Authoring and verifying vehicle configuration rules

Anna Tidstam* and Johan Malmqvist

Department of Product and Production Development
Chalmers University of Technology
SE-412 96 Göteborg, Sweden
Tel: +46 (0) 31 772 82 88
tidstam@chalmers.se

*Corresponding author

Abstract: Vehicles require large set of configuration rules defining which vehicles that are allowed to be built. Incorrect rules may have expensive consequences, e.g. faulty configurations with missing parts in production. This paper aims to investigate industrially applied methods for authoring and verifying configuration rules, specifically to understand the difficulties that potentially may lead to faulty configurations. The research method was mainly by interviewing design engineers and product structure specialists at three large automotive companies operating in-house developed Product Data Management systems for the product structure. Both roles want configuration rules that are easy to read based on their specific needs. However, their needs differ due to different daily working activities. Our main contribution is to formally define the authoring methods that we have found during the interviews, and to analyze their strengths and weaknesses during the verification activity.

Keyword: configuration rules, authoring methods, verification methods

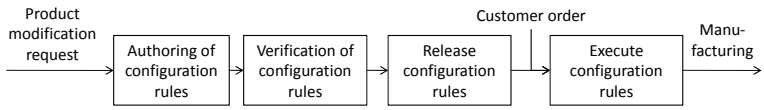
1 Background

This paper aims to investigate industrially applied methods for authoring and verification of vehicle configuration rules, specifically to understand the difficulties that potentially may lead to faulty vehicle configurations and inefficiencies in the configuration rule development process.

Configuration rule modifications are usually requested by new development projects, facelifts or modified market offering, but may also be requested due to discovered quality issues. The configuration rules are authored using certain methods, and are then verified before the release, see Fig. 1. Both design engineers and product structure specialists are involved in authoring and verification of configuration rules. These roles have different daily activities, which generates different preferences in authoring methods. There is also a risk of misunderstandings between the two roles when developing the configuration rules. When the configuration rules are released they may be used within the order system, where they are executed when verifying that customer orders are allowed to be manufactured. The paper's scope covers the authoring and verification methods of the configuration rules, which is in contrast to the more commonly researched topic the execution of the configuration rules as in [1, 2, 3]. Efficient authoring and verification of

configuration rules is becoming increasingly important, due to increasing industrial needs and increased capability of supporting these needs by formal verification techniques. The industrial needs are at most automotive companies increasing due to the vehicle complexity and harsh competition.

Figure 1 Schematic picture from the change initiation of the product structure to manufacturing, with this paper's scope marked in dotted line.



Some papers claim to know how to identify the flawed configuration rules, for example [3, 4]. Others claim to have automated the support for debugging of configuration rules, for example [5]. These authors do however not address the complete issue, since what the authors are discussing is the algorithm for how to calculate which vehicle configurations that are allowed to be built according the configuration rules. Looking at the automotive industry, there is nothing flawed about a vehicle configuration that is not allowed to be manufactured, if it should not be according the strategy from the product planning department or the design engineers' feasibility studies. Following this line of argumentation, paper [4] states that it is yet to be proven for how to verify that the product structure is correct. In the future maybe, the configuration rules will contain enough information to fully automate the verification of configurations. This paper studies the design engineers and product structure specialists reasoning when authoring and verifying configuration rules. The results shows that both authoring and verification is done by visually examining the configuration rules, which is an activity not yet described in the literature at least to the authors' awareness.

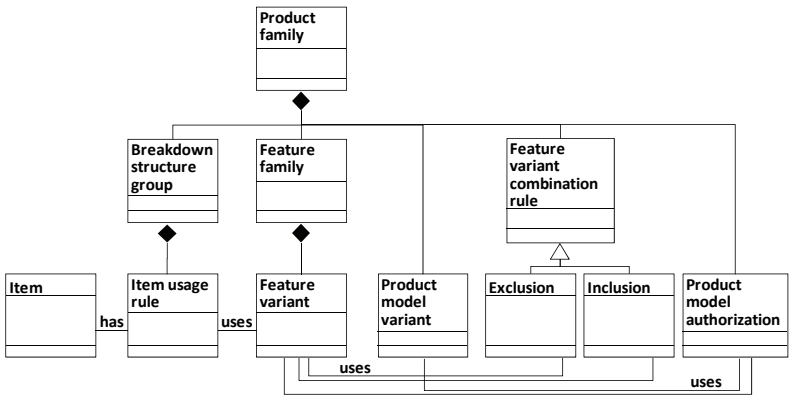
The remainder of the paper is organized as follows. The definitions used are presented in Section 2. Research questions, data collection methods and interview strategy are presented in Section 3. Section 4 presents the results from the study, Section 5 the validation of the results, Section 6 the conclusions and Section 7 the future work.

2 Definitions

Vehicle configuration requires two product structures, one feature (variant)-oriented that is related to the options that a customer selects in sales configuration, and one item-oriented structure that is related to design and manufacturing. Therefore, vehicle configuration requires several different kinds of configuration rules. The definitions used in this paper are presented in Fig. 2, but have also previously been described in [6]. The top node of the information model is the product family, defining e.g. all vehicle models sharing the same platform. Within the product family there may be several product models, e.g. the basic and luxury version of a medium-sized vehicle. The vehicles within a product model is specified using features, which were mentioned as early as in 1982 by Mather [7], who claimed that the features are necessary when the product variety is too large to define a product number for each developed product. Selecting one feature

variant from each feature family creates the vehicle specification. The feature variants are variable product features, e.g. the "exterior colour white" or "exterior colour red".

Figure 2 Information model for the product structure commonly used in the automotive industry, adapted from [6].



The allowed combination of feature variants are controlled by three types of configuration rules called exclusions, inclusions and product model authorizations. There is a fourth rule type called "item usage rule", which declares for which feature variant combinations the items, e.g. components, documents or interfaces, are used. The configuration rules are thereby:

- Product model authorization rules define for which product model variants (e.g. Volvo V70, BMW 3 Sedan etc) a specific feature variant (e.g. sunroof) is allowed.
- Feature variant combination rules that define prescribed ("inclusions") or forbidden ("exclusions") combination of feature variants.
- Item usage rules, which define for what feature variant combinations a certain item, should be used.

The first two bullets are called variant combination rules and exist at the studied automotive companies but the company strategies differ with respect to the extent a certain rule type is used.

The following section will describe how the configuration rules may be written in propositional logic. The constituents are the variable values called feature variants and logical operators such as NOT ($\neg$), AND ($\wedge$), OR ($\vee$) and IMPLIES ($\rightarrow$). Exclusions declare that for example the feature variant "19 inch tyre" cannot be combined with feature variant "21 inch wheel". The formalized definitions of configuration rules are:

> *Let X be the set of feature families (variables) $X = \{x_1, x_2, \ldots, x_N\}$, where N is the number of feature families. Let D be the set of corresponding feature variants (domains) $D = \{D_1, D_2, \ldots, D_N\}$ such that $x_i \in D_i$ and $D_i = \{a_1, a_2 \ldots, a_M\}$ where M is the number of feature variants for the set D_i.*

Exclusion: *Let the scope S be defined as feature family indices in a restriction. An exclusion is a constraint of the form:*

$$\neg \bigwedge_{i \in S} x_i = a_i, \text{ where } a_i \in D_i \quad (1)$$

Inclusion: *An inclusion is a constraint of the form:*

$$x_i = a_i \Rightarrow x_j = a_j, \text{ where } x_i \in D_i, x_j \in D_j \text{ and } i \neq j. \quad (2)$$

Product model authorization: *Let C be an arbitrary positive integer. Let the variable describing the product model be denoted y, with $y \in P$ and $P = \{ p_1, p_2 \ldots, p_L \}$ where L is the number of product models. The product model authorizations for product model p_i for feature family x_k is a constraint of the form:*

$$y = p_i \Rightarrow \bigvee_{j=1}^{C} x_k = a_j, \text{ where } p_i \in P \text{ and } a_j \in D_k \quad (3)$$

Van Veen described the maintenance benefit of using item usage rules for translating product specification using features into bill-of-materials, hence he used the definition "item specification" [9]. Vehicles have a high number of allowed bill-of-materials, and it according to Veen it is more efficient to use item usage rules describing a product family instead of manage every single bill-of-material separately. The allowed vehicle configurations are defined by the variant combination rules, while the item usage rules populate allowed vehicle configurations with items.

Item usage rule: *Let the scope S be defined as the feature family indices for the feature variants in an item usage rule. Let B be the complete set of items with $B = \{b_1, b_2, \ldots, b_K\}$, where K is the number of items. The item usage rule to the item b_j is the constraint of the form:*

$$\bigwedge_{i \in S} x_i = a_i \Rightarrow b_j, \text{ where } a_i \in D_i \quad (4)$$

The described automotive information model is fairly limited since it is only using logic operators and not the mathematical operators, e.g. {>, +} as in [8]. The authoring and verification methods that are going to be described in this paper are therefore only valid within the automotive industry and possibly for the business production strategy "assemble-to-order" [10].

3 Research method

This desired result of his study is formalization of the methods used when authoring and verifying configuration rules. The stated research questions with this motivation are:

RQ1: How are configuration rules authored and which variations exist?

RQ2: What are the strengths and weaknesses of different authoring methods?

RQ3: How are missing/incorrect configuration rules detected?

RQ4: What are the strengths and weaknesses of the verification methods?

The research questions were studied at three automotive original equipment manufacturers, which are all large enterprises according to EU definitions [11]. The companies' Product Data Management systems, henceforth PDM systems, for managing the product structure have been developed in-house. Development of variant-rich products, together with a long experience of the PDM systems, makes the companies suitable for studying challenges when authoring and verifying configurations.

3.1 Data collection methods

The interview sessions included interviewees' demonstrations of authoring and verification methods. Documents were also studied, i.e. the guidelines for how to update the product structure. Also, the product structure itself was very useful to study when evaluating the findings. It may therefore be concluded that multiple methods have been used to validate the findings.

3.2 Interview strategy

In total, 20 semi-structured interviews were conducted, which lasted approximately 2 hours each. The interviews were carried out by one or two researchers together with one employee from one of the automotive companies. The interviewees were equally distributed from the roles:

- Design engineer with >20 years experience of the PDM system;
- Design engineer with <5 years experience of the PDM system;
- Product structure specialist with focus on item usage rules;
- Product structure specialist with focus on variant combination rules.

An initial analysis based on the results from four interviews was reviewed by the industrial reference group, consisting of representatives from the product structure specialists, and suggestions for modifications to the interview guide were agreed upon.

4 Results

The results and conclusions were presented in a workshop where all automotive companies participating in the study were attending. The following sections contain the study's results in subsections Role activities, Authoring instructions, Visualization of item usage rules and variant combination rules, Authoring variations and Verification methods.

4.1 Role activities

The interviewees were asked how many hours that were spent on reading, authoring and verifying variant combination rules and item usage rules. The design engineers spend about the same hours independently of experience, in average around 5 hours/week, and the time is usually equally distributed between variant combination rules and item usage rules. There are two types of product structure specialist. Either occupied full-time on reviewing the request of variant combination rules modifications, or occupied full-time

reviewing the item usage rules modifications. The later type spends the time equally between item usage rules and variant combination rules.

4.2 Authoring instructions

The instructions for authoring configuration rules were found both from the interviews and the guidelines for how to update the product structure. The recommended practice is at the studied companies to author "as short rules as possible", and to use the replace functionality as much as possible. The length of an item usage rule or a configuration rule is defined as the number of feature variants used in a single rule. The automotive industry suffers from enormous amount of configuration data, which motivates the need of keeping the rules as short and as few as possible. The use of the replace function is due to the traceability when updating the configuration rules. The risk with using the replace function is the sometimes insufficient analysis of the variant combination rules. The variant combination rules are constantly changing which may result in necessary modifications to the existing rules sets. The next section is about how the variant combination rules and item usage rules are displayed.

4.3 Visualization of item usage rules and variant combination rules

The next two subsections are describing the display of item usage rules respectively the variant combination rules. The item usage rules and the variant combination rules are in the automotive industry displayed in different views in the PDM system, and sometimes not even stored in the same IT system. They are also used to a different extent by different roles.

4.3.1 Matrix versus list of item usage rules

At the studied automotive companies, the item usage rules are presented using two variants of visualization formats, both as a matrix and as list, see Fig. 3. The 18INCHTYRE, 20INCHTYRE, STDWHEEL and SPAREWHEEL are feature variant codes. The item usage rule matrix contains crosses which mean that these feature variants are included in item usage rules. The item usage rule is a logic expression for when a certain item should be used. The first row of the item usage rule matrix is equivalent to the first row in the item usage rule list. 18INCHTYRE and 20INCHTYRE are the feature variants from the same feature family. STDWHEEL and SPAREWHEEL are also the feature variants from the same feature family. The length of item usage rule for ITEM002 is 2, equal to the number of crosses in the second row of the item usage rule matrix. The example shown is a illustrative simplified description of the visualization of item usage rules which holds for all three studied automotive companies, but the real industrial examples includes of course more tyre items and more feature variants in the item usage rules. The benefit of using the matrix format is especially beneficial when comparing item usage rules with many feature variants.

Figure 3 Item usage rule list as well as item usage rule matrix for three tyres.

Item usage rule list:		*Item usage rule matrix:*		FEATURE VARIANTS: 18INCHTYRE	20INCHTYRE	STDWHEEL	SPAREWHEEL	
		ID:	*Description:*					*Length:*
IF(18INCHTYRE)THEN(ITEM 1)	ITEMS:	ITEM001	Tyre	x				1
IF(20INCHTYRE & STDWHEEL)THEN(ITEM 2)		ITEM002	Tyre		x	x		2
IF(20INCHTYRE & SPAREWHEEL)THEN(ITEM 3)		ITEM003	Tyre		x		x	2

x = feature variant included in item usage rule

4.3.2 List of variant combination rules

At the three studied automotive companies, the variant combination rules are presented as lists, see Fig. 4. The variant combination rules consist of product model authorizations (declaration of allowed feature variants for product models), restrictions (using NOT operator) and inclusions (using IF-THEN operator for feature variants).

Figure 4 List of variant combination rules.

List of variant combination rules:
Product model authorization: IF(Model X) THEN (STDWHEEL OR SPAREWHEEL)
Restriction: NOT(18INCHTYRE & STDWHEEL)
Inclusion: IF(18INCHTYRE) THEN(SPAREWHEEL)

4.4 Authoring variations

All the studied companies have all three types of variant combination rules, but there is a difference with respect to the extent the different types are used. However, even though there are some differences in how variant combination rules are used, the main difficulty still holds since it is how to combine the variant combination rules with the item usage rules. This difficulty is also one of the reasons to why authoring variations exist, as there is a potential of using an overlap between the two classes of rules.

The most common authoring method is to use the replace function, but then there are three variations in how to author configuration rules identified. These will be described in the following subsections, where possible with the formal definition followed by an example and discussion of strengths and weaknesses. It should be underlined that the methods' definitions do not provide any instructions for how they may be implemented in IT applications. Several important evaluation criterions for authoring methods were found from the initial four interviews:

1. Time to find (starting to search → finding);
2. Time to interpret (finding → using);
3. Time to maintain (assumed proportional to update frequency);
4. Time to verify (correctness and completeness checks).

4.4.1 Overlapping documentation

Using overlapping documentation is similar to saying that the candle is red and of wax, when all candles are made of wax. The method's name use the word "overlapping" since the item usage rules repeat some of the information that the variant combination rules states. Overlapping variant combination rules may also occur, and are then repeating information that other variant combination rules states. To avoid overlapping documentation is to use the shortest length of configuration rules. The example in Fig. 5 shows item usage rules for tyres in the matrix format used in one of the studied companies. The example is comparing with and without overlapping documentation. With overlapping documentation the item usage rule for the first ITEM001 is using both 18INCHTYRE and SPAREWHEEL, while without overlapping documentation it is only 18INCHTYRE. For ITEM001, the usage of the SPAREWHEEL is not necessary because of the restriction stating NOT(18INCHTYRE & STDWHEEL).

Figure 5 Item usage rules with and without overlapping documentation due to variant combination rules.

FEATURE VARIANTS:

Item usage rule matrix:

	ID:	*Description:*	18INCHTYRE	20INCHTYRE	STDWHEEL	SPAREWHEEL	
ITEMS:	ITEM001	Tyre	x			x	With overlapping documentation
	ITEM002	Tyre		x	x		
	ITEM003	Tyre		x		x	
	ITEM001	Tyre	x				Without overlapping documentation
	ITEM002	Tyre		x	x		
	ITEM003	Tyre		x		x	

Feature variant combination rule list:
NOT(18INCHTYRE & STDWHEEL)

x = feature variant included in item usage rule

Using overlapping documentation does not prevent vehicles from being correctly built, but avoiding overlapping documentation is one method for reducing the length of rules and to show that the configuration rules have been analyzed which is beneficial during the verification task. Using the example in Fig. 5, avoiding overlapping documentation shows that ITEM001 is covering all allowed vehicle configurations with 18INCHTYRE. However, with overlapping documentation there seems to be an item missing for the 18INCHTYRE and STDWHEEL. Due to less data to manage, some product structure specialists are claiming that the time is also low for interpreting the item usage rules when avoiding overlapping documentation, see Fig. 6. However, avoiding overlapping documentation means using fewer feature variants for the item usage rules, which makes the item usage rules more difficult to interpret and find for design engineers.

Figure 6 Evaluation of strictly avoiding overlapping documentation in terms of time consumption.

	Evaluation for avoiding overlapping documentation	
	Design engineers	Product structure specialists
Time to interpret	High	Low
Time to maintain	-	-
Time to find	High	-
Time to verify	-	Low

The formal definition of overlapping documentation is presented in propositional logic in formula (5). The formula states that the left hand side is equivalent to the right hand side. The check if an item usage rule or a configuration rule is authored with overlapping documentation may be done by using formula (5) together with e.g. SAT solver.

> ***Overlapping documentation:*** *Let the scope T be the feature variants of an item usage rule, and let the scope S be another set of feature variants. The scope T is an overlapping documentation if the following statement is valid:*
>
> $$\bigwedge_{i \in S} x_i = a_i \Leftrightarrow \bigwedge_{i \in T} x_i = a_i, \text{ where } S \subset T \qquad (5)$$

Formula (5) is also valid for variant combination rules, since this would only negate both sides of the equivalence.

4.4.2 High-level feature variants

Using high-level feature variants reduces the number of configuration rules. It is similar to saying that the feature family "outfit colour" is black, instead of saying that the "trouser colour" is black, the "sweater colour" is black and the "shoe colour" is black. In Fig. 7, use of high-level feature variant STDWHEEL results in 1 item usage rule, compared to 2 item usage rules when not using the high-level feature variant. The new feature family with variants LOW, BASIC and HIGH DURABILITY describes the the tyre characteristics. The equivalence between with and without usage of high-level feature variant is in this example due to the variant combination rules.

Figure 7 Item usage rules for tyres with and without use of high-level feature variant STDWHEEL.

Item usage rule matrix:

			FEATURE VARIANTS:							
	ID:	*Description:*	18INCHTYRE	20INCHTYRE	STDWHEEL	SPAREWHEEL	LOW DURABILITY	BASIC DURABILITY	HIGHDURABILITY	
ITEMS:	ITEM002	Tyre		x	x					Use of high-level feature variants
	ITEM002	Tyre		x				x		Without use of high-level feature variants
	ITEM002	Tyre		x					x	

x = feature variant included in item usage rule

Feature variant combination rules list:
IF(STDWHEEL) THEN (BASIC DURABILITY or HIGH DURABILITY)

One of the difficulties with using high-level feature variants is that the relations are not at all studied companies explicitly documented in the PDM system. High-level feature variants may instead be found from analyzing variant combination rules. However, the high level feature variants were appreciated by product structure specialists who are constantly suffering of enormous amount of data, see Fig. 8. There is no clear opinion from the design engineers since the usage of high-level feature variants gives various effects in different cases.

Figure 8 Evaluation of using high-level feature variants in terms of time consumption.

	Evaluation for using high-level feature variants	
	Design engineers	Product structure specialists
Time to interpret	-	Low
Time to maintain	-	Low
Time to find	-	Low
Time to verify	-	-

It is possible to define high-level feature variants with propositional logic:

> ***High-level feature variants:*** *Let the scope S be the indices for a set of feature variants from D_k, with $S \neq \{0\}$. The feature variant a_i is a high-level feature variant compared to a_j with $j \in S$ if the following statement is valid:*
>
> $$x_i = a_i \Rightarrow \bigvee_{j \in S} x_k = a_j, \text{ where } i \neq k \qquad (6)$$

4.4.3 Building blocks of item usage rules

Using consistent selection of feature variants for the item usage rules may create small "building blocks", which then may be used when allowed according the variant combination rules. The building block may not necessarily avoid overlapping documentation, but aims to have a fixed number of feature variants, see Fig. 9. This method is by far the most used authoring method. The new feature variant codes are FAMILY VERSION and SPORT PACKAGE which are only used without using the building-block method. The equivalence with and without the usage of the building block method in this example is not proven by presenting the variant combination rules.

Figure 9 Item usage rules for tyres with and without use of building block method, here without the necessary variant combinations for proving equivalence with and without using the method.

FEATURE VARIANTS:

Item usage rule matrix:		FAMILY VERSION	SPORT PACKAGE	18INCHTYRE	20INCHTYRE	STDWHEEL	SPAREWHEEL	LOW DURABILITY	BASIC DURABILITY	HIGH DURABILITY	
ID:	*Description:*										
ITEMS: ITEM001	Tyre			x		x					Using building-block method
ITEM002	Tyre				x	x					
ITEM003	Tyre				x		x				
ITEM001	Tyre		x	x							Without using building-block method
ITEM002	Tyre				x	x					
ITEM003	Tyre	x			x			x			

x = feature variant included in item usage rule

The item usage rules and variant combination rules are as independent as they can be, which results in item usage rules requiring low update effort, both by design engineers and product structure specialists, see Fig. 10. Independent means that no variant combination rules analysis is required when authoring the item usage rules. Another aspect is that the design engineers search and find the item usage rules fairly easy when e.g. all tyres are documented with either STDWHEEL or SPAREWHEEL. Concerning the formal definition of this method, the only criteria is a consistent and minimized set of feature families used for an item usage rule set. The method is decreasing the time consumption for the verification activity, since it becomes easier to compare similar exclusions or similar item usage rules.

Figure 10 Evaluation of the building-block method in terms of time consumption.

	Evaluation for building-block method: Design engineers	Product structure specialists
Time to interpret	-	-
Time to maintain	Low	Low
Time to find	Low	-
Time to verify	Low	Low

4.5 Verification methods

Some of the authoring methods described have a positive impact on the verification efficiency. However, the methods for verification using manual inspection are not found to be documented at the studied companies. The verification tasks have been divided into two sections depending on if they are investigating the allowed vehicle configuration or the population of items on these.

4.5.1 Verification of allowed feature variant combinations

The verification of allowed feature variant combinations is most commonly done by the product structure specialist, but may also be done by the responsible design engineer. Either the variant combination rules are analyzed by reading their formulation, or the allowed feature variant combinations are generated. As is stated in [9], the correctness of each configuration rule may be validated separately, but in large sets of variant combination rules it can be very difficult to interpret the "implicit rules". The implicit rules are constraints which are not explicitly expressed by a rule, but which follow from a combination of explicitly defined rules. The generated allowed feature variant combinations take both explicit and implicit rules into account. Several interviewees mentioned that they found the generated allowed feature variant combinations easier to read, but this verification method is not mandatory and very few design engineers use it. One of the reasons may be that it is difficult to find the explanations to why (which variant combination rules) a feature variant combination is not allowed without IT system support or other deficiencies in the functional core of the IT system. Another reason to why the verification method is not used may also be the interface since the allowed feature variant combinations does not show any information about item usage rules or items.

The allowed feature variant combinations may be compared to the design engineers' knowledge about which feature variant combinations should be allowed and which should not, see Fig. 11. The first row is the allowed feature variant combination 18INCHTYRE & STDWHEEL & BASIC DURABILITY according to the variant combination rules. The allowed feature variant combinations show the effect of all variant combination rules, and may be easier to analyze instead of looking at product model authorizations, exclusions and inclusions separately. If there are allowed feature variant combinations that should be restricted, then there are feature variant combination rules missing and vice versa.

Figure 11 The allowed feature variant combinations and the variant combination rules.

FEATURE VARIANTS:

Allowed feature variant combination list:	18INCHTYRE	20INCHTYRE	STDWHEEL	SPAREWHEEL	LOW DURABILITY	BASIC DURABILITY	HIGH DURABILITY
Allowed feature variant combination 1	x		x			x	
Allowed feature variant combination 2	x		x				x
Allowed feature variant combination 3		x	x			x	
Allowed feature variant combination 4		x	x				x
Allowed feature variant combination 5		x		x	x		
Allowed feature variant combination 6		x		x		x	
Allowed feature variant combination 7		x		x			x

Variant combination rule list:
NOT(18INCHTYRE & SPAREWHEEL)
IF(STDWHEEL) THEN (BASIC DURABILITY or HIGH DURABILITY)

x = feature variant included in allowed feature variant combination

4.5.2 Verification of allowed items

The verification of allowed items is the activity to make sure that every allowed feature variant combination is populated with the correct items. Typical errors show up either as vehicle specifications missing necessary items, or items not used at the assembly line. Assuming correct variant combination rules from the previous section, it is faulty item usage rules that cause these errors to occur.

The visual verification of allowed items is mainly done by studying item usage rules, but if necessary also the variant combination rules to realize which vehicle configurations that are allowed. In the previous presented Fig. 7, there is an example where it is clear that the item usage rules do not provide enough information for verifying that only one ITEM001 is allowed for every allowed vehicle configuration. It is necessary to know if STDWHEEL is allowed in combination with BASIC or HIGH DURABILITY to realize if multiple items are allowed for the same vehicle configuration. In the example shown, there will always be two ITEM001 for some of the allowed vehicle configuration.

4.6 Discussion of results

At the studied companies, each configuration rule is analyzed if it can be shortened, called avoiding overlapping documentation. This is the method that has shown the clearest discrepancy between design engineers and product structure specialists. However, the by far most common authoring method is to replace what already exists with new rules slightly modified. This is mainly due to the otherwise time-consuming and difficult analysis of variant combination rules. From this discussion, it may be concluded that the main difficulty for design engineers is how to make sure that all types of configuration rules together describes the by product planning department requested and buildable vehicle specifications.

Both the verification of allowed feature variant combinations and allowed items requires the capability to analyze the variant combination rules. As a consequence, it was found that the verifications are primarily conducted by product structure specialists. These findings motivate the need for facilitating the configuration rule analysis by adequate system support.

5 Generalization

The automotive information model used in this paper has been verified to exist at the studied automotive companies and previous paper [6] contains a literature review on the subject of typical automotive information model. This information model contains both feature variants and items, which are describing vehicle configurations by using configuration rules. During both authoring and verification of item usage rules, the main difficulty found was how to make sure that the types of configuration rules together describe the by product planning department requested and buildable vehicle specifications. This difficulty occurs due to the automotive information model, and is then founded on the model's validity which has been proven well-funded. The comparison between the companies is the main strategy for obtaining generalization of the results.

Another similarity between the studied automotive companies is the structure of the PDM system. The item usage rules and variant combination rules are in different views or even different systems, which further make the combination of item usage rules and variant combination rules difficult and therefore preferences of certain authoring methods.

6 Conclusions

The literature review showed that the authoring and verification methods of configuration rules described in this paper are rarely studied. Authoring variations have been identified, where readability is put against compactness and maintainability. Repeatedly arguments for using an interface consisting of a matrix format are presented. We have also shown that the main difficulty is to combine variant combination rules with item usage rules, and the traditional interface to the PDM system consisting of a database viewer should therefore be challenged. With the formalization of the authoring and verification methods, there is a potential for higher degree of automation of these activities which

would facilitate the work for both product structure specialists and design engineers. We have shown that the time spent on reading, authoring and verifying configuration rules is significant for design engineers, and full-time job for product structure specialists, which motivates realizing the automation potential and thereby reducing development costs.

7 Future work

Since the different users show different needs when interpreting the configuration rules, it is impossible to verify configuration rules without an analysis support that makes short and few rules more informative and understandable. The methods presented in this paper are essential when developing this analysis support that would facilitate the validation of configuration rules.

Acknowledgment

This work was carried out at the Wingquist Laboratory VINN Excellence Centre within the Area of Advance – Production at Chalmers, supported by the Swedish Governmental Agency for Innovation Systems (VINNOVA). The support is gratefully acknowledged.

References

1 Sinz, C., Kaiser, A., Küchlin, W. (2003) `Formal methods for the validation of automotive product configuration data', *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 17, No. 1, pp.75--97.

2 Astesana, J. M., Cosserat, L., Fargier, H. (2010) `Constraint-based modeling and exploitation of a vehicle range at Renault's: requirement analysis and complexity study', *ECAI 2010 Configuration workshop*, pp.33--39.

3 White, J., Schmidt, D. C., Bendavides, D., Trinidad, P., Ruiz-Cortés, A. (2008) `Automated Diagnosis of Product-Line Configuration Errors in Feature Models', *Software Product Line Conference, SPLC '08,* pp. 225--234.

4 Batory, D. (2005) `Software Product Lines - Feature Models, Grammars, and Propositional Formulas', *Lecture Notes in Computer Science*, Springer, Vol. 3714, pp.7--20.

5 Felfernig, A., Friedrich, G., Jannach, D., Strumptner, M. (2004) `Consistency-based diagnosis of configuration knowledge bases', *Artificial Intelligence*, Vol. 152, No. 2, pp.213--234.

6 Tidstam, A, Malmqvist, J. (2010) `Information Modelling in Automotive Product Development', *Proceedings of NordDesign 2010*.

7 Mather, H., "Bills of Materials, Recipes and Formulations", *Wright Publishing Company Inc.,* Atlanta, USA, 1982.

8 Oracle Configurator Developer, (2000) "User's Guide", Release 11i, www.oracle.com

9 Van Veen, E., "Modelling product structures by generic bills-of-material", *Elsevier Science Inc.,* New York, NY, USA, 1992.

10 SAP, "Assemble-to-order", www.sap.com

11 European Commission (2003) `Commission Recommendation concerning the definition of micro, small and medium-sized enterprises', *Official Journal of the European Union*, No. 124: 36.