CHALMERS

Developing Vehicle Configuration Rules

ANNA TIDSTAM

Division of Product Development
Product and Production Development
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2012

THESIS FOR THE DEGREE OF LICENTIATE OF ENGINEERING

Developing Vehicle Configuration Rules

Anna Tidstam

Department of Product and Production Development
CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden 2012

Developing Vehicle Configuration Rules
ANNA TIDSTAM

ISSN 1652-9243
Report No. 61

Department of Product and Production Development
Division of Product development
Chalmers University of Technology
SE-412 96 Gothenburg, Sweden
Telephone + 46 (0)31-772 1000

Cover:
The cover illustration illustrates the suggested user interface (right) when developing vehicle configuration rules, as compared to the currently used (left).

Chalmers Reproservice
Gothenburg, Sweden 2012

Abstract

Vehicles are sold in many variants with different engine horsepower, wheel dimensions, type of headlights etc. Vehicle customers specify individual vehicles by the selection of such features during the so-called “sales configuration” process. Logic expressions called vehicle configuration rules are often utilized for automating the sales-to-delivery process.

The development of vehicle configuration rules includes authoring and evaluation of these rules. The goal is to make sure that the configuration rules specify so-called “valid” configurations. Valid in this context is a user-defined state based on perception of domain specialists, e.g. design engineers of brakes. The aim of this thesis is to create methods for configuration rules development which efficiently ensure that configurations are buildable, by making the configuration rules development process more time-efficient and less error-prone.

The problem is that the industrial visualization tools for developing configuration rules, henceforth CR visualization tools, are argued to be difficult to use. Consequently, the users find it difficult to validate configuration rules. The problem is approached in this thesis from scratch by first suggesting a generalized information model for vehicle configuration rules. The information model was derived from industrial studies and literature reviews. Then, user studies at three vehicle manufacturing companies were conducted in order to formalize the authoring methods and to study existing CR visualization tools. Limitations of current CR visualization tools were identified, which were addressed in a new CR visualization tool. This new tool uses one single user interface, which eliminates the swapping between windows. Moreover, the alternative authoring methods and potentially missing items are visualized. Some activities when evaluating the configuration rules are thereby facilitated. The new CR visualization tool has been iteratively developed and evaluated through formative usability tests. The test results were positive in terms of real users correctly conducting test tasks, appreciating the new CR visualization method and predicting a more time-efficient CR development process. The planned future work includes more usability tests to study whether there are any unforeseen usability threats.

Keywords: configuration, vehicle configuration rules, visualization tools, usability tests, design automation, product data management, PDM

Acknowledgements

First of all, I would like to thank my supervisor Professor Johan Malmqvist for his continuous encouragement. I am also grateful to my academic collaborators Alexey Voronov and Knut Åkesson from the Automation research group at the Department of Signals and Systems at Chalmers, Fredrik Ekstedt from the Fraunhofer-Chalmers Research Centre for Industrial Mathematics and Lars-Ola Bligård from the Division of Design and Human Factors at Chalmers.

This project has been dependent on access to in-house developed PDM systems to be able to conduct the research work. From the vehicle manufacturing companies I would like to thank Jan Nykvist for patiently sharing his exclusive knowledge about configuration rules as well as his historical anecdotes; James Keefe for not being hesitant to try new ideas; Camilla Norström for collaborating in multiple case studies; Samir Mesihovic for initiating the research project. Special thanks to Elna Holmberg for arranging the financial possibility for me to continue my work as a PhD candidate at Chalmers. Other industrial people I would like to thank are Bertil Turesson, Håkan Jonsson, Johan Tärbo and Joakim Olsson for giving me the opportunity to conduct user studies at more vehicle manufacturing companies and thereby generalize my research results. Thanks to all design engineers and product structure specialists participating in the numerous workshops and industrial studies!

The willingness to share datasets of industrial configuration rules is gratefully acknowledged. I would like to take this opportunity to thank Ragnar Sjöblom and Henrik Utter for their assistance with that.

This work was carried out at the Wingquist Laboratory VINN Excellence Centre for Virtual Product Realization at Chalmers University of Technology, supported by the Swedish Governmental Agency for Innovation Systems (VINNOVA). The work was supported by the VINNOVA FFI programme – Strategic Vehicle Research and Innovation. The financial support is gratefully acknowledged.

Finally I would like to thank those who are close to me for their support during the years, especially my family.

Appended papers

The appended Papers A, B and C have been written for this Licentiate thesis. Here, each paper is presented with a brief description of its intention and content, as well as the authors' individual contributions. The full-length papers can be found in the Appendix.

Paper A Tidstam, A. & Malmqvist, J. (2010), "Information Modelling for Automotive Configuration", *Proceedings of NordDesign 2010,* Gothenburg, Sweden, 25-27 August, pp. 275-286.

Paper A introduces a generalized information model defining the core elements required for discussing the development of vehicle configuration rules. The aim of Paper A was to compare the information models for vehicle configuration rules as identified in empirical studies with theoretical product structures proposed in the literature, thereby identifying differences and similarities. Johan Malmqvist and Anna Tidstam planned the paper. Anna Tidstam collected the empirical data through interviews at a vehicle manufacturing company. Anna Tidstam analyzed the data with assistance from Johan Malmqvist. The paper was written by Anna Tidstam and reviewed by Johan Malmqvist and product structure specialists from three vehicle manufacturing companies.

Paper B Tidstam, A. & Malmqvist, J. (2011), "Authoring and Verification of Vehicle Configuration Rules", *Proceedings of Product Lifecycle Management (PLM11),* Eindhoven, The Netherlands, 11-13 July.

Paper B presents a user study with product structure specialists and design engineers from three vehicle manufacturing companies. The aim of Paper B was to understand the difficulties when authoring and verifying configuration rules as well as the variations of authoring methods. The paper was planned by Anna Tidstam and Johan Malmqvist. Anna Tidstam and an employee from one of the studied companies collected the empirical data through user studies. Anna Tidstam analyzed the data and wrote the paper. The findings were validated in workshops with product structure specialists from the studied vehicle manufacturing companies. The paper was reviewed by Johan Malmqvist.

Paper C Tidstam, A., Bligård, L-O., Ekstedt, F., Voronov, A., Åkesson, K. & Malmqvist, J. (2012), "Development of Industrial Visualization Tools for Validation of Vehicle Configuration Rules", *Proceedings of Tools and Methods of Competitive Engineering (TMCE2012),* Karlsruhe, Germany, 7-11 May.

Paper C discusses the configuration rules (CR) visualization needs when inspecting configuration rules found at three vehicle manufacturing companies. The paper also describes a demonstrator using a new CR visualization method which has been developed and evaluated by usability tests. Anna Tidstam and Johan Malmqvist planned the paper. Anna Tidstam collected the CR visualization needs together with one employee from a studied vehicle manufacturing company. Anna Tidstam coded the demonstrator together with Alexey Voronov and Fredrik Ekstedt. Anna Tidstam planned the usability tests together with Lars-Ola Bligård. Anna Tidstam and one employee from the studied vehicle manufacturing company conducted the usability tests. The findings were validated in workshops with product structure specialists from three vehicle manufacturing companies. The paper was written by Anna Tidstam. All co-authors reviewed the paper.

Acronyms

AC	Air conditioner
AP214	Application protocol 214
ARC	Area of relevance and contribution
BOM	Bill-of-Material
CAD	Computer-aided design
CM	Configuration management
CR	Configuration rule
CSP	Constraint satisfaction problem
DRM	Design research methodology
DSM	Design structure matrix
IDEF	Integration definition for function modelling
IEC	International electrotechnical commission
ISO	International organization for standardization
KDD	Knowledge discovery in databases
PDM	Product data management
PFMP	Product family master plan
PLM	Product lifecycle management
PMR	Product modification request
QFD	Quality function deployment
RG	Research goal
RQ	Research question
STEP	Standard for the exchange of product data
UML	Unified modelling language

Table of contents

1 Introduction

This introductory chapter describes the background as well as defining the problem. The research work is motivated and the research goals are presented. Finally, the scope and the outline of the thesis are described.

1.1 Background

This section gives an introduction to the terms frequently used in this Licentiate thesis: “configuration”, “configuration rules”, “development” of configuration rules, as well as the “verification” and “validation” of configuration rules.

1.1.1 Configuration

According to Hvam et al. (2007), “configuration” is possible when the product range is based on “modules”. Examples of modules in a vehicle could be the “engine”, the “steering wheel” and the “seats”. The product is configured by selecting, combining and possibly adapting a set of standard modules. The engine module is available in different engine sizes, horsepowers etc. This means that there are many unique configurations possible. A well-known definition of configuration is:

Configuration is a special case of design activity where the artifact being configured is assembled from a set of pre-defined components that can only be connected together in certain ways (Mittal & Frayman, 1989).

The definition applied to this Licentiate thesis assumes that the “artifact” to configure is the vehicle, and “a set of constraints” is constituted by the configuration rules. The term “components” is replaced in this thesis with “items” and “feature variants". The items may consist of components, drawings and other documents etc. Selecting one feature variant from each "feature family" creates the product variant specification; see Fig. 1.1. The feature variants are variable product features, e.g. the "exterior colour white" or "exterior colour red". For vehicles, it is common that the feature variants have codes, e.g. "RC-ROUGH" and "RFUEL490". The customer orders are specified by using a sales configurator. The configurator is a "tool which supports the product configuration process so that all the configuration rules are guaranteed to be satisfied" (Hedin et al., 1998). The customer order consists of a specified selection for every feature variant.

For a manufacturing company of a variant-rich configurable product, it is impossible to keep every configuration of a complete vehicle in stock. Vehicle manufacturing companies today often use a production strategy called "Assemble-to-Order". This production strategy means that assembly of the vehicle is postponed until a customer order has been received, although single components or sub-assemblies may be manufactured in advance (Blecker & Abdelkafi, 2006). The idea of assembling vehicles to order has been described as early as 1983 in the Toyota Production System (Monden, 1983) as a method for producing vehicles "pulled" by the customers, thereby reducing overproduction or "waste". The customer must accept a certain delivery time as the vehicle has to be assembled, but is then offered a vehicle according to his/her specification. There are vehicle manufacturing companies that are selling vehicles based only on customer orders (Hertz et al., 2001).

Following the definition of configuration by Mittal & Frayman, the feature variants may be combined only in certain ways. The allowed configurations are defined by the

Product variant specification:

RC-ROUGH	USIGNS	NR-80EC	UALARMB	COOLC40
RFUEL490	VW2600	UTRACON	SUNV-SMO	FUELSEC2
6*2	FAA10	DETECT-S	WRITEPAD	FTANK-AL
04-TLV	AUD-MED2	BBOXC-BA	UPAXLE	EM-ECO6B
TMPF166	UCBRKT	URTOLL	FAP3040	TRANSF-H
UBULBKIT	RADI-NAR	UBUNKBOX	INST-MED	HL-ASYMR
DASH-BLU	USPEEDDU	ENGPROT	UCBRKTOW	BBOX-L
ENG-VE13	FUELTFIL	PHONE	INST-MED	TWEETER
APF-ENH	FCAP-L	HORN-JER	BUGNET	...

Feature variant:	**Feature family:**	**Description of feature variant:**
RC-ROUGH	Road condition	Badly maintained road
6*2	Axle arrangement	6 wheels thereof 2 driving
RFUEL490	Fuel tank at the RHS	490 liters right side fuel tank

Figure 1.1: Vehicle specification from selection of feature variants (terminology slightly modified from Lindroth, 2011).

configuration rules, which are the topic of the next section.

1.1.2 Configuration rules

For product families with many feature variants, so-called variant-rich products, there may be hundreds of thousands of configuration rules. Vehicles are typically variant-rich products. The configuration rules, aided by a configurator, prevent the user from selecting feature variants that are not available in combination, e.g. “17 inch tires” with “20 inch rims”. The original definition of configuration rules for computers, with minor adaptation to the terminology used in this thesis, is as follows:

Configuration rules denote which combinations of feature variants are obliged and which combinations are forbidden (Euwe & Schuwer, 1993).

In Haag (1998) there is a distinction between high-level and low-level configuration. The high-level configuration is the selection of feature variants, while the low-level configuration is about selection of items. This thesis also considers the logic expressions required for assigning items to configurations as configuration rules.

The configuration rules are primarily created during vehicle development projects

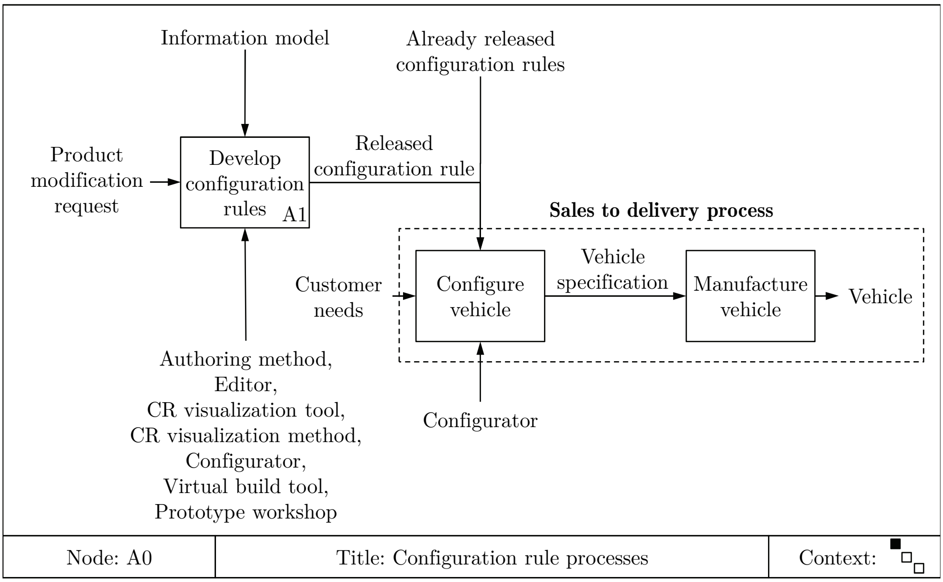


Figure 1.2: Process model for the development and use of vehicle configuration rules.

(Mesihovic, 2004a). The development of configuration rules precedes the sales to delivery process, as in Fig. 1.2. The sales to delivery process starts with customer needs and ends with a manufactured vehicle. The input to the development of vehicle configuration rules is typically a product modification request, which may require new or modified configuration rules to be fulfilled. The output of the development of configuration rules is released configuration rules.

The performance requirements for the configuration rules development process increase with quantity and complexity (Lamberti, 2011). The configuration complexity of the products may be considered to be the multiplicity of number of offered products with the product complexity. Spacecraft are complex products, e.g. consisting of many items, but are not offered in many product variants. Computers and telephones are offered in many product variants, but have a relatively small number of items. The product complexity of vehicles is moderate in comparison to spacecraft, but vehicles are offered in a higher number of product variants. Lamberti argues therefore that the requirements on configuration rules are higher for vehicles than for spacecraft, computers and telephones.

Researchers have found that it is not humanly possible to keep a vehicle configuration rule set absolutely defect-free (Sinz et al., 2003). The objective is to minimize the errors. The activities for avoiding errors in the configuration rules are called verification and validation. Those activities take place during the configuration rules development process, henceforth CR development process, which is the topic of the next section.

1.1.3 CR development process

The development of vehicle configuration rules may be described as a three-step process: the authoring, evaluation and release of vehicle configuration rules, as shown in Fig. 1.3. When the configuration rules are released they may be used for vehicle configuration in the sales to delivery process. The CR development process is initiated by a product modification request. The request is written using the terminology that the information model for vehicle product structures provides, e.g. there may be requests of new feature variants or items. For fulfilling the product modification request, there may be a need for authoring vehicle configuration rules, either by modifying existing ones or by creating new ones. The configuration rules are authored using certain methods and tools, and the configuration rule suggestion is then evaluated before the release. The three-step process potentially iterates between the evaluation and authoring of configuration rules, causing revisions of the rules. After some potential iteration with such revisions, the outcome will be verified and validated configuration rules.

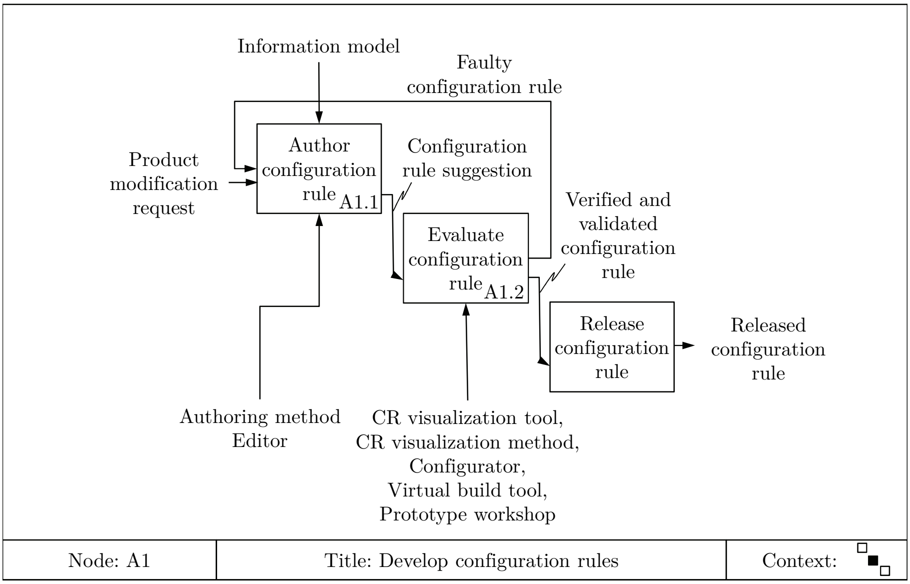


Figure 1.3: Process model for the CR development process.

The next section will describe the verification and validation activities by first describing the terms' definitions for systems or software, and then by applying those definitions to knowledge bases.

1.1.4 Verification and validation of configuration rules

To avoid time-consuming and costly errors during the assembly of the vehicle, the vehicle configurations have to be verified and validated during vehicle development projects. Examples of potential errors are configurations permitted by the sales configurator which are not manufacturable. There may, for example, be insufficient space to fit the components, or components that are missing in the Bill of Material. Another potential error is that faulty or missing configuration rules make the sale configurator reject vehicle configurations that should be permitted, which results in a loss of income.

In systems or software engineering, the terms "verification" and "validation" have been defined in standards (ISO/IEC, 2008). Those standards are general definitions not only for the CR development process, but also for the development of product designs, software etc. The definitions are:

Verification is the confirmation, through the provision of objective evidence, that specified requirements have been fulfilled.

Validation is the set of activities ensuring and gaining confidence that a system is able to accomplish its intended use, goals and objectives.

In other words, verification is about fulfilling requirements, while validation is about gaining confidence that objectives are fulfilled. Let us now discuss the meaning of requirements and objectives during the CR development process.

As has been previously discussed, the configuration rules set may be compared to a knowledge base, which is a special kind of database. For knowledge-based systems, the terms verification and validation have definitions that will be used for this Licentiate thesis (O'Keefe, 1993):

Verification: *Domain-independent* technique that may detect peculiar configuration rules, e.g. redundant configuration rules. Every verification check of configuration rules may be written down to requirements. Configurators offer possibilities for automated verifications if those requirements are formalized into logic expressions.

Validation: The configuration rules are assumed to be valid if they are consistent with domain specialists' perceptions of which vehicles should be allowed to be configured. The validation is therefore a *domain-dependent* technique. The domain specialists are people with specialist knowledge about the domain, e.g. the design engineer of brakes (Tuhrim et al., 1988). Currently the validation of configuration rules is not possible to fully automate since the knowledge and objectives of the domain specialists are not formalized requirements.

After the configuration rules have been verified and validated, they are released. The release process may be further subdivided, e.g. there is a release for engineering and a release for production (Watts, 2008). The release status indicates in which context the configuration rules are allowed to be used. The next section will describe the problem utilizing a section division of the configurable product structures, the CR development process and the CR visualization tools.

1.2 Problem analysis

The CR development process is argued to be an error-prone process and partly relies on manual time-consuming manual inspections (Baumeister & Freiberg, 2010). The following sections will describe the problems related to the product structure, the process itself and the CR visualization tools used.

1.2.1 Configurable product structures

The research on information models for product structures is extensive, e.g. compromising the generic BOM (Van Veen, 1992), high-level and low-level configuration (Haag, 1998) and the standard AP214 (ISO, 2004). Those models are an important foundation, but cannot fully explain the difficulties during the vehicle CR development process. To understand these difficulties, the value and complexity of the data that they involve must be taken into account. Having a good information model for product structures including the configuration rules as a foundation, it should be possible to identify factors that are potentially causing errors to occur, as well as the high time-consumption. In other words, it is currently not known which properties of the vehicle configuration rules result in difficulties, and whether they are the same properties at all vehicle manufacturing companies.

The addressed problem should not be the information itself, but the methods and tools used during the CR development process. There are researchers within the area of configuration who argue that it is the number of configuration rules which is causing them to be time-consuming to develop (Hami-Nobari & Blessing, 2005; Huang et al., 2008). Those authors prescribe a more sophisticated language for the configuration rules, e.g. by including relational expressions ("less than", "more than") for the rules, which is a quite drastic proposal potentially affecting the sales to delivery process negatively. To cite one discussion with a product structure specialist from a vehicle manufacturing company: "It has always been forbidden to touch the house of cards since it may fall." Instead, the generalized description of vehicle product structures needs to be treated as a prerequisite when identifying which factors that are causing difficulties.

The next section discusses the CR development process and identifies two of its activities which put high demands on the design engineers.

1.2.2 CR development process

There is one activity during the development of knowledge bases which has been identified to be especially error-prone and time-consuming. The activity is called the "knowledge acquisition bottleneck", and includes the formalization of configuration rules. Therefore, it may be assumed that the formalization of configuration rules is a difficult activity at the vehicle manufacturing companies. Experience and knowledge of existing products play an important role (Huang et al., 2008). However, it is not known why the formalization of configuration rules is argued to be difficult, nor whether computerized tools may facilitate the development, e.g. by semi-automation.

Another time-consuming problem during the evaluation activity of the CR development process is the iteration of inspections and computations; see Fig. 1.4 (an

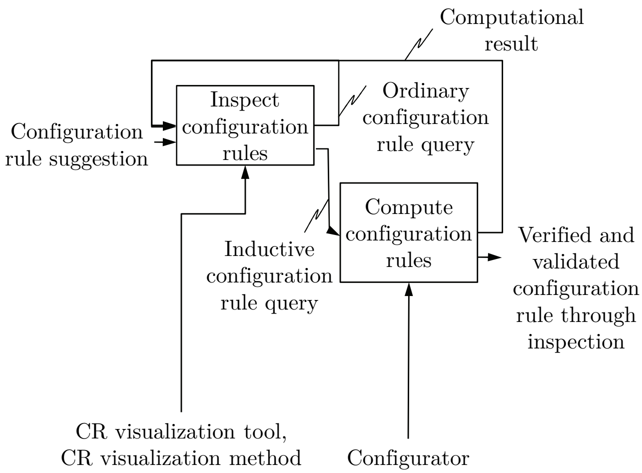


Figure 1.4: The iterative process for inspecting configuration rules.

extract of the process model A1.2). After the inspection of configuration rules has taken place, configuration rule queries have to be stated in order to access or compute the configuration rules. Certain typical rule queries for verifying configuration rules have been exemplified in the literature (Sinz et al., 2003), while configuration rule queries for validation have not been studied. The computational results may be inspected using various formats, e.g. tables of configurations, yes/no to satisfiability queries etc. The identified problem is that a substantial amount of experience is required for stating the configuration rule queries for validating the configuration rules. To cite one discussion with a product structure specialist from a vehicle manufacturing company: “Partly you have to know the answer before you state the question.” Some experienced design engineers know exactly what to ask, while others leave the querying activity to product structure specialists. Those specialists then state control questions to the design engineers if required. The inspection and computational tasks are supported by a CR visualization tool and a configurator. The next section will discuss the problems identified with existing CR visualization tools.

1.2.3 CR visualization tools

One problem with the existing vehicle CR visualization tools is the lack of ease of use, e.g. due to low readability. At the truck manufacturer Scania, the less frequent users describe the company’s CR visualization tool as inaccessible (Pak, 2011). Knowledge about how the tool works and what it can be used for is not widely spread. What is happening is that instead of learning how to use it, the less frequent users are asking the more experienced users to provide the information they want. The effect is that more people are involved than necessary, and that it is more difficult to determine which data are outdated.

An example of a user interface from an industrial CR visualization tool for vehicle configuration rules is shown in Fig. 1.5. The long text string starting with "–((MU5/…" is a configuration rule including 63 feature variants. Each of the codes 3-4 characters long corresponds to one feature variant. To cite one discussion with a design engineer from a vehicle manufacturing company: "It is a source of pride if you know how to find the information needed."

There are two factors that have been identified as potentially causing the CR visualization tools to be difficult to use: the CR visualization method and the iterations between inspection and computations. The CR visualization tools at vehicle manufacturing companies have been supplied with graphical user interfaces on top of the direct displays from the mainframe systems, but little attention has been given to evaluating the CR visualization methods. It should also be investigated whether the CR visualization method somehow could reduce the required number of configuration rule queries.

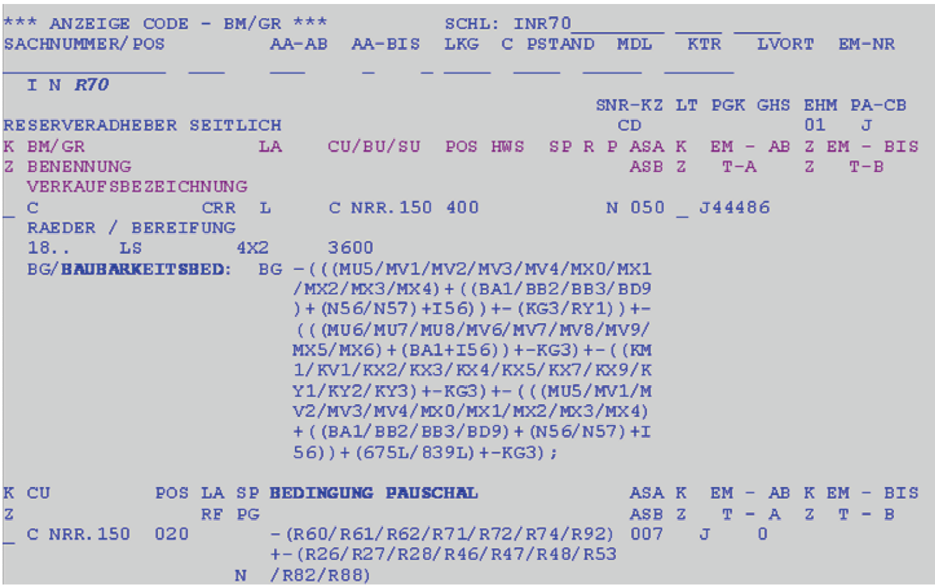


Figure 1.5: Text-based CR visualization, here using only feature variants. From the mainframe PDM system at Daimler, adapted from (Hami-Nobari & Blessing, 2005).

An easy-to-use CR visualization tool is especially needed for less frequent users, e.g. design engineers who able to validate the configuration rules but are also occupied with many other duties. The product structure specialists are today often required to support design engineers for computation of the configuration rules.

The identified problems motivate the research goals described in the next section.

1.3 Purpose

The general purpose of this Licentiate thesis is to identify factors that are causing difficulties when developing configuration rules, to study the development methods and to create tools that facilitate the CR development process. The facilitation should be expressed in terms of number of errors, time efficiency and learnability. By facilitating the CR development process, the hypothesis is that the number of errors discovered at the assembly line decreases. How the purpose of the Licentiate thesis can be accomplished is formulated with research goals in the following section.

1.4 Research goals

The extensive research about product structures is suffering from lack of synthesis, especially the categorization of different types of product structures. The aim of reviewing the product structures is to gain an understanding of what may distinguish different types of product structures, but most importantly to find a common language when discussing the problems during the CR development process. The aim is formulated as a research goal (abbreviated “RG”):

RG1: Propose a suitable information model for vehicle product structures in order to find a common language when discussing the CR development process.

The lack of description of methods for authoring configuration rules as well as the verification checks through inspection prevents improvements to the CR development process. The aim of studying the methods used when developing configuration rules is formulated as a research goal:

RG2: Formalize the methods used when authoring and verifying configuration rules, in order to open up the possibility to utilize computerized process support.

The currently used CR visualization methods at vehicle manufacturing companies do not fully take advantage of the capacity of configurators during the CR development process. The aim of evaluating the CR visualization tools is to find improvements that increase the domain specialists' ability to verify and validate the vehicle configurations. This aim is addressed by a research goal:

RG3: Develop an easy-to-use CR visualization tool that utilizes formalized authoring methods and thereby makes the CR development process more time-efficient and its activities less error-prone.

These research goals are fulfilled by studying the existing literature as a foundation and then stating more specific research questions for the needed research work. The literature is discussed in Chapter 2 with reviews on whether the contributions can be applied to this Licentiate thesis' context. The research questions are further discussed in Chapter 3 in order to show how they relate to existing literature.

1.5 Scope

The research goals will be fulfilled within the scope of this Licentiate thesis. The most important limitation is that only vehicle manufacturing companies have been studied. The configuration rules studied are therefore only used for configuring vehicles. The scope of the research work is further discussed in the following sections by referring to the CR development process.

1.5.1 Authoring of vehicle configuration rules

The focus of this Licentiate thesis during the authoring of configuration rules is on the formalization activity. The formalization of configuration rules uses both the information model and the authoring methods, which are two of the main topics in this thesis. The roles that have been studied are product structure specialists and design engineers, which are the two roles involved in the formalization of configuration rules. Other roles, such as industrial designers, are assumed to collaborate with the design engineers prior to the formalization of configuration rules.

1.5.2 Evaluation of vehicle configuration rules

The activities within the thesis' focus during the evaluation of configuration rules are the inspection and computation. The use of computations, however, is only to semi-automate the inspection, which is in contrast to the more common research topic of the development of computational methods (as in e.g. Sinz, 2003; Astesana et al., 2010; White, 2008). This Licentiate thesis is a result of a collaborative project in the Wingquist Laboratory at Chalmers University of Technology (Chalmers, 2011). A configurator capable of computing vehicle configuration rules has been developed within the research project by other researchers (Voronov et al., 2011). The CR visualization tool developed in this thesis uses that configurator for generating the visualizations.

The inspection is regulated by the CR visualization method implemented in a CR visualization tool. The studied CR visualization methods are limited to the methods found at the studied vehicle manufacturing companies.

The scope of this Licentiate thesis excludes empirical testing by virtual or physical builds, which is one of the activities for verifying and validating configuration rules. This limitation has been motivated in a previous study (Fleischanderl, 2000) where the inspection of vehicle configuration rules as logic expressions was preferred over their representation as CAD data when authoring and maintaining the vehicle configuration rules.

1.6 Outline

The remainder of this Licentiate thesis is organized as follows. Chapter 2 describes the frame of reference and identifies the research needs. Chapter 3 describes the research method including the discussion of the more specific research questions that were derived from the identified research needs. Chapter 4 summarizes the research results and includes summaries of the appended papers. Chapter 5 gives an analysis and a discussion of the results in relation to the research questions. Chapter 6 presents the conclusions and finally Chapter 7 proposes some specific directions for future work. The Appendix contains the full-length versions of the research papers.

2 Frame of reference

In this chapter important research from reference areas is discussed. The references are discussed following a framework of engineering information management systems. First the configurable product structures are discussed, followed by a description of the CR development process. Then, the vehicle CR visualization methods are described, as well as the roles which are developing the configuration rules. Finally, the research needs are identified and analyzed.

2.1 Framework for reference areas

This section presents the framework that is used for structuring the reference areas for this thesis. The framework, shown in Fig. 2.1, describes engineering information management systems and categorizes their different aspects: “processes”, “information”, “information systems” and “organization” (Svensson et al., 1999). All categories in this framework together contribute to its purpose, and none of the parts can be approached as separate phenomena. The information has here been limited to configurable product

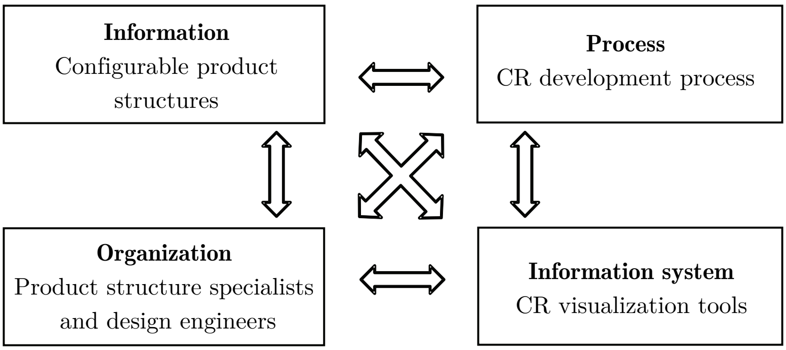


Figure 2.1: Framework for describing technical information systems, adapted from (Svensson et al., 1999).

structures, the process to the CR development process, the organization to the product structure specialists and the design engineers and finally the information system to the CR visualization tools.

All parts of the framework are relevant reference areas. This Licentiate thesis mainly contributes to three of them. The contribution may be visualized by using an Area of Relevance and Contribution (ARC) diagram (Blessing & Chakrabarti, 2009); see Fig. 2.2. The reference areas in bold letters have been actively contributed to with this Licentiate thesis. Other reference areas have been found important for discussing the research results.

According to Hubka & Eder (1988), design research can focus either on the technical system, e.g. a vehicle, or on the processes of designing those vehicles. This Licentiate thesis belongs to the latter processes. The process that is studied for this research work is the CR development process. The vehicle configuration rules are stored in a technical information system.

The following sections 2.2-2.4 discuss areas of the engineering information

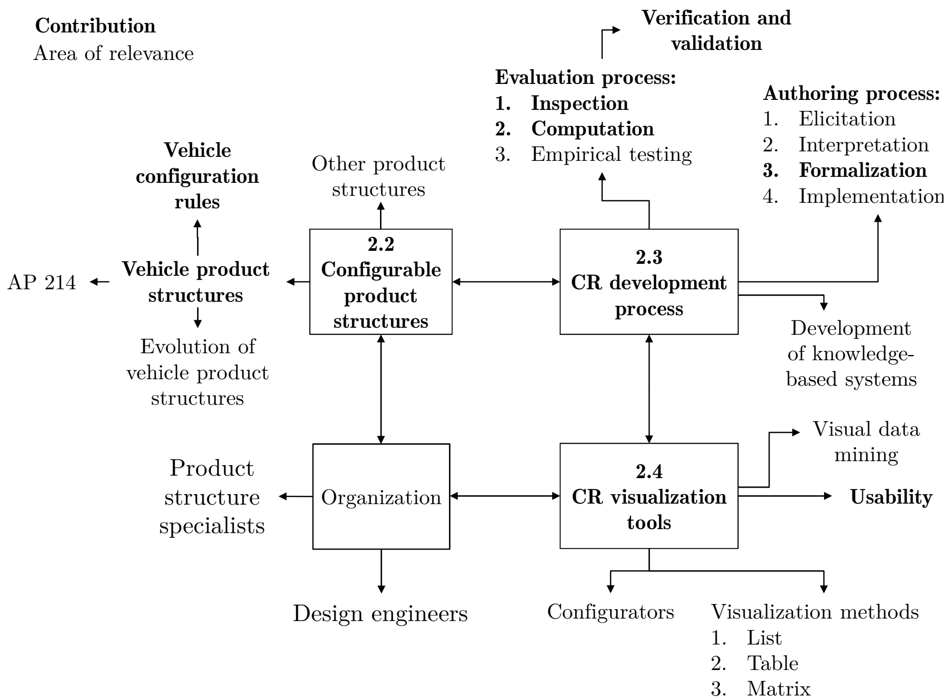


Figure 2.2: Relevant reference areas including where this Licentiate thesis has contributed to in bold letters.

management framework.

2.2 Configurable product structures

This section first defines the term product structure, and then describes some information models for configurable product structures. Based on the discussions during this section, it will be possible to describe the research needs in the end of this chapter. The selection of product structure studied is limited to the ones that have been proven or claim to be applicable for vehicles.

2.2.1 Product structure

One common definition of the term “product structure” is usually referred to as a Bill of Material, which is a “list of all assemblies, subassemblies and parts that form a product” (Garwood, 1988). A more general definition has a wider scope, where “parts” have been exchanged for the more abstract term “elements”, and the “list” has been exchanged for the term “interrelationship”:

The product structure is a set of elements and their interrelationships that describe how a product is built up from a particular viewpoint. (Svensson & Malmqvist, 2002)

The product structure is one aspect of the product “representation”, also called product “model”. With the aim of being more detailed about the definition of product structures, the seven dimensions of the product representation according to Collier (2001) are described here:

1. **Hierarchy**: The hierarchy is the decomposition of large entities into smaller ones, i.e. the breakdown structure of the product from modules to sub-modules.
2. **Variants:** Sets of entities that share characteristics but have alternatives; e.g. all product models describe chassis layouts but the chassis layout is either hatchback, coupé etc. There are different types of variants, e.g. product variants, product model variants, feature variants and part variants.
3. **Interfaces:** Information that defines how entities interact with others, e.g. how the engine module is mounted on the chassis.
4. **Views:** Presenting information for emphasizing one aspect of the product representation, e.g. listing the feature variants may be sufficient as a sales view. Other views may be the engineering and manufacturing views.
5. **Versions:** The records of changes to an entity, e.g. part version B is succeeding part version A.
6. **Status:** The maturity of an entity, e.g. whether it is released for production.

7. **Effectivity:** The conditions for when information may be actively used; e.g. between week 42 in 2012 and week 12 in 2013, the factory may assemble vehicles with a specified part.

The first three dimensions – hierarchy, variants and interfaces – define relations between entities by decomposition, mutual exclusivity and modularity. The configuration rules unfortunately do not fit within those three dimensions, even though they are a fundamental constituent of the product structure.

The fourth dimension, "views", is how information from the product structure is organized and selected with a particular purpose or viewpoint. The last three dimensions define the mechanisms' integrity, and may be considered as attributes of the entities and their relations.

The views described in Collier (2001) are also mentioned in e.g. Isaksson et al. (2000), where they are exemplified by geometry representations, manufacturing information and simulation information. Collier further describes 12 views of the product representation in his "Twelve-Fold way", see Fig. 2.3. The triangle in the "Twelve-Fold way" shows how the design activities can be described as a transformation of two representations, e.g. the order specifications and the manufacturing capabilities gives the manufacturing process designs. The product structure as used in this thesis primarily includes the completeness view and the preferred configurations. The completeness view includes all items that are "present or absent" in a product configuration. The preferred configurations include all products that are offered to the markets, e.g. feature variants, product families etc. It may be assumed that Collier also includes the configuration rules in the preferred configurations view. The following

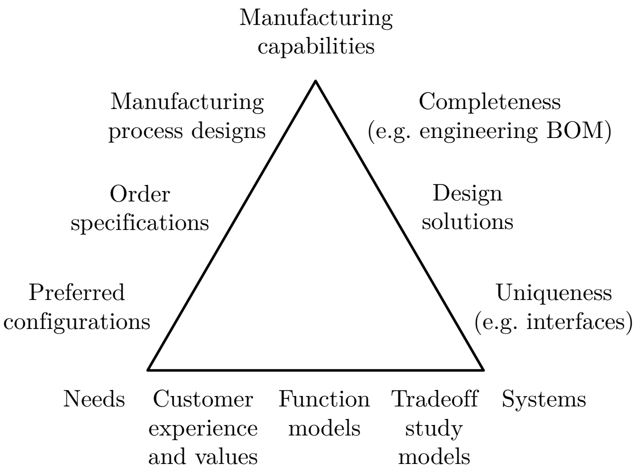


Figure 2.3: The "Twelve-Fold way" of representing products (Collier, 2001).

sections describe some configurable product structures that include more than vehicle configurations and item structure in the definition of product structure. The configurable product structures are categorized into the generations described in the next section.

2.2.2 Characteristics of configurable product structures

The characteristics of configurable product structures are described by their "information models". An information model is a representation of concepts, relationships, constraints, rules, and operations to specify data semantics (Lee, 1999). The information models may be either conceptual or implementation models following the framework in Duffy & Andreasen (1995). According to this framework, the conceptual information model is an early stage of an implementation model. The later is used for implementations into computerized tools. This section will illustrate the evolution of product structures as in Claesson (2006) by showing both conceptual and implementation information models.

The first three generations of product structures shown in Fig. 2.4 are the historical evolution of product structures, while the fourth is the product structure Claesson prescribes which is called configurable components. The evolution of product structures started from "closed" product structures, where every product variant had its own part list and later a part hierarchy. The first description of computerized implementation of a part hierarchy instead of a part list was published in 1969 (IBM, 1969). The hierarchical

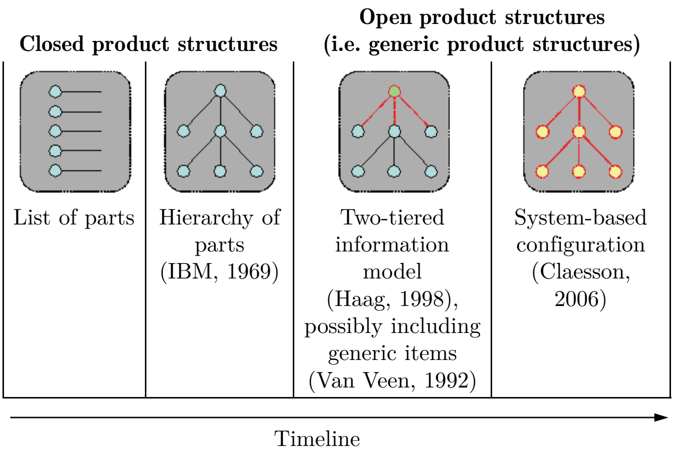


Figure 2.4: Evolution of product description methodologies with increased flexibility, adapted from (Claesson et al., 2001).

database contained the product structure for the Apollo programme, i.e. the Saturn V rocket and the Apollo space vehicle (IBM, 2004). The closed product structures evolved into “open” product structures capable of representing all product variants in one single product structure. This generation of product structures appeared in the early 1980s due to the increasing number of product variants to manage. The term “feature variants” was introduced, since the product variety was too large to define a product number for each developed product configuration (Mather, 1982). The term “two-tiered information model” was introduced by Haag (1998) as a definition of the product configuration first on feature variants and then on items. The two-tiered information model must have a generic item structure which could be instantiated from the product configuration specified by feature variants. Some generic item structures contain generic items; i.e. not only the item structure, but also the single generic item, could be instantiated to an item variant. The two-tiered information model and the generic item structure with generic items will be described in the following two examples.

2.2.3 Two-tiered product structure information models

One example of a two-tiered information model is described by Mesihovic & Malmqvist (2004b); see Fig. 2.5. The first tier, the feature domain, is visualized like a tree with the product family at its root. The product family here consists of several “product groups” (product models), e.g. Volvo V70 and S60. Those product groups are based on “modules”. The engine module may consist of several engine assemblies in different sizes and powers, defined by feature variants; i.e. the feature families and the feature variants are created based on variations in the modules. The definition of a “module” is a group of components that have well-defined interfaces to other components (Erens & Wortmann, 1996). Components are here interpreted as either feature variants or the items constituting the Bill of Material, i.e. parts, software, documents etc. The purpose of creating modules is that the module combinations increase the number of unique product configurations. If a vehicle were completely geometrically and functionally modular, there would be no need of vehicle configuration rules at least not from a technical perspective. The vehicles are not, however, completely modular (Baldwin & Clark, 2000); e.g. an engine with a certain torque will break a gearbox designed for a lower torque. The reasons why configuration rules exist from a technical perspective are e.g. clashes and strength issues. The second tier is the item domain where the items required for a vehicle specification are instantiated from the generic item structure. The blue lines indicate some relation between the objects which may be the configuration rules.

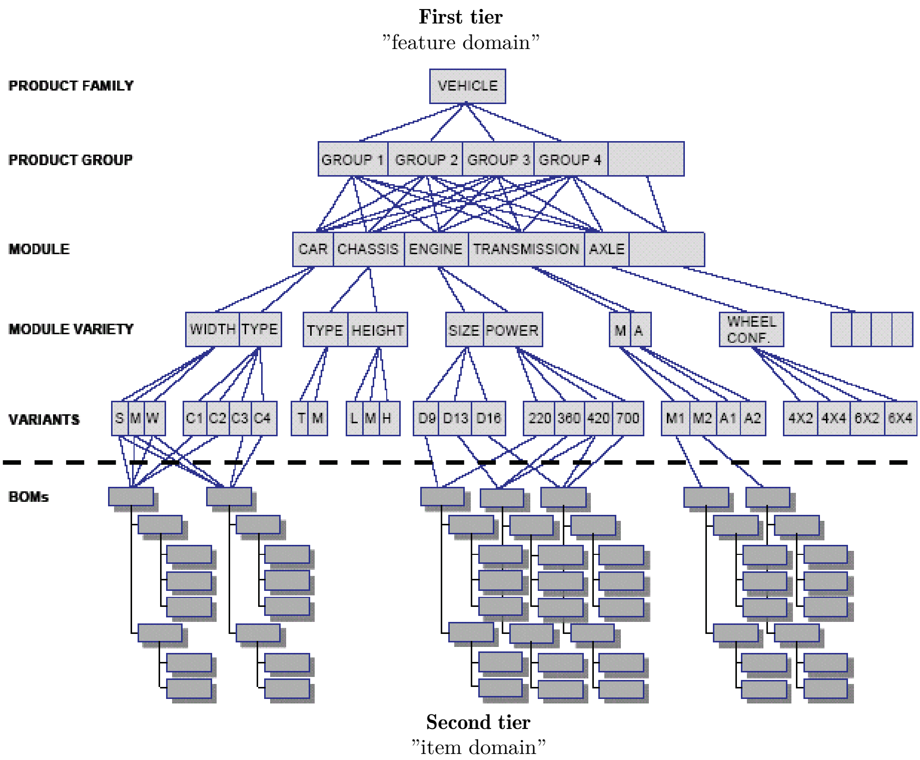


Figure 2.5: Example of two-tiered product structure with instantiation of generic item structure. The configuration rules are marked out with blue lines. Adaptation from (Mesihovic & Malmqvist, 2004b).

Another example of a two-tiered product structure is the generic Bill of Material, henceforth the generic BOM (Van Veen, 1992). The generic BOM, according to Van Veen, contains generic items, which are mutually exclusive items. There are examples of generic item structures in the vehicle manufacturing industry which use the generic items that Van Veen prescribed; see Fig. 2.6. The vehicle model is here the top node of the item structure, which consists of several “main modules”. Those are then divided into “modules”, which may also be divided further into “sub-modules”. In a sub-module, there are several “positions” which may have mutually exclusive “position variants”. The generic item may therefore be identified as the “position”. In this example, the feature domain as well as the configuration rules is left out.

In an attempt to describe a two-tiered information model containing more details about the configuration rules, the standard of implemented product structure

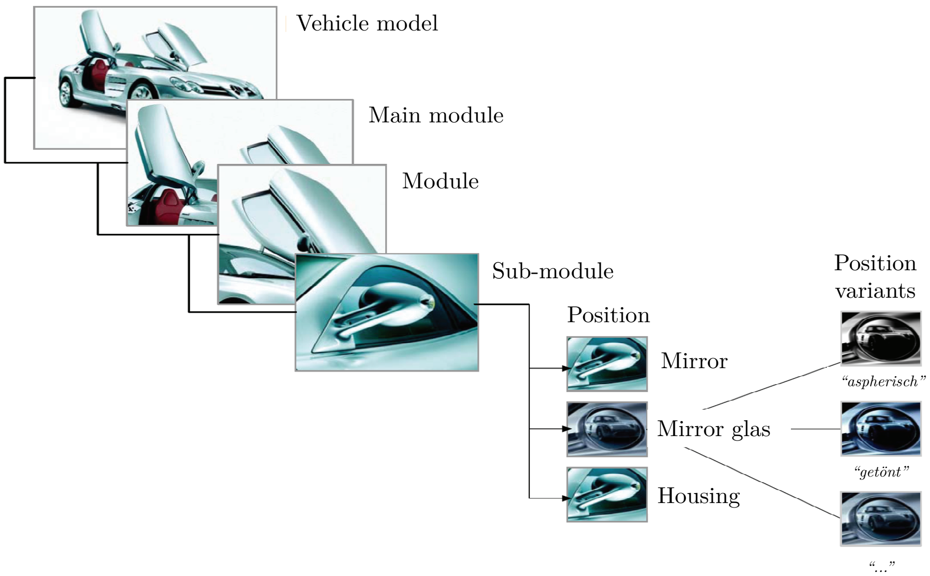


Figure 2.6: Generic item structure with generic items ("positions variants") for the mirror glass (Lamberti, 2011). The corresponding implementation model for this example of generic items also contains the first tier and configuration rules.

information models, STEP, is studied. STEP denotes "Standard for the Exchange of Product Data" (ISO 10303). One of the application protocols of STEP is AP214 – "Core data for automotive mechanical design processes". The class diagram for the standard contains several classes that are common for vehicle product structures; see Fig. 2.7. There are examples where it is demonstrated that AP214 is applicable for modelling product structures from the vehicle manufacturing industry, e.g. at PSA Peugeot Citroën (Viel, 2003). Feature families, feature variants and configuration rules (see Section 1.1.1) have corresponding classes in AP214. The figure shows classes named "specification_category" (equal to the feature family), "specification" (equal to the feature variant), as well as the "specification_condition" (equal to the configuration rules). The condition type for the "specification_condition" may be either "part_usage" (which controls the usage of parts) or "technical" (technical dependences between Specifications). AP214 is described on many pages, and it is very difficult to get an overview of which classes are used when developing the configuration rules.

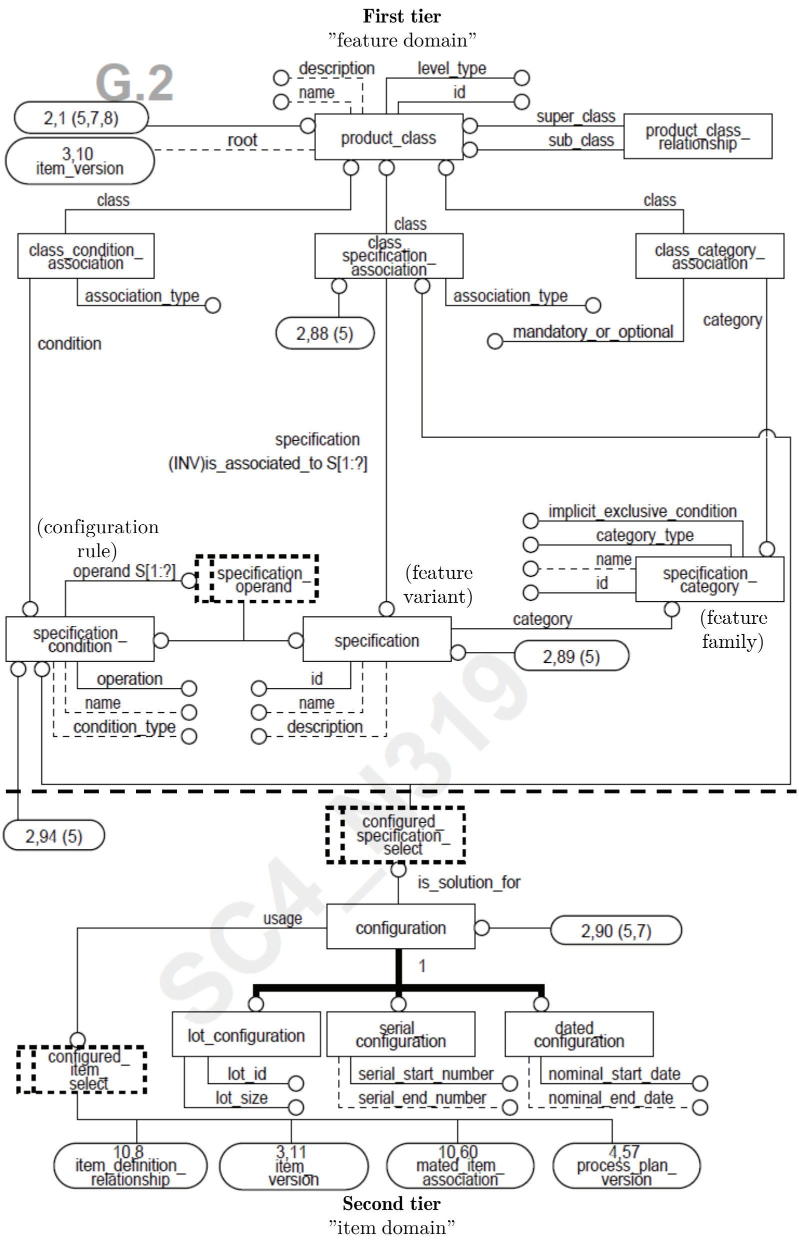


Figure 2.7: EXPRESS-G representations of objects used for the management of diversity in AP214 (ISO, 2004).

2.2.4 System-based information models

Let us turn to the fourth generation of product structures, which was the system-based product structure (Claesson, 2006). System-based configuration is achieved by incorporating interfaces between subsystems whose interactions configure the product. For a single product configuration, there are no iterations between the specification of feature variants and the instantiation of the generic item structure in the two-tiered information model. With a system-based information model, the feature variants are interactively specified, as the subsystem interacts through their interfaces. Each subsystem contains a generic item structure.

In Collier (1999) there is a description of how interfaces controlled by Interface Control Objects model the parts to product configuration; see Fig. 2.8. This research was a result of long company visits, among others to Ford Motor Company. The characteristic of a system is that there are no roots or leaves, as on a tree. The system architecture is also used in the configurable component approach (Claesson et al., 2001).

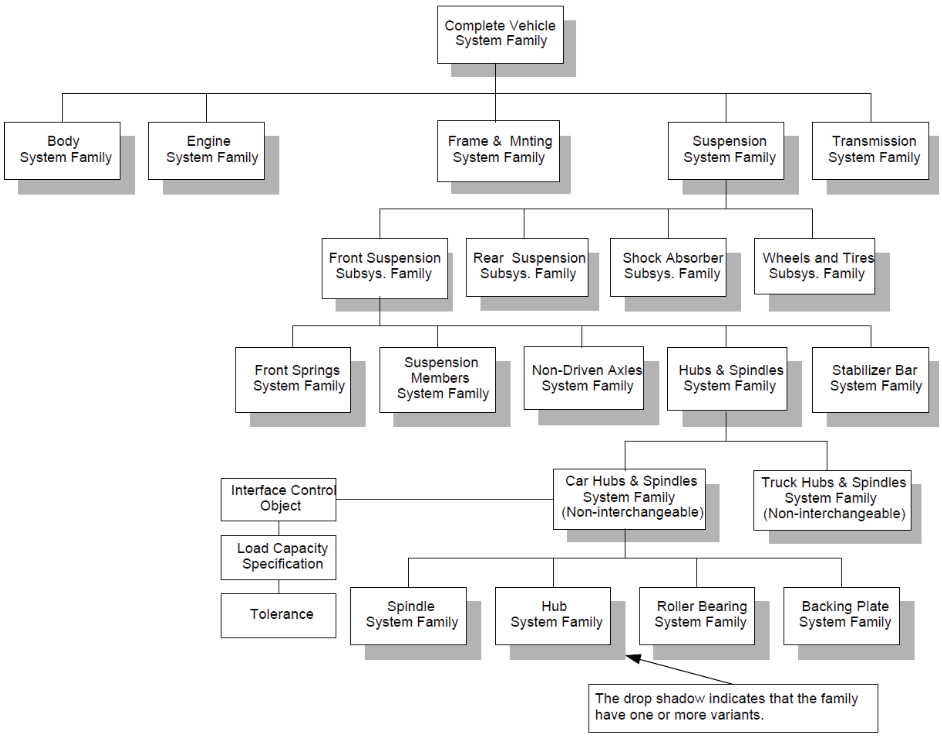


Figure 2.8: System-based partial product structure with interface control objects executing the configuration (Collier, 1999).

The system architecture is more strongly applied here, as the configuration rules are local residents of the configurable component. In Collier (1999) the configuration rules were still global for the entire product family. The configurable component approach also uses interfaces (called CRI) between configurable components; see Fig. 2.9. The configurable components also contain the function-means tree as well as performance models, which are not shown in the figure. The configurable component approach therefore addresses the challenge of capturing the design rationale for the product, requiring manual data collection (Huang et al., 2007). The configurable component also has a parameter interface (VPI) with feature variants that could be used for the configuration rules inside the configurable component.

The configurable component approach utilizes the function-means tree; see Fig. 2.10. The function “dry clothes” is decomposed and determined by means such as “revolving drum”. The means are realized by parts. The function-means tree was first described by Tjalve (1979), and should be interpreted as a graphical representation of Hubka’s law which states that the developments of functions and parts are conducted in parallel. The function-means tree therefore describes the design activity rather than the

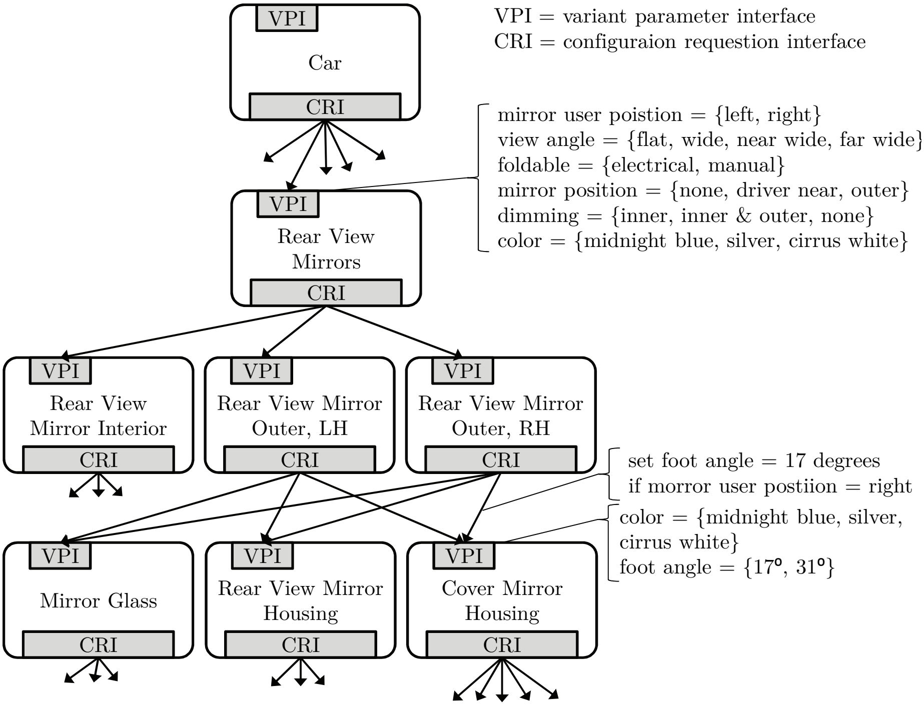


Figure 2.9: The approach “configurable components” (Claesson et al., 2001).

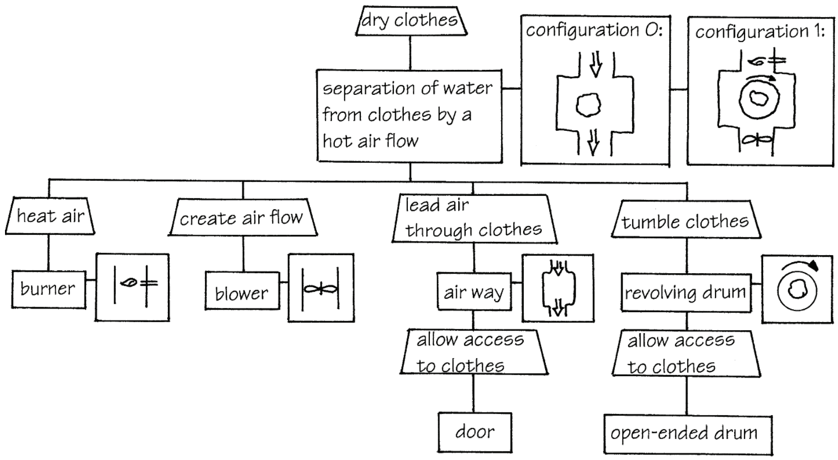


Figure 2.10: A function/means tree for a tumble dryer, adapted from (Hansen, 1995).

configuration rules. The function-means tree is similar to a matrix-based method called axiomatic design (Suh, 1990). The function-means tree may also be divided into two domains, functions and means (sometimes called "organs"), as in the chromosome model (Andreasen, 1992). The domains then have "causal links" stating the dependences between the domains.

One product structure which is difficult to place in the evolution of product structures is described in Soininen et al. (1998). The product structure is based on "components", "resources", "connections" and "ports". There is an example of a computer described in this product structure in Männistö el al. (2001b). In Soininen et al. (1998), it is argued that the suggested product structure is a synthesis of the known configuration approaches, and therefore is the most generic information model presented at the date of publication. It is, however, not obvious whether there are both items and feature variants in the example. For vehicle product structures, it may be questioned whether all the elements of the prescribed product structure are necessary or whether introduction of any of them facilitates the CR development process. The information model elements of configurable product structures have been defined in Section 1.1.1. Further classification of the configuration rules may be found in Paper A, Section 4.1.2.

2.2.5 Conclusions

It was discovered when studying the configurable product structure literature that most of information models described were conceptual, i.e. it is not possible to implement a configurator using only the conceptual model. The few examples found of implementation models, e.g. AP214, are too detailed and require specialist competence to fully understand how to follow the information model. The conceptual models often lack rigor when it comes to describing configuration rules, e.g. the information model in Mesihovic & Malmqvist (2004b). The AP214, which is an implementation model, is described in a long document. The generalizing attempt described in Soininen (1998) for creating a general product structure adds classes that may not be applicable to the vehicle manufacturing industry, and does not distinguish between the feature and item domain. The development of configuration rules is already found to be difficult, and it is hard see the benefit of introducing a more complex information model. A preferable approach would be to generalize vehicle product structure with the limitation to the classes that are necessarily used during the CR development process.

2.3 CR development process

This section continues from the description of the CR development process in Sections 1.1.2-1.1.3. The aim here is to discuss the methods that are used when developing configuration rules. These methods were found by studying the research on development of knowledge-based systems. The section starts with a discussion about automation of the development process, followed by a more detailed discussion of authoring and evaluation methods for configuration rules. Finally, the conclusions drawn from this section are stated.

2.3.1 Automation of the product development process

According to Cederfeldt (2007), the automation of the product development processes can be divided into two types: information management (e.g. storage and retrieval) and knowledge processing (e.g. computations). An information system managing the configuration rules and capable of doing computations with them is automating the information management as well as the knowledge processing. The automation type defined as information management is not studied in this Licentiate thesis, which has previously been mentioned in the delimitation of not studying configuration management principles. The knowledge processing is within the scope of the Licentiate thesis, if it can be used for making the CR development process less error-prone and more time-efficient.

According to Sunnersjö (1994), there are two criteria when evaluating the automation potential of the product development process. For a high automation potential, the “product variety” should be high and the product knowledge be “mature”; see Fig. 2.11. The product variety is suggested to be measured by the ratio of developed to manufactured products. Sunnersjö defines the metric for product knowledge maturity as the percentage of "known rules in relation to all rules". If all known rules were equal to all rules, the product development process would be possible to automate completely. To reach this maturity, the rules have to be formalized, e.g. by the use of logic expressions. The "fan ventilator system" has to be developed for every customer, e.g. with simulations of fan capacity. Each developed fan ventilator system is manufactured once, and therefore the product variety is high. The process maturity is also fairly high due to the use of simulations. The development of the "heavy welded steel structure" is apparently not as automated as for the fan ventilator system, but is also manufactured once for every developed product. The two aspects product variety and process maturity will now be discussed in the application to development of configuration rules:

- **Automation potential due to product variety:** The products that were studied in Cederfeldt (2005) gave values within the range 0-100%. The products that were studied were standard products with "dimensional variations" (customized products), e.g. shop equipment, heavy welded steel structures, fan ventilator systems etc. Configurable and variant-rich products, such as vehicles, have a higher

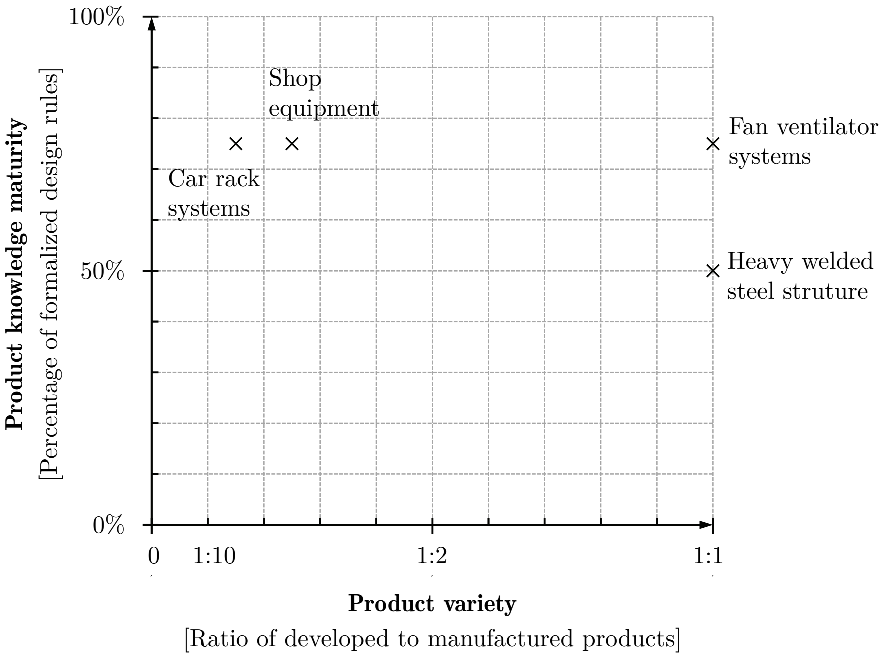


Figure 2.11: Automation potential as function of process maturity and product variety, adapted from (Cederfeldt, 2005).

number of developed product variants than the manufactured product variants. According to this reasoning, vehicles have a high automation potential due to the product variety. There are also alternative criteria for the product variety, e.g. the development cost as percentage of the revenue (Cederfeldt, 2007). This measurement would indicate whether the design process is costly compared to the income. Collection of data from annual reports gave 3% for Fiat S.p.A (Fiat, 2010), 4% for Ford Motor Company (Ford, 2010) and 5% for AB Volvo (Volvo, 2010) and Daimler AG (Daimler, 2010). Those figures are very low percentages, but the amounts are in billions of euro/dollars. When assessing the automation potential from a cost perspective, it would be more meaningful to look at the activities that may be automated and the cost for those, e.g. the number of hours spent on developing configuration rules.

- **Automation potential due to product knowledge maturity:** An increased product knowledge maturity gives higher automation potential. A component of the product knowledge is its development methods. The formalization of development methods facilitates the use of computerized tools. Formalized rules are here defined as logic expressions. The product knowledge is gained and used during the development process. An example of a vehicle development process is shown in Fig. 2.12, where the process has been divided into "finding of product" and then "development of product". The finding of product involves the early stages of the product development, which includes the technical description of product configurations.

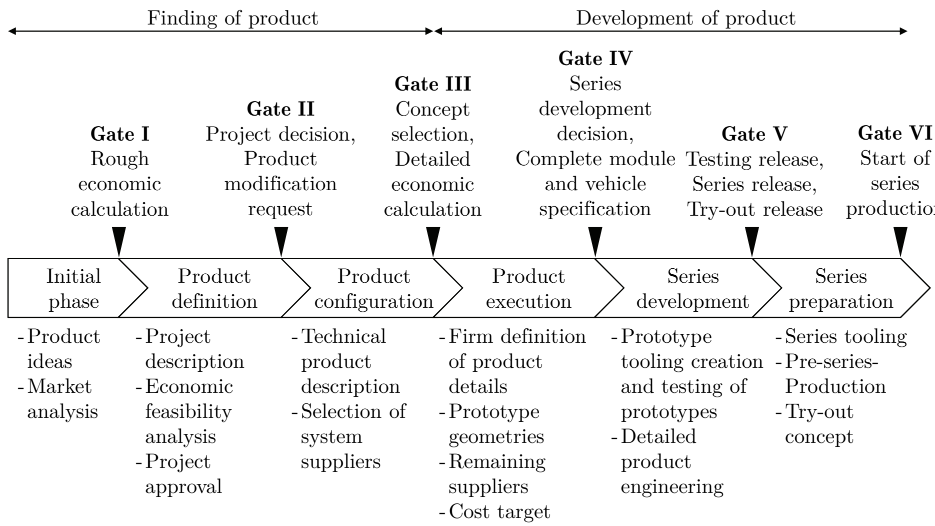


Figure 2.12: Example of vehicle development project phases (Oeltjenbruns et al., 2000).

The development of product includes the generation of geometries as well as the complete documentation of module and vehicle specifications. One of the sub-processes of the product documentation is the CR development process. By referring to the activities during the product development as engineering activities, Catic (2011) studied the automation potential by the use of knowledge-based engineering. As the CR development can be compared to development of a knowledge-based system (Soininen et al., 2001), the vehicle CR development process consists of the three phases: author, evaluate, and release vehicle configuration rules (Neubert, 1993). The vehicle CR development process is mature when the authoring and evaluation methods have been formalized. As the number of formalized rules for developing configuration rules increases, the potential for automation also increases. The research work for this Licentiate thesis aims at an increased product knowledge maturity, which is the second criterion for automation potential.

A study with 12 Swedish companies collected, according to six categories, the motives for implementing development process automation (Amen et al., 1999). The usability factors addressed in this Licentiate thesis, for a computerized tool automating the development process, are measured in terms of errors (how many and how severe), learnability (how easy it is to accomplish a task the first time) and efficiency (how quickly) (Nielsen, 1993). Comparing the motives for development process automation with the addressed usability factors gives some matches:

- Laborious design tasks: 75% (time efficiency)
- Quality assurance: 58% (number of errors)
- High repetition frequency: 50% (time efficiency)
- Lead time minimization: 50% (time efficiency)
- Highly optimized design: 50%
- Establish knowledge bank (17%)

The most frequent motive for development process automation is to make it more time- efficient. The sought-after facilitation of the CR development process should be measured in terms of number of errors, time efficiency and learnability. The increased time efficiency and the decreased number of errors were found as motives for automation. Increased learnability, however, is not addressed as a motive for automation. The development process automation may therefore address two of three criteria for the Licentiate thesis' purpose. This motivates the further study of the authoring and evaluation methods in the next sections in order to assess its automation potential.

2.3.2 Authoring of configuration rules

The vehicle configuration rule set is an example of a “knowledge base”. A definition of knowledge as it is used here is that it includes objects, concepts and relations that are assumed to exist in some area of interest (FOLDOC, 2011). Here, the knowledge is the configuration rules. The authoring of vehicle configuration rules may be described with the general process for knowledge acquisition presented by Neubert (1993), i.e. the four-step process of “elicitation”, “interpretation”, “formalization” and “implementation”. Those steps were later used in development processes for knowledge-based systems, e.g. the “model-based and incremental knowledge engineering (MIKE) approach (Angele et al., 1998). The shift between informal and formal description of knowledge in the MOKA approach, i.e. Methodology and tools Oriented to Knowledge-based engineering Applications (MOKA, 2000). The following sections describe the authoring process in IDEF0 model terms and give an example of how an authoring of a configuration rule takes place.

The IDEF0 diagram for the authoring process is shown in Fig. 2.13. The authoring process is initiated by the product modification requests requiring new or modified

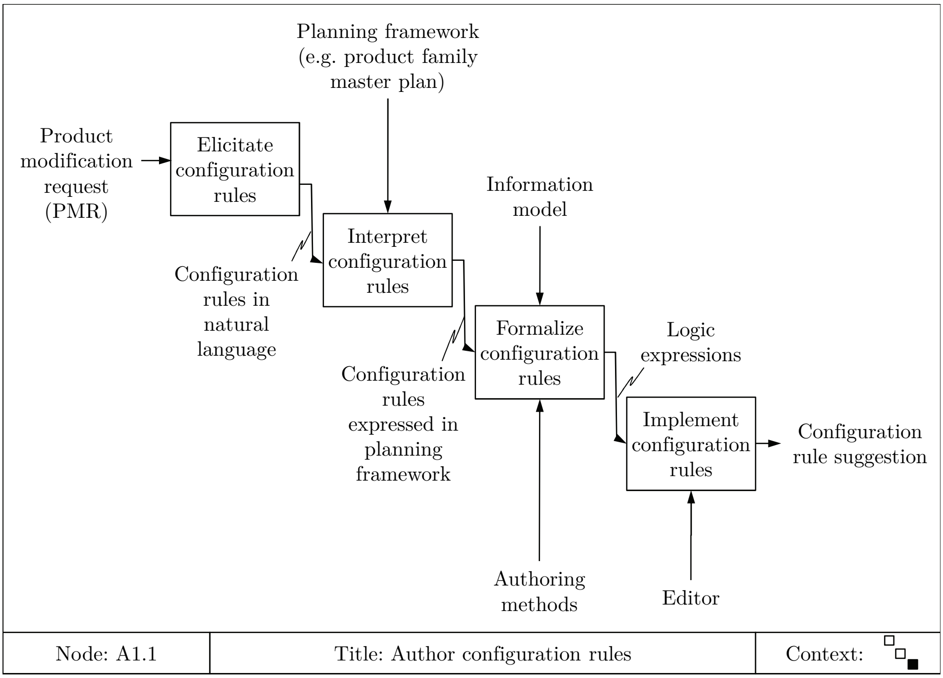
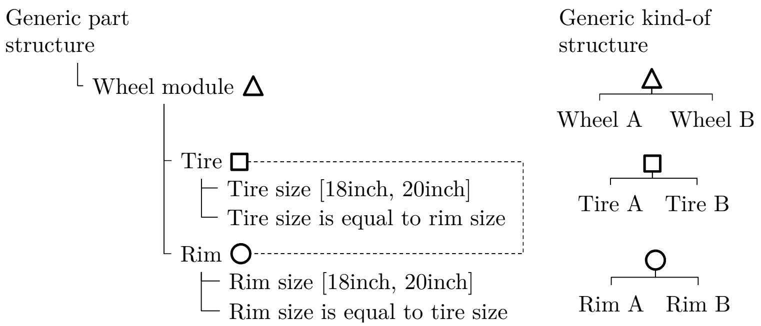


Figure 2.13: Process model for the authoring of configuration rules.

configuration rules. The elicitation is the activity to create an informal description of the configuration rules, often in natural language format. The natural language phrasing of the configuration rules may be interpreted to result in a planning framework, e.g. a product family master plan (Kvist, 2010). The product family master plan includes a visual representation of the configuration rules. A product family master plan cannot directly be used in configurators, as e.g. the configuration rules are simplified. Other types of planning frameworks may be matrices and trees that are used for planning the configuration rules, but not used for the implementation of configuration rules. Those planning frameworks are a foundation for the formalization of configuration rules. Finally, the formalized rules may be implemented in an information system, e.g. a configurator, by using en editor.

The activities during the authoring of configuration rules are illustrated with an example in Fig. 2.14. The first activity is the elicitation of the configuration rules, i.e. a formulation in natural language of the new configuration rules. The next activity is the interpretation of the natural language formulation into a planning framework, e.g. the product family master plan, henceforth PFMP (Mortensen, 2000; Harlou, 2006). The planning framework gives an overview of the configuration rules, but is not sufficient for the their implementation. They must first be formalized into logic expressions. The PFMP has been applied in industrial cases, where one of the evaluations was: “PFMP does not tell us how the configuration rules are maintained, but it is the foundation for being able to maintain them” (Mortensen et al., 2000). Planning frameworks for the configuration rules do exist at vehicle manufacturing companies, as interpretations of

1. **Elicitation:** “The tire has the same diameter as the rim, either 18 or 20inch.”
2. **Interpretation:**



3. **Formalization:** “NOT(20tire AND 18rim)”, “NOT(18tire AND 20rim)”
4. **Implementation:** Typing “–(20tire+18rim)” and “-(18tire+20rim)” in the editor

Figure 2.14: The authoring process of configuration rules.

the project prerequisites and planning of vehicle projects. They are not, however, sufficient for authoring and maintaining the configuration rules.

A problem often cited as occurring during the authoring process is the “knowledge acquisition bottleneck” (Schreiber et al., 1993). It has been found very difficult to extract and formalize the necessary information from the people having the knowledge. The knowledge acquisition bottleneck is addressed in this thesis by study of the formalization step and its use of authoring methods.

2.3.3 Evaluation of configuration rules

The configuration rule suggestion from the authoring process is the input to the evaluation process; see Fig. 2.15. The outcome of the evaluation process is “verified” and “validated” configuration rules. For knowledge-based systems, the terms “verification” and “validation” have definitions that will be used for this Licentiate thesis. Verification is the search for structural errors or errors of form in the system, while validation is the search for errors of substance in the system (Juristo & Morant, 1998). The distinction between validation and verification used is that the verification fulfils some stated requirement, such as that every vehicle should have a steering wheel – while validation

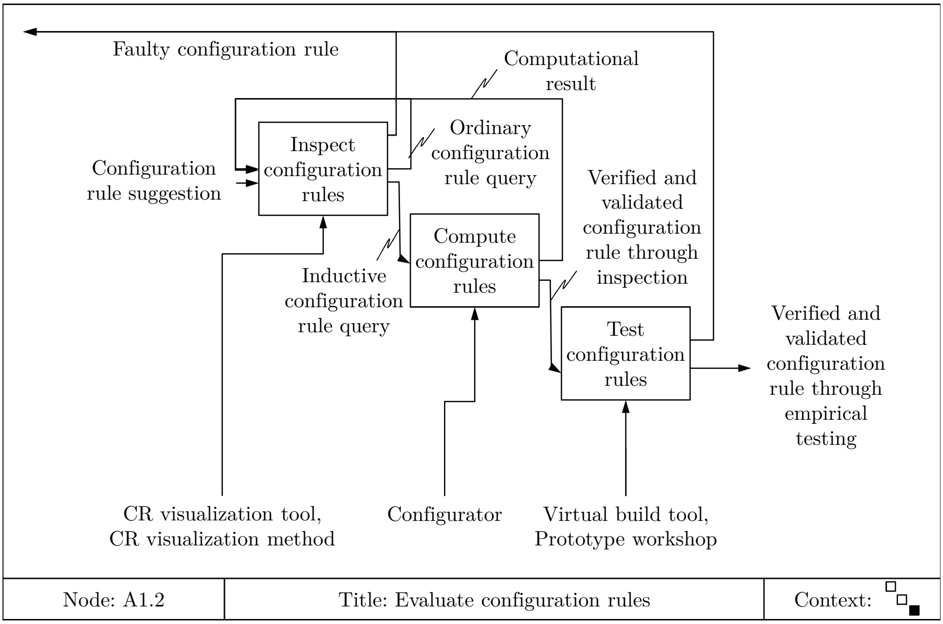


Figure 2.15: Process model for the evaluation of configuration rules.

is about developing the right vehicle, e.g. every vehicle should have the correct steering wheel. An example of verification and validation for the natural language: verification of grammatical errors may be automated but the sentence does not have a valid message if put into the wrong context. The activities for verification and validation of knowledge bases can, according to Meseguer & Preece (1995), be clustered into three activities. These activities have been further elaborated by Baumeister & Freiberg (2010) and have here been applied to vehicle configuration rules. The first activity in the three-step evaluation process is the “inspection”, followed by the “computation” and “empirical testing” of vehicle configuration rules. The following sections will describe those three activities.

Inspection of configuration rules

Visualization supports humans in dealing with decisions that cannot yet be automated (Wong, 1999). The inspection of configuration rules is the visual examination conducted with a CR visualization tool. The inspection of configuration rules takes place by visualizing the configuration rules for a “domain specialist” (Tuhrum et al., 1988), e.g. design engineers of frame suspensions, brakes, engines etc., or other roles with specialist knowledge about a domain, e.g. a “product structure specialist” who reviews the configuration rule suggestions.

According to Mesihovic & Malmqvist (2004a), many industrial companies have dedicated roles within product development for reviewing the configuration rule suggestions from the design engineers. This reviewing role is in this thesis called product structure specialists. The product structure specialists analyze and execute the configuration rule suggestions. There are many stakeholders communicating with the product development department for discussing the configuration rules, e.g. finance, marketing, manufacturing and product planning. This Licentiate thesis, however, assumes that roles involved in the authoring and evaluation of configuration rules are limited to the design engineers and the product structure specialists.

The inspection detects mistakes in the content of the configuration rules. By inspecting the configuration rules, the domain specialist may *validate* the configuration rules by comparing his/her knowledge about which vehicles should be allowed to be built with the CR visualization. The configuration rules are assumed to be valid if they are consistent with domain specialists’ perceptions of which vehicle configurations should be allowed to be built and the items assigned to those. By inspecting the configuration rules the domain specialists may further identify errors. These errors are potentially caused by not only one but several interacting configuration rules. As long as this error detection involves users that are comparing their not yet formalized knowledge with the implemented configuration rules, this activity is referred to as validation of configuration rules. The user may *verify* the configuration rules by detecting errors in the product structure such as redundant configuration rules if this detection is formalized and possible to implement in a computerized tool.

Computations of configuration rules

The computations are performed by first stating a configuration rule query, which may be either ordinary or inductive (Boulicaut & Masson, 2005). Ordinary rule queries are used to access data, e.g. by searching for all configuration rules containing a specified feature variant. Inductive rule queries ask for information not directly found from the data itself, but from data processing e.g. by using a configurator. An illustrative example of an inductive rule query is whether the “fuel tank 60 liters” may be combined with “adaptive headlights”. Those two feature variants are not directly related with a single configuration rule, but with a configurator it is possible to detect if there are any chains of configuration rules that are restricting the feature variant combination. Complex configuration rule chains, which consist of many interacting configuration rules, are difficult to assess without computations, e.g. the use of configurators. When the configuration rules are evaluated in computations, they are executed for checking some aspect of configuration rules, e.g. whether every vehicle has exactly one steering wheel. Some of the *verification* checks that are done manually through inspection may be automated using computations if the checks are formalized into a computerized tool. Also, the result of the computations may be inspected by a domain specialist for *validating* the configuration rules.

Iterations to the authoring process take place if any faulty or missing configuration rules are discovered from the computations. When there are no further iterations needed, the configuration rule suggestion has through inspection become verified and validated configuration rules. The automated verification of knowledge bases has been studied thoroughly in the past (e.g. by Preece, 1998; Gaupta, 1990; Ayel & Laurent, 1991). In Sinz et al. (2003) it is described how formal methods may be applied for configuration rules verification, but the authors admit that in industrial applications the validation activity requires additional knowledge not yet formalized.

There are also process models where the computations are conducted prior to the inspection, e.g. in the viewpoint from the literature on knowledge discovery in databases, henceforth KDD. The knowledge discovery process aims at finding simplified representations, i.e. patterns. First, the user needs to get an overview of the data. In the simplified overview, the user identifies interesting patterns and then the user needs to drill down and access details of the data (Kleim, 2002). This is also referred to as the “information-seeking mantra” (Shneiderman, 1996).

There are several process models in the literature for knowledge discovery (e.g. Fayyad et al., 1996; Anand & Buchner, 1998; Cios et al., 2005; Cabena et al., 1998; and Wirth & Hipp, 2000). Among these, the most detailed description of activities and their outcomes is the process model suggested by Fayyad et al.; see Fig. 2.16, which here has been modelled using IDEF0. The input of the knowledge discovery process is simply data, which then are transformed into knowledge as the process output. The first activity is to select the data that are going to be analyzed. They may e.g. be a set of item usage rules. This selection of data will be pre-processed and transformed, e.g. following strategies for missing data fields. The transformation of data prepares the data for the analysis, and is dependent on the data mining methods. “Data mining” is the analysis of large-scale data and aims to extract useful information. One of the data mining methods is the generation of “association rules” for discovering interesting relations between variables. The visual presentation of the association rules gives a visual data mining method. The association rules were first used at supermarkets (Agrawi et al., 1993); e.g. the rule IF (“butter” AND “bread”) THEN “milk” indicates that customers that were buying butter and bread most likely also bought milk. This example is based on statistics, but the data mining methods may also be based on logics. The pattern created from association rules should be simpler than an

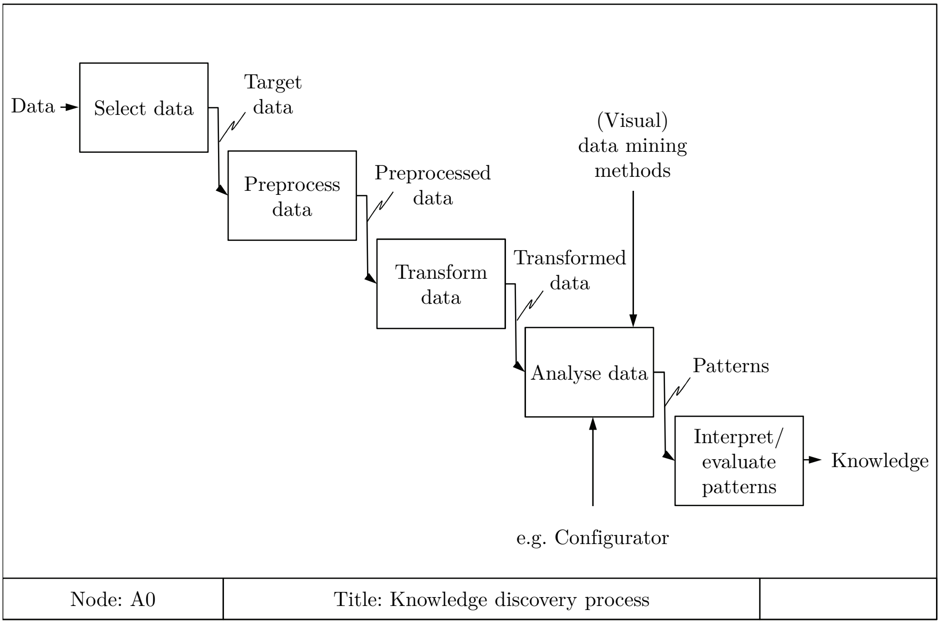


Figure 2.16: Knowledge discovery process, adapted from (Fayyad et al., 1996).

enumeration of data that were required when generating the association rules. The association rules may therefore not be equal to the configuration rules themselves. The application of association rules to vehicle configuration rules is not, however, described in the existing literature. The last activity is an interpretation and evaluation of the patterns found from the data analysis, where the discovered associations are studied. The knowledge discovery process is modelled without iterations, but those may occur between any of the activities.

Empirical testing of configuration rules

The empirical testing is used for testing the configuration rules by executing them on sample vehicle configurations and then building these vehicle configurations either virtually or physically. Virtual builds for empirical testing of sample vehicle configurations is studied by Fuxin (2005). To guarantee complete correctness, testing has to be exhaustive: every potential input should be tested. This is not feasible for complete vehicle configuration rules sets, so empirical testing only studies a subset of all allowed vehicle configurations. The vehicle configuration rules that are empirically tested then become verified and validated through empirical testing.

The last activity in the CR development process is the release of configuration rules, which is described in the next section.

2.3.4 Release of configuration rules

The CR development process as described in this thesis starts with an approved change request. The CR development activities aim at implementing the change, e.g. a modification of the configuration rules. As the engineering change has been implemented, it is released for the next phase, the “manufacturing implementation of change” (VDA, 2010). Thereby the configuration rules are handed over to the users in the next development phase.

2.3.5 Conclusions

The automation potential during the CR development process would increase if the methods used were formalized. Attempts to apply existing literature on knowledge-based systems have yielded some alternative generic process models; e.g. the process for development of knowledge-based systems has some alternative activities compared to the process for knowledge discovery in databases. The studied process models are, however, far too general for being able to describe formalized methods. For example, during the inspection of configuration rules, it is not clear what is inspected and why.

2.4 CR visualization tools

The aim of this section is to show some examples of industrial CR visualization tools in order to categorize their CR visualization methods. The section starts by describing the information systems where the industrial CR visualization tools are found. Then some examples of CR visualization methods are described, classified into list-based, table-based and matrix-based methods. Finally, conclusions are drawn.

2.4.1 Industrial CR visualization tools

Information systems are manual or computerized ways of handling information. An information system may be used for the collection, processing, storage, search, retrieval, transfer and presentation of information (Duffy & Andreasen, 1995). The information system with the industrial CR visualization tool is called a Product Data Management system, henceforth PDM. Product Data Management is the use of software or other tools to track and control data related to a particular product. The configuration rules are stored in the database of the PDM systems. When commercial PDM systems became available in the late 1980s, they were promoted as being able to increase the efficiency of the information management process (CIMData, 2001). Their use was initially focused on management of documents, e.g. CAD files. The central database of PDM systems may also manage metadata such as “owner” of a file and release status of the components, control check-in and check-out of the product data as well as modelling and maintaining the “product structure”. Two categories of data can be distinguished in a PDM system (Guyot et al., 2006):

- metadata, i.e. “data about data”, representing the product structure, but possibly also containing revisions, which people have access to the data etc.;
- files which are linked to the metadata, e.g. CAD files or other documents.

Improved capabilities for configurator support have been introduced in major PDM systems such as Windchill (PTC, 2011), Enovia Variant Configuration Central (Enovia, 2011) and Teamcenter (Siemens, 2011). Fig. 2.17 shows proposed system architecture for the use of configurators in a PDM system. There are two interfaces, the “system-developer” interface and the “end-user” interface. The system-developer interface is used for visualizing the product structure, e.g. configuration rules and metadata for components (creator, versions, etc.). The tool for visualizing the configuration rules is in this thesis called a CR visualization tool.

The following three sections will describe the CR visualization methods that are applicable for the industrial CR visualization tools. A CR visualization method is a “systematic and rule-based graphical representation aiming to acquire insights, develop an elaborate understanding or communicate experiences” (Lengler & Eppler, 2007).

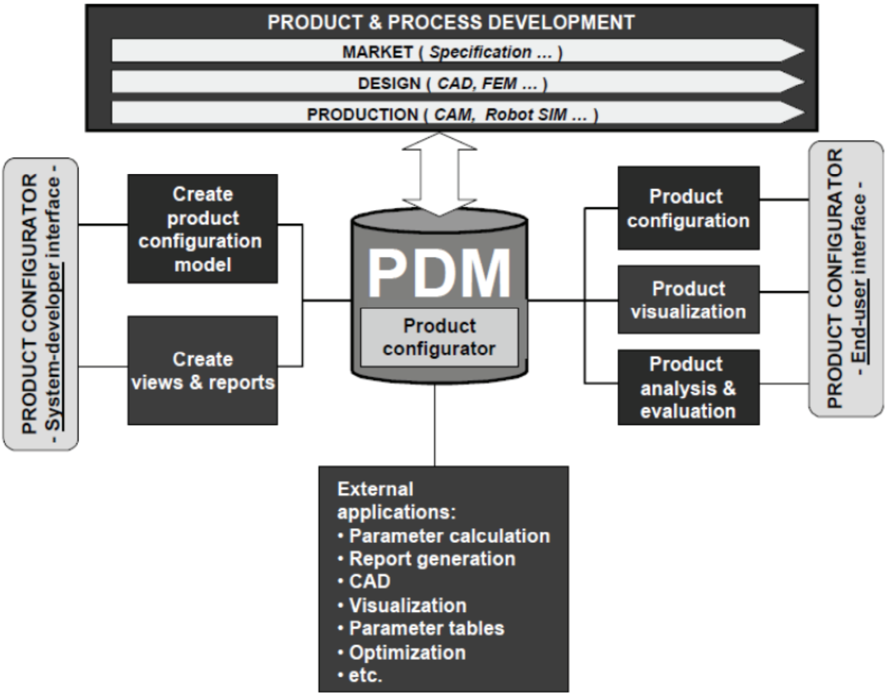


Figure 2.17: Product configurators used in the PDM system
(reprinted from Mesihovic & Malmqvist, 2000).

Examples of CR visualization methods are tables, pie charts, mind maps, Gantt charts, decision trees etc. The visualization methods that will be discussed here are the list, table and matrix, as they are the methods currently used in CR visualization tools.

2.4.2 List-based CR visualization methods

The majority of the commercially available configurators support the sales configuration process (Sabin & Wiegel, 1998). A sales configurator is often based on list menus or checkboxes. One example of a sales configurator is shown in Fig. 2.18. The selection of feature variants is marked out with a filled blue square, e.g. for the "Smokers package". The combination of feature variants potentially results in conflicts or implications, as the selection of "Adaptive headlights" will automatically also imply the selection of "Xenon headlights" and "Headlight wash". The configuration rules in this example are based on logic operators.

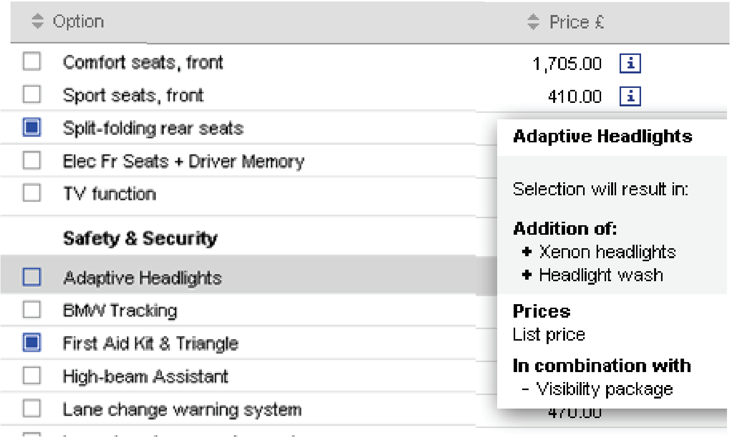


Figure 2.18: List-based sales configurator (BMW, 2011)

Another example of a commercial configurator is Tacton, whose list-based visualization of configuration rules is shown in Fig. 2.19. In the example, the first configuration rule is that "voltage" must be equal to or less than the "total_resistance" multiplied by the "current". The example shows some more advanced rule features, e.g. continuous feature variants are allowed. Also, the use of mathematical operators, e.g. the multiplication, is in contrast to the logic operators used in the previous example of vehicle configuration.

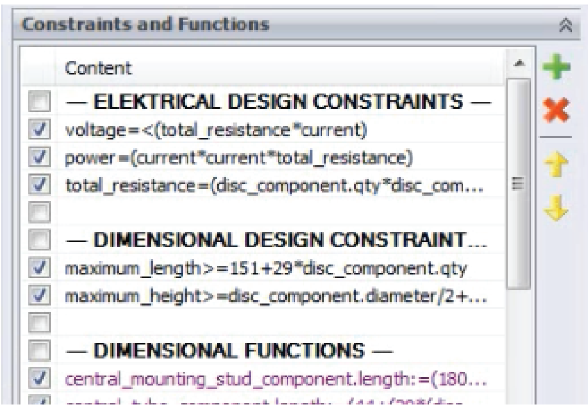


Figure 2.19: List of configuration rules in commercial configurator (reprinted from Tacton, 2011).

2.4.3 Table-based CR visualization methods

A user interface for configuration rules using a table-based CR visualization method is displayed in Fig. 2.20 (Lamberti, 2011). The user had searched for the bill of material [Stückliste] for the generic item with ID “0500” describing the “mirror shell” [Spiegelschale] in the sub-module “exterior mirrors” [Spiegel aussen]. There are 13 alternative mirror shell items, which are mutually exclusive. In the table showing the bill of material, the first column “POS” shows the ID of the generic item. The second column “PV” is the Item ID, e.g. “010” and “015”. The third column “ST” is the quantity. The fourth column “Teil” is the part number, e.g. “A 221 810 27 21”. The description of the item is found in the column “Benennung”. The configuration rules are written in a string format, with the column heading “Codebedingung”. The feature variant codes are written in a format three characters long, e.g. “234” and “494”. The configuration rule for POS “0500” and PV “015” should be read as IF(“234”) THEN(“A 221 810 13 21”), where the latter is a part number. When looking through the configuration rules in the column “Codebedingung” it becomes clear that some appear twice, e.g. the “800”, even though the 13 items should be alternative. What is not shown in the example is that there are more classes of configuration rules. Each ID for “POS” and “PV” has its configuration rules, and they have to be taken into account when studying the configuration rules found in the bill of material.

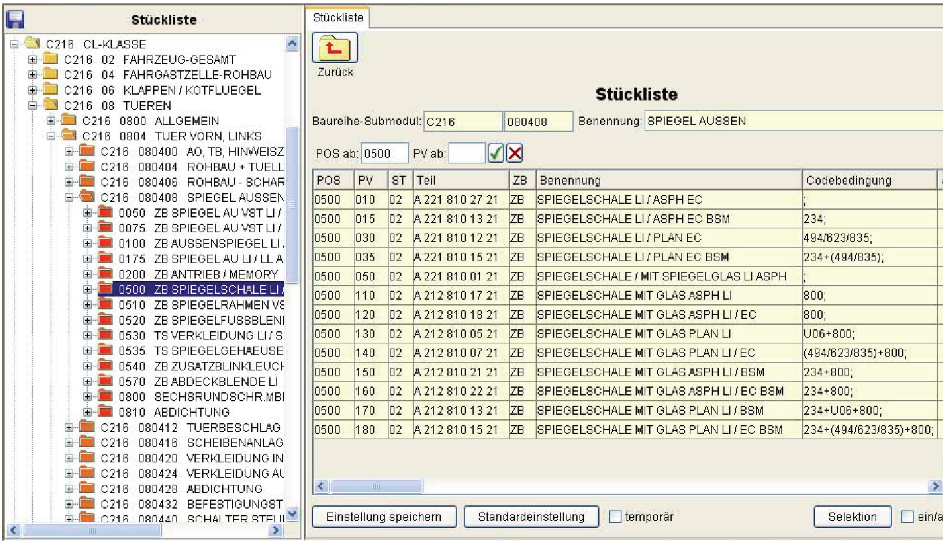


Figure 2.20: Table-based CR visualization tool at Daimler (reprinted from Lamberti, 2011).

2.4.4 Matrix-based CR visualization methods

Much work has previously been done on matrix-based visualization methods, e.g. the Quality function deployment (Akao & Mizuno, 1994), Design structure matrix (Steward, 1981), “K- and V-Matrix” (Bongulielmi et al., 2001) etc. Those visualization methods may be characterized by their use of intra-domain or inter-domain matrices, see Fig. 2.21. Intra-domain matrices shows the dependencies within a domain, e.g. between feature variants. The intra-domain matrix is always square. The inter-domain matrices show dependences between domains, e.g. between items and features variants. The inter-domain matrix may be rectangular.

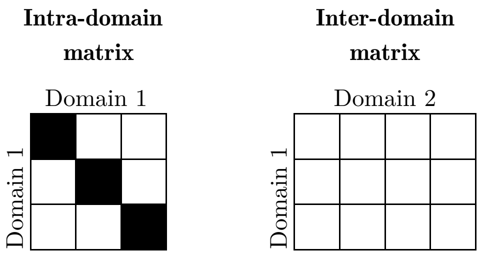


Figure 2.21: Comparison between intra-domain matrix and inter-domain matrix (adapted from Malmqvist, 2002).

The purpose of the matrix-based visualization methods found in the literature is often an optimization of the product structure, e.g. finding potential modularization (Maurer, 2007; Luh et al., 2011). The aim of an intra-domain matrix is to analyze the relationships between elements within a domain, but vehicle configuration requires both the feature and item domain. In Bongulielmi et al. (2002) there is a description of how two intra-domain matrices may be combined with an intra-domain matrix to be visualized, according to Bongulielmi et al., a “major part” of the configuration rules; see Fig. 2.22. The visualization method is called the K- and V-Matrix and has been designed to be applicable to product configuration. The K stands for configuration [“Konfiguration”, in German], while the V stands for compatibility [“Verträglichkeit”, in German]. The inter-domain configuration rules shown with the table-based CR visualization method in Fig. 2.20 would be able to visualize in the K-matrix. The two compatibility matrices (V-Matrix) show intra-domain configuration rules. The entire set of intra-domain configuration rules cannot be visualized with a matrix-based approach if a single configuration rule states relationships between more than pairs of feature variants or pairs of items.

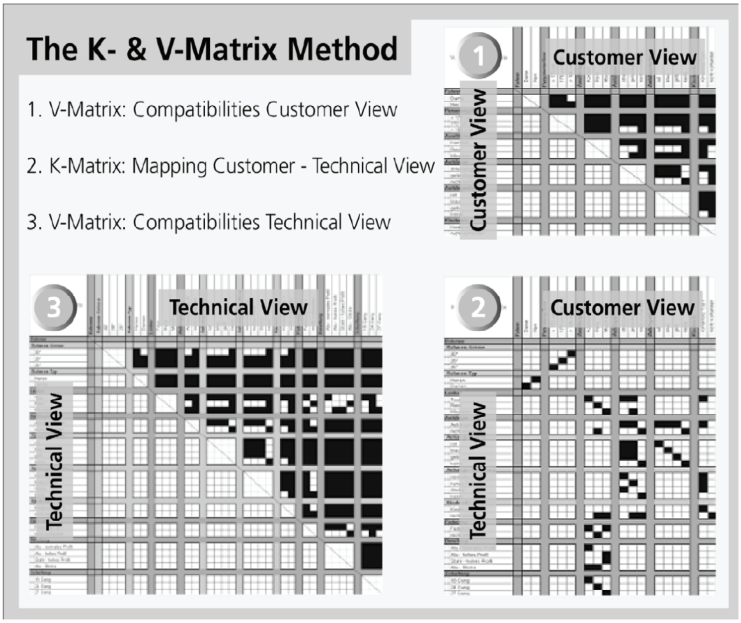


Figure 2.22: The K- and V-matrix (Bongulielmi et al., 2002).

The benefit of using a matrix compared to the table and list is, according to Bongulielmi et al., that the matrix is able to represent presence or absence of a relationship.

2.4.5 Conclusions

The language in which the configuration rules are expressed affects the potential use of CR visualization methods, as a list of vehicle configuration rules may easily be transformed into a matrix in contrast to a list of mathematical configuration rules. The applicable visualization methods for vehicle configuration rules therefore range between the list-, table- and matrix-based methods.

The CR visualization methods that have been described have different properties in terms of scalability and readability; see Fig. 2.23. The matrix-based CR visualization method has its strengths in its patterns that are created from the configuration rules. The scalability of the matrix is dependent on the number of feature variants or items. When adding a configuration rule to a matrix-based visualization, the size of the matrix does not change if the pattern itself may visualize the modification. In the worst case, the matrix has to grow by as many columns or rows as the introduced number of

	SCALABILITY When configuration rules are added…	**READABILITY**
List	… the list becomes one row longer, but not necessarily wider.	Listed configuration rules are read from left to right as any other word sentence.
Table	… the table gets one more row, but not necessarily more columns.	The columns may separate the item and feature domains.
Matrix	… grows in rows or columns with as many items/feature variants that are introduced.	Patterns may be used to show similarities and differences between configuration rules.

Figure 2.23: Comparison between the CR visualization methods.

feature variants or items. The table does not show any patterns but may separate feature variants and items into different columns which is good for the readability. The table grows in length as more configuration rules are added, but as in the case of the industrial table-based CR visualization tool shown in Fig. 2.20, not necessarily in number of columns. The list has the same scalability property, but the advantage of the list's readability is that the configuration rules are read from left to right like any other word sentence.

The studied CR visualization methods show strengths and weaknesses depending on whether it is one single configuration rule that should be visualized, or whether there is a set of configuration rules. For one single configuration rule, the most efficient visualization is the list. For a set of configuration rules, the patterns created from the matrix make it easier for humans to distinguish similarities and differences between the configuration rules.

2.5 Research needs

This Licentiate thesis contributes to the research areas of configurable product structures, CR development methods and CR visualization tools. This section discusses identified research needs in those parts of the engineering information management framework. The next chapter will formulate research questions which are motivated by the research needs.

2.5.1 Configurable product structures

The conclusion from the frame of reference stated that there is no generalized information model suitable when discussing the vehicle CR development process. The classification of configurable product structure information models followed the definition in Duffy & Andreasen (1995) and Neubert (1993), i.e. the conceptual model is a simplified version of the implementation model. The conceptual models have been created with different purposes in mind, e.g. to facilitate the authoring process of configuration rules as in the case of the PFMP, or to communicate where the product structure needs to express variability in a product family as in Mesihovic & Malmqvist (2004b). The purpose of this thesis is to identify the difficulties during the vehicle CR development process and to find methods and tools to support those. The sought-after information model should therefore be a generalization of the implementation models for vehicle product structures, which does not necessarily imply a simplification. The generalized information model should contain the information elements that are necessarily used during the vehicle CR development process. The difficulties during the CR development process should be possible to discover and discuss based on the generalized information model, e.g. difficulties from the relationships between the information elements, their attributes etc.

2.5.2 CR development process

Process models from the development of knowledge-based systems have been found applicable also for the CR development process. This generalized CR development process, however, is not detailed enough for taking advantage of automation potential; e.g. the authoring methods are not described in a formalized manner. Neither are the methods used when evaluating the configuration rules concretized; e.g. how configuration rules are inspected, and why, should be described in enough detail. For obtaining a more mature vehicle CR development process, with a higher automation potential, there is a need of formalizing the authoring and visualization methods.

2.5.3 CR visualization tools

The implemented configuration rules are visualized in commercial configurators and industrial visualization tools using methods based on lists, tables or matrices. Vehicle configuration rules are large-scale data expressing relationships between and within the feature and item domain. Consequently, both intra-domain and inter-domain matrices are required to visualize the configuration rules. Inspecting the configuration rules is considered to be time-consuming, potentially due to CR visualization tools' multiple user interfaces. The CR visualization tools have not been thoroughly evaluated in the literature, but the use of matrices with patterns seems promising. There is a need to study how to improve the industrial CR visualization tools. To fulfil the purpose of this thesis, the currently used visualization tools should also be systematically developed aiming at higher usability and time efficiency.

3 Scientific approach

This chapter first describes the research questions, which are derived from the research goals and the identified research needs described in previous chapters. Then the research setup and process are described, followed by the research framework. After that, the research process is described with a discussion of the research methods. Finally, the validation approach describes the actions taken for reducing the impact of validity threats.

3.1 Research questions

The research questions align with the different parts of the engineering information management framework applied to this Licentiate thesis. The studied parts of the framework concern configurable product structures (*information*), CR development process (*process*) as well as the CR visualization tools for developing configuration rules (*information system*). Each research question is addressed by more detailed research questions which are found in the appended papers.

3.1.1 Configurable product structures

The first research question concerns the information model for vehicle product structures. The related research goal (RG1) is to find a suitable information model as a common language when discussing the CR development process. The identified research need showed that there are generalizing attempts for product structures, and standards for how to document vehicle product structures. The generalizing attempts, however, have not focused on the vehicle product structure, specifically the configuration rules that guide the specification and instantiation of such structures. The AP214 standard for vehicle product structures fulfils its purpose of being applicable during implementation, but has been found lacking guidelines for how to develop vehicle configuration rules. The information model should be able to represent the information needed during the CR development process. A suitable information model should also be able to describe similarities and differences to other types of product structures. As both the research goal and the identified research needs address the issues of a missing generalized information model for vehicle product structures, the following research question (abbreviation: "RQ") was stated:

RQ1: What are the characteristics of a generalized information model for vehicle product structures suitable for supporting the CR development process?

The information model found will then be used when describing the development methods during the CR development process. The next section describes the research questions that address the CR development methods.

3.1.2 CR development process

The research needs stated for the CR development process addressed the development methods that could be formalized in order to semi-automate the process. As the CR development process is iterative, there is a need for describing the complete process in order to understand the difficulties, e.g. why the iterations occur. The set of research questions addressing the CR development process is addressing both the complete process, and, more specifically, the authoring variations and their formalization due to the related research goal (RG2). The research questions have been formulated as follows:

RQ2a: How are vehicle configuration rules developed?

RQ2b: Which authoring methods exist?

RQ2c: How can these authoring methods be formalized?

The CR visualization tools play a central role in the CR development process, as it is used both for inspecting the authoring variations and for verifying and validating the configuration rules. The CR visualization tools are therefore addressed with a research question in the following section.

3.1.3 CR visualization tools

The research need for the CR visualization tools is fundamental, as there is a lack of such efficient and user-friendly tools. The CR visualization methods were approached from scratch by studying which methods are used in the industrial CR visualization tools. The related research goal (RG3) is to develop an easy-to-use CR visualization tool that utilizes the formalized authoring methods, which indicates the aim of automating the CR development process in order to make it more time-efficient and less error-prone. The CR visualization method for an efficient CR development process is studied by the research question:

RQ3: How can CR visualization tools be improved?

The next sections motivate and describe the methods that have been used during the research process.

3.2 Research setup and process

This research project was conducted with participation of three vehicle manufacturing companies. The collaboration took place within the frames of FFI, "Fordonstrategisk Forskning och Innovation" ["Strategic vehicle research and innovation", in Swedish] with a vehicle manufacturing company as project leader. This assured a strong commitment within the industry for the research project. The project constellation was based on three vehicle manufacturing companies, the two Chalmers research departments "Product and product development" and "Signals and systems", and the Chalmers-Fraunhofer research centre for industrial mathematics. The research product studies vehicle configuration from a holistic viewpoint, e.g. processes, information systems, algorithms etc. The focus of this Licentiate thesis is on the CR development process and the vehicle product structure, while the researchers from the Signals and systems research department as well as Fraunhofer-Research centre have focused on the algorithms.

3.3 Research framework

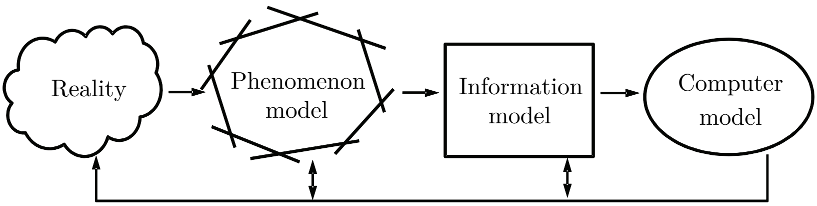
The research work for this Licentiate thesis ultimately aims to develop a CR visualization tool making the CR development process more time-efficient and less error-prone. How the CR visualization tool impacts the reality of the CR development process may be described by using the framework presented in Fig. 3.1 of Duffy & Andreasen (1995). They suggest that the "reality" is first described with "phenomenon models". The phenomenon model is based on theories or frameworks, e.g. the development of knowledge-based systems. These phenomenon models are then developed where appropriate into information models, which are possible to utilize in the CR visualization tools for supporting the CR development process. Information models are based on information theory, such as object-oriented modelling or the unified modelling language UML. Computer models are based on computational theories or languages.

The important looping arrows within the research framework indicate the necessary iterative refinements of the perception of reality to be able to develop an efficient tool.

This Licentiate thesis applied the research framework when studying the vehicle CR development process. The "reality" of developing configuration rules was studied by conducting user studies and the industrial PDM systems. A phenomenon model was modeling the CR development process, aided by the theory of knowledge-based systems as well as the process modelling language IDEF0. The generalized information model captures the product structure information used during the CR development process. The last step in the iteration cycle was the computer model, which was a CR visualization tool supporting the development methods. The CR visualization tool was iteratively developed by repeating the iteration cycle, which included going back to the reality and studying how the users would develop configuration rules aided by the tool.

Moreover, in design research there is a hypothesis that any developed tool will make an impact upon the design process itself. If the CR visualization tool were implemented in the CR development process, the reality of the process would change. There is therefore a potential for evaluating the effects on the CR development process after an implementation of the CR visualization tool. This last step is related to this Licentiate thesis' general purpose and success criteria: to make the CR development process more efficient and less error-prone.

The next section first describes the research process and then shows how



Activity:	Develop vehicle configuration rules.	Model the CR development process.	Generalize the information model.	Implement the formalized methods, Develop visualization tool.
Based on:	User studies. industrial PDM Systems.	IDEF0, knowledge-based systems.	UML class diagram.	Java, C#, Excel, ...

Figure 3.1: Research framework and its application in this Licentiate thesis, adapted from (Duffy & Andreasen, 1995).

triangulation has been used to ensure the validity of the qualitative research results.

3.4 Research process

The research process is described following the framework of Blessing & Chakrabarti (2009). The research work starts with a research clarification, followed by two descriptive studies separated by a prescriptive study. The shifts between descriptive and prescriptive studies are according to Blessing and Chakrabarti typical for design research projects. The shifts between descriptive and prescriptive phases may also be found in the previously described research framework of Duffy & Andreasen (1995): the descriptive study I is an interpretation of reality, and the descriptive study II occurs at later iteration cycles as an evaluation after a support has been designed.

This Licentiate thesis started with a descriptive study I as soon as the project plan was written. The scope of a paper sometimes included more than one single type of study (see Fig. 3.2), which is marked out with a curly bracket. The following sections will describe the research process in chronological order. Following the DRM framework for the research type "development of support based on comprehensive study of the existing situation", the described stages would be sufficient.

3.4.1 Research clarification

According to Blessing & Chakrabarti (2009), the research in engineering design aims to increase the ability to produce a successful product. A number of questions arise in the area of product improvement:

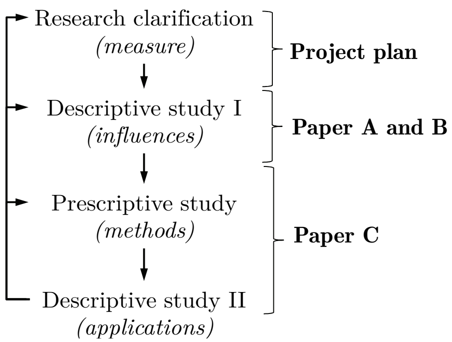


Figure 3.2: Research process following the stages of the DRM framework (adapted from Blessing & Chakrabarti, 2009).

1. What is a successful product?
2. How is a successful product created?
3. How can the chances of an improved product be increased?

The questions lead to issues that can be captured by using an "impact model". This model serves as a plan for how the chances for a successful product can be increased. The impact model created for this thesis is shown in Fig. 3.3. The "key factors" are influencing factors that seem to be the most useful factors to address in order to improve the CR development process if the research results were deployed in practice. The impact model starts with the two key factors "formalized CR development methods" and the "number of window swaps for using the CR visualization tool". Formalization of the CR development methods increases the automation potential, which together with fewer window swaps may increase the CR visualization tool's ease-of-use. The measurable success factor is the number of errors, the learnability as well as the time efficiency when using the CR visualization tool. The success factor of a lower number of misbuilds in the factory is the ultimate goal of the research project. However, due to the difficulties in isolating this factor from other influencing factors, it is instead the measurable success factor that will be observed. It is assumed that that it is possible to judge the success of the research project by the measurable success factor.

3.5 Validation approach

Validity depends on the relationship between the conclusions and the reality. A validity threat is a possible source of error. Validity as part of the research design is the strategy for how to identify and try to rule out such threats. This section describes the actions for lessening the impact of validity threats for this research project, following the

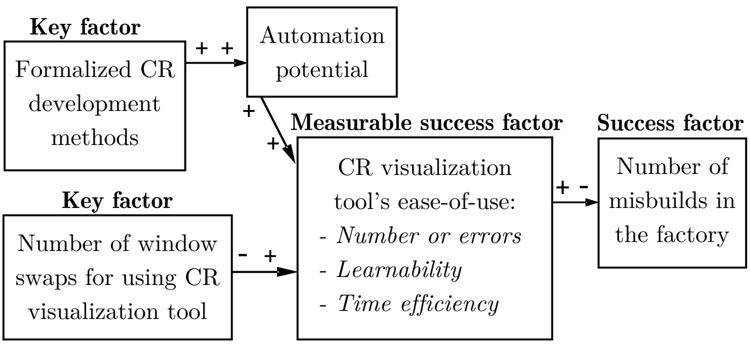


Figure 3.3: Impact model for this thesis (adapted from Blessing & Chakrabarti, 2009).

categorizations by Maxwell (2005). How the validity threats have been addressed in the research work is described in the following sections, but first a short description of each kind of validity threat will be given:

- **Researcher bias:** There are two ways that researchers may be biased (subjective), either by selecting data that fit the researchers' existing theories or by omitting data.
- **Reactivity:** The influence of the researcher on the setting or individuals is known as reactivity. What the interviewee says is always influenced by the interviewer and the interview setting. It is possible to prevent undesirable consequences such as avoiding leading questions.
- **Intensive long-term involvement:** Long-term participant observation gives a more complete view of the specific situations than any other method. Repeated observations and interviews, as well as the sustained presence of the researcher in the setting studied, can help to rule out spurious associations and premature theories.
- **"Rich data":** Long involvement and intense interviews enable collection of "rich" data, which means that the data are detailed and varied enough to reveal the picture of what is going on.
- **Respondent validation:** Systematically collecting feedback about the data and the conclusions from the people that are studied.
- **Intervention:** The researcher intervenes in the study by experimentally manipulating the factors that may affect the research results.
- **Searching for discrepant evidence and negative cases:** Results that do not match the conclusions can point at important defects.
- **Triangulation:** Collecting information from various sources and diverse settings using a variety of methods. Triangulation reduces the risk of chance associations and of systematic biases due to a specific method.
- **Quasi-statistics:** Quantitative statements such as "rare" and "often" should be backed up by quantitative support, e.g. 5 hours/week. The term quasi-statistics was first used by Becker (1970), and is provided for support to qualitative claims as well as assessing the amount of evidence in the data.
- **Comparison:** Comparing the obtained results with existing results, between control groups or at different points in time may contribute to the interpretability of the results.

3.5.1 Paper A: Information modelling for automotive configuration

The results of Paper A is a generalized information model for the vehicle product structure, and a comparison between different other product structures theories. The results were achieved by first outlining a draft of a generalization of the vehicle product structure. Then, an empirical study was conducted at a vehicle manufacturing company. Since the author has had an intensive long-term (~1 year) involvement at the studied vehicle manufacturer, there was a great potential for collecting "rich" data. When discussing the information model with design engineers and product structure specialists, it was beneficial to understand the company's feature variant codes. There were also occasions arranged when the author had the possibility to learn the meaning of those codes in the physical vehicle.

The purpose of the empirical study was to map out the company's product structure information model, aiming at validating the generalization and identifying additional issues in the area. The information sources for the empirical study were (1) the PDM system documentation at the vehicle manufacturing company, (2) the PDM learning material at the company and (3) discussions during work group meetings with academic and industrial representatives. One researcher was present at the vehicle manufacturing company during a long period to further understand their configuration approach. The generalization was refined and the characteristic of the generalization was used as a classification framework for existing research on product structure. This was followed by a more comprehensive literature analysis where the constructs of various theoretical models were contrasted against the basic framework and industry issues. In a sense, there was a shift back and forth between theory and empiric in an iterative process. A table showing all studied product structure information models was finally created, where strengths and weaknesses could be pointed out and analyzed. There was also a validation of the generalized information model through discussions with representatives from two other vehicle manufacturers.

One bias validity threat that stems from the research is company A's wish not to practically evaluate any alternative information models for vehicle configuration rules, e.g. the configurable component approach (Claesson, 2006) or the product model introduced in Soininen et al. (1998). The reason for not considering alternative information models was also practical. Basing the empirical and demonstration activities on another product structure theory would require comprehensive information collection and modelling by the researcher. The modelling and collection of information would have to include reformulation of configuration rules, collection of functional requirements etc. This was not deemed feasible within the scope and setup of the current research project.

3.5.2 Paper B: Authoring and verification of vehicle configuration rules

The result of Paper B is a description and formalization of the authoring and verification methods found at three vehicle manufacturing companies. The findings of Paper B aim at increasing the CR visualization tool's ease of use, by adding the formalization of CR development methods and extending the currently used CR visualization method. The data collection was primarily interviews, including the users' demonstrations of authoring and verification methods. In total, 20 semi-structured interviews were conducted, which lasted approximately two hours each. With the aim of reducing the influence of the interviewer, there was an interview guide with a checklist of what the interviewee should be informed about before the interview. One of the checklist's topics was the statement that there are no right or wrong answers to the interview questions and the only interesting answers are the interviewee's views and opinions. By this statement it was emphasized that the interviewee should speak freely about his/her opinions instead of trying to guess the "right" answer to the interview questions. The number of interviewers was also limited to only two persons in the interview sessions. This was another effort to reduce the interviewers' influence. The interviews were carried out by one or two researchers together with one employee from one of the three studied vehicle manufacturing companies. The benefit of being two interviewers is to be sure to collect the data, e.g. to be able to take notes while asking the interview questions. Another benefit is the possibility to make an analysis directly after the interview, to compare interpretations about what was said and draw some preliminary conclusions. This can also be seen as a specific type of triangulation. The interviewees were from the three companies and equally distributed among 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 feature variant combination rules.

Respondent validation was used as verbatim transcripts were sent out to the interviewees who then gave feedback on both the data and the keypoints.

Another source of data was the document study of guidelines for how to update configuration rules. Moreover, the configuration rules themselves were very useful to study when evaluating the findings. It may therefore be argued that multiple methods have been used to validate the findings.

The descriptive user study also compiled the users' difficulties and views on the current CR visualization methods, which initiated data collection for Paper C.

3.5.3 Paper C: Development of industrial visualization tools for validation of vehicle configuration rules

The results of Paper C are a suggestion and evaluation of a CR visualization tool showing how the current CR visualization methods may be extended in order to make the CR development process more time-efficient and less error-prone. The addressed factor from the impact model is the ease of use for the CR visualization tools. The starting point for Paper C was the user study conducted for Paper B. The user study included both interviews and observations, and was conducted at three large vehicle manufacturing companies operating in-house developed CR visualization tools. The interviewees were either design engineers or product structure specialists. It was not explicitly asked for, but it was found during the user study that several of the interviewees were not able to answer typical configuration rule queries with the current CR visualization tools. Several users did rely on personal communication with other staff members instead of the CR visualization tools. Based on 10 interviewees with design engineers, 40% declared that they relied primarily on personal communication. A successful outcome of this Licentiate thesis would be to find a CR visualization method where all users are able to answer the typical configuration rule queries correctly as well as doing so more time-efficiently.

The next step was to study the literature for similar CR visualization methods and configuration rule queries. Then a demonstrator was created, based on a modified CR visualization method proposed by several interviewees during the user study. Four users were selected for the formative usability tests, following the guidelines not to conduct large numbers of experiments but to extract as much information as possible from every user. The tests were based on industrial data and real users for the application, e.g. design engineers and product structure specialists from the studied vehicle manufacturing companies. The industrial data were exported from a database. A Java program written in Eclipse then computed, by using SAT4J, the configuration rules in order to present the CR visualization in Excel. The user interface was programmed in C# by using Visual Studio.

The test setup was based on a selection of configuration rules that a design engineer had identified as “his” interesting configuration rules. The test was therefore based on 15 item usage rules as well as the computation results from the other classes of configuration rules. Three databases of industrial vehicle configuration rules were available, but only data from company A were used in the tests; see Fig. 3.4. Company A has the highest number of configuration rules, and there should not be any capacity problems if conducting usability tests with data from company B or C. From the initiation of the research project, company A had claimed that its configuration rules were far more complex than the configuration rules from the other two companies (B

<table>
<tr><th></th><th></th><th>Company A</th><th>Company B</th><th>Company C</th></tr>
<tr><td></td><td>Product family</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td></td><td>Product models</td><td>20</td><td>6</td><td>3</td></tr>
<tr><td></td><td>Feature families</td><td>500</td><td>200</td><td>500</td></tr>
<tr><td></td><td>Feature variants</td><td>3 000</td><td>800</td><td>4 000</td></tr>
<tr><td rowspan="5">Configuration rules</td><td>Product model authorizations</td><td>20 000</td><td rowspan="3">1 000*</td><td>4 000</td></tr>
<tr><td>Inclusions (IF-THEN, AND)</td><td>300</td><td>8 000</td></tr>
<tr><td>Exclusions (NOT, AND)</td><td>30 000</td><td>1 000</td></tr>
<tr><td>Item usage rules (IF-THEN, AND)</td><td>400 000</td><td>9 000*</td><td>20 000</td></tr>
<tr><td>Length limitations</td><td>10</td><td>100</td><td>5</td></tr>
<tr><td></td><td>Items</td><td>40 000</td><td>7 000</td><td>20 000</td></tr>
</table>

*) IF-THEN, AND, NOT and OR are used.

Figure 3.4: Test data from three vehicle manufacturing companies.

and C). Most of the research work has taken place at company A. By studying Fig. 3.4, it is concluded that the number of configuration rules was higher at company A, but the length of the configuration rules was significantly shorter than for company C. It is therefore debatable whether so much attention should have been given only to company A. The potential impact on the research results is more thoroughly discussed in Paper C.

First a pilot test with one user was conducted to estimate the adequacy of test questions' number as well as question formulations. Then three usability tests were conducted, separated into two test series by a redesign of the demonstrator. The last step was to conclude the test results.

The time efficiency was measured once the users had some experience of performing tasks from the typical configuration rule query. The errors were defined as filling in the wrong answers to the tasks on the test sheet.

3.5.4 Summary of research methods

A summary of the research methods used in Papers A-C is given in Fig. 3.5. To what extent a method was applied can be described by how many vehicle manufacturing companies (A-C) were involved. As seen in the figure, there is a potential for triangulation since the papers use a variety of research methods. Triangulation

	Workshops with product structure specialists	Documentation/ information system access	User interviews	User observations	Formative usability tests (including interviews)
Paper A	Company A, B and C	Company A	Company A		
Paper B	Company A, B and C	Company A, B and C	Company A, B and C	Company A, B and C	
Paper C	Company A, B and C	Company A, B and C			Company A

Figure 3.5: Research methods used for the papers.

strengthens the evidence of the research results. For example, Paper C about the development of industrial CR visualization tools has applied three research methods: “workshop with product structure specialists”, “documentation/ information system access” and “formative usability tests (including interviews)”. Triangulation is also about the study of diverse settings, which has been achieved by selecting interviewees from different vehicle manufacturers and different domains, e.g. brakes, software, electronics, suspensions etc.

4 Results

This chapter presents the main results of the research with an extended summary of the papers. The papers may be categorized according to the engineering information management framework, starting with the configurable product structures (information) in Paper A, continuing with the methods used during the vehicle CR development process (process) in Paper B and then the CR visualization tools (information system) in Paper C.

4.1 Paper A: Information modelling for automotive configuration

Following the engineering information management framework of information, information system and processes, this paper is about the "information".

4.1.1 Background and research questions

Today, many vehicle manufacturing companies do not use commercial Product Data Management (PDM) systems for CR development. Instead, they are relying on unique in-house developed PDM systems: Spectra at Scania (Johansson & Eriksson, 2007), KOLA at AB Volvo, KDP at Volvo Cars Corporation (Ikaros, 2012), Smaragd at Daimler AG (Hospach et al., 2002), SIGNE at Renault SA (Normile et al., 2001) and GPDS at General Motors (Pyle, 2010). We did, however, notice signs of significant similarities in the product configuration approaches within the truck and car manufacturers that we collaborate with. Our hypothesis was therefore that there is a potential for creating a generalization of the product configuration methods and information models used in the automotive industry. An adjacent task is to pinpoint more precisely in what ways automotive product configuration models are different from those presented in the literature. The specific research questions were:

RQ1: Which elements and relations are included in published theoretical product configuration information models?

RQ2: What product information is used in automotive configuration in practice, considering specifically the feature and item structures? What practical issues can be identified?

RQ3: What are the similarities and differences between the practical and theoretical product configuration information models?

The research questions were answered by an empirical study at a vehicle manufacturing company, as well as a literature review. The information sources for the empirical study were (1) the system implementation documentation of the PDM system at the vehicle manufacturing company, (2) the PDM learning material at the company and (3) discussions during work group meetings. One researcher was present at the automotive firm site during a long period to further understand their configuration approach. A table showing all studied product data information models was finally created, where strengths and weaknesses could be pointed out and analyzed.

4.1.2 Result: Information model for vehicle product structures

Currently there is no commonly accepted information model for product structures. A generalized information model of the vehicle product structures was therefore developed; see Fig, 4.1. The top node in the Unified Modelling Language (UML) class diagram is the “product family”, including e.g. all vehicles sharing the same platform. Within the product family there may be several “product model variants”, e.g. sedan or station-wagon versions of a vehicle. The vehicle information model mainly consists of the part-oriented (item) structure with “items” (parts, documents etc) as well as the feature-oriented structure with “feature variants” (engine size, with or without cup holder etc.). Several classes of configuration rules, i.e. logic expressions, state relationships between/within those structures:

- **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 to be chosen.
- **Feature variant combination rules** define prescribed (“inclusions”) or forbidden (“restrictions”) combination of feature variants.
- **Item usage rules** define for what feature variant combinations a certain item should be used.

The classes in the information model are the typical core objects of a vehicle product structure. On the level of abstraction that has been presented, the model holds generally among vehicle manufacturing companies. Those core objects have been used when characterizing product structures in the literature review.

The review of both theoretical and empirical product information models has shown that the framework of item structure, item usage rules, feature variants and feature variant combination rules presented in Fig. 4.1 is capable of describing all earlier published information models. The objects of earlier published information models for product structures can be classified into one of the elements in the framework. Different information models have different strengths: generic BOMs (Van Veen, 1992) for item structures and item usage rules, and configurable components (Claesson, 2006) for configuration rule modelling.

4.1.3 Conclusions

Practical challenges emerge from method and process aspects rather than the information content. Frameworks for product configuration models need to go beyond pure information modelling to include methods for writing, e.g. “good” configuration rules and item usage rules, in order to increase their utility for industry.

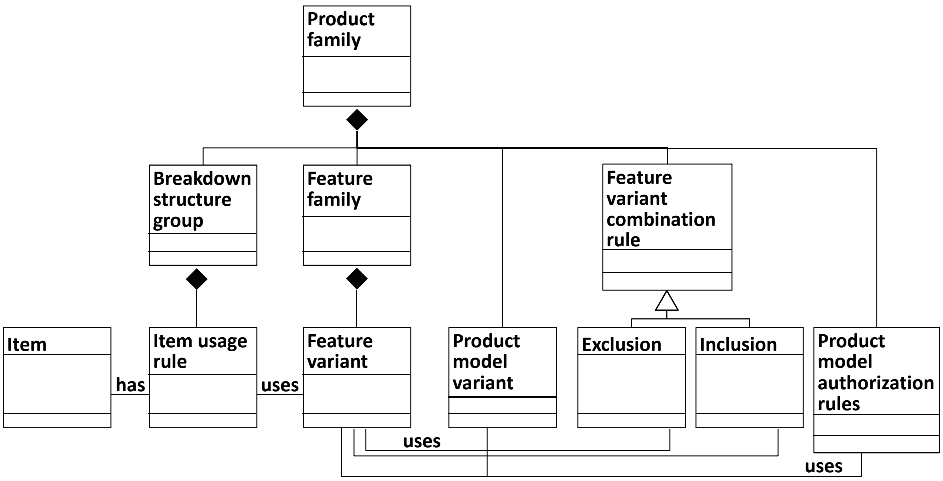


Figure 4.1: UML class diagram of vehicle product structures in Paper A.

4.2 Paper B: Authoring and verification of vehicle configuration rules

Following the engineering information management framework of information, information system and processes, this paper is about the “process”.

4.2.1 Background and research questions

The conclusion from Paper A stated that the practical challenges emerged not only from the information model itself, but from the methods and processes when developing vehicle configuration rules. As a response to the conclusion, Paper B aimed to investigate industrially applied methods for authoring and verification of vehicle configuration rules, specifically to address the difficulties that potentially may lead to faulty vehicle configurations and inefficiencies in the CR development process. The specific research questions were:

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 answered by studying three vehicle manufacturing companies. In total, 20 semi-structured interviews were conducted with both design engineers and product structure specialists, which lasted approximately 2 hours each. The interview sessions included interviewees’ demonstrations of authoring and verification methods. Documents were 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.

4.2.2 Result: Authoring and verification methods

Configuration rule modifications are usually requested by new development projects, facelifts or modified market offerings, 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. 4.2. This CR development process differs slightly in terminology used in this Licentiate thesis, as verification here is an activity and not an outcome of the evaluation process. When the configuration rules are released they may

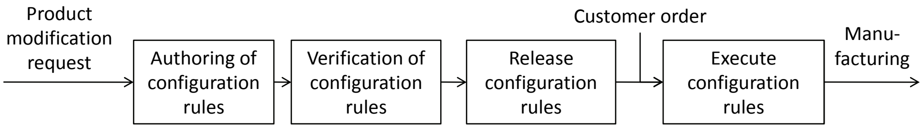


Figure 4.2: Schematic picture from the change initiation of the vehicle configuration rules to the manufacturing of vehicles

be used within the order system, where they are executed when supporting a customer to specify a vehicle while verifying that customer orders are allowed to be manufactured. Both design engineers and product structure specialists are involved in authoring and verification of configuration rules. These roles have different daily activities, which generate different preferences in authoring methods. The authoring variations that were found are:

- *Overlapping documentation:* The overlapping documentation occurs when there are two or more configuration rules giving the same information, even though they are not identical. Overlapping documentation can often be avoided by shortening the configuration rules.
- *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.
- *Building-blocks:* 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 feature variant combination rules.

Eliminating overlapping documentation for item usage rules is a sign that the users have already found and verified the feature variant combination rules. This is a typical situation where the verification of item usage rules requires knowledge about the other classes of configuration rules. In other words, a common verification task is to make sure that there are no gaps between allowed vehicle configurations and the items that populate those configurations. The verification methods have been further elaborated in section 2.3.3.

4.2.3 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 traded against compactness and maintainability. Arguments for using a user interface consisting of a matrix rather than a text-based format were repeatedly identified. The traditional user interface to the industrial CR visualization tools consists of a text-based database viewer, e.g. an output from the MS-DOS command prompt or the file manager Windows Explorer. We have shown that the main difficulty is to combine feature variant combination rules with item usage rules, which require a configurator, and the traditional user interfaces to CR visualization tools should therefore be challenged. With the formalization of the authoring and verification methods, there is a potential for a 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 a full-time job for product structure specialists, which motivates realizing the automation potential and thereby reducing development costs.

4.3 Paper C: Development of industrial visualization tools for validation of vehicle configuration rules

Following the engineering information management framework of information, information system and processes, this paper is about the "information system".

4.3.1 Background and research questions

The conclusions from Paper B stated that the industrial CR visualization tools needed to combine feature variant combination rules with item usage rules. From the interviews, it was found that at the three studied vehicle manufacturing companies, 4 of the 10 design engineers participating in the interview study stated that they relied primarily on personal communication instead of the CR visualization tool. Based on those findings, Paper C aims to find a CR visualization tool that is easier to use, and thereby makes the CR development process less time-consuming and less error-prone. This paper addresses the research questions:

RQ1: What are the strengths and weaknesses of current industrial CR visualization tools used when validating vehicle configuration rules?

RQ2: Which CR visualization tool addresses those weaknesses?

RQ3: What benefits and limitations would such a tool provide?

One of the outcomes from Paper B was a formalization of the authoring methods used, which is an automation potential that initiated the further development of the industrial CR visualization tools. The pre-study for the development of a CR visualization tool included a description of current CR visualization tools, typical configuration rule queries, CR visualization needs and discussion on design considerations. The literature on CR visualization tools and methods helped in defining the industrial CR visualization tools as well as giving alternative references. Then a demonstrator was created for developing and evaluating the suggested CR visualization method. A positive outcome of formative usability tests (Hix & Hartson, 1992) would show that all test participants are able to perform the tasks with the demonstrator without errors as well as doing so time-efficiently. The tests were based on industrial data and the participants were real users for the application, i.e. design engineers and product structure specialists from the studied automotive company.

4.3.2 Results: Demonstrator development and testing

The demonstrator displays item usage rules exported from the company's configuration rules database, while the other classes of configuration rules are processed with the aid of a configurator. The item usage rules are visualized with black crosses and should in Fig. 4.3 be read as: IF(a1 AND b1 AND c3 etc.) THEN(ITEM1). The processed configuration rules generate the pink fills, "?" and "!" in the CR visualization tool. All classes of configuration rules are thereby visualized in one single user interface.

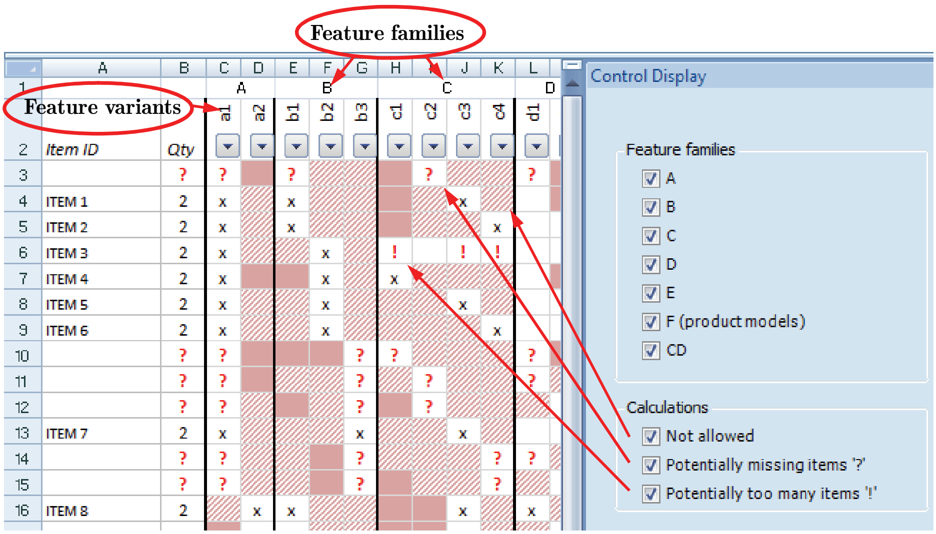


Figure 4.3: User interface of the demonstrator in Paper C.

The user interface facilitates both the authoring and the verification of configuration rules: (1) Authoring methods for item usage rules are distinguished by inspecting the pink fills. Alternative authoring methods are therefore realized directly from the visualization; e.g. the item usage rule for ITEM4 should be reduced, as the feature variant “a1” is the only allowed feature variant from that feature family according to the restricting configuration rules. (2) Verifications of the configuration rules are visualized by the exclamation and question marks, which show configurations with potentially missing or too many items.

The usability tests resulted in identification of numerous critical incidents that increased the understanding of what the demonstrator was possible to show and which elements should be modified or taken away. The test participants predicted that the greatest value of the suggested CR visualization method was the increased confidence of the users. Since the configuration rules already were processed with a configurator, computations e.g. for potentially missing items had been automated. The demonstrator therefore reduces the risk for the users of finding wrong answers. Also, a major positive predicted outcome is the increased time efficiency by avoiding iteration loops between design engineers and product structure specialists.

4.3.3 Conclusions

The analysis of industrial CR visualization tools and the related user study have shown that there is a potential for facilitating the CR development process. Usability tests of the demonstrator have shown that the identified weaknesses are possible to address. The outcome of the usability test was successful:

- **Decreased numbers of errors:** All users fulfilled the tasks correctly. A decrease in errors is predicted during the post-session interviews due to automation and an improved understanding of the different configuration rule classes.
- **Easier to learn:** Negative critical incidents clarified the first tests, and could mostly be eliminated through development of the demonstrator.
- **Improved time efficiency:** The measurement of response times showed that test participants who had gained some experience with the demonstrator performed the benchmark tasks in the order of seconds. Although there was no comparative response time measurement, time efficiency increased according to the post-session interviews.

5 Discussion

This chapter discusses the research results and contributions in relation to the research questions and the purpose of the thesis. The research process is discussed concerning aspects such as generalizability.

5.1 Answering research questions

The research questions address the different parts of the engineering information management framework, which, applied in this Licentiate thesis, are identified as "configurable product structure" (information, RQ1), "CR development process" (process, RQ2) and "CR visualization tools" (information system, RQ3). The research questions will be discussed in relation to the purpose of the thesis, which is to identify factors that are causing difficulties during the CR development in order to create tools that facilitate the CR development process. Each of the research questions is answered with a discussion of contribution to the general purpose.

RQ1: What are the characteristics of a generalized information model for vehicle product structures suitable for supporting the CR development process?
The characteristics of the information model for vehicle product structures have been found by studying the literature as well as the PDM systems used at three vehicle manufacturing companies. The vehicle configuration is rule-based and relies on the execution of configuration rules. The core elements of the vehicle information model are the domains "feature" and "items", and the intra- and inter-domain "configuration rules". The intra-domain configuration rules are in this thesis called item usage rules. The inter-domain configuration rules in the feature domain are called feature variant combination rules and product model authorization rules. There are two types of feature variant combination rules: inclusions (IF-THEN) and exclusions (NOT). This is in contrast to much earlier research on product structures, where often e.g. there are no objects called item usage rules.

The set of questions for the second research question (RQ2a-c) addresses the vehicle CR development methods.

RQ2a: How are vehicle configuration rules developed?
The CR development process is initiated with a product modification request requiring new or modified configuration rules. The authoring of configuration rules is the first activity in the CR development process, followed by an evaluation and release. The authoring process of configuration rules may be subdivided into the elicitation, interpretation, formalization and implementation of configuration rules. The evaluation of configuration rules may be subdivided into the inspection, computation and testing of configuration rules. The CR development process is potentially iterative between the authoring and the evaluation steps as well within those two activities. The outcome of the CR development process is "valid" and "verified" configuration rules. The configuration rules are assumed to be valid if they are consistent with domain specialists' perceptions of which vehicles should be allowed to be built. The domain specialists are people with specialist knowledge about the domain, e.g. the design engineer of brakes. The configuration rules are verified when the computations currently conducted by product structure specialists are done, e.g. computations for potentially missing or too many items as well as redundant configuration rules.

Iteration in the CR development process may be argued to be especially time-consuming if it requires input from different roles, as in the case of the "knowledge acquisition bottleneck" which involved both domain specialists and product structure specialists. A more time-efficient CR development process would minimize the iterations, and especially the iterations where both domain specialists and product structure specialists are involved. We have not found any strict guidelines for which role should be responsible for which CR development process activity, except that the product structure specialists should fulfil the domain specialists' requests. Our approach has been to formalize the authoring methods and to make the CR visualization tools easier to use, thereby making the domain specialists more independent of the product structure specialists.

RQ2b: Which authoring methods exist?
One of the reasons why iterations potentially take place in the CR development process is the user preferences for certain authoring methods. The design engineers and product structure specialists from the three studied vehicle manufacturing companies described several authoring methods that are used when authoring vehicle configuration rules. Those methods may be classified into three types of authoring methods, which are shown in Fig, 5.1. Shorter (with fewer feature variants) configuration rules are easier to use during manual inspections and computations, while longer configuration rules have more precise content. The authoring method in between these two extremes is the

	← Short configuration rules	Consistent use of feature families	Longer configuration rules →
Authoring method:	Using high-level feature variants	Using building-block method	Using overlapping documentation
Benefit:	Efficient for computations	Easy to follow	Precise configuration rules
Drawback:	May be difficult to interpret/find the configuration rules	-	Inefficient for computations

Figure 5.1: Authoring variations on a length scale of the configuration rules.

"building-block" method, which is a consistent use of feature families for a set of configuration rules.

RQ2c: How can these authoring methods be formalized?
The general purpose of this Licentiate thesis is partly to create tools that facilitate the development of configuration rules. The authoring methods that were discussed in RQ2b are not possible to implement in a computer tool until a formalization of logic expressions has been made. For a complete description of the formalization, see Paper B. The authoring methods are interchangeable, i.e. there is no right or wrong authoring method when studying which vehicle configurations become allowed and the items assigned to those.

The last research question addresses the CR visualization tools.

RQ3: How can the CR visualization tools be improved?
There are several classes of vehicles configuration rules, and each of those is visualized using separate user interfaces in the studied industrial CR visualization tools. The more classes of vehicle configuration rules, the more user interfaces are required. This is an identified factor causing difficulties when using those tools. The demonstrator was developed with a CR visualization method inspired by a currently used matrix-based method, which was found at one of the studied vehicle manufacturing companies. In contrast to currently used CR visualization tools, the new suggested tool consisted of only one single user interface. The compromise for being able to use only one single interface is that the CR visualization tool shows a simplification of the configuration

rules. The simplification is created by using “association rules”, i.e. relationships between item usage rules and the other classes of configuration rules, which is a technique from one of the most common established visual data mining methods. It then becomes possible to show the formulation of item usage rules with its feature variant combinations. The feature variant combinations are computed from the product model authorizations rules and the feature variant combination rules. For more information about those computations, see Paper C. The simplification is made to overcome the difficulty in visualizing the formulation of item usage rules, product model authorization rules and feature variant combination rules at the same time.

The computations are today done in practice by iterating configuration rules queries and inspecting computational results; see Fig. 5.2. The design engineers who are not capable of stating configuration rule queries independently are supported in this activity by product structure specialists. For stating a configuration rule query, the user has to know which feature families should be included in the query, and know how to limit the scope to get a manageable answer. This should be contrasted to the fact that, with the current industrial CR visualization tools, among 10 design engineers as many as 40% stated that they primarily relied on oral communication or validation by “others’” before their own use of the industrial CR visualization tools.

The user evaluation of the demonstrator was positive and the predicted outcome from an industrial implementation was a more time-efficient and less error-prone vehicle CR development process.

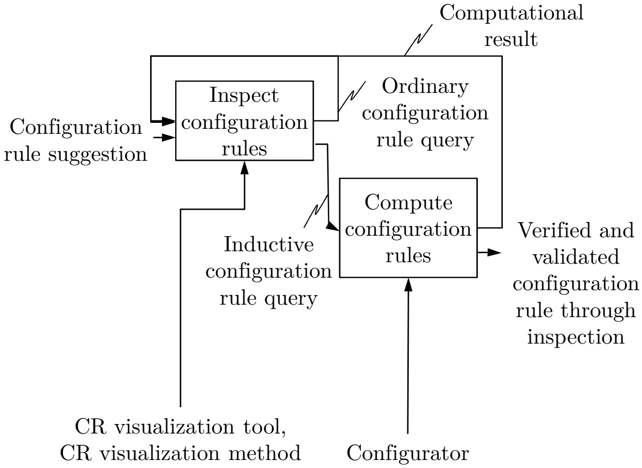


Figure 5.2: The configuration rule query as an input for the computation of configuration rules.

5.2 Contribution

The contribution of this Licentiate thesis concerns the research on product structure and on methods used when developing vehicle configuration rules, as well as the CR visualization tools. The contributions of the research for the thesis are described in the following sections:

- *Generalization of the information model for vehicle product structures.* The vehicle product structure contains both the items and the feature domain, and the configuration rules are both inter- and intra-domain configuration rules for those domains. This is in contrast to much research on product structures, where often e.g. there are no intra-domain configuration rules, here called item usage rules. (Configurable product structures, Paper A)
- *Formalized methods for authoring configuration rules.* The authoring methods are described by using logic expressions. As vehicles are variant-rich products, the economic potential for automating the development of vehicle configuration rules is high. Automation requires formally defined methods using logic expressions. (Configuration rules development process, Paper B)
- *Different roles – different CR visualization needs.* The authoring variations as well as the verification methods found have been evaluated for strengths and weaknesses based on different users. What was found is that different use of the configuration rules motivates different user preferences, which in this context may be interpreted as CR visualization needs. (Configuration rules development process, Paper B)
- *CR visualization methods.* Based on an evaluation of the currently used CR visualization tools it was possible to develop a new CR visualization method. The new tool has been tested and makes it easier and more time-efficient to answer the typical configuration rule queries.
- *Clarification of the definitions for verification and validation during the development of vehicle configuration rules.* The research on vehicle configuration has so far mainly concerned the computation and execution of vehicle configuration rules. The use of the conducted research during the development of configuration rules has been classified into the verification and validation activities, by making a comparison to knowledge-based systems. (Sections on CR development process in this thesis)

5.3 Goal fulfilment

The purpose of the research project was to create tools that reduce the difficulties during the CR development process in order to make the process more time-efficient and less error-prone. The demonstrator is the tool that has been developed and evaluated with real users to achieve a more time-efficient and less error-prone development of configuration rules.

5.4 Generalizability

"Generalizability" refers to the prediction based on recurrent experience. If something occurs frequently, it is expected that it will continue to do so in the future. There are three types of generalization claims according to Runkel & McGrath (1972). Together, they increase the strength of a claim for generalizability. The different types of generalization claims are:

1. *Different circumstances.* Compared to most other products, vehicles are variant-rich and each vehicle variant is consists of many items. The vehicle configuration is therefore a suitable application when studying the difficulties during the CR development process.

The research work for this Licentiate thesis has studied three vehicle manufacturing companies, which is arguably sufficient for claiming generalization of the vehicle CR development process. The vehicle manufacturing companies are, however, using in-house developed CR visualization tools and the published literature on those is very sparse.

2. *Different measurements.* The tests that have been conducted with the suggested CR visualization tool were based on four typical configuration rule queries during the CR development process. The queries were found both during user studies and from the literature. It was noted from comments by the users that the benefits of an easier-to-use CR visualization tool goes beyond those four configuration rule queries. The suggested CR visualization tool may potentially support those configuration rule queries without further development. The post-session interview questions during the usability tests asked for benefits and drawbacks of the suggested CR visualization tool without referring specifically to the configuration rule queries from the test. Future work could further evaluate the benefits of the suggested CR visualization tool.

3. Different test groups. The issues of having research results valid only within the selected test group has been addressed by selecting users with an aim of heterogeneity, e.g. different roles and different years of experience.

The conclusion regarding generalizability is that the three types of generalization claims have been addressed within the application of vehicle manufacturing companies. The broadening of project scope to other application areas is left as potential future work.

5.5 Summary

This chapter has answered the main research questions and then evaluated the outcome by discussing this Licentiate thesis' goal fulfilment and contributions. Finally, the process of how to achieve the research results has been discussed concerning generalizability.

6 Conclusions

In this thesis, the problem of an error-prone and time-consuming CR development process has been approached from scratch by studying the product structure information model as well as the industrial CR visualization tools.

The aim of this research work was to identify factors that are causing difficulties during the CR development process, and to find methods and tools that address those. The measurable success factor was defined as an easier-to-use CR visualization tool resulting in a more time-efficient and less error-prone CR development process.

The first step was to find a generalized information model for vehicle product structures suitable for representing the information needed when developing configuration rules. From the study of the CR development process, methods for authoring configuration rules have been formalized and implemented into a CR visualization tool. The new tool facilitates the CR development process by visualizing configuration rules together in one single user interface. The necessary computations for creating this visualization were previously done iteratively by manually stating configuration rule queries. By performing this automation, the CR visualization tool becomes easier to use for both domain specialists and product structure specialists. Usability tests have indicated that the CR development process then becomes more time-efficient and less error-prone.

The most important conclusions drawn in this Licentiate thesis are:

- **Configurable product structures:** With increased product variety, vehicle product structures have evolved from unique BOMs for every vehicle configuration to rule-based product structures that are instantiated by executing the configuration rules. The two-tiered information model with its intra- and inter-domain configuration rules has been found to complicate the CR development process. There is an ambiguity in how to author configuration rules, as well as difficulties in how to visualize different classes of configuration rules.
- **CR development process:** The product modification request requiring new or modified configuration rules initiates the CR development process The potentially

iterative activities include authoring, evaluation and release of configuration rules. In order to make the CR development process more time-efficient and less error-prone, a reduction of required iterations has to be achieved. By formalizing the authoring methods, it was possible to visualize alternative methods in a CR visualization tool, increasing the users' confidence in how to interpret the configuration rules. Another source of iterations is the manually stated configuration rule queries that could be reduced, visualizing the different classes of configuration rules in one single user interface.

- **CR visualization tools:** The industrial CR visualization tools are based on either lists, tables or matrices. The matrix-based CR visualization method has been found to have the advantage of increasing readability. The currently used matrix-based CR visualization method could be easier to use and better support the CR development process by (1) including the visualization of authoring methods, and (2) visualizing different classes of configuration rules in one single user interface. Evaluations together with the real users of the industrial CR visualization tools have shown that the new CR visualization tool efficiently supports the users when developing the configuration rules and makes the activity more time-efficient and less error-prone.

7 Future work

The research presented in this Licentiate thesis has three interesting areas that can be addressed as future work: comparative tests, more diverse settings and further development of the demonstrator.

7.1 Comparative tests of CR visualization tools

The formative usability tests that have been conducted in this research work contained post-session interviews for collecting the users' prediction of the demonstrator's usability. Planned future work concerns primarily the further testing of the demonstrator. The quantitative measurements during the tests were positive: configuration rule queries were answered in the order of seconds. From a statistical viewpoint, critique may be directed towards the low number of test users. The formative usability tests that have been conducted serve their purpose during the development of the demonstrator. However, the evidence of the demonstrator's success currently consists partly of statements made during user interviews. It would be possible to achieve more reliable test results if a comparative test (industrial CR visualization tools versus the CR visualization tool developed for this Licentiate thesis) could be conducted.

7.2 More diverse settings

Even though three vehicle manufacturing companies have been studied, the cultural variety among these is rather limited. Culture influences the CR development process rules, since the users act within a cultural environment (Röse et al., 2001). The research for this Licentiate thesis has taken place in Sweden, which yields a fairly low rate of uncertainty avoidance compared to e.g. Germany and Japan. This factor would be interesting to study in relation to how the configuration rules are authored and evaluated.

It would also be interesting to study whether the demonstrator may be applicable to other industries than the vehicle manufacturing industry. The vehicle configuration information model consisting of a rule-based product structure has been found to be rather industry-specific. The applicability of the demonstrator to other industries should be investigated. However, the method of formative usability tests for developing the demonstrator has been successful, and can be studied as to whether it can be re-used.

7.3 Development of the demonstrator

There is also a potential in looking into whether the demonstrator can be improved, e.g. by addressing the remaining critical incidents such as the debugger. Other potential additions are the CR visualizations of feature variant combination rules. Those configuration rules are currently used only for doing computations.

References

Agrawal, R., Imielinski, T. & Swami, A. (1993), "Mining Association Rules between Sets of Items in Large Databases", *Proceedings of International Conference on Management of Data* (*SIGMOD),* Washington, USA, 26-28 May, pp. 207-216.

Akao, Y. & Mizuno, S. (1994), "QFD: The Customer-driven Approach to Quality Planning & Deployment", *Asian Productivity Organization,* Tokyo, Japan.

Amen, R., Rask, R. & Sunnersjö, S. (1999), "Matching Design Tasks to Knowledge-based Software Tools – When Intuition Does Not Suffice", *Proceedings of ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC99),* Las Vegas, USA, 12-15 September.

Anand, S., & Buchner, A. (1998), "Decision Support Using Data Mining", *Financial Times Pitman Publishers,* London, UK.

Andreasen, M. M. (1980), "Syntesemetoder på Systemgrundlag – Bidrag till en Konstruktionsteori", [Synthesis Methods on a Systems Foundation], *PhD thesis,* Lund Technical University, Lund, Sweden.

Andreasen, M. M. (1992), "Designing on a "Designer's Workbench (DWB)", *Proceedings of the 9th WDK workshop,* Rigi, Switzerland, March.

Angele, J., Fensel, D., Landes, D & Studer, R. (1998), "Developing Knowledge-based Systems with MIKE", *Automated Software Engineering,* Vol. 5, pp. 389-418.

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", *Proceedings of European Conference on Artificial Intelligence (ECAI),* Lisbon, Portugal, 16 August, pp. 33-39.

Baldwin, C. Y. & Clark, K. B. (2000), "Design Rules: The Power of Modularity", *MIT Press,* Cambridge, USA.

Barnes, J., Conrad, K., Demont-Heinrich, C., Graziano, M., Kowalski, D., Neufeld, J. & Zamora, J. (2005), "Generalizability and Transferability", http://writing.colostate.edu/guides/research/gentrans [accessed 2011-11-01].

Baumeister, J. & Freiberg, M. (2010), "Knowledge Visualization for Evaluation Tasks", *Knowledge Information Systems, Springer-Verlag,* London, UK.

Becker, H. S. (1970), “Sociological Work: Method and Substance”, *Aldine*, Chicago, USA.

Blecker, T. Abdelkafi, N. (2006), “Variety Management in Assemble-to-order Supply Chains”, *Munich Personal RePEc Archive*, pp. 39-56.

Blessing, L. & Chakrabarti, A. (2009), “DRM, a Design Research Methodology”, *Springer-Verlag,* London, UK.

BMW (2011), “Build your own car”, http://www.bmw.co.uk [accessed 2011-03-04].

Bongulielmi, L., Henseler, P., Puls, C. & Meier, M. (2001), “The K- & V-Matrix Method - An Approach in Analysis and Description of Variant Products”, *Proceedings of International Conference on Engineering Design (ICED),* Glasgow, UK, 21-23 August, pp. 571-578.

Bongulielmi, L., Henseler, P., Puls, C. & Meier, M. (2002), “The K- & V-Matrix-Method in Comparison with Matrix-Based Methods supporting Modular Product Family Architectures”, *Proceedings of NordDesign,* 14-16 August, Trondheim, Norway.

Boulicaut, J-F. & Masson, C. (2005), “Data Mining Query Languages”, In Maimon, O. & Rokach, LL. (Eds.), *The data mining and knowledge discovery handbook,* Springer Science Business Media, New York, USA.

Cabena P., Hadjinian, P., Stadler, R., Verhees, J. & Zanasi, A. (1998), “Discovering Data Mining: from Concepts to Implementation”, *Prentice Hall Saddle River,* New Jersey, USA.

Catic, A. (2011), “Knowledge-based Engineering in Product Development Process”, *PhD thesis,* Chalmers University of Technology, Gothenburg, Sweden.

Cederfeldt, M. (2005), “Planning Design Automation Systems: Criteria and Guidelines” *Licentiate thesis,* Chalmers University of Technology, Gothenburg, Sweden.

Cederfeldt, M. (2007). “Planning Design Automation: A Structured Method and Supporting Tools”, *PhD thesis,* Chalmers University of Technology, Gothenburg, Sweden.

Chalmers (2011), “Configuration Rule Management”, http://www.chalmers.se/ppd/EN/reserach/product-development/ongoing-projects/ [accessed 2011-10-19]

Cios, K. & Kurgan, L. (2005). “Trends in Data Mining and Knowledge Discovery”. In Pal, N.R. & Jain, L.C. (Eds.), *Advanced Techniques in Knowledge Discovery and Data Mining,* Springer Verlag, London, UK.

Claesson, A. (2006), “A Configurable Component Framework Supporting Platform-based Product Development”*, PhD thesis,* Chalmers University of Technology, Gothenburg, Sweden.

Claesson, A., Johannesson, H. & Gedell, S. (2001), “Platform Product Development: Product Model a System Structure Composed of Configurable Components”, *Proceedings of ASME Design Engineering Technical Conferences (DETC),* Pittsburgh, USA, 9-12 September.

Collier, W. (1999), "A Common Specification for Systems-based Product Modelling", *DH Brown Associates.*

Collier, W. (2001), "Fundamentals of Shared Product Structure", *DH Brown Associates.*

Daimler (2010), "Annual Financial Report 2010", http://www.daimler.com/Projects/c2c/channel/ [accessed 2011-12-29]

Duffy, A. H. B. & Andreasen, M. M. (1995), "Enhancing the Evolution of Design Science", *Proceedings of International Conference on Engineering Design (ICED),* Prague, Czech Republic, 22-24 August.

Enovia (2011), "Variant Configuration Central", http://www.3ds.com [accessed 2011-04-01].

Erens, F. & Wortmann, H.C. (1996), "Generic Product Modeling for Mass Customization", *Implementation Road Map,* Ann Arbor, USA.

Euwe, M.J. & Schuwer, R.V. (1993), "Configuration of Complex Products", *Computers in Industry – Knowledge Engineering,* Vol. 21, No. 1, pp. 1-10.

Fayyad, U., Platetsky-Shapiro, G. & Smyth, P. (1996), "From Data Mining to Knowledge Discovery in Databases", *AI Magazine,* Vol. 17, No. 3, pp. 37-54.

Fiat (2010), "Fiat S.p.A Annual Report (2010)", http://www.fiatspa.com/en-US/investor_relations/financial_reports [accessed 2011-12-29]

Fleischanderl, G. (2000), "User Interface Requirements for Knowledge Acquisition and Modeling", *Proceedings of European Conference on Artificial Intelligence (ECAI),* Berlin, Germany, 21 - 22 August.

Foldoc, (1994), "knowledge", Free on-line dictionary of computing, http://foldoc.org/knowlege [accessed 2011-12-08]

Ford (2010), "Annual Report 2010 – Ford Motor Company", http://corporate.ford.com/doc/ [accessed 2011-12-29]

Frawley, W.J., Platetsky-Shapiro, G. & Matheus, C.J. (1992), "Knowledge Discovery in Databases: An Overview", *AI Magazine,* Vol. 13, No. 3, pp. 57-70.

Fuxin, F. (2005), "Evolution and Communication of Geometry Based Product Information within an Extended Enterprise", *PhD thesis,* Luleå University of Technology, Luleå, Sweden.

Garwood, D. (1988), "Bills of Materials: Structured for Excellence", *Dogwood Publishing,* Marietta, USA.

Guyot, E., Charles, S., Eynard, B. & Gallet, T. (2006), "Towards PLM Based Digital Mock-up Exchange in Collaborative Product Development", In Ghodus, Rp., Dieng-Kuntz, R. & Loureiro (Eds.), *Leading the web in concurrent engineering, IOS press,* Amsterdam, Netherlands.

Haag, A. (1998), “Sales Configuration in Business Processes”, *Intelligent systems and their applications*, Vol. 13, No. 4, pp. 78-85.

Hami-Nobari, S. & Blessing, L. (2005), “Effect-oriented Description of Variant Rich Products”, *International Conference on Engineering Design (ICED),* Melbourne, Australia, 15-18 August.

Hansen, C. T. (1995), “An Approach to Simultaneous Synthesis and Optimization of Composite Mechanical Systems”, *Journal of Engineering Design,* Vol. 6, No. 3, pp. 249-266.

Harlou, U. (2006), “Developing Product Families Based on Architectures”, *PhD Thesis,* Technical University of Denmark, Lyngby, Denmark.

Hedin, G., Olsson, L. & McKenna, J. (1998), “Product Configuration Using Object Oriented Grammars”, *Lecture Notes in Computer Science, Springer-Verlag,* Berlin Heidelberg, Germany, pp. 107-126.

Helo, P. T. (2006), “Product Configuration Analysis with Design Structure Matrix”, *Industrial Management & Data Systems,* Vol. 106, No. 7, pp. 997-1011.

Hertz, S., Johannsson, J. K. & de Jager, F. (2001), “Customer-oriented Cost Cutting: Process Management at Volvo”, *Supply Chain Management,* Vol. 6, No. 3, pp. 128-141.

Hix, D. & Hartson R. H. (1992), “Formative Evaluation: Ensuring Usability in User Interfaces”, *Technical Report,* Virginia Polytechnic Institute & State University, Blacksburg, USA.

Hospach, H-J., Katzenmaier, J., Wasmer, A. & Viel, C. (2002), “SAM - The Smart PDM Tool for Suppliers of DaimlerChrysler and Renault”, *Product Data Journal,* Vol. 2, pp. 17-19.

Huang, Y., Liu, H., Ng, W. K., Lu, W., Song, B. & Li, X. (2008), “Automating Knowledge Acquisition for Constraint-based Product Configuration," *Journal of Manufacturing Technology Management,* Vol. 19, No. 6, pp. 744-754.

Hubka, V. & Eder, W. E. (1988), “The Theory of Technical Systems”, *Springer Verlag,* Berlin, Germany.

Hvam, L., Mortensen, N. H. & Riis, J. (2007), “Product Customization”, *Springer Verlag,* Berlin, Germany.

IBM (1969), “Information Management System/360 for the IBM system/360 Program Description”, *IBM*, White Plains, USA.

IBM (2004), “History of IMS: Beginnings at NASA”, http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/index.jsp?topic=/com.ibm.imsnews.doc/newslnewsle/v902/v902.htm [accessed 2011-12-13]

Ikaros (2012), “KDP/KOLA”, http://www.ikraos.se/kdp-kola, [accessed 2012-01-26].

Isaksson, O., Fuxin, F., Jeppsson, P., Johansson, H., Johansson, P., Katchaounov, T., Lindeblad, M., Ma, H., Malmqvist, J., Mesihovic, S., Sutinen, K., Svensson, D. & Törlind, P.

(2000), “Trends in Product Modelling - An ENDREA Perspective”, *Proceedings for Produktmodeller,* Linköping, Sweden, pp. 65-88.

ISO (2004), “ISO 10303”, *International Organization for Standardization,* http://www.iso.org [accessed 2011-03-04].

ISO/IEC (2008), “ISO/IEC 12207:2008 - Systems and Software Engineering”, http://www.iso.org [accessed 2011-12-01].

Johansson, F. & Eriksson, M. (2007), “3D-modellen som Informationsbärare” [3D Model as Information Carrier], *Thesis for the Degree of Master (One Year),* Halmstad University, Halmstad, Sweden.

Jonassen Hass, A. M. (2003), “Configuration Management Principles and Practice”, *Addison-Wesley Professional,* Boston, USA.

Juristo, N. & Morant, J.L. (1998), “Common Framework for the Evaluation Process of KBS and Conventional Software”, *Knowledge-Based Systems,* Vol. 11, No. 2, pp. 145-159.

Kleim, D. A. (2002), “Information Visualization and Visual Data Mining”, *IEEE Transactions on visualization and computer graphics,* Vol. 7, No. 1, pp. 100-107.

Kvist, M. (2010), “Product family assessment”, *PhD thesis,* Technical University of Denmark, Lyngby, Denmark.

Lamberti, R. (2011), “Effiziente Kreativität – Prozesse und Methoden in der Automobilindustrie”, [Efficient Creativity – Processes and Methods in the Automotive Industry], Lecture presented in Informationstechnik und Wissensverarbeitung, Karlsruher Institut für Technologie, Karlsruhe, Germany, 2011.

Lee, Y.T. (1999), “Information Modeling: From Design to Implementation”, *Proceedings of the Second World Manufacturing Congress (WMC),* Durham, UK, 27-30 September, pp 315-321.

Lengler, R. & Eppler, M. (2007), “Towards a Periodic Table of Visualization Methods for Management”, *Proceedings of Graphics and Visualization in Engineering (GVE),* Clearwater, USA, 3-5 January.

Lindroth, P. (2011), “Product Configuration from a Mathematical Optimization Perspective”, *PhD thesis,* Chalmers University of Technology, Gothenburg, Sweden.

Luh, D-B., Ko, Y-T. & Ma, C-H. (2011), “A Structural Matrix-based Modelling for Designing Product Variety”, *Journal of Engineering Design,* Vol. 22, No.1, pp. 1-29.

Malmqvist, J. (2002), “A Classification of Matrix-based Methods for Product-modeling”, *Proceedings of Design Conference (DESIGN),* Dubrovnik, Croatia, 14-17 May, pp. 203-210.

Mather, H. (1982), “Bills of Materials, Recipes and Formulations”, *Wright Publishing Company Inc.,* Atlanta, USA.

Maurer, M. (2007), “Structural Awareness in Complex Product Design”, *PhD Thesis,* Technical University of Munich, Munich, Germany.

Maxwell, J. M. (2005), “Qualitative Research Design”, *Applied Social Research Method Series,* Vol. 42.

McKay, A., Erens, F. J. & Bloor, M. S. (1996), “Relating Product Definition and Product Variety”, *Research in Engineering Design,* Vol. 2, pp 63-80.

Meseguer, P. & Preece, A. D. (1995), “Verification and Validation of Knowledge-based Systems with Formal Specifications”, *The Knowledge Engineering Review,* Vol. 10, pp 331-343.

Mesihovic, S. & Malmqvist, J. (2000), “Product Data Management (PDM) System Support for the Engineering Configuration Process”, *Proceedings of European Conference on Artificial Intelligence (ECAI),* Berlin, Germany, 20-25 August.

Mesihovic, S. & Malmqvist, J. (2004a), “On the Development and Sales-delivery Process of Configurable Products”, *Licentiate Thesis,* Chalmers University of Technology, Gothenburg, Sweden.

Mesihovic, S. & Malmqvist, J. (2004b), “A Process-oriented Approach for Management of Product Configuration Models”, *Proceedings of ASME Design Engineering Technology Conference (DETC),* Paper No DETC2004-57384, Salt Lake City, USA, 28 September - 2 October.

Mittal, S. & Frayman, F. (1989), “Towards a Generic Model of Configuration Tasks”, *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI89),* San Mateo, USA, pp. 1395-1401.

MOKA (2000), “MOKA: A Framework for Structuring and Representing Engineering Knowledge”, http://webl.eng.coventry.ac.uk/moka [accessed 2012-01-23].

Monden, Y. (1983), “The Toyota Product System”, *Productivity press,* Portland, USA.

Mortensen, N.H. (2000) “Design Modelling in a Designer’s Workbench”, *PhD thesis,* Technical University of Denmark, Lyngby, Denmark.

Mortensen, N.H., Yu, B., Skovgaard, H.J. and Harlou, U. (2000), “Conceptual Modeling of Product Families in Configuration Projects”, *Proceedings of 14th European Conference on Artificial Intelligence (ECAI),* Berlin, Germany, 20-25 August, pp. 68-73.

Männistö, T., Soininen, T. & Sulonen, R. (2001), “Modelling Configurable Products and Software Product Families”, *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI-2001),* Seattle, USA, 4-10 August.

Neubert, S. (1993), “Model Construction in MIKE (Model Based and Incremental Knowledge Engineering)”, *Lectures in Computer Science,* Vol. 723, pp. 200-219.

Nielsen, J. (1993), “Usability Engineering”, *Academic Press,* San Diego, USA.

Normile, D., Tyrka, K., Bulkeley, D. & Bak, D. (2001), “Design Anywhere, Build Anywhere”, http://www.designNews.com/document.asp?doc_id=216157, [accessed 2012-01-06]

OMG (2010), “UML Superstructure Specification”, http://www.omg.org [accessed 2011-03-29].

O’Keefe, R. M. (1993), “Expert System Verification and Validation: A Survey Tutorial”, *Artificial Intelligence Review,* Vol. 7, pp 3-42.

Oeltjenbruns, H., Bracht, U., Neidlein, G. & Frisch, M. (2000), “Standardized Product Creation Process in Commercial Vehicle Development”, *SAE Technical paper,* DOI:10.4271/ 2000-01-3168.

Pak, C. (2011), “Hantering av Dataflöde i Produktutveckling”, [Management of Information Flow in Product Development], *Master Thesis*, Royal University of Technology, Stockholm, Sweden.

Pugh, S. (1996), “Creating Innovative Products Using Total Design”, *Addison Wesley Publishing,* Reading, USA.

Preece, A. D., Talbot, S. & Vignollet, L. (1997), “Evaluation of Verification Tools for Knowledge-based Systems”, *International Journal of Human-Computer Studies,* Vol. 47, pp. 629-658.

PTC (2011), “Windchill”, http://www.ptc.com [accessed 2011-04-01].

Pyle, M.J. (2010), “Imitation in Large Complex Organizations: When Does Copying Become Learning”, *PhD thesis,* University of Michigan, Ann Arbor, USA.

Runkel, P. J. & McGrath, E. J. (1972), “Research on Human Behavior: A Systematic Guide to Method”, *Holt, Rinehart and Winston, Inc.,* New York, USA.

Röse, K., Liu, L. & Zühlke, D. (2001), “Design Issues in Mainland China: Demands for a Localized Human-machine-interaction Design”, *Proceedings of 8th Symposium on Analysis, Design, and Evaluation of Human-Machine Systems,* Kassel, Germany, 18-20 September, pp. 17–22.

Sabin, D. & Weigel, R. (1998), “Product Configuration Frameworks – A Survey”, *IEEE Intelligent Systems,* Vol. 13, No. 4, pp. 42-49.

Schreiber, G., Wielinga, B. & Breuker, J. (1993), “KADS – A Principled Approach to Knowledge-based System Development”, *Academics Press,* London.

Sheiderman, B. (1996), “The Eye Have it: A Task by Data Type Taxonomy for Information Visualization”, *IEEE Symposium on Visual Languages,* Boulder, USA, 3-6 September, pp. 336-343.

Siemens, “Teamcenter”, http://www.plm.automation.siemens.com [accessed 2011-04-01].

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.

Soininen, T., Tiihonen, J., Männistö, T. & Sulonen, R. (1998), “Towards a General Ontology of Configuration”, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing,* No. 12, pp. 357-372.

Soininen, T., Niemelå, I., Tiihonen, J. & Sulonen, R. (2001), “Representing Configuration Knowledge with Weight Constraint Rules”, *Proceedings of the Association for the Advancement of Artificial Intelligence fall symposium (AAAI 2001),* North Falmouth, USA, 2-4 November.

Steward, D. (1981), “The Design Structure System: A Method for Managing the Design of Complex Systems”, *IEEE Transaction on Engineering Management,* Vol. 28, No. 3, pp. 79-83.

Suh, N. P. (1990), “The Principles of Design”, *Oxford University Press,* New York, USA.

Sunnersjö, S. (1994), “AID-Features: A New Tool for Design Automation”, *Proceedings of Lancaster International Workshop on Engineering Design (CACD’94),* Lancaster, UK, 11-13 April, pp. 241-258.

Svensson, D., Malmström, J., Pikosz, P. & Malmqvist, J. (1999), “A Framework for Modelling and Analysis of Engineering Information Management Systems”, *Proceedings of ASME Design Engineering Technical Conferences (DETC),* Paper No DET99/CIE-9006, Las Vegas, USA, 12-15 September.

Svensson, D. & Malmqvist, J. (2002), “Strategies for Product Structure Management at Manufacturing Firms”, *ASME Journal of Computers in Science and Engineering,* Vol. 2, No. 1, pp. 50-58.

Tacton, “TactonStudio”, http://www.tacton.se [accessed 2011-03-04].

Tjalve, E. 1979, “A Short Course in Industrial Design”, *Hodder Arnold,* London, UK.

Tuhrim, S., Reggia, J.A. & Floor, M. (1988), “Expert System Development: Letting the Domain Specialist Directly Author Knowledge Bases”, In Hendler, J.A. (Eds.), *Expert systems: the user interface,* Albex Publishing Corporation, Norwood, USA.

Van Veen, E. A. (1992), “Modelling Product Structures by Generic Bill-of-Materials”, *Elsevier Science Inc.,* New York, USA.

VDA (2010), “VDA 4965 – ECM Recommendation”, *Strategic Automotive Product Data Standards Industry Group (SASIG),* http://www.vda.de/en/publikationen/publikationen_downloads [accessed 2012-02-13].

Viel, C. (1993), “Management of Product’s Diversity: Industrial Validation of AP214 Reference Model”, *International Journal of Computer Applications in Technology,* Vol.18, Nos. 1-4, pp. 62-77.

Volvo (2010), "The Volvo Group Annual Report 2010", http://www.volvogroup.com/group/global/en-gb/investors/reports/annual_reports/ [accessed 2011-12-29]

Voronov, A., Åkesson, K., Ekstedt, F. (2011), "Enumeration of Valid Partial Configurations", *Proceedings of Workshop on Configuration (IJCAI 2011)*, Barcelona, Spain, 16-22 July, pp. 25-31.

Watts, F. (2008), "Engineering Documentation Control Handbook: Configuration Management in Industry", *William Andrew,* Norwich, USA.

White, J., Schmidt, D. C., Bendavides, D., Trinidad, P., Ruiz-Cortés, A. (2008), "Automated Diagnosis of Product-line Configuration Errors in Feature Models", *Proceedings of Software Product Line Conference (SPLC),* Limerick, Ireland, 8-12 September, pp. 225-234.

Wirth, R. & Hipp, J. (2000). "CRISP-DM: Towards a Standard Process Model for Data Mining", *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining (PADD2000),* Manchester, UK, 11-13 April, pp. 29–39.

Wong, P.C. (1999), "Visual Data Mining", *IEEE Computer Graphics and Applications*, Vol. 19, No. 5, pp. 20-21.

Appended papers A-C

Paper A Tidstam, A. & Malmqvist, J. (2010), "Information Modelling for Automotive Configuration", *Proceedings of NordDesign,* Gothenburg, Sweden, 25-27 August, pp. 275-286.

Paper B Tidstam, A. & Malmqvist, J. (2011), "Authoring and Verification of Vehicle Configuration Rules", *Proceedings of Product Lifecycle Management (PLM),* Eindhoven, The Netherlands, 11-13 July.

Paper C Tidstam, A., Bligård, L-O., Ekstedt, F., Voronov, A., Åkesson, K. & Malmqvist, J. (2012), "Development of Industrial Visualization Tools for Validation of Vehicle Configuration Rules", *Proceedings of Tools and Methods of Competitive Engineering (TMCE2012),* Karlsruhe, Germany, 7-11 May.

CHALMERS UNIVERSITY OF TECHNOLOGY
SE - 412 96 Gothenburg, Sweden
Telephone: + 46 - (0)31 772 10 00
www.chalmers.se