. The main argument was that the matrix-based visualization kept the Boolean algebra simple as it did not allow any OR operator. With a simple Boolean algebra, the configuration rules are becoming easier-to-read for all users working with configuration rules and that would make the development of configuration rules more efficient.

6 Future work

The three examples tested in this paper were selected on the basis of the users' wishes to test the visualization methods' strengths and weaknesses on challenging examples. Challenges in this context mean that the users had either a high number of configuration rules, or a high number of feature variants in each configuration rule. This is similar to a stress test, where a concept is tested under extreme conditions. A suitable continuation would be to build a prototype that the users could test under normal work conditions, which could capture more opinions of strengths and weaknesses of the two visualization methods.

Acknowledgements. This work was carried out at the Wingquist Laboratory VINN Excellence Centre for Efficient Product Realization at Chalmers University of Technology, supported by the Swedish Governmental Agency for Innovation Systems (VINNOVA). Further financial support was received from ProViking. For supervision during the industrial stay the first author would like to thank Hr. Eiffinger.

References

1. Barker, V.E., O'Connor, D.E.: Expert systems for configuration at Digital: XCON and beyond. Communications of the ACM 32(2):298--318 (1989)
2. Soininen, T., Niemelä, I.: Developing a declarative rule language for applications in product configuration. Practical Aspects of Declarative Languages, Lecture Notes in Computer Science, vol. 1551, pp. 305--319 (1998)
3. Bachant, J., McDermott, J.: R1 revisited: Four years in the trenches. AI Magazine 5:21--32 (1984)
4. Mackinlay, J.: Automating the design of graphical presentations of relational information. ACM Transactions on Graphics 5(2):110--141 (1986)
5. Bongulielmi, L., Henseler, P.C., Puls, C. and Meier, M.: The K- & V-Matrix-Method in Comparison with other Matrix-based Methods supporting Modular Product Family Architectures. In Sigurjonsson, J.B. (eds.) Proceedings of NordDesign, Trondheim, (2002)
6. Harlou, U.: Developing product families based on architectures – Contribution to a theory of product families, PhD thesis, Department of Mechanical Engineering, Technical University of Denmark. (2006)
7. Schuh, G., Jonas, I.: Variantenreduzierung im Verbund – Praktikable Methode zum Variantenmanagement: Ein Leitfaden zur Beherrschung der Variantenvielfalt. Verbundinitiative Automobil Nordrhein-Westfalen, Düsseldorf, Germany (1997)
8. Tidstam, A., Malmqvist, J.: Authoring and Verifying Vehicle Configuration Rules. In: (eds.) Proceedings of 8th International Conference on Product Lifecycle Management, 11-13 July, Eindhoven, The Netherlands. (2011)
9. Hami-Nobari, S., Blessing, L.: Effect-oriented Description of Variant Rich Products. In: (eds.) Proceedings to International Conference on Engineering Design (ICED2005), Melbourne, Australia, 15-18 August (2005)